UNIVERSIDAD CARLOS III DE MADRID

Ph.D. THESIS

Rational Exchange Protocols

Author:
Almudena Alcaide Raya

Supervisors:
Dr. D. Juan M. Estévez Tapiador
Dr. D. Arturo Ribagorda Garnacho

Computer Science Department

Leganés, November 2008

UNIVERSIDAD CARLOS III DE MADRID

TESIS DOCTORAL

Protocolos de Intercambio Racional

Autor:
Almudena Alcaide Raya

Directores:
Dr. D. Juan M. Estévez Tapiador
Dr. D. Arturo Ribagorda Garnacho

Departamento de Informática

Leganés, Noviembre 2008

TESIS DOCTORAL

RATIONAL EXCHANGE PROTOCOLS

Autor: Almudena Alcaide Raya

Directores: Dr. D. Juan M. Estévez Tapiador
Dr. D. Arturo Ribagorda Garnacho

Firma del Tribunal Calificador:

Firma

Presidente:

Vocal:

Vocal:

Vocal:

Secretario:

Calificación:

Leganés, de de

Agradecimientos

Sin la ayuda de mi director de tesis, Juan M. Estévez Tapiador, este trabajo nunca hubiera existido. Creo que le he dicho ya mil veces gracias, pero que sirva este primer párrafo para agradecerle su esfuerzo una vez más. Espero no olvidar nunca todo lo que he aprendido trabajando contigo.

Gracias a Arturo Ribagorda Garnacho por su apoyo desde que entré en esta universidad en el 2002 y por haber codirigido esta tesis, a Julio C. Hernández Castro que ha hecho muchas veces de director en funciones y a Benjamín Ramos, que siempre está para echarme una mano.

A todo el grupo de seguridad del departamento de Informática de la Universidad Carlos III de Madrid, por su constante apoyo y paciencia, mi más sincero agradecimiento.

Gracias a mis padres Antonio y Maruja, a mis hermanos Toñín y Ángel, a Silvia y a Cova, a Thomas, a Sofía, a Andrés y a Grace, que mucho antes que yo supieron que sin duda, yo acabaría este proyecto.

Por último gracias a mis amigas porque, simplemente, no podría tener mejores compañeras de viaje.

Of Many Worlds in this World

Just like unto a Nest of Boxes round,
Degrees of sizes within each Boxe are found.
So in this World, may many Worlds more be,
Thinner, and lesse, and lesse still by degree;
Although they are not subject to our Sense,
A World may be no bigger then two-pence.
Nature is curious, and such worke may make,
That our dull Sense can never finde, but scape.
Margaret Cavendish, 1653.

El cuerpo consiente
dijo la mente.
El mundo giró un grado,
de regalo.
Úrsula Martí, 2008.

A mis hijos Thomas y Grace.

Abstract

An *exchange protocol* describes a sequence of steps by which several entities are capable of exchanging certain pieces of information in a particular context. *Rational*–exchange protocols serve that core purpose with several important advantages over the existing exchange paradigms, those referred to as *fair*–exchange solutions. Traditional fair–exchange protocols impose strong restrictions on the protocol execution context. They ensure fairness to participants but at the expense of entities such as TTPs (trusted third parties) having to be involved in the exchange. By contrast, rational schemes, although not ensuring fairness, assure that rational entities would have no reason to deviate from the steps described in the protocol and, have the enormous advantage of not needing the services of a TTP. Rational–exchange protocols therefore represent the only viable option in many modern ad–hoc and unstructured environments.

The main goal of this thesis is to apply concepts from Game Theory to both the analysis and design of rational–exchange protocols. In our opinion, significant contributions have been made in both directions:

- In terms of the formal analysis of these schemes, our work has focused on the proposal of two extensions to an existing formalism. The viability and effectiveness of our proposals is corroborated by the application of both formalisms to the analysis and verification of several exchange schemes.

- With regard to the design of rational protocols, our approach is based on applying heuristic search to automate the process, and to generate exchange protocols which can be proven rational within an underlying game theoretical framework.

  Experimental work is carried out to illustrate the proposed methodology in a particular three-entity exchanging scenario as well as in several randomized environments. Different heuristic techniques are implemented and their results compared, measuring success rates and the average number of protocols evaluated until an optimal solution is obtained. Furthermore, as a result of this experimental work, a whole family of multi–party rational exchange protocols is presented.

Resumen

Durante siglos el comportamiento racional de la especie humana ha sido extensamente estudiado por filósofos, sociólogos, psicólogos, etc. Considerado siempre como un concepto abstracto, a mediados del siglo veinte el desarrollo de la Teoría de Juegos proporcionó, por primera vez, un marco matemático para la definición formal del comportamiento racional de las entidades participantes de un juego. A partir de entonces la Teoría de Juegos se ha convertido en el modelo matemático que sustenta importantes resultados en campos tan diversos como la Biología, la Economía, la Inteligencia Artificial o la Criptografía.

Este trabajo se encuentra englobado dentro del campo de la Criptografía Racional. La Criptografía Racional nace de la aplicación de los resultados teóricos sobre juegos al campo de la Criptografía. Nielsen et al. en [Nielsen et al., 2007] establecen una relación de los avances más significativos llevados a cabo hasta el momento en esta área de reciente creación. En particular, especialmente relevantes para esta tesis serán los trabajos de Syverson [Syverson, 1998] y Buttyán et al. [Buttyán, 2001] centrados respectivamente en el diseño y análisis formal de protocolos seguros de intercambio racional.

0.1 Protocolos Criptográficos

Un protocolo seguro o criptográfico consiste en una sucesión de instrucciones dadas a un conjunto de entidades para llevar a cabo un cometido común. Estos protocolos, además, verifican una serie de propiedades de seguridad que vienen definidas por el objetivo concreto y el tipo de protocolo.

Ejemplos de protocolos seguros son los de distribución de claves secretas, para los que propiedades esenciales de seguridad son la confidencialidad y la integridad de las claves distribuidas, o los protocolos seguros de intercambio justo.

0.1.1 Protocolos Seguros de Intercambio Justo

Un protocolo seguro de intercambio justo describe una sucesión de pasos que han de seguir un conjunto de entidades para el intercambio de un conjunto de tokens de información, de tal manera que, ningún participante puede acabar el protocolo de manera desventajosa, esto es, habiendo enviado su token y no habiendo recibido nada a cambio. Además, la seguridad ofrecida por este tipo de protocolos queda garantizada incluso en entornos de ejecución en los que los participantes puedan ser agentes maliciosos.

Los protocolos seguros de intercambio justo son especialmente relevantes por el gran número de instancias y aplicaciones en las que son utilizados hoy en día: comercio electrónico, firma digital de contratos, servicios de correo certificado, etc. Sin embargo, existe una amplia clase de entornos en los que no pueden utilizarse.

La principal razón de este hecho se encuentra en un resultado formal de Pagnia y Gärtner [Pagnia and Gärtner, 1999] que establece que todo protocolo seguro de intercambio justo ha de involucrar a una entidad de confianza, llamada tercero de confianza (Trusted Third Party o TTP), que será la encargada de proteger a las entidades honestas participantes del intercambio.

En este marco, redes y servicios electrónicos ad-hoc sin infraestructuras preestablecidas, sistemas descentralizados, dispositivos móviles de baja capacidad computacional, operaciones críticas de respuesta inmediata, etc., representan un reto desde el punto de vista de la seguridad. Más en concreto, es relevante la ausencia de terceros de confianza en cualquiera de los contextos anteriormente citados, que puedan garantizar un intercambio justo en el que se apoyan muchas operaciones esenciales. En este tipo de entornos, los protocolos seguros de intercambio *racional* representan en la actualidad la opción más viable, ya que estas soluciones ofrecen la enorme ventaja de no necesitar los servicios proporcionados por una TTP.

0.2 Protocolos Seguros de Intercambio Racional

Un protocolo seguro de intercambio racional dicta una serie de pasos a un conjunto de entidades para que puedan intercambiarse una serie de tokens de información de tal manera que, si bien el protocolo no asegura que entidades honestas no puedan acabar en situación desventajosa con respecto a sus oponentes, lo que sí garantiza es que ninguna otra entidad obtendrá beneficio de ello. Los protocolos racionales operan sobre la premisa de que las entidades participantes sean racionales y tengan como objetivo maximizar su propio beneficio. En este supuesto, si manifestar un comportamiento malicioso con respecto a otros participantes no incrementa el beneficio obtenido, las entidades racionales no tendrán motivación alguna en hacerlo.

Finalmente, cabe destacar que el primer y único protocolo de intercambio racional que se encuentra en la literatura es el protocolo de Syverson [Syverson, 1998]. Este protocolo fue formalmente analizado por Buttyán et al. utilizando un formalismo basado en Teoría de juegos [Buttyán, 2001].

0.2.1 Análisis Formal y Diseño de Protocolos Seguros de Intercambio Racional

Distintos problemas son claramente identificados en cada una de estas áreas.
Por un lado, en el área del análisis formal de protocolos seguros de intercambio racional nos encontramos que:

- Existe tan sólo un formalismo para el estudio formal de este tipo de protocolos [Buttyán, 2001] y, además,
- El formalismo, basado en Teoría de Juegos elemental, es muy básico y limita su capacidad al análisis de esquemas muy sencillos en entornos muy controlados.

Por otro lado, en la parte del diseño de este tipo de soluciones encontramos que:

- Existe una ausencia total de metodologías para el diseño de este tipo de protocolos.
- Como resultado, existe una ausencia de propuestas y soluciones de intercambio racional y, por último,
- Son necesarios esquemas escalables que puedan dar solución a problemas de intercambio en entornos multiparte.

El trabajo realizado en esta tesis tiene por objetivo dar solución a cada uno de los puntos anteriores.

0.3 Contribuciones Principales

Esta tesis se encuentra dividida en dos partes claramente distinguibles.

Parte I: Análisis Formal de Protocolos Seguros de Intercambio Racional

Las aportaciones originales en este campo son:

1. La extensión, basada en juegos de información imperfecta, del modelo de análisis de Buttyán.
2. La extensión, basada en juegos de información incompleta o juegos bayesianos, del modelo de análisis de Buttyán .

De ambos modelos podemos resaltar las siguientes características:

1. Ambos formalismos suponen un avance en el estudio formal de los protocolos de intercambio racional, proporcionando una herramienta de análisis más completa y precisa en la que se pueden capturar fácilmente variables del entorno de ejecución de un protocolo, como son la desconfianza entre los participantes, la reputación de los mismos o el estado de la red de conexiones.

2. Dentro del campo global del análisis formal de protocolos resulta especialmente novedoso la representación, que en las propuestas descritas en este trabajo se hace, del comportamiento impredecible de un posible adversario, no encontrándose este comportamiento en ningún caso predefinido o limitado.

3. Los modelos extendidos son escalables a cualquier número de participantes y número de tokens de información que aquellos deseen intercambiarse.

4. Ambos modelos se han aplicado al análisis formal del protocolo racional de Syverson. Los resultados son manifiestamente significativos ya que difieren de los obtenidos hasta ahora con el esquema de análisis de Buttyán.

5. Además, se ha realizado un estudio formal basado en juegos bayesianos, de un protocolo racional de intercambio de contenidos en una red peer to peer.

Parte II: Diseño Automatizado de Protocolos Seguros de Intercambio Racional

En el campo del diseño de este tipo de protocolos podemos destacar las siguientes aportaciones:

1. La definición de una metodología formal de diseño de protocolos seguros y multiparte de intercambio racional.

2. La definición de una taxonomía de problemas y protocolos multiparte de intercambio racional. La taxonomía se basa en las posibles coaliciones entre entidades participantes y en los programas de incentivos de los que puedan formar parte.

3. La reducción del problema de diseño de este tipo de protocolos a un problema de optimización probabilística.

4. Por último, el desarrollo e implementación de una técnica, basada en una búsqueda heurística, para el diseño automatizado de protocolos multiparte de intercambio racional. Los protocolos sintetizados según estas técnicas son

además formalmente racionales, al existir para todos ellos una demostración formal de racionalidad basada en Teoría de Juegos e integrada en el proceso de síntesis.

Contents

List of Figures

List of Tables

Symbols and Terminology

Cryptography

$E_K(m)$	Symmetric encryption of message m using a secret key K.
$E_{K^{-1}}(m)$	Asymmetric encryption of message m using a private key K^{-1}.
$h(m)$	Cryptographic hash function applied to message m
k_{P_i}	Entity's P_i public key.
$k_{P_i}^{-1}$	Entity's P_i private key.
NRO	Non–Repudiation of Origin.
NRR	Non–Repudiation of Receipt.
PKI	Public Key infrastructure.
$PRNG$	Pseudo random number generater.
$sig(k^{-1}, m)$	Digital signature on m using private key k^{-1}, i.e. $E_{k^{-1}}(h(m))$.
TTP	Trusted Third Party.
$w(\cdot)$	WSBC (Weakly Secret Bit Commitment) function.

Game Theory

$(\preceq_i)_{i \in P}$	Preference relation for player P_i.
γ_{P_i}	Item belonging to player P_i.
EGT	Evolutionary Game Theory.
$EP_{P_i}(s)$	Expected payoff for player P_i when following strategy s.
$EP_{P_i}(\rho)$	Expected payoff for player P_i when following probabilistic strategy profile defined by ρ.
$EP(P_i, s, \alpha)$	Expected payoff for player P_i when following strategy s and considering conjecture α.
ESS	Evolutionary Stable Strategy.
G_Π	Protocol game of imperfect information representing protocol Π.
G_Π^B	Bayesian protocol game representing protocol Π.
G_{Sy}^B	Bayesian protocol game representing Syverson's protocol.
G_{RP}^B	Bayesian protocol game representing content access protocol.

GT	Game theory
IPD	Iterated Prisoner's Dilemma.
NE	Nash Equilibrium.
P_d	Game player type dishonest.
P_h	Game player type honest.
PBE	Perfect Bayesian Equilibrium.
PD	Prisoner's Dilemma.
$\mathcal{T}$	Represents a type–profile space.
$\mathcal{T}_i$	Represents a type space for player P_i.
$u^-_{P_i}$	What γ_{P_i} is worth to player P_i.
$u^+_{P_i}$	What an item γ_{P_j} is worth to player P_i.

Mathematical Notation

$x!$	Factorial of an integer x.
$\mid x \mid$	Absolute value of integer x.
$\lfloor a/b \rfloor$	Quotient of the integer division between integers a and b.
$\Delta(X)$	It denotes the space of probability distributions over the set X.
$\varsigma(a, b)$	Function which returns a tuple of two components. The first component corresponds to the quotient of the integer division between a and b. The second component is the remainder of that division.
$E(X)$	Represents the expected value of a random variable X.
ln	Natural (or Napierian) logarithm.
$mod(a, b)$	Reminder of the division between integers a and b.
$\mathcal{M}_{n\times m}$	Set of $n \times m$ matrices.
$r(x, y)$	Fitness autocorrelation function.
$\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{R}^n$	Real numbers, reals greater than 0, n–tuples of reals.
$var(X)$	Represents the variance of a random variable X.

Protocol Synthesis

3–RES protocol	A three entity rational–exchange security protocol.
`ACC`	Item accessible.
`BENEF`	Item with the potential to increase payoff value.
$\hat{b}_i$	Maximum utility value for entity P_i.
$\bar{b}_i$	Minimum utility value for entity P_i.
$b_{i,j}$	Component of benefit matrix B.

B	Benefit matrix.
`COST`	Holding onto this item will decrease payoff value.
du_i	Differential utility for entity P_i.
$dU(S^{\Pi})$	Global differential utility when executing protocol Π.
D	List of description tokens.
$h_{i,j}(t)$	Component of state matrix $H(t)$.
$H(t)$	State matrix at step t of the protocol.
`LOST`	Lost item.
m	Number of items to be exchanged.
M–RES protocol	A multi–party rational–exchange security protocol.
`NEG_DR`	A negative dependency relationship.
`NO_ACC`	Item non–accessible.
`NO_COST`	No cost item.
n	Number of protocol steps.
o_i	Item involved in the exchange.
$\mathcal{O}$	Set of items.
$\mathcal{P}$	Set of protocol participants.
P_i	Protocol participant.
$r_{i,j}$	Component of dependency matrix R.
R	Dependency matrix.
$R\_DENSITY$	Density factor of the dependency matrix.
$s_{i,j}^{\Pi}$	Component of protocol matrix S^{Π}.
S^{Π}	Protocol matrix. It represents protocol Π.
t	Protocol step.
$u_i(t)$	Utility value for entity P_i at step t in the protocol.
`UNKNO`	Item unknown.
v	Number of protocol participants.
Y	List of payment tokens.

Peer to Peer

n_i	Node n_i.
$P2P$	Peer to peer.
$s_{n_i}(m)$	Node n_i's signature over content m.
$s_{n_i}^{n_j}(m)$	Node n_i's signature on m concatenated with n_j's identity.

Syverson's Protocol

α	Level of confidence entity B has on Syverson's protocol design.
β	Level of uncertainty entity A has over B's behavior at step two in the protocol.
δ	Probability that entity B assigns to the event of A sending m_3 at round three of the protocol.
$desc_{item_k}$	Description of $item_k$.
F_A	Penalty player A has to pay when proved to be the author and sender of a forged message m_1.
gm_k	Represents action send message *garbage* m_k.
m_k	Represents action send message m_k.
$quit_{P_i}$	Represents action taken by player P_i to quit the protocol.
ugm_k	Represents action send message *unpredictable garbage* m_k.
$u^-_{P_i}$	What $item_i$ is worth to entity P_i.
$u^+_{P_i}$	What a particular $item_j$ is worth to entity P_i.
$u^+_{P_i}(r)$	What a particular $item_j$ is worth to entity P_i at round r of the protocol.

Evolutionary Computation

α	Cooling factor.
$Avg.\ NPE$	Average number of protocols evaluated.
$F(\cdot)$	Fitness function.
GA	Genetic algorithm.
HC	Hill Climbing.
ILS	Iterated local search.
MIL	Moves in inner loop.
MR	Moving rate.
$N(S)$	Set of solutions in the neighborhood of solution S.
SR	Success rate.
SA	Simulated Annealing.
T	Temperature.
T_0	Initial temperature.
TS	Tabu search.

Chapter 1

Motivation, Scope and Objectives

This chapter serves to contextualize this thesis within the field of Rational Cryptography, and more particularly within the areas of the design and analysis of rational exchange security protocols.

For centuries, rationality in humankind has been extensively studied in areas of Philosophy, Sociology and Psychology always considered as an abstract concept. In the 1950's Game Theory provided a mathematical formalism to explain and define rational behavior and from then on it rapidly became the basis for theoretical models in other areas such as Economics, Biology and Artificial Intelligence. By contrast, relatively novel is the application of Game Theory and the concept of rationality to the area of Cryptography and in particular, to the analysis and design of security protocols.

1.1 Rational Cryptography

Rational Cryptography is the research area in which methodologies and techniques of Game Theory are applied to Cryptography.

At the heart of Rational Cryptography is the re-definition of the concepts of adversary and adversarial model. Traditionally, from a security point of view, an entity or participant involved in any type of electronic operation was considered to be either honest (behaving correctly) or dishonest, that is, intentionally and maliciously deviating from the instructions given for the completion of the task. Furthermore, a dishonest entity could have been motivated by the chances of attaining informational advantage over their opponents or by simply causing damage to third parties. In this context, Cryptography, as well as many other techniques, was deployed to protect properly behaved parties from malicious agents.

As this type of adversary could ultimately represent a worst-case attacker, heavy and elaborated constructions were designed to offer honest participants enough guarantees of security. Many of these operational schemes included trusted third parties (TTP) that would monitor operations to detect malicious actions and finally restore or compensate damage caused to honest participants. Although these techniques have been very successful at securing electronic operations, a recurring factor has always been high costs in terms of resources, time and the infrastructure needed for deployment. Unfortunately, in some instances such heavy requirements simply cannot be provided. Decentralized systems, light-weight devices with lower computational power, speed critical operations, ad-hoc e-services and networks, etc., all represent a challenge from a security point of view, difficult to overcome with the techniques and schemes currently available.

Rational Cryptography represents an alternative to some of the existing schemes for some specific environments in which it is feasible to modify the adversary model by introducing a new type of entity: a *rational* entity. Basically, rational agents aim to maximize their own benefit. A rational entity will only perform a task if and when it is in their own self-interest. Rational (self-interested) parties cannot be considered honest and dishonest or, good and bad. Rational agents behave exclusively in line with their expectations and objectives and, by being able to model these we will be able to predetermine their conduct.

When considering a traditional worst-case adversary (also referred to as the Dolev-Yao adversary model [Dolev and Yao, 1983]), the design of security schemes had to be such that agents were unable to misbehave as the security mechanisms would either prevent all possible deviations or detect (and therefore penalize) such conducts. By contrast, when considering an adversarial model based on rational entities, the designed schemes have to be such that entities do not misbehave on the sole assumption that misbehaving does not render any benefit. Furthermore, in the traditional adversarial model, entities had to be protected "externally" by applying recovery procedures or recurring to third parties to ensure safety. In a rational model, agents cannot react to attacks, they can only react to threats by abandoning the scheme before putting themselves in disadvantageous situations. Rational Cryptography can be seen as a trade-off of security for feasibility and resource economization.

Adversarial models based on rationality can be composed of all rational entities or, some mixed models have also been proposed where a fraction of the parties is assumed to be rational and the rest assumed to collude and behave arbitrarily, even against their own interests. In this thesis, we will only consider rational agents, selfish and self-interested, aimed solely at maximizing their own returns.

Rational Cryptography has already been applied to many common cryptographic scenarios. A summary report can be found in [Nielsen et al., 2007]. These are some of the applications:

- *Auction protocols and techniques*,
  [Cramer et al., 2001, Bogetoft et al., 2006].
- *Collaborative benchmarking and forecasting*,
  [Khetawat et al., 1997, Atallah et al., 2004].
- *Exchange protocols*,
  [Syverson, 1998, Buttyán, 2001].
- *Fair division of goods*,
  [Lipton et al., 2004, Bezáková and Dani, 2005].
- *Function evaluation*,
  [Izmalkov et al., 2005].
- *Multi–party computation*,
  [Abraham et al., 2006, Lysyanskaya and Triandopoulos, 2006].
- *Network Routing*,
  [Roughgarden and Tardos, 2000, Roughgarden, 2005].
- *Polling*,
  [Ambainis et al., 2004, Moran and Naor, 2006].
- *Secret sharing protocols*,
  [Halpern and Teague, 2004, Gordon and Katz, 2006].

In particular, some of the work in this thesis will focus on extending and complementing the work in [Syverson, 1998] and [Buttyán, 2001], applying Game Theory results to the analysis of rational exchange protocols.

1.2 Cryptographic Protocols

Over recent years there have been an overwhelming amount of new security protocols published at a variety of forums and conferences. Both fields, *Analysis* and *Design* of security protocols have been of enormous interest and proliferation of ideas.

In the next sections we will give a succinct overview of some of the most relevant aspects regarding each one of these fields. This brief introduction will further assist in contextualizing the contribution of our work.

1.2.1 Analysis of Cryptographic Protocols

The definition of formal models to validate and verify cryptographic protocols has been an area of intense development. Since the definition of the Dolev-Yao adversary model [Dolev and Yao, 1983], many tools and techniques have been developed to prove the correctness of a protocol. Informally, a security protocol is assumed to be correct when it satisfies all its goals, requirements and, security properties. However, correctness does not mean that the protocol offers protection against every possible type of attack [Bicakci and Baykal, 2003]. In fact, to attack a security protocol, we only need to step out of the set of restrictions imposed by the model used to verify its properties [Dennin, 1999].

Tools to formally validate classic security requirements such as confidentiality, authentication or integrity have been extensively studied. By contrast, the formal analysis of more recently defined security properties such as fairness, non-repudiation or timeliness, is still pending a global solution[1].

Several issues pose an extra challenge to the already complex task of protocol formal verification. Firstly, the constant emergence of new security properties and secondly, that many of these requirements cannot be modeled and represented using the formal tools and techniques currently available. Consider multi–party contract signing protocols as an example of a new service difficult to formalize using any of the old methods.

Different approaches have been considered when developing formal validation methods: *Abstract logics* [Burrows et al., 1990, Syverson and van Oorschot, 1994b]; *Communicating sequential processes language* [Roscoe, 1995]; *Inductive theorem proving method* [Paulson, 1998]; *Game theoretical models* [Kremer and Raskin, 2000, Buttyán and Hubaux, 2001, Buragohain et al., 2003]; *Model checking* [Lowe, 1997]; *Process algebra* [Abadi and Gordon, 1997]; *State exploring techniques* [Meadows, 1991], etc. Furthermore, most of the tools have been developed merging and combining several of those formalisms.

Moreover, the most powerful constraint to overcome when defining tools to formally analyze security protocols is known as the *Protocol Insecurity Problem* and has been the focus of study for decades. The protocol insecurity problem arises when one tries to answer a simple but very relevant question: Is it possible to formally decide whether a cryptographic protocol is secure or not?

[1] Section 1.2.2 provides a description for these and other common security properties.

The global idea evolves around the following issue: if the analytical model used to validate a given protocol imposes strong restrictions on entities and their capabilities to interact, then it might be possible to formally prove that a protocol is secure within such a controlled environment. On the other hand, relaxing the context in which entities interact and analyzing more realistic scenarios makes the insecurity problem harder and ultimately, undecidable. In this regard, important results have been obtained and are summarized in the following statements:

1. Secrecy is an undecidable security property [Durgin et al., 1999, Amadio et al., 2002, Delaune, 2006].
2. Protocol insecurity is in NP for an intruder that can exploit the properties of the XOR operator [Chevalier et al., 2003].
3. The testing problem for multi–party protocols of the Dolev-Yao type is NP-Hard [Even and Goldreich, 1983].
4. The protocol insecurity problem with finite number of sessions is NP-Complete [Rusinowitch and Turuani, 2001].

1.2.2 Design of Cryptographic Protocols

It is not only the formal analysis of security protocols that has been a challenge to researchers in recent years, but also the design and definition of such schemes.

Not all protocols serve the same purpose. The following list represents a small sample of the type of services that security protocols provide. This is a list in constant expansion as commercial and business driven organizations seek to satisfy market demands.

List of Protocol Services

- *Access control*
- *Entity authentication*
- *Auction and barter services*
- *Certified e-goods delivery*
- *Certified mail delivery*
- *Contract signing*
- *E-billing*

- *E-commerce services*
- *E-lottery*
- *E-voting*
- *Session Key establishment schemes*
- *Etc.*

Furthermore, each of the services that a security protocol provides is usually accompanied by a wide set of desired properties and characteristics. These properties are difficult to define but also, in many cases, their definition is not always consistent throughout the literature. Some effort has gone into classifying them according to different criteria (*safety properties* and *liveliness properties* [Lamport, 1977]) however, there still seems to be different understanding of which ones are the required security properties for each kind of service. Finally, some of these properties are optional and will be specific to a given kind of security service, while others are essential and indispensable in all instances of a cryptographic protocol.

As an example, what follows is a list with the informal description of some of the most common protocol security properties in alphabetical order.

List of Protocol Security Properties

- *Abuse-free* [Wang et al., 2005]: A protocol preserves the abuse-free property when it is impossible for a single entity, at any point during the protocol execution, to prove to an outside party, that she has the power to terminate (abort) or successfully complete the protocol. Since many protocols give participants the option to invoke aborting or recovery sub–protocols, a new abuse-free property has recently been defined. A fair protocol is *strongly abuse-free* if before the protocol ends, no party is able to prove to an outside party that his/her opponent is participating in the protocol.

- *Composability* [Meadows, 2003a]: Usually, a protocol is executed at the same time and in the same environment as many other protocols (simultaneous and interleaving sessions). In these kind of scenarios it must be ensured that a message or messages from one protocol could not be used to subvert other protocols' goals. That is, it is necessary to prove that no protocol in the collection will accept a message sent by another protocol in the collection.

- *Computational efficiency:* The protocol must be efficient given the computational and technological power of the different entities involved.

- *Correctness* [Bicakci and Baykal, 2003, Meadows, 2003b]: Traditionally, a protocol has been considered correct when, at completion time and under a predetermined set of hypothesis, a number of specific goals have been achieved such as, an entity being authenticated, a session key provided, or a message being sent in a confidential way.

- *Fairness* [Kremer et al., 2002, Nenadic et al., 2004]: Typically, an exchange protocol satisfies fairness when no participant can terminate the execution of the protocol in a disadvantageous situation with respect to other participants and, regarding the protocol exchanging goals. Several variants to this property exist:

  - **Computational fairness** [Kremer, 2003]: A protocol is computationally fair when, at any point during the protocol execution, if an entity quits early or misbehaves, for every participant there is the same amount of computational work involved in getting to an informational advantageous position.
  - **Probabilistic fairness** [Kremer, 2003]: A protocol satisfies this property when at any stage of the protocol, the probability of one of the entities having informational advantage is at most ϵ, a fixed predefined parameter.
  - **Strong fairness** [Franklin and Tsudik, 1998, Zhang et al., 2004]: Strong fairness is ensured when no participant entity can gain any informational advantage by quitting early or otherwise misbehaving during the protocol execution.
  - **True fairness** [Kremer et al., 2002]: True fairness is ensured when the protocol provides strong fairness and, if the exchange is successful, the non-repudiation evidences produced during the protocol are independent from how the protocol is executed (with or without the involvement of a trusted third party).
  - **Weak fairness** [Franklin and Tsudik, 1998, Kremer et al., 2002, Zhang et al., 2004]: Weak fairness is satisfied when a participant entity could terminate the execution of the protocol in a disadvantageous situation but at least, that entity can prove that other parties have misbehaved.

- *Message confidentiality:* This property is satisfied when messages exchanged during the protocol execution are such that no other entity, apart from the intended recipient, is able to disclose the content of such messages.

- *Message integrity:* This property is satisfied when messages exchanged during the protocol execution are such that no attacker is able to modify the content of such messages, without the intended recipient noticing the fraud.

- *Non–repudiation evidence generation and key revocation scheme:* Protocol security also depends on good management and good design of non–repudiation evidences. Deficiently designed non–repudiation evidences could result in successful attacks on sound protocols [Alcaide et al., 2005]. Moreover, it is necessary to be able to identify whether a signature, present in a non–repudiation evidence, was generated before or after revocation.

- *Non–repudiation of delivery* [ISO/IEC13888-3:1997], [Kremer et al., 2002]: Non–repudiation of delivery is intended to provide evidence that the recipient/s received the message.

- *Non–repudiation of origin* [ISO/IEC13888-3:1997], [Kremer et al., 2002, Onieva et al., 2003]: Non–repudiation of origin is intended to provide evidence that an entity is the originator and the sender of a given message.

- *Non–repudiation of submission* [ISO/IEC13888-3:1997]: In indirect communication a delivery agent is involved in transferring messages from an originator to one or more recipients. Non–repudiation of submission is intended to provide evidence that the originator submitted the message for delivery.

- *Protocol participant confidentiality* [Franklin and Tsudik, 1998, Kremer et al., 2002]: A security protocol satisfies this property when one or more participants of the protocol can be ensured confidential and anonymous participation.

- *Robustness* [Husdal, 2004]: Robustness means the ability to stay on course and to accommodate unforeseen events.

- *Secrecy* [Durgin et al., 1999]: A protocol is considered to satisfy the secrecy property when it ensures message confidentiality and integrity during simultaneous and interleaving sessions of the same protocol.

- *Timeliness* [Kremer et al., 2002]: A protocol is considered to satisfy timeliness when it satisfies some required security features during current execution and during all future instances of the same protocol. In other words, the protocol finishes for honest participants in a finite amount of time.

- *Viability* [Kremer and Raskin, 2000]: A protocol is viable if honest participants always succeed in exchanging the expected evidences.
- *Etc.*

Again, this is a list in constant expansion. Most frequently, the need to add extra security features to existing schemes stems from the latest attacks and threats on security services. In this regard, many authors have argued that only experience and time can give us a clear picture of what is needed to design secure protocols ([Meadows, 1994, Dennin, 1999]).

1.3 The Fair Exchange Problem

When focusing on a specific type of cryptographic protocol other problems of similar characteristics arise. For example, numerous cryptographic schemes are defined to provide entities with a mechanism to exchange items in a *fair* manner. Definition 1.3.1 gives a formal definition of this type of protocol.

Definition 1.3.1 (Fair-exchange security protocol)**.** *A fair-exchange security protocol is a cryptographic protocol allowing several parties to exchange commodities in such a way that, even when one or more entities deviate from the protocol description, none of the well-behaved entities will finish the protocol in a disadvantageous situation, that is, having sent their items and not receiving the appropriate items in return.*

Interest in this class of protocol stems from its importance in many services that rely on electronic transactions where disputes among parties can take place. Examples of these include digital contract signing, certified e-mail, exchange of digital goods and payment, etc. Moreover, assurance of fairness is fundamental when the exchanged items include any kind of evidence of non–repudiation, for this constitutes a key service in most of the previously mentioned applications. As a result, fair non–repudiation has experienced an explosion of proposals in recent years (see [Kremer et al., 2002] for an excellent survey).

Unfortunately, there is no established protocol by which a number of parties can exchange items in a fair manner, exclusively by themselves, and assuming that misbehaving parties participate in the protocol. Pagnia and Gärtner provide a formal treatment of this problem in [Pagnia and Gärtner, 1999]. The underlying idea can be intuitively sketched avoiding technical details: during the protocol execution, eventually one of the parties has to go first in providing her item to the other party. At that point, the first agent falls into in a unfair situation of which a misbehaving party can take advantage.

Therefore, the simplest protocol that can provide true fairness relies on the use of a trusted third party. The role of the TTP varies from one class of fair-exchange protocols to another according to its involvement. In schemes based on an *in-line* TTP (e.g. [Bahreman and Tygar, 1994]), this acts as a delivery authority which is involved in every message exchanged. The main drawback of these schemes is the heavy reliance on the involvement of the TTP, which can typically become a bottleneck. To avoid this inefficiency, some authors proposed the use of an *on-line* TTP (e.g. [Abadi et al., 2002]). Here, the TTP is involved during each protocol execution, but not necessarily in every message exchanged between parties. A third step towards reducing the role of the TTP was the introduction of *off-line* TTPs (see [Asokan et al., 1997, Asokan et al., 1998]). In these protocols –sometimes referred to as *optimistic fair exchange*–, parties try to carry out the exchange by themselves, and only appeal to the TTP in case of misbehavior of a dishonest party, or whenever a failure occurs during the protocol execution.

However, as mentioned before, recent computing paradigms (for example, ad hoc and peer–to–peer networks) pose a challenge from the point of view of the security mechanisms that should be applied as in many cases, the operation of these systems rely on a complete lack of fixed infrastructure. Generally, it is not realistic to assume that services such as those provided by a TTP will be available in these environments. In this context, the notion of *rational exchange* becomes especially interesting as in particular, rational–exchange protocols have the main advantage of not needing a trusted third party.

As for fair exchange protocols, the following formal definition will serve to unify understanding of this type of protocol.

Definition 1.3.2 (Rational–exchange security protocol)**.** *A rational–exchange security protocol is a cryptographic protocol allowing several parties to exchange commodities in such a way that, if one or more parties deviate from the protocol description, then they may bring other correctly behaving participants to a disadvantageous situation, but they cannot gain any advantages in doing so.*

In particular, this thesis will focus on the analysis and automated design of rational–exchange protocols.

1.4 Heuristic Search

An heuristic search is concerned with the finding of optimal solutions to *very difficult* problems. An heuristic approach is usually taken when deterministic algorithms cannot produce an answer to a given problem, for example, if their running times are non–polynomial or simply the number of inputs is so huge that a polynomial

algorithm would take too long to deliver the appropriate solution. This is the case in multi–party exchange scenarios, where the design space of cryptographic exchange protocols grows exponentially as the number of participants or the number of items increase. In these scenarios, an heuristic technique is sometimes able to produce optimal solutions within feasible and tractable computational settings. In fact, heuristic algorithms have already been successfully applied to the design of cryptographic protocols.

Some of the work in this thesis will focus on the automated design (by means of heuristic techniques) of rational–exchange security protocols.

1.5 Scope of Our Work

This thesis represents a contribution to the Rational Cryptography field. In particular, our work will focus on the analysis and design of rational–exchange cryptographic protocols. Some effort will go into extending and complementing the work in [Syverson, 1998] and [Buttyán, 2001], whilst other aspects will regard the application of non–standard computation to the automated design of multi–party rational–exchange cryptographic protocols.

We now explain in detail the objectives of this work.

1.6 Objectives

In the previous sections we have described what is a remarkably challenging scenario.

Globally, existing security schemes find it very difficult to satisfy the security demands of heavily decentralized and structure–less computing frameworks. Existing cryptography protocols cannot be applied to secure new computational environments, usually due to a shortage of available resources. In particular, *fair* exchange security protocols are in need of a replacement as the required presence of a TTP and the services it provides cannot be guaranteed in these environments.

Moreover, not only the design of new schemes presents difficulties but also the formal verification of these new protocols. Current formal analytical tools are of no use to validate the new security properties defined in recent cryptographic solutions.

Finally, multi–party scenarios make the design space of cryptographic exchange protocols grow exponentially to a scale difficult to explore through manual design methodologies.

All in all, Rational Cryptography combined with heuristic search seem to provide a suitable frame for the development of a different approach.

We have created this thesis within the framework just described and with very clear objectives in mind.

Objective 1:

To extend and enhance existing model based on Game Theory ([Buttyán, 2001]), for the formal analysis of rational–exchange security protocols.

Game Theory has already been identified as a suitable tool for the formal analysis of rational–exchange protocols ([Buttyán, 2001]). In this work Buttyán defines a formal analytical model, based on Game Theory, for the analysis of rational–exchange protocols. The formalism defines rationality in terms of the Nash equilibria found in a game constructed from the protocol description. The study is sound and correct but it is also limiting in terms of scalability and adaptability to the analysis of rational–exchange protocols in different execution environments. In our opinion, despite the effort made by Buttyán and despite the similarities with fair exchange, rational exchange still poses a challenge regarding its formal verification.

Our new formalism will extend Buttyán et al.'s work in two main areas:

- It will allow for some protocol contextual information to be taken into consideration to formally analyze a rational–exchange protocol. Factors such as participant reputation, protocol robustness or network reliability, which are usually determinant of the outcome of a rational–exchange protocol, will be represented within the analytical framework.
- Additionally, the analytical model will not impose any restrictions on participant's capabilities to misbehave or deviate form a protocol description. By contrast, the formalism will allow us to easily represent any participant unpredictable behavior.

Two different Game Theory concepts will be applied to extend Buttyán's model:

- *Games of imperfect information and Nash equilibrium perfect in sub-games.*
- *Games of incomplete information* or *Bayesian games and perfect Bayesian equilibrium.*

Objective 2:

The definition of a new methodology to incorporate validation and contextual information into the design of rational–exchange security protocols.

Some authors have already suggested integrating, in one single process, protocol analysis and design ([Meadows, 2003b, Kremer, 2003]). In our opinion not only analysis and design need to be compounded but also, the real scenarios where the protocols are going to be implemented and executed. As experience has shown,

a protocol cannot be safely taken out of a given environment and applied in a completely different setting.
The new methodology that we propose will have the following characteristics:

- The definition of a set of linear structures will allow us to represent any given exchange problem.
- Contextual information such us the capability of coalition between participants in the execution environment and whether participant entities are part of an incentive scheme, will be easily parameterized within such set of structures.
- Intrinsic to the design methodology will be a mathematical reasoning based on Game Theory for the synthesis of provably rational–exchange cryptographic protocols.

Objective 3:

The definition of a process based on heuristic techniques
for the automated design of multi–party rational–exchange security protocols.

The number of possible ways in which various entities can exchange a series of items can grow exponentially as the number of items or the number of participants increases. In this scenario, we will define a meta–heuristic search technique, for the automated exploration of large rational protocol design spaces, far greater than could be considered using manual design.
Our technique will offer the following features:

- The process will make use of the formalism described as Objective 2 to explore the space of rational–exchange solutions for a given multi–party exchange problem.
- The process will be highly scalable, versatile and it will be based on meta–heuristic Simulated Annealing algorithm.

1.7 Organization

The document is divided in three parts.

1.7.1 Part I: Game Theoretical Analysis of Rational–Exchange Protocols

Four chapters (Chapters 2, 3, 4 and 5) constitute the second part of this thesis:

Chapter 2: Syverson's Rational–Exchange Protocol and Buttyán et al.'s Game–Theoretical Model

Buttyán et al. introduced in [Buttyán and Hubaux, 2004] a mathematical model based on Game Theory under which rationality can be formally defined and properties of rational–exchange protocols can be analyzed. As an example, the model was used to analyze Syverson's rational–exchange protocol [Syverson, 1998]. In this chapter we present and examine both, Buttyán et al.'s model and Syverson's protocol. We also carry out a cryptanalysis of Syverson's scheme and propose an enhanced version.

Chapter 3: A Model based on Dynamic Games of Imperfect Information

In this chapter, we present a contribution consisting of extending Buttyán et al.'s model to capture relevant aspects involved in the execution of a rational–exchange protocol. We base our extension on games of imperfect information.

Chapter 4: A Model based on Bayesian Games

In this chapter, a new extension to Buttyán et al.'s work is described. This time, our formalism is based on Bayesian games.

Chapter 5: Bayesian Analysis of a Secure P2P Content Distribution Protocol

In this chapter we apply our model based on Bayesian games to the analysis of a secure content P2P distribution protocol. The protocol is proven to be a rational protocol for which it is possible to predict all possible outcomes.

1.7.2 Part II: Automated Design of Multi–party Rational–Exchange Security (M–RES) Protocols

Three chapters (Chapters 6, 7 and 8) make up this part:

Chapter 6: Introduction to the Automated Synthesis of Cryptographic Protocols

The aim of this chapter is to give the reader a basic understanding of what an heuristic search technique is. Also in this chapter, we will briefly describe existing work on applying heuristic search to the automated synthesis of security protocols.

Chapter 7: Foundations for the Automated Synthesis of M–RES Protocols

In this chapter, we describe the formal foundations for the automated synthesis of multi–party rational–exchange security protocols (M–RES protocols). We define the appropriate structures to describe any given multi–party exchange problem in need of a rational solution. Furthermore, a taxonomy is provided which will help in identifying the type of problem we are encountering and, what type of solution the synthesis process is going to produce.

Chapter 8: Heuristic Synthesis of v–RES Protocols

This chapter serves to apply the formalism described in Chapter 7 to the parametrization of a particular three entity rational exchange problem. We also define an heuristic search technique, based on Simulated Annealing, for the synthesis of rational–exchange solutions. Finally, we provide the formal analysis of a family of M–RES protocols. The analysis is based on Game Theory using the formal model described in Part I.

Chapter 9: Solving More Complex Problems

In this chapter we apply the techniques and automated tools for the synthesis of M–RES solutions for more complex exchanging problems. As a result of the experimentation a series of rational solutions are presented, which serve to give solution to several multi–party randomized exchange problems. For all these protocols a formal proof of rationality exists, based on Game Theoretical concepts.

1.7.3 Part III: List of Contributions, Conclusions and Future Work

Chapter 10: Summary and List of Contributions

In this chapter we present a summary of the main contributions obtained in the process of developing this thesis, a summary of the main conclusions and open issues and future work in relation to all previous chapters. Finally, the author presents a list of publications containing parts of this thesis.

Appendix A: Principles on Game Theory

This appendix offers a basic introduction to Game Theory principles.

Part I

Game Theoretical Analysis of Rational–Exchange Protocols

Chapter 2

Syverson's Rational–Exchange Protocol and Buttyán et al.'s Game–Theoretical Model

2.1 Introduction

In 1998, P. Syverson introduced the idea of *rational exchange* as an alternative to *fair exchange* for those scenarios where the use of a TTP (trusted third party) is not allowed or not feasible ([Syverson, 1998]). Informally, a rational–exchange protocol cannot provide fairness but it ensures that rational (i.e. self-interested) parties would have no reason to deviate from the protocol, as misbehaving does not result beneficial.

Moreover, in 2001 Buttyán et al. identified Game Theory principles as a powerful and suitable tool to formalize rationality in exchange protocols, in particular in Syverson's rational–exchange protocol ([Buttyán and Hubaux, 2001], [Buttyán, 2001]).

In this chapter, we will describe Syverson's scheme and introduce Buttyán et al.'s analytical model. We will detail how the model was applied to analyze Syverson's protocol highlighting the strengths and weaknesses of such a formalism. Besides, the protocol, as it was first described by its author, presents some significant vulnerabilities and several attacks can be successfully mounted against the scheme, showing how protocol participants can end up in undesired situations. In the next sections, we will describe the aforementioned attacks and suggest how to fix the scheme. Finally, a formal verification of the enhanced version will be specified using BAN logic as a formal tool ([Burrows et al., 1990]).

2.1.1 Chapter Organization

The chapter is organized as follows. In Sections 2.2 and 2.3 we introduce Syverson's scheme and Buttyán et al.'s model respectively. In Section 2.4, we describe some flaws present in the protocol and describe how they can be exploited to mount three different attacks. Section 2.4.5 is devoted to explain how the protocol can be fixed in order to eliminate previous vulnerabilities. Section 2.5 serves to present some concepts used to give formal proof of the enhanced scheme. Section 2.6 describes weaknesses and limitations of the formal model defined by Buttyán et al. Finally, we conclude the chapter with Sections 2.7 and 2.8 in which we briefly describe some results on the complexity of Nash equilibria computation and, summarize the main conclusions of this chapter, respectively.

2.2 Syverson's Protocol Description

The scheme presented by Syverson in 1998 is illustrated in Fig. 2.1. It consists of three messages exchanged between two different entities. Next are the main components and the adopted notation:

- A and B denote the two protocol parties, with private keys k_A^{-1} and k_B^{-1}, respectively.
- We assume that $item_A$ and $item_B$ are the items to be exchanged, being $desc_{item_A}$ a description of $item_A$. (There is no equivalent description for $item_B$ because the scheme was introduced to serve as a payment protocol, in such a way that $item_B$ has the role of the payment for buying $item_A$).
- Moreover, $E_k(m)$ is a symmetric encryption algorithm that encrypts message m with key k.
- Likewise, $sig(k_i^{-1}, m)$ provides a digital signature on m using private key k_i^{-1}. All messages exchanged are cryptographically signed by the corresponding sender.
- Finally, $w(\cdot)$ is a WSBC (Weakly Secret Bit Commitment) function [Syverson, 1998]. For our analysis, it suffices to know that $w(x)$ keeps x secret, but it can be broken in acceptable bounds on time with reasonable resources.

In step one of the protocol, A sends B $item_A$ in a encrypted form. Next, B sends A $item_B$ in return, along with acknowledgement of the first message. Finally, A sends the appropriate key k and acknowledgement of the second message.

$$
\begin{array}{ll}
A \rightarrow B: & m_1 = (desc_{item_A}, E_k(item_A), w(k), \sigma_1) \\
B \rightarrow A: & m_2 = (item_B, m_1, \sigma_2) \\
A \rightarrow B: & m_3 = (k, m_2, \sigma_3)
\end{array}
$$

where:

$$
\begin{aligned}
\sigma_1 &= sig(k_A^{-1}, (desc_{item_A}, E_k(item_A), w(k))) \\
\sigma_2 &= sig(k_B^{-1}, (item_B, m_1)) \\
\sigma_3 &= sig(k_A^{-1}, (k, m_2))
\end{aligned}
$$

Figure 2.1: Syverson's rational–exchange protocol.

We now proceed to analyze some aspects of the protocol just described. These will help in understanding the type of scenarios where the protocol is suitable to use, as Syverson's protocol is not always appropriate.

- Note that after message m_1, B cannot access encrypted $item_A$ unless it first accesses the encryption key k. For this, B would need to break $w(k)$.
- Therefore, B can only disclose the encrypted $item_A$ if m_2 has been sent (this is, payment has gone through) and A has responded with message m_3. As a result, A could send a forged $item_A$ and still receive payment in return.
- Also note that, at step three, A might fail to send message m_3 or she might not send it for a long time.

The first deterrent against A delaying sending message m_3 is that A gains nothing by doing so, except a bad reputation that could ruin her business. In the case of A sending B the wrong $item_A$, B holds message m_3 as a proof of such misbehavior. However, an important issue arises from both of the previous statements: *during the protocol execution both participants must exchange irrevocable evidences to be able to prove each other's misbehavior.* For example, an scheme on entity A's reputation can only be implemented when it is not possible for B to accuse A of misbehaving if A was honest, and vice versa. A fourth message could be added in which customer B acknowledges timely receipt for message m_3. Likewise, for B to be able to prove in front of an external judge that A sent an invalid $item_A$, B must hold irrevocable proof of such a message.

Given the observations above, the author of the protocol identifies scenarios where the scheme could be used for:

1. If the vendor A is selling relatively low value items, so it is not worth it for the customer (in terms of computational cost or the inconvenience of delay) to break the encryption to recover the item;

2. The vendor A might be selling something that might be of timely and diminishing value, such as short term investment advice or regularly changing lists of bargain items for sale; or

3. The protocol might begin one step earlier with a signed customer request for $item_A$. The vendor A can then take the chance of trading with unknown customers and refuse to service customers who repeatedly fail to pay.

2.3 Buttyán et al.'s Formal Model

Syverson's rational–exchange protocol was analyzed by Buttyán et al. in [Buttyán, 2001]. For readability and completeness, we first provide a summary of the game–theoretical model proposed by Buttyán et al. Please refer to [Buttyán, 2001], [Buttyán and Hubaux, 2001], [Buttyán et al., 2002] and [Buttyán and Hubaux, 2004] for further details.

Where possible, we have adopted the same notation. Also at this point, the reader should already be familiar with some of the concepts in Game Theory provided in Sections A.1 and A.4.

2.3.1 Protocol Games

Buttyán et al. introduced the concept of *protocol game* as a way to represent an exchange protocol. The protocol game of an exchange protocol is intended to model all possible interactions of the protocol participants, even the potentially misbehaving actions (i.e., those different from the ones prescribed by the protocol).

A protocol game is constructed from the protocol description as follows:

- Each of the parties involved in the protocol, including the network, becomes a player of the protocol game. A different set of strategies is associated to each different player.

  From this point onwards and throughout this document, we will refer to protocol participants and players indistinctly.

- The network is considered to be reliable, which means that it correctly delivers messages to their intended destinations within a constant time interval. Therefore, the network has only one fixed strategy consisting of delivering messages to players.

- The rest of the participants have the strategies to *quit*, *do nothing*, *send a message* following the steps described in the protocol or *send a message deviating* from the protocol description.

- Each player can send messages which have been defined as *compatible* with the protocol, this is, messages which are within the context of the protocol. The set M_π of messages compatible with a protocol π is formally defined within the model. Although the participants can alter the order in which those messages are sent, the model does not allow the protocol parties to run multiple instances of the protocol in parallel (i.e., they do not consider interleaving attacks). Furthermore, neither eavesdropping nor message manipulation are considered in the model.

- *Information sets* for players (information available to players at each step in the protocol game) are defined in terms of their local state. Buttyán et al. formally define both, the structure of such sets as well as the way they are updated according to the actions observed during the game. These sets are defined as singletons, so all protocol games constructed following Buttyán et al.'s formalism are of *perfect* information.

- Finally, a payoff function $y_i(\cdot)$, is established for every player. In particular, the model considers two players P_1 and P_2 and two items to be exchanged denoted by γ_{P_1} and γ_{P_2}. What γ_{P_1} is worth to P_1 and P_2 is denoted by $u^-_{P_1}$ and $u^+_{P_2}$, respectively. Likewise, what γ_{P_2} is worth to P_1 and P_2 is denoted by $u^+_{P_1}$ and $u^-_{P_2}$, respectively. In this way, the values u^+_i and u^-_i can be viewed as the potential gain and loss of player $i \in \{P_1, P_2\}$ in the game.

  When the protocol game is over, every participant can assess the profit or the loss they have incurred by using this payoff function. The function takes the local state of every participant at the time the game is over and calculates an outcome value (the highest profit represents the most preferable protocol outcome). Buttyán et al. introduce this concept in their model as follows. Given a terminal sequence of actions q, the payoff function for player i is defined as $y_i(q) = y^+_i(q) - y^-_i(q)$, where functions $y^+_i(q)$ and $y^-_i(q)$ represent the gain and the loss player i has incurred, respectively. In general, these functions can be defined as follows:

$$y^{\oplus}_i(q) = \begin{cases} u^{\oplus}_i & \text{if } \phi^{\oplus}_i(q) = \texttt{true} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

  where $\oplus \in \{+, -\}$. The purpose of boolean functions $\phi^{\oplus}_i(q)$ is to capture those conditions under which each partner gains/losses control over the items. Thus, $\phi^+_i(q) = \texttt{true} \Leftrightarrow$ player i gains access to γ_j $(i \neq j)$, and $\phi^-_i(q) = \texttt{true} \Leftrightarrow$ player i losses control over γ_i, where $i, j \in \{P_1, P_2\}$.

2.3.2 Security Properties

Informally, a two–party rational–exchange protocol is an exchange protocol in which both parties are motivated to follow the protocol faithfully as misbehaving does not result beneficial. If one of the parties deviates from the protocol, then she may bring the other correctly behaving party to a disadvantageous situation, but she cannot gain any advantages in doing so.

Buttyán et al. define the concept of *rationality* in terms of a Nash equilibrium of the protocol game. It is required that the strategies that correspond to the behavior described by the protocol form a Nash equilibrium of the protocol game and that no other Nash equilibrium is strongly preferable by any other participant.

Other properties such as fairness, effectiveness, termination, gains closed, and safe back–out are also formally defined within the model. In particular, the proof of the protocol rationality relies on the fact that the protocol must be:

- Closed for gains: The closed for gains property is satisfied when for every possible outcome of the game q, it holds that $y_A^+(q) \geqslant 0 \Rightarrow y_B^-(q) \geqslant 0$ and $y_B^+(q) \geqslant 0 \Rightarrow y_A^-(q) \geqslant 0$. Put simply, this property establishes that if a party A gains access to an item belonging to the other party B, then B must lose control over the same item and vice versa.

- It must also satisfy the safe back–out property: The safe back–out property is satisfied when for every possible sequence of actions q, if party A's strategy was *always quit*, then A loses nothing by following sych a strategy (i.e. $y_A^-(q) = 0$). In the same way, if B's strategy is to *always quit*, then $y_B^-(q) = 0$.

In our opinion the model is consistent and correct, even though, as we will see, it is easy to step out of it and break that way the rationality property. It is also possible to break the closed for gains property to attack rationality.

2.3.3 Syverson's Protocol within Buttyán et al.'s Model

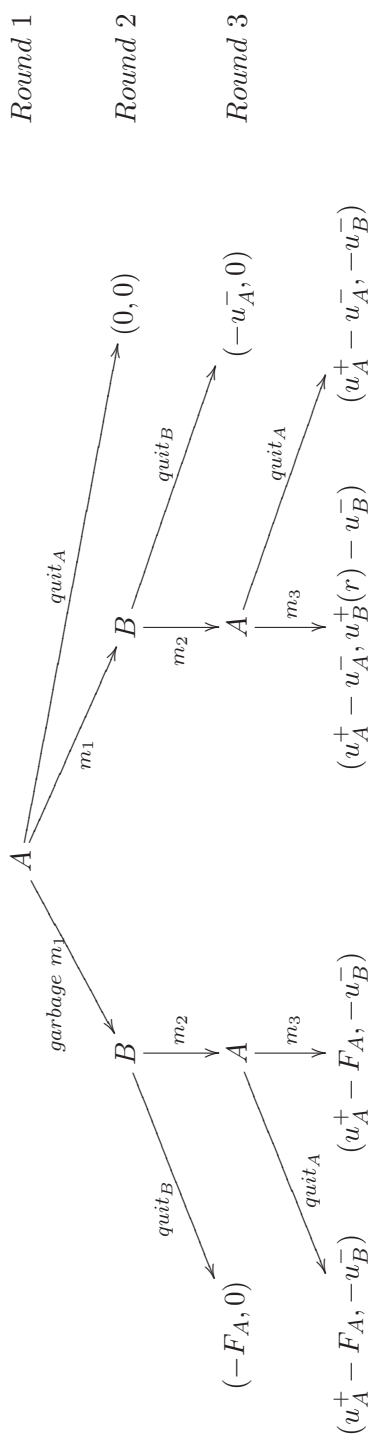


Figure 2.2: Representation of Syverson's protocol game in extensive form.

The protocol game derived by Buttyán et al. from Syverson'protocol description is shown in Fig. 2.2. The protocol game is represented in *extensive–form* (i.e. a tree) showing all possible moves each participant can make and all the different outcomes.

The vectors assigned to each terminal node represent the values of the payoff function (also called *utility function*) for A (first value) and B (second value) when A and B follow the path of strategies to finish the protocol at that end. The payoff function for each participant can take different values defined by Buttyán et al. as follows: u_A^+ and u_B^+ reference how much $item_B$ and $item_A$ are worth to parties A and B respectively. In a similar way, the values u_A^- and u_B^- denote the values that $item_A$ and $item_B$ are worth to A and B, respectively.

The total payoff values assigned at each terminal node are computed as follows:

- Entity A obtains a positive value u_A^+ in all tree branches in which entity A gains access to $item_B$. By contrast, entity A obtains a negative value u_A^- if along the branch, A losses control over message m_1 .

- The value of F_A represents the penalty A has to pay when sending a forged message m_1 (denoted as *garbage* m_1). Although entity B can end the protocol holding a fake $item_A$ and, this could only be discovered at step three of the game (once payment has gone through), penalty F_A serves to deter such malicious behavior.

- In a similar way, entity B obtains a positive value $u_B^+(r)$ in all tree branches in which entity B gains access to $item_A$. Factor r specifies the round number in the game. Moreover, entity B obtains a negative value u_B^- if along the path, B losses control of message m_2.

- Besides, the payoff values are defined to satisfy the following relations: $u_A^- < u_A^+ < F_A$ and $u_B^- < u_B^+(r)$.

- Finally, there is not penalty for entity B's misbehavior as this can be detected by entity A at execution time. In this case, entity A would quit the protocol which does not result beneficial for entity B.

Buttyán et al. established that the path in tree which would result in an exchange of items $item_A$ and $item_B$ constitutes a Nash Equilibrium so, by definition, neither of the players would want to deviate from it. Therefore, any two rational parties following Syverson's protocol description would terminate the protocol having exchanged their items.

In this regard, Buttyán's model serves to:

1. Formally define rationality, and

2. Prove that Syverson's protocol is a rational–exchange protocol.

2.4 Cryptanalysis of Syverson's Protocol

In this section we will show how Syverson's protocol, as originally described by its author, presents some vulnerabilities not detected by Buttyán et al.'s formalism and how some attacks on the scheme can be successfully mounted. Additionally, at the end of this section we will provide an enhancement to the original protocol which we will formally prove correct in a subsequent analysis.

2.4.1 Observations

All messages involved in the protocol are cryptographically signed (see Fig. 2.1). Since such cryptographic algorithms are assumed to be unbreakable, the primary focus for attackers or penetrators is on the possibility to reuse messages, even when they are not able to read them or to produce them by themselves. By replaying old messages, dishonest parties can impersonate other entities, mislead other participant's actions or obtain confidential information. We will see how it is possible in some instances, to reuse messages during a Syverson's protocol execution.

The following observations will help in the understanding of the overall cryptanalysis:

1. Message m_1 could be used as a proof of A's misbehavior. Indeed, due to the nature of $w(k)$, if A randomly generates a ciphertext ϵ to include in m_1, A can be penalized whenever the commitment $w(k)$ is broken and k disclosed. Therefore, this kind of misbehavior can always be proven to a judge. In this regard and from B's point of view, the protocol provides a sort of weak fairness. However, also note that m_1 ensures B that A is the author of such a message, but it does not guarantee that A is also the sender of such a message. It is not until step 3 of the protocol that B holds a valid NRO (Non–Repudiation of Origin) token for $item_A$. Therefore, message m_1 could be used to prove that A once generated a forged message, but m_1 cannot be used to prove that A is actually the sender of such a message in the current instance of the protocol.

2. Message m_2 could serve as a NRR (Non–Repudiation of Receipt) token for message m_1 as well as a NRO of $item_B$. B's signature on message m_2 ensures A that B received $item_A$ and that B has proceeded with the sending of $item_B$. Message m_2 could always be used as a proof of B's misbehavior in the protocol.

3. Message m_3 could serve as a NRR token for message m_2 as well as a NRO of $item_A$. A's signature on message m_3 ensures B that A received $item_B$ and that A has generated and sent m_1 with the correct key. Message m_3 could always be used as a proof of A's misbehavior in the protocol. A might not send the third message, or not do it for a long time, but A gains nothing by doing that apart from a poor reputation that could damage her business. The context in which to execute this protocol should then be a regularly repeated scenario.

The protocol, therefore, when rationally executed could provide rational exchange of non–repudiation evidences. However, the non–repudiation evidences would have to be linked to each particular protocol run to serve the purposes of non–repudiation in future disputes. Since A is asked to generate a fresh key k for each run of the protocol, k could be the unique label to reference each different protocol run and the corresponding evidences. In particular, notice how, although it is possible to determine whether m_2 and m_3 are fresh, this is not the case for m_1 unless some enhancements are made.

Next, we describe three possible attacks on the original scheme that illustrate the aforementioned observations and which motivated us to propose an enhanced version of the protocol.

2.4.2 Attack 1

Consider the following scenario, where $P(Q)$ means that party P acts impersonating the role of party Q:

$$\begin{array}{rcll} A & \rightarrow & B & : m_1 = (desc_{item_A}, E_k(item_A), w(k), \sigma_1) \\ B(A) & \rightarrow & C & : m_1 = (desc_{item_A}, E_k(item_A), w(k), \sigma_1) \\ C & \rightarrow & B(A) & : m_2 = (item_C, m_1, \sigma_2) \end{array}$$

This attack is based on B impersonating A, sending the same message m_1 to C and receiving $item_C$ in return. B would have to quit the protocol after receiving the payment as she has no key to send to C. Although C has paid a full price for $item_A$, by the time that k is disclosed to C, $item_A$ would be of very little value to C. The customer C could only present message m_1 to prove A misbehaved. However, A will claim that m_1 was never intended for C and that she was not part of such a communication. Indeed, there is nothing in m_1 linking A and C as participants on the same protocol run. To overcome this attack, new restrictions would have to be placed over the communicating network or amendments should be made to the structure of m_1.

2.4.3 Attack 2

Let us suppose the following simplistic scenario: A is selling an access code to enable the viewing of a football match on a private television network. Let us suppose that A and B carried out a successful Syverson's protocol execution and that they properly exchanged the encrypted access code $E_k(item_A)$, $item_B$ and the corresponding key k in messages m_{11}, m_{12}, and m_{13}, respectively. The access code that B has bought from A is obviously of timely diminishing value, but B could still have time to impersonate A and sell the access code to other customers, receiving payment in return:

$$\begin{array}{rcll} B(A) & \rightarrow & C & : m_{21} = m_{11} = (desc_{item_A}, E_k(item_A), w(k), \sigma_A) \\ C & \rightarrow & B(A) & : m_{22} = (payment_C, m_{21}, \sigma_C) \\ B(A) & \rightarrow & C & : m_{23} = m_{13} = (k, m_{12}, \sigma_A) \end{array}$$

In this scenario, by the time C receives message three and realizes that there is a fraud going on, C has no evidence of such a fraud to present in front of a judge and has got the key k to decrypt the football match access code and watch the match. However, A could claim that C is watching a program without a license and take action against her. If the number of reselling codes is large, the scale of the fraud would make it impractical to pursue each of the individuals watching the match without license. Furthermore, trying to trail back the origin of such messages would be practically impossible. Again, the nature of the communicating network would have to change or the content of the first message amended.

2.4.4 Attack 3

If a vendor sends the customer a message m_1 containing garbage (i.e, a ciphertext which does not correspond with the actual $item_A$), the vendor is indeed providing the customer with evidence of such a form of cheating. Message m_1 could be presented to a judge and the vendor would be charged with the appropriate penalty. Such a penalty could greatly exceed the value of the goods, so the vendor is completely discouraged from performing such a scheme. However, the vendor could not be sued and penalized twice for the same offence and, on these terms, a vendor A could carry on sending the forged message m_1 to many others customers, receiving payments in return. These new angry customers would only have message m_1 to blame vendor A. Vendor A would claim that she never sent m_1 to them and that they must have got it from the first resentful customer. As a matter of fact, there will be nothing in m_1 to prove that A is using the same forged message all over again. A's reputation would therefore stay untouched.

2.4.5 Fixing the Protocol

Even though the replay attacks one to three described in the previous section correspond to simple deviations from the protocol description, they represent real threats to parties using the scheme to exchange their items. In e-commerce transactions, neither vendor A nor customer B would want to take the risk of being cheated.

However, previous weaknesses can be avoided if a better cryptographic evidence is constructed. This can be done in many ways. Probably the easiest one is just by including the identity of B in m_1, thus linking the message with its intended receiver. Since A is asked to generate a fresh key k for each protocol run, the tuple (k,A,B)[1] could be the unique label to associate each m_1 with the corresponding protocol execution:

$$A \rightarrow B : m_1 = (\mathbf{B}, desc_{item_A}, E_k(item_A), w(k), \sigma_1) \tag{2.2}$$

where:

$$\sigma_1 = sig(k_A^{-1}, (\mathbf{B}, desc_{item_A}, E_k(item_A), w(k))) \tag{2.3}$$

Note how this modification suffices to prevent attacks one to three. Now, in attack one, an entity C would have sent a payment to a false entity A. With the new structure of message m_1, C would know that m_1 is newly formulated by A (since A is asked to create a fresh key k for each instance of the protocol) and that C is the intended recipient. Therefore, A could not claim that it was not part of the protocol run.

In a similar way, this also prevents attack two, for entity B can establish whether the other participant is able to provide key k in the last message of the protocol. Attack three is also easy to prevent, as entity B can tell if the message is an old message that entity A is trying to replay in a new protocol run.

Next is the formalization of all concepts described in this section.

2.5 Formal Analysis of the Enhanced Version

Proof of our new scheme's correctness will be based on establishing the freshness of messages m_1, m_2, and m_3. Therefore, any type of replay attack with messages from outside the current execution (old replayed messages) will automatically be rejected, in particular attacks one to three. *Interleaving attacks* (a type of replay attack occurring when two instances of the same protocol are running simultaneously) are not being considered in our analysis as Syverson's protocol participants are assumed

[1] We assume that A's identity is implicit in m_1, since the message contains A's signature.

to run only one instance of the protocol at the time. No other form of attack is considered, as messages one to three are digitally signed by algorithms which are assumed to be cryptographically secure.

Below, only those steps of the formal process which are relevant to our enhancement are explicitly shown. Notice how this formal proof could not have been performed on the original protocol as freshness of message m_1 was not guaranteed.

2.5.1 Preliminaries

In this section, we briefly introduce some concepts that will be used throughout our analysis of the scheme proposed by Syverson. In particular, we will outline a well known formalism used to analyze security protocols (BAN logic) and we will use it to clarify the protocol's main security properties.

A Brief Overview of BAN Logic

Burrows, Abadi and Needham made a significant effort in 1989 defining a logic for the analysis of security protocols [Burrows et al., 1990]. BAN logic is a *logic of beliefs*. An inference process develops from a set of initial beliefs to a set of final goals for each protocol participant. Inference rules are defined as part of the logic.

Next, we introduce a few concepts and two of the BAN logic inference rules, which will be enough for the proofs presented in this chapter.

- ***BAN Notation:***
  - $\sharp(M)$: Formula M is fresh, that is, M has not been sent in a message at any time before the current run of the protocol.
  - $P \models M$: Entity P believes M, i.e.: P may act as if M is true.
  - $P \mid\sim M$: P once said M.
- ***BAN Inference Rules:***
  1. Freshness Verification Rule (FVR): This rule expresses that if a message is fresh, then the originator of such a message still believes in it:

$$\frac{P \models \sharp(M), \quad P \models Q \mid\sim M}{P \models Q \models M} \tag{2.4}$$

  2. Encrypted Freshness Verification Rule (EFVR): If a message or part of a message is known to be fresh, then the encrypted message must also be fresh.

$$\frac{\sharp(M)}{\sharp(\{M\}_K)} \tag{2.5}$$

Given an entity P and a message M, the statement "P *said* M" ($P \mid\!\sim M$) implies entity P having said or sent message M at some point in the past. By contrast, the statement "P *believes* M" ($P \mid\!\equiv M$) implies entity P to have said or sent M during the current protocol execution –typically taken from the initial point of the protocol run–, so M is *fresh* and A still believes in M. This distinction is crucial for our analysis.

2.5.2 Freshness of Messages and Replay Attacks

Replay attacks consist of the capture of a message –or a piece of a message– that is used at a later time, and probably with a different semantics. *Freshness* of messages is a common and relevant element in security-related protocols, in particular because of its importance as a mechanism to prevent replay attacks. Within the context of a protocol, freshness of a message will guarantee such a message belongs to that specific protocol instance and that it has never been used before in any other instances.

Linking a message to a particular protocol run is commonly obtained by the use of Timestamps in messages and Timestamping Certification Authorities. Other methods are also implemented, as the use of nonces (randomly created identifiers generated fresh by a participant for each protocol instance [Needham and Schroeder, 1978]), counter values, numbers provided by synchronized pseudo-random number generators, or fresh encryption. See [Gong, 1993] for a detailed description of each of them. However, message replay can take place in many different forms (see [Syverson, 1994] for a full classification and taxonomy) and usually more than one of these mechanisms has to be implemented to prevent the protocol from one or another form of replay attack. Freshness of messages is therefore a difficult and very important matter. In any given protocol, the recipient entity of any message should be able to determine whether the message received is fresh. Particularly, our cryptanalysis of Syverson's protocol is based on the impossibility for entity B to determine freshness of message m_1.

Freshness of m_1

When entity B receives m_1, B knows A's public key and is able to verify A's signature on m_1. Once B verifies the signature, B can be sure that the originator of that message was entity A. In BAN logic notation, we would express:

$$B \models A \mid\sim m_1 \tag{2.6}$$

Furthermore, B can see their name as part of the message so B is convinced she is the intended recipient. The item $desc_{item_A}$ serves B to identify m_1 as unique. Entity A has signed a message where $item_A$ has been encrypted and B is the intended recipient. Entity B believes this message could not have been used in any other instances of the protocol of which she was not part. Therefore, the combined tuple $(B, desc_{item_A})$, which is part of message m_1, is fresh: $B \models \sharp(B, desc_{item_A})$. Then, the following formula can be inferred applying the EFVR:

$$B \models \sharp(m_1) \tag{2.7}$$

Note that if B was buying the same $item_A$ twice, then entity B would have to verify that the two message components, B and $E_k(item_A)$, were never bound together in any of the previous instances. Recall that entity A is forced to generate a new key k for each new run.

Therefore, in any given case, applying FVR to formulæ (2.6) and (2.7) we obtain:

$$B \models A \models m_1 \tag{2.8}$$

which ensures freshness of m_1.

Freshness of m_2

When entity A receives m_2, A knows B's public key and is able to verify B's signature on m_2. Once A verifies the signature, A can be sure that the originator of that message was entity B. In BAN logic notation we would express:

$$A \models B \mid\sim m_2 \tag{2.9}$$

Moreover, A can see message m_1 as part of message m_2. Entity A generated m_1 as step one of the protocol so A believes m_2 is fresh as it could not have been generated in any other previous instances of the protocol. In BAN logic notation we have:

$$A \models \sharp(m_2) \tag{2.10}$$

Now, applying FVR to formulæ (2.9) and (2.10), we obtain a proof of freshness for m_2:

$$A \models B \models m_2 \tag{2.11}$$

Freshness of m_3

This part of the formal verification is exactly the same as for the freshness of message m_2. Therefore, mirroring the previous steps, we can conclude that m_3 is also fresh:

$$B \models A \models m_3 \tag{2.12}$$

2.6 Weaknesses of Buttyán et al.'s Model

In our opinion, Buttyán et al.'s model presents as major contribution the identification of Game Theory as an appropriate framework in which rational–exchange protocols can be formally analyzed. However, we are able to identify a few drawbacks in its definition which lead to a very restricted model.

In the following sections, we will briefly describe in more detail some of those limitations. The first two observations in Section 2.6.1 are related to the way participant actions and behavior are modeled, i.e. the computational model according to which messages are analyzed. Furthermore, in Section 2.6.2 we find basic Game Theory as a too narrow formalism for real-world protocols. In particular, we strongly believe that uncertainty plays a major role in rational exchange and, therefore, it should be somehow incorporated into the reasoning model.

2.6.1 Flaws in Local Computations and History Records

In Section 2.4, we described a number of vulnerabilities in Syverson's protocol allowing several attacks against the scheme. The reason why these flaws were not detected by Buttyán et al.'s analysis is not related to the reasoning model itself (i.e. Game Theory), but to a inherent hypothesis: the model assumes that messages are well constructed from a security point of view. In other words, in the process of formalizing participant local actions, the logic applied is limited to message compatibility and not message content. We further elaborate on this topic in what follows.

In Buttyán et al.'s model, an exchange protocol is considered to describe a set of local computations Π_j, one for each participant j of the protocol. Typically, each program Π_j contains instructions to wait for messages that satisfy certain conditions or to generate events such as to send a message m to a given participant p. In particular, when the program Π_B is described for entity B, no test is defined to determine whether m_1 is an old message being reused. In this regard, the model assumes that evidences are fresh and well constructed, thus failing to reflect the actual content of message m_1 as it is described in the protocol. This erroneous assumption is subsequently reflected in the protocol game. The result is that

the formal notation and structures used do not allow entities to verify essential properties such as freshness of a message, originator, sender or intended receiver.

Furthermore, it is specified that each player creates a history record of all the events that were generated by her and the round number of their generation. Possible entries in the history record file of protocol participant A would be $\mathsf{send}(m_1, B)$ or $\mathsf{rcv}(m_2)$, in round r. Based on the entries stored in this record, each player is either allowed or not allowed to send a particular compatible message. For instance, a valid digital signature $sig(k_A^{-1}, m)$ can only be generated by A. Therefore, B can send a message containing $sig(k_A^{-1}, m)$ iff B received a message containing $sig(k_A^{-1}, m)$ *earlier* in the current protocol execution.

As the model was defined, this history record is newly created for each protocol run, so information received in previous runs is discarded at the end of each execution. However, any (malicious) participant of the protocol could have compatible messages from previous runs and will be able to use them, as they are perfectly compatible with the protocol. In particular, attacks 1 and 2 described in Section 2.4 are based on the use of evidences obtained in different protocol runs.
Summarizing, in this first section we have identified two aspects of the model which can induce to erroneous protocol analysis.

- First, the construction of the protocol game must take into account the fact that messages could not be well constructed. For this, a number of additional tests on messages have to be carried out, beyond those aimed at ensuring compatibility with the current protocol round.

- Second, these security checks must include messages received in previous runs, and not only those corresponding to the current protocol execution.

However, improving the model with respect to the above aspects is out of the scope of this thesis. For our purposes, we will use the corrected version of Syverson's protocol described in Section 2.4.5.

Other aspects represent further limitations to the formalism presented by Buttyán et al. These are related to the lack of expressiveness of some of the formal parameters defined in the model and are described in the next section.

2.6.2 Limitations in Expressiveness

A fundamental aspect that remains unaddressed by Buttyán et al.'s analytical model is the lack of expressiveness when dealing with uncertainty, especially to model contextual information. Due to their very nature, uncertainty plays a major role in rational protocols. Not in vain, it has been stated in several occasions that the context in which the protocol is to be executed should be carefully checked. Roughly,

this means that some environmental factors, such as how much we trust other(s) participant(s) or the degree of reliability of the network, should be in some way incorporated into the security analysis.

Example 1

To illustrate this idea, consider the following example. As we can see in Fig. 2.2, A is not motivated to be fair to B in the last round of the protocol. Therefore, A could threaten B to execute $quit_A$ or to delay sending m_3 to B. Then, B would be safer quitting the protocol before round 2 and aborting the exchange. The best response that A can give to B's quit strategy is to quit as well. Therefore $s^* = (quit_A, quit_B)$ is also a Nash equilibrium of the protocol game of the model, which could be the most preferable protocol outcome under threatening situations.

In order to resolve this issue, A could have an incentive to be fair to B in the last round of the protocol. This incentive may be a kind of "reputation factor", securely managed by external parties, publicly known and which would have an effect on entity A's payoff function.

At a more practical level, suppose that, in the past, A has been honest (i.e. she has sent m_3 at step 3) in the 75% of the exchanges performed. How can this information be taken into account by other entities to decide whether or not to initiate a new protocol run with A? Moreover, this decision will also depend on the values that the exchanging items are worth to both parties. For instance, if the item is very important to player B, B would assume the risk of exchanging with a questionable party.

In Buttyán et al.'s model, the protocol participants could not bring their past experience or their beliefs into the current protocol instance. We found this inappropriate and unrealistic.

Example 2

In Buttyán et al.'s model –as in many others reasoning models– the network is considered an additional participant of the protocol. However, its behavior is limited to always deliver messages. Even though a reliable network can be assumed in many circumstances, it would also be interesting to count on a formalism in which more complex behaviors can be analyzed.

This feature is particularly relevant in the case of rational–exchange protocols, for their most probable execution environments could be hostile to, at least, one of the participants. For instance, in a mobile ad hoc network, two devices that are not in each other's range must rely on intermediary nodes to carry out a multi-hop

communication. In this case, assuming that each node that compose the network will behave well is a hypothesis that simplifies the analysis, but it is unrealistic.

Apart from modeling the actions that the network can perform as a participant (e.g. deliver or not deliver messages), it would also be informative to take into account *beliefs* about its behavior. This would allow us to distinguish between a network which is highly reliable (e.g. 99% of the messages are properly delivered) and a network which is highly non–reliable.

In this thesis, part of our effort will go into extending Buttyán et al.s model, so uncertainty and contextual information can be easily considered and parameterized, in an enhanced version of the original formalism.

2.7 Complexity on Computing Nash Equilibria

Finally, in this section we describe additional considerations which establish further limitations on formalisms based on Game Theory, such as the one just described. Game Theory, and in particular Buttyán et al.'s model, provides an excellent framework in which to analyze rationality. The basic idea of representing protocols as games and, identifying rational protocol outcomes with Nash equilibria of the protocol game, represents a powerful analytical tool.

However, existing results on the complexity of computing Nash equilibria impose significant limitations to the aforementioned approach.

On the one hand, the existence of at least one Nash equilibrium in a finite game is a well known and easily demonstrable result (see Theorem A.1.1). On the other, finding just one Nash equilibrium if a finite normal–form game has been formally proven to be *hard* (in particular PPAD-complete[2], even in the two–player case ([Conitzer and Sandholm, 2004]).

Moreover, many other questions related to the computation of Nash points are also proven to be extremely difficult to answer. For example, counting the number of equilibria points in a finite game is $\sharp\mathcal{P}-hard$ ([Conitzer and Sandholm, 2004]) [3]; find which of the players pure strategies receives positive probability in the equilibrium is also an $\mathcal{NP}$ problem ([Conitzer and Sandholm, 2004]); or given a pure strategy, to decide whether it is played in any Nash equilibria of the game is $\mathcal{NP} - hard$ ([Gilboa and Zemel, 1989]). Similar results are encountered for Bayesian games for which to determine whether there exists a pure-strategy Bayesian equilibrium is proven to be $\mathcal{NP} - complete$ ([Conitzer and Sandholm, 2004]).

Another topic of interest is how the protocol game is presented. Most results on

[2]PPAD is a complexity class, standing for "Polynomial Parity Arguments on Directed graphs".

[3]$\sharp\mathcal{P}$ is the set of counting–problems associated with the decision–problems in the set $\mathcal{NP}$. A $\sharp\mathcal{P}$ problem is at least as hard as the corresponding $\mathcal{NP}$ problem.

complexity are enunciated for normal–form games (games are represented as tables). However, two main issues make these results extendable to extensive–form games (games represented as trees): (1) A normal–form game can always be represented in an extensive–form. However, an algorithm to convert a extensive–form game onto a normal–form runs in exponential time. This indicates that the computational problem on extensive–form representations cannot become any easier than under normal–form. And (2), the computational complexity increases significantly when the games are dynamic of imperfect information with information sets of more than one element.

The best-known algorithm for finding a Nash equilibrium in finite games in normal–form is the Lemke-Howson algorithm ([Lemke and Howson, 1964]), which has however been proven to have exponential running times in some instances. Significant efforts have also gone to resolve games in extensive–form representations. However, most of the results are related to the design of efficient algorithms for the computation of approximations to equilibrium points in zero–sum two–player games ([Koller and Megiddo, 1992], [Gilpin et al., 2007], [Gilpin and Sandholm, 2007]).

2.8 Conclusions

In this chapter, we have demonstrated how Syverson's scheme suffers from some weaknesses due to an inappropriate design of the cryptographic evidences. Several attacks are described showing that the protocol can lead to undesired situations for any of the two parties involved in the exchange. We have also suggested how to fix the scheme and we have given a formal security proof of the enhancement. The proof is based on guaranteeing freshness of all protocol messages, thus ensuring rejection of all forms of replay attacks. We have used BAN logic to prove the correctness of our enhancement.

Additionally, we have presented Buttyán et al.'s formal model used to analyze Syverson's scheme. We recognized Buttyán's formalism as a powerful tool to reason about rationality in exchanging schemes. Nevertheless, other aspects of the model were identified as weak and limiting. In particular, one of those aspects has motivated us to propose two extended versions of the formalism, to be described in the next chapter of this thesis.

Finally, results on the complexity of computing Nash equilibrium of finite games, impose further restrictions to any analytical tool based on a Game theoretical approach.

Chapter 3

A Model based on Dynamic Games of Imperfect Information

3.1 Introduction

In the previous chapter we presented Syverson's rational exchange protocol. In Section 2.4, we carried a cryptanalysis of Syverson's scheme and proposed and enhanced version. For all intents and purposes and from this point onwards in this document, we will always refer to the corrected version of Syverson's protocol described in Section 2.4.5.

Also in Chapter 2, we introduced Buttyán et al.'s formal model and, although the model was found correct and consistent, we also found it presented several drawbacks limiting the overall scope of the formalism. For example, the lack of expressiveness when dealing with uncertainty, in particular when trying to model contextual information, was considered to be too restrictive for real life exchanging scenarios.

In this chapter, we present a contribution which consists in extending Buttyán et al.'s model, to capture relevant aspects involved in the execution of a rational–exchange protocol. We will formalize and analyze the effect that factors such as participant reputation, protocol robustness or even unpredictable participant behavior, have on the outcome of the protocol execution. To the best of our knowledge, it is the first time that such parameters have been formalized and considered when analyzing security protocols, therefore providing an extended analysis framework that goes beyond the cryptographic properties of the scheme.

3.1.1 Chapter Overview

Our approach is based on representing protocol games as dynamic games of *imperfect* information. In simple terms, a dynamic game is one of imperfect information if a player does not know exactly what actions other players took up to that point in the game. Intuitively, if it is my turn to move, I may not know what every other player has done up to the current point.

In this type of games, players' beliefs over other participants' previous actions can be taken into account when making the optimal decision at any given point during the protocol execution. Several Game Theory results allow us to predict the outcome of such a game and therefore, the outcome of the protocol it represents when executed by rational entities. Despite the analysis becoming more complex than by using basic Game Theory, we find it more realistic and more powerful. Readers are referred to Appendix A for a detailed exposition on Games of Imperfect Information.

3.1.2 Chapter Organization

The chapter is organized as follows. In Section 3.2, we provide a global description of our extended model detailing the differences with the original formalism. In Section 3.3, we apply the new model to the analysis of Syverson's protocol. In Section 3.4 we describe the main conclusions.

3.2 Extended Model based on Dynamic Games of Imperfect Information

In this section, we formalize those aspects of Buttyán et al.'s model previously described in Section 2.3.1 extending such a formalism by the use of:

(I) *Imperfect information in protocol games:* Represented as information sets with more than one element in which entities form conjectures about other players' previous actions.

(II) *Randomized strategies:* Strategies based on probability distribution functions over players's set of actions.

Next, we give a formal definition of each one of these amendments followed by a detailed description and a relation of the differences between our work and the original formalism.

3.2.1 Extensions to the Model

(I) Imperfect Information in Protocol Games

The concept of *protocol game* was introduced by Buttyán et al. as a way to represent an exchange protocol (see Section 2.3.1). As mentioned before, the protocol game of an exchange protocol is intended to model all possible interactions of the protocol participants, even the potentially misbehaving actions (i.e., those different from the ones prescribed by the protocol).

In the following definitions, we will formally define the process of deriving a protocol game from the description of a given exchange protocol. We will describe such a process according to Buttyán et al.'s work ([Buttyán, 2001]) but we will also include several novel components part of our extension to the original model. Whenever possible we will use the same notation as in [Buttyán, 2001] and Appendix A.

We start by unifying notation regarding two–entity exchange protocols.

Definition 3.2.1 (Two–entity exchange protocol)**.** *We notate a two–entity exchange protocol as a tuple* $\Pi = \langle P, \mathcal{O}, T \rangle$ *where:*

- $P = \{P_1, P_2\}$ *is the set of protocol participants,*
- $\mathcal{O}$ *is the set of all items/tokens being exchanged during the protocol execution. Let* $M = \mathcal{O}^*$ *be the set of all possible messages that can be constructed by concatenating zero or more items from* $\mathcal{O}$*, and*
- T *is an ordered collection of* n *protocol steps describing the scheme, each of the form:*

$$\begin{aligned} & t : P_i \rightarrow P_j : m_t \\ & \textit{with} \quad t = 1 \ldots n, \quad i, j \in \{1, 2\} \quad \textit{and} \quad m_t \in M \end{aligned} \tag{3.1}$$

The following game of imperfect information can be constructed to represent any given two–entity exchange protocol denoted as previously.

Definition 3.2.2 (Extended protocol game)**.** *Given a two–entity exchange protocol* $\Pi = \langle P, \mathcal{O}, T \rangle$*, the following protocol game of imperfect information denoted as* G_Π *is defined to represent such a protocol:*

$$G_\Pi = \langle P, A, Q, p, (\mathcal{I}_i)_{i \in P}, (\preceq_i)_{i \in P} \rangle \tag{3.2}$$

where:

- $P = \{P_1, P_2\}$ *is a set of players.*

- *A and Q are the set of actions and set of action sequences respectively, satisfying:*

  a1. $\epsilon \in Q$, *where* ϵ *is the empty sequence.*

  a2. if $q = (a_k)_{k=1}^{t} \in Q$ *and* $0 < w < t$, *then* $q = (a_k)_{k=1}^{w} \in Q$

  a3. if $q = (a_k)_{k=1}^{t} \in Q$, $0 < t < n$ *and* $a \in A$ *then* $q \cdot a$ *denotes the action composed by* q *followed by* a.

  a4. A finite sequence of actions $q \in Q$ *is said to be terminal if there is no* $a \in A$, *such that* $q \cdot a \in Q$. *The set of terminal sequences of actions is denoted by* Z.

  a5. $A(q) = \{a \in A : q \cdot a \in Q\}$ *denotes the set of available actions after action sequence* $q \in Q \backslash Z$.

  a6. In particular, for every $q = (a_k)_{k=1}^{t-1} \in Q \backslash Z$, $A(q) = \{quit,\ m_t,\ gm_t,\ ugm_t\}$ *where,*

  - m_t *represents action send message* m_t, *where* m_t *is as described in the protocol.*
  - gm_t *represents action send message* gm_t, *where* gm_t *represents a predictable deviation from message* m_t.
  - ugm_t *represents action send message* ugm_t, *where* ugm_t *represents an unpredictable deviation from message* m_t.

  a7. p *is the player function. It assigns a player* $p(q) \in P$ *to every non–terminal sequence* $q \in Q \backslash Z$. *The interpretation is that player* $p(q)$ *has the turn after the sequence of actions* q.

- $\mathcal{I}_i$ *is an information partition for player* $P_i \in P$. *It is a partition of the set* $\{q \in Q \backslash Z : p(q) = i\}$ *satisfying:*

  b1. If sequences q *and* q' *are in the same information set* $I_i \in \mathcal{I}_i$, *then* $A(q) = A(q')$.

  b2. In particular, for any sequence $q \in Q \backslash Z$, *sequences* $(q \cdot send\ m_t)$ *and* $(q \cdot send\ ugm_t)$ *are in the same information set, so* $A(q \cdot send\ m_t) = A(q \cdot send\ ugm_t)$.

  b3. If sequences q *and* q' *are in the same information set* $I_i \in \mathcal{I}_i$, *then player* P_i *is forced to define a probability distribution* α_i *over every action sequence in* I_i, *so* $\sum_{q \in I_i} \alpha_i(q) = 1$.

- *Finally,* $\preceq_i$ *is a preference relation of player* $P_i \in P$ *on* Z.

The most common representation of a game of imperfect information is a tree (see Section A.4 for a detailed description of games in extensive–form). All possible action sequences in set Q are represented by branches in the tree and every new action added to an existing sequence is represented by a new edge directed to a new node. Terminal nodes represent different outcomes of the game.

Note that, information sets I_i describe the information available to player P_i at every stage of the game. The protocol game just described has been defined such that one information set can cover several nodes. This means that, a player reaching such a set does not know in which particular node of the information set she is –or equivalently– she does not know the last action of her rival. Nodes which belong to the same information set are represented with a dashed line across the game tree.

Moreover, for those information sets I_i, with more than one node it is necessary to specify some player beliefs. Formally, these beliefs are represented by a probability distribution function α_i over the nodes belonging to the information set I_i.

Finally, all sequences $q \in Z$ are the possible outcomes of the game. The preference relations $\preceq_i$ establishes which outcomes are preferred by player P_i. Thus, if $q, q' \in Z$ and $q \preceq_i q'$, then player P_i prefers q' to q.

(II) Randomized Strategies

Buttyán et al.'s original model will also be enhanced by the use of mixed or randomized strategies. A *randomized* or *mixed* strategy is a strategy which chooses randomly between possible moves. For each player, there is a probability distribution over the set of possible moves at each step in the game. Each probability value will correspond to how frequently each move is chosen, being possible for players to assign probability zero to one or more moves. Playing a mixed strategy should be understood in contrast to playing a pure strategy, where a player follows a single strategy with probability one and assigns probability zero to all other options. Next are the formal definitions of these concepts within the model being described. The reader should refer to Section A.1.3 of the Appendix A for further details on mixed and randomized strategies.

Definition 3.2.3 (Pure strategy)**.** *A pure strategy for player P_i is defined as a function s_i that assigns an action in $A(q)$ to each non–terminal action sequence $q \in Q \backslash Z$ for which $p(q) = i$, with the restriction that:*

$$s_i(q) = s_i(q') \quad \forall q, q' \in I_i \tag{3.3}$$

In other words, function s_i assigns the same action to all action sequences belonging to the same information set.

We denote the set of all strategies of player P_i by S_i. Since a strategy s_i assigns the same action to every action sequence q that belongs to the same information set I_i, we sometimes write $s_i(I_i)$ instead of $s_i(q)$.

Definition 3.2.4 (Strategy profile)**.** *A strategy profile is a vector of pure strategies s_i, one for each player in the game, where $s_i \in S_i$.*

Definition 3.2.5 (Mixed strategy)**.** *Being S_i the set of all possible pure strategies for player P_i, a mixed strategy for P_i is a probability distribution function $p_i \in \Delta(S_i)$.*

Definition 3.2.6 (Probabilistic strategy profile)**.** *A probabilistic strategy profile is a vector of mixed strategies p_i, one for each player in the game.*

3.2.2 Comparison with Buttyán et al.'s Model

Various aspects of the formalism proposed in Section 3.2.1 make it differ from Buttyán et al.'s original model presented in Chapter 2. These are:

i. **Information sets with more than one element:** Buttyán et al. considered information sets of only one element. This meant that players were always aware of other player's previous moves. In particular, given a two–entity exchange protocol the original model assigns entities actions to *quit* the game, send message m_t as defined in Section 2.2, or send garbage message gm_t as described in Section 2.3.3, being these last two messages fully distinguishable by the recipient. By contrast, in our model information sets are defined such that for each message sent in the protocol (action taken by a player in the protocol game) there will be an information set of at least two elements (Condition *b.2* of Definition 3.2.2) which will be undistinguishable for the recipient's point of view.

ii. **Entity's misbehavior representation:** In the original formalism, entities's misbehavior is limited to sending forged messages (*garbage* m_t) which are always detected and penalized[1] as shown in Fig. 2.2. By contrast, our specification includes a new action consisting in sending unpredictable garbage message ugm_t. If message m_t is composed of several items/tokens, an unpredictable garbage message could differ from m_t in only one or in more than one of the components. Message ugm_t could vary from m_t only in content and/or it could also be semantically different. In fact, message ugm_t represents whatever a malicious entity is able to configure to forge a valid

[1] Note that this is only applicable when the enhancement described in Chapter 2 is added to the protocol description, so that at the end of the protocol, entity B holds a valid token to evidence A's misbehavior.

m_t message. Therefore, our model takes into account the possibility of an unpredictable malicious message being sent instead of m_t (specified in the protocol description) of which we cannot anticipate the content or its nature.

Those game strategies including action *send unpredictable garbage* m_t will play an important role in our new analysis of Syverson's protocol.

iii. **Player's conjectures:** Moreover, in our new model, entities are forced to form conjectures about previous player's moves. Probability distribution functions are defined for each non–singleton information set in the game. Besides, conjectures over information sets can be interpreted in several ways according to the particular protocol and its analysis. When appropriate, distribution α_i could represent the level of confidence an entity P_i can place on the robustness of the protocol design, for example assigning a negligible probability value to the action of sending message ugm_t when the protocol design offers high levels of trust.

iv. **Player's beliefs:** Finally, in Buttyán et al.'s model, all strategies are considered to be pure. In other words, at every decision point in the game, players have no doubt about which action to take next should they reach such a point. The game equilibrium is therefore an equilibrium on pure strategies whose existence is not always guaranteed. By contrast, at relevant decision points during a protocol game our model is able to consider and manage randomized strategies. Moreover, in finite games, the existence of at least one Nash equilibrium point with randomized strategies is ensured by an important result (see Theorem A.1.1). An application of this will be depicted when analyzing Syverson's scheme in our extended model.

3.3 Syverson's Rational–Exchange Protocol as a Game of Imperfect Information

In order to illustrate the proposed formal model we will apply the formalism described in Section 3.2.1 to the analysis of Syverson's scheme. Section 3.2.1 provided the guidelines to represent a rational–exchange protocol as a game of imperfect information. In this section, we will present a series of definitions which will serve to represent Syverson's protocol game within such an extended Game theoretical model.

Furthermore, we will also describe a series of parameters used to define the level of uncertainty players hold about certain aspects of the game. Formally, some of these parameters will be used to define probability distribution functions over

non–singleton information sets, while others will represent randomized strategies for the players involved in the game. Finally, Syverson's protocol game of imperfect information will be represented in extensive–form.

Definition 3.3.1 (Set of players)**.** *We denote P the set of players in Syverson's protocol game where P=$\{A,B\}$.*

Definition 3.3.2 (Player's set of actions)**.** *The set of actions available to players A and B in Syverson's protocol game of imperfect information is:*

$$A = A_A \cup A_B \tag{3.4}$$

where:

$$\begin{array}{ll} A_A = & \{quit_A, m_1, gm_1, ugm_1, m_3\}, \\ A_B = & \{quit_B, m_2\} \end{array} \tag{3.5}$$

and,

- *$quit_X$ represents action quit the protocol for an entity X.*
- *m_k represents action send message m_k.*
- *gm_k represents action send garbage gm_k.*
- *ugm_k represents action send unpredictable garbage ugm_k.*

Definition 3.3.3 (Pure strategies for player A)**.** *The complete set of pure strategies for player A, denoted as S_A, is defined as the set of tuples:*

$$\begin{array}{ll} S_A = & \{(m_1, m_3),\ (m_1, quit_A), \\ & (gm_1, m_3),\ (gm_1, quit_A), \\ & (ugm_1, m_3),\ (ugm_1, quit_A), \\ & (quit_A, \cdot)\} \end{array} \tag{3.6}$$

where the first component of each tuple represents the action taken by player A at step one of the protocol and the second component represents the action taken by A at step three of the protocol.

Definition 3.3.4 (Pure strategies for player B)**.** *The complete set of pure strategies for player B, denoted as S_B, is defined as the set of actions:*

$$S_B = \{m_2, quit_B\} \tag{3.7}$$

where each strategy represents the action taken by player B at step two of the protocol.

3.3.1 Modeling Uncertainty

By introducing imperfect information and randomized strategies, we are providing the mechanisms to consider and measure a completely new set of variables when formally analyzing an exchange protocol. We will be able to evaluate the effect that factors such as participant reputation, protocol robustness, network reliability or participant past experience, have on the outcome of a protocol execution.

In this regard, for the analysis of Syverson's protocol three novel parameters are defined to capture a series of factors surrounding the protocol execution. These are essential parameters when trying to predict and anticipate the outcome of Syverson's rational–exchange and they are used to describe two different areas of uncertainty:

- The different beliefs that each entity holds about the protocol robustness and,
- The level of trust between participants.

The conjecture about the protocol's robustness is especially important as what we present here is a novel way to express unpredictable participant behavior. Since the definition of the Dolev-Yao adversary [Dolev and Yao, 1983], in all formal models participant unpredictable behavior has always been tailored and limited to a series of possible malicious actions. Our approach is based on expressing, within a single parameter, the whole set of unknown vulnerabilities that a protocol might present.

Below we provide a detailed description of each one of these new variables. As stated before, these parameters will be used to formally define probability distribution functions over non–singleton information sets and to represent randomized strategies for the players involved in the game.

Parameter α

The chances of successfully attacking a particular protocol (a malicious entity being able to derive from the protocol description without being detected) get reduced if the protocol is a well known scheme, on which reasonable levels of testing and trialling have been performed. By contrast, the probability of unpredictable malicious behavior taking place at execution time increases if the scheme is new or if it has a reputation for poor and insecure performance.

In the case of Syverson's protocol, B is the entity taking a greater risk, so B must be confident that the protocol is well designed and that A cannot deviate from it without being noticed. In this regard, our model captures the level of uncertainty that participant B holds over the robustness of Syverson's protocol. A certain probability α will be considered as the level of confidence customer B has in the protocol's design. It represents the possibility that a forged message sent by A could

actually be part of the protocol execution, enabling A to finish the protocol in a advantageous position over entity B.

As previously mentioned, in the original model A's misbehavior was limited to sending a forged message *garbage* m_1 which would always be detected and penalized. By contrast, in our model we also consider the possibility of ugm_1 being sent instead of m_1 of which we cannot anticipate the content or its nature.

Parameter β

Furthermore, entity A will also be asked to conjecture about entity's B behavior during the protocol. We will define a parameter β which will capture the uncertainty participant A has over participant B's behavior at step two in the protocol. B could, at step two, continue or quit the protocol. In our model, A will assign a certain probability β to the event of B sending m_2 at round two of the protocol. Hence, $(1-\beta)$ represents the probability of B misbehaving at round two by quitting the execution. Note that A can always verify the freshness of message m_2 and its originator, so B cannot cheat sending the wrong m_2 as this will always be detected and punished. B's signature on m_2 ensures A that B received $item_A$ and that B proceeded sending $item_B$. Message m_2 could always be used as a proof of $B's$ misbehavior in the protocol (message m_2 is a Non–Repudiation of Origin token for entity A).

Parameter δ

In a similar way, as we mentioned before A could quit or delay sending m_3 in the last round of the protocol. This forces entity B to form a new conjecture over A's behavior besides the initial one captured by parameter α. This new conjecture could be based on past experience, A's reputation or any other factor. This way, our model will evaluate the uncertainty B has over A's behavior at the last step in the protocol.

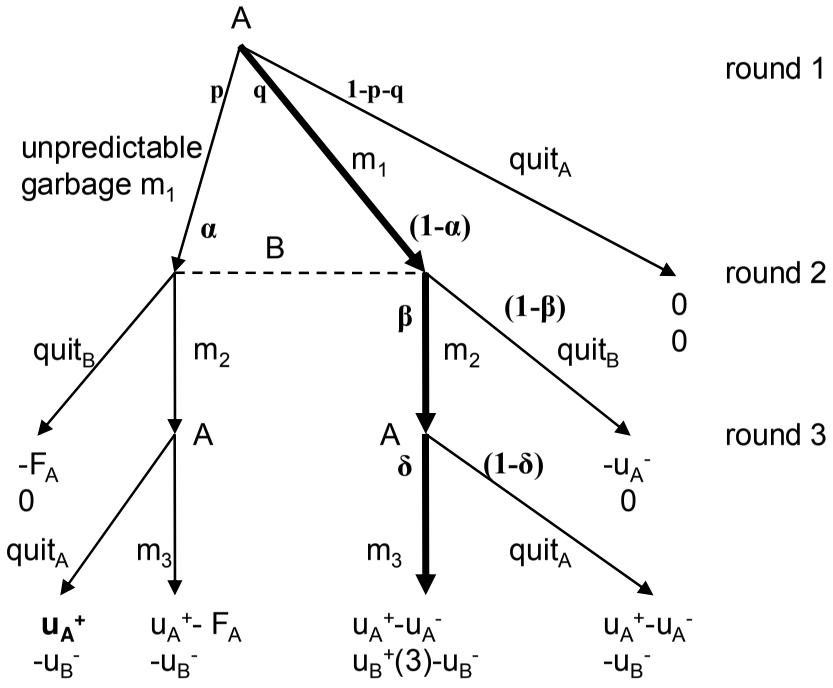
Formally, a certain value δ will define the probability that entity B assigns to the event of A sending m_3 at round three of the protocol. Consequently, the value $(1-\delta)$ determines the probability entity B assigns to the event of A getting delayed in sending m_3 or not sending it at all.

Most likely, entity A will be using Syverson's protocol to interact with a variety of entities in different occasions. This information (represented by a reputation factor or similar) will help entity B to adjust an accurate value for δ.

Note that in a different instance, parameter δ could be given a complete different meaning. It could be defined as the formal representation of the level of trust entity

B places on the network. The reliability of the network can be taken into account when trying to anticipate protocol participant behavior.

3.3.2 Protocol Game in Extensive–Form



Where B's conjectures are:
α = prob{A is able to misbehave at round one}
δ = prob{A sends m_3 at round three of the protocol}
and A's conjectures are:
β =prob{B sends m_2 at round two of the protocol}

Figure 3.1: Syverson's protocol as a game of imperfect information.

By considering Syverson's protocol game as a game of imperfect information, we are enabling both entities, A and B, to evaluate the conjectures about each other, and also about the correctness and robustness of the protocol. These conjectures are represented by probabilities α, β, and δ previously introduced and can be formalized as follows:

- Parameter α represents a probability distribution function over the non–singleton information set composed by actions m_1 and ugm_1 at stage one of the protocol.
- Parameter β represents a probability distribution over the set of pure actions for entity B. It therefore represents a randomized strategy for entity B.
- Finally, parameter δ represents a probability distribution over entity's A set of actions at step three of the protocol. Therefore it is part of entity A's mixed strategies.

See Fig. 3.1 for a representation of the imperfect information Syverson's protocol game in extensive–form. Fig. 3.1 extends Fig. 2.2 showing a completely new scenario, as well as being able to evaluate uncertainty at each step of the game represented in the tree. Payoff values are defined as in Section 2.3.3. In Fig. 3.1, there exists the possibility that A could send B a forged message ugm_1, for which A would obtain message m_2 in return, and for which entity A will not be fined or penalized. This would only be possible by stepping outside the previous model and assuming that there are still vulnerabilities in the protocol design. A may well identify such flaws and try to take advantage of them. However, it is not always clear that flaws exist and that entity A could recognize them. So the uncertainty B holds over the protocol correctness and entity's A behavior at step one in the protocol can be captured and modeled by this new branch in the tree and the parameter α.

Note that α is composed by two different probabilities (p, q). Probabilities p and q represent the original B's conjectures over entity A's behavior at step one. Parameter α is calculated using the Bayes rule, so $\alpha = p/(p + q)$.

To simplify the tree, we have omitted the predictable gm_1 path, previously shown in Fig. 2.2 and from which entity A would always be deterred as it would always be punished.

The dashed line across B's nodes represent the uncertainty that B holds over A's previous move (formally, both nodes belong to the same information set). Entity B might be incapable of recognizing A's unpredictable misbehavior after step one of the protocol. B assumes a risk when deciding what to do at round two as it does not know what move A has made. If A tries to deceive B and A is detected, then

B would quit the protocol. In that case, A would be penalized with the same value F_A as if she had sent gm_1 in the original model.

In the following section, several calculations will establish the criteria for A and B to be participants of the protocol depending on the values of the new set of parameters. We will see how the values that $item_A$ and $item_B$ are worth to entities A and B also play an important role in the decision-making process. Additionally, we will calculate where entities reach an equilibrium in the new protocol game.

3.3.3 Expected Payoff for Entities A and B

Let $EP_B(m)$ represent the expected payoff for entity B after sending message m. Likewise, $EP_B(quit_B)$ represents the expected payoff for entity B after quitting the protocol. Entity B can then formulate the following considerations for each one of the two possible pure strategies at round two of Syverson's protocol:

$$\begin{aligned} EP_B(quit_B) = &\quad 0 \\ EP_B(m_2) = &\quad \alpha * [(1-\delta) * (-u_B^-) + \delta * (-u_B^-)] + \\ &\quad (1-\alpha) * [\delta * (u_B^+(3) - u_B^-) + \\ &\quad (1-\delta) * (-u_B^-)] \\ = &\quad u_B^+(3) * \delta * (1-\alpha) - u_B^- \end{aligned} \tag{3.8}$$

Note that:

$$EP_B(m_2) \geqslant EP_B(quit_B) \Leftrightarrow \delta * (1-\alpha) * u_B^+(3) - u_B^- \geqslant 0 \tag{3.9}$$

Therefore:

$$EP_B(m_2) \geqslant EP_B(quit_B) \Leftrightarrow \delta * (1-\alpha) \geqslant u_B^-/u_B^+(3) \tag{3.10}$$

The graph shown in Fig. 3.2 (a) represents the function $\delta * (1-\alpha)$. For values α and δ for which the graph is over the value $u_B^-/u_B^+(3)$, the best strategy for B would be to carry on with the exchange and follow the protocol description. Below that line, B's best strategy is to quit, as otherwise the expected payoff value would be less than zero.

In a similar way, A can formulate the following considerations. For each possible strategy that A can follow, the expected payoff would be:

$$
\begin{aligned}
EP_A(quit_A, \cdot) &= 0 \\
EP_A(m_1, quit_A) &= \beta * (u_A^+ - u_A^-) + (1-\beta) * (-u_A^-) \\
&= \beta * u_A^+ - u_A^- \\
EP_A(m_1, m_3) &= \beta * (u_A^+ - u_A^-) + (1-\beta) * (-u_A^-) \\
&= \beta * u_A^+ - u_A^- \\
EP_A(ugm_1, quit_A) &= \beta * (u_A^+) + (1-\beta) * (-F_A) \\
&= \beta * (F_A + u_A^+) - F_A \\
EP_A(ugm_1, m_3) &= \beta * (u_A^+ - F_A) + (1-\beta) * (-F_A) \\
&= \beta * u_A^+ - F_A
\end{aligned}
\tag{3.11}
$$

Note that we have omitted the strategies (gm_1, m_3) and $(gm_1, quit_A)$ which appeared in the original representation tree. They do not affect the following rationale, as they are strictly dominated strategies, i.e. there are no possible conjectures over entity B's behavior which would make a rational entity A choose such strategies. The strict dominance comes from a negative payoff (expected payoff) given by the expression $\beta * u_A^+ - F_A$. Furthermore, the strategy (ugm_1, m_3) is also a strictly dominated strategy with a payoff value less than zero. However, the strategy $(ugm_1, quit_A)$ plays an important role, as there will be a threshold value for β to establish whether A, having the opportunity to attack the protocol, would take the risk to be detected at the first step of the protocol.

From previous expressions, we obtain the following relationship:

$$
EP_A(ugm_1, quit_A) \leqslant EP_A(m_1, m_3) \Leftrightarrow \beta \leqslant (F_A - u_A^-)/F_A \tag{3.12}
$$

This is, for a rational entity A to be motivated to behave accordingly to the protocol description, the expected payoff by doing so must be higher than the expected payoff obtained when misbehaving. Misbehavior is only profitable when $\beta \geqslant (F_A - u_A^-)/F_A$.

Fig. 3.2 (b) shows the intersection between the space of values for α, δ and the new threshold for A's conjecture, β. From a Decision Theory point of view, the shadowed area represents the space of values where a rational exchange may take place. This is the space for which entities, individually committed to achieve the best results for themselves without considering the other participant's reactions, would decide to carry on with the exchange.

There exist a strong correlation between variables α, β and δ as they will be affected by the same publicly known reputation factors or other similar parameters. That is, if A's reputation is not good or if it is the first time A participates in an exchange, B will show a high level of distrust, but A will be aware of this and will adjust the value of β accordingly. When considering a repeated scenario, the total

profit for participants A and B is calculated as an average of the profits obtained at each one of the protocol executions. Misbehaving will then have a global impact on the total expected payoff.

3.3.4 Interpretation and Graphical Representations

The following figures will help to better understand the implications that calculations in section 3.3.3 will have in real scenarios. Equations (3.10) and (3.12) express in mathematical terms the relationship between the different parameters defined in the formal model. Representing those relations graphically will assists in identifying such links.

Let us consider a rational merchant *Alice* and a rational buyer *Bob*, who want to exchange $item_A$ and $item_B$ using Syverson's protocol. Let us also assume that these two entities have never exchanged any items previously to this point and that they have never been participants in the same protocol run before, in particular Syverson's protocol. The following graphs will explain phenomena such as why in some instances, merchant *Alice*, having found vulnerabilities in Syverson's protocol, will be rationally forced to ignore these and carry on with the actual protocol description. The graphs will also describe the context in which *Bob*, not trusting merchant *Alice*, will still be part of the protocol if the profit *Bob* expects to obtain is sufficiently high.

From *Bob*'s point of view, calculations in Section 3.3.3 –particularly equation (3.10)– provide a threshold between sending message m_2 or quitting the protocol at stage two. By contrast, from *Alice*'s point of view, calculations in Section 3.3.3 –particularly equation (3.12)– provide a threshold between behaving honestly at protocol stage one or taking the risk of committing fraud.

The following analysis can be carried out from the buyer's point of view.

Analysis of the Correlation between Cost and Profit for B

Let us assume Syverson's protocol to be reasonably robust and flawless. In this case, *Bob* will probably assign α a low value, for example of $\alpha = 0.1$, (α is the probability of *Alice* breaking the protocol and sending *Bob* an unpredictable fraudulent message m_1). Moreover, δ is the parameter representing the level of trust entity *Bob* places on merchant *Alice* delivering message m_3 as the last step in the protocol. For different values of δ, equation (3.10) gives us the upper bound for the ratio between cost u_B^- and profit $u_B^+(3)$. Within this range, Bob would be rationally forced to enter the protocol as the expected payoff would be higher than the one obtained by quitting.

In particular, Fig. 3.3 shows how when the level of trust on merchant *Alice* is high (e.g. $\delta = 0.9$), the limit for the ratio $u_B^-/u_B^+(3)$ increases (cost could be a high

proportion of the profit). In the case of *Bob* not being able to trust *Alice*, the limit for the ratio decreases (cost could only be a small proportion of the profit). In this last context, *Bob* would only be part of the protocol when the margin between profit and lost is sufficiently high.

In Fig. 3.4, different values of α are considered. The parameter α represents the level of trust that entity B places on the robustness and correctness of the protocol. Considering as well different values of δ (level of trust on merchant *Alice* to deliver message 3), the shadowed area is showing the set of values for the ratio between cost and profit, for which *Bob* would rationally choose to participate in the protocol with *Alice*.

Analysis of the Correlation between α and δ

Equation (3.10) does also establish the correlation between parameters α and δ (entity's B conjectures). Fig. 3.5 shows how for each given ratio between cost and profit for exchanging items $item_A$ and $item_B$, the level of trust *Bob* places on the protocol's robustness will determine the pattern for rational behavior.

As an example, consider the case when the cost is 10% of the profit *Bob* obtains when exchanging $item_B$ ($u_B^-/u_B^+(3) = 0.1$). Then, for *Bob* not to quit the protocol at step 2, α must be below the limit $[1 - 0.10 * (1/\delta)]$, given by equation (3.10).

Likewise, when the cost is 90% of the profit that *Bob* obtains exchanging $item_B$, α must be below the limit $[1 - 0.90 * (1/\delta)]$ given by the same equation (3.10). Figure Fig. 3.5 shows how the threshold for parameter α decreases when the margin between cost and profit decreases and vice versa. In other words, when the benefit expected is high, *Bob* will be part of the protocol even when it is quite probable that the protocol presents vulnerabilities. The shadowed area represents the values for which *Bob* will rationally choose to be a participant in the protocol.

As mentioned before, from *Alice*'s point of view, calculations in Section 3.3.3 and in particular equation (3.12) provide a threshold between behaving honestly and taking the risk of committing fraud at stage one in the protocol. *Alice*, having found vulnerabilities in the protocol design will rationally choose to ignore them and carry on with the protocol description, when β (probability of *Bob* sending m_2) drops below the limit given by equation (3.12). In other words, *Alice* would only try to cheat on *Bob* when she has enough guarantees that *Bob* is going to respond sending m_2 instead of quitting at stage two. The probability of *Bob* detecting the fraud and quitting the protocol must be sufficiently low for A to take advantages of possible protocol design faults. What follow is an analysis from entity *Alice*'s point of view.

Analysis of the Correlation between Cost and Penalty for A

Fig. 3.6 shows how the limit given for β in equation (3.12) is dependant on the ratio between the cost of sending $item_A$ (u_A^-) and the penalty for committing fraud (F_A).

Equation (3.12) can also be expressed as:

$$\begin{gathered} EP_A(ugm_1, quit_A) \leqslant EP_A(m_1, m_3) \Leftrightarrow \\ \beta \leqslant 1 - u_A^-/F_A \Leftrightarrow u_A^-/F_A \leqslant 1 - \beta \end{gathered} \tag{3.13}$$

Therefore, for example when $\beta = 0.2$, *Alice* is rationally forced to behave honestly if the cost of sending $item_A$ does not exceed 80% of the value of the penalty. Otherwise, if the cost exceeds 80% of the value of the penalty and *Alice* knows how to break the protocol, she would be better off by doing so in the context of a single protocol execution. Likewise when $\beta = 0.9$, merchant *Alice* is rationally forced to behave honestly if the cost does not exceed 10% of the value of the penalty. In other words, if it is unlikely that B detects the fraud and quits, and the cost of sending $item_A$ exceeds 10% of the value of the penalty, A is rationally forced to commit fraud as it is the most profitable option. In Fig. 3.6 the shadowed areas represent the space of values for which, given different values of β, *Alice* will ignore possible protocol flaws and behave honestly with entity B, sending the correct m_1 at stage one of the protocol.

Figure Fig. 3.7 does also represent the correlation expressed by equation (3.12), though in terms of value β and the ratio between cost and penalty.

Summarizing, the correlations just studied have served to:

(1) Establish the existing relation between the different areas of uncertainty surrounding Syverson's protocol execution; and

(2) Establish the criteria for *feasible* solutions from each player's individual point of view.

    Formally, an outcome is feasible if it can be obtained as a combination of actions in the game, or more generally from a probability distribution over combinations of actions. However, most of these possible outcomes can only occur when pre-execution agreements take place between players as they do not conform a Nash equilibrium of the game. The computation of Nash equilibria points (rational solutions) require further considerations.

3.3.5 Nash Equilibrium

In Section 3.3.3, we analytically calculated the set of values for α, β and δ for which rational entities A and B would unilaterally decide to be participants of Syverson's

protocol. The result is expressed by equations (3.10) and (3.12). Furthermore, we also represented graphically such a space as the intersection between the set of values for entity's B conjectures over A's behavior (α, δ) and the set of values for A's conjectures over B's behavior (β) (see Fig. 3.2 b). However, because other participant's actions have an impact on the overall expected–payoff value of all players, further refinements to these values are needed to evaluate the consequences of all possible actions.

In this section we formally compute the Nash equilibrium for the new protocol game depicted in Fig. 3.1.

As in previous sections, we first consider α to be a fixed value $\alpha_0 > 0$, known by both entities, as the probability of the protocol being faulted and vulnerable. This value will be equivalent to the probability of entity A being able to send ugm_1 at stage one of the protocol. Finally, F_A is a public and fixed value, greater than zero.

A Nash equilibrium of the Syverson's protocol game requires of the following formalisms:

Definition 3.3.5 (Probabilistic strategy profile)**.** *A probabilistic strategy profile in Syverson's protocol game is a vector $s = (p, q, \beta, \delta)$ where:*

- *p and q define a probability distribution function over the set of individual strategies that entity A holds at step one of the protocol game.*
- *β defines a probability distribution function over the set of individual strategies that entity B holds at step two of the protocol game.*
- *δ defines a probability distribution function over the set of individual strategies that entity A holds at step three of the protocol game.*

When appropriate, a probabilistic strategy profile in Syverson's protocol game can also be represented by the tuple (α, β, δ) where $\alpha = p/(p + q)$. For example, the tuple (α, β, δ) represents a probabilistic strategy profile, in which *Alice* chooses to send unpredictable garbage at step one of the protocol with probability α, *Bob* sends m_2 at round two with probability β and finally, *Alice* sends m_3 at round three with probability δ.

The functions *expected payoff* $EP_A(\cdot)$ for entity A and *expected payoff* $EP_B(\cdot)$ for entity B, defined in Section 3.3.3, can also be defined to express the expected payoff obtained by individual entities when following a specific probabilistic strategy profile:

Definition 3.3.6 (Expected payoff for entity A)**.** *Given a probabilistic strategy profile in the Syverson's protocol game defined by the tuple (α, β, δ), the expected*

payoff obtained by entity A when A and B follow such profile is:

$$\begin{aligned} EP_A(\alpha,\beta,\delta) = \quad & -\alpha * F_A + \alpha * \beta * F_A + \beta * u_A^+ \\ & -\alpha * \beta * \delta * F_A - u_A^- + \alpha * u_A^- \end{aligned} \tag{3.14}$$

Definition 3.3.7 (Expected payoff for entity B)**.** *Given a probabilistic strategy profile in the Syverson's protocol game defined by the tuple (α,β,δ), the expected payoff obtained by entity B when A and B follow such profile is:*

$$EP_B(\alpha,\beta,\delta) = \beta * [u_B^+(3) * \delta * (1-\alpha) - u_B^-] \tag{3.15}$$

Definition 3.3.8 (Nash equilibrium of Syverson's protocol game)**.** *Let S be the space of all probabilistic strategy profiles in Syverson's protocol game, each one of them represented by a vector (p,q,β,δ).*

A tuple $(\alpha_0,\beta^*,\delta^*) \in S$, where $\alpha_0 = p/(p+q) > 0$, represents a Nash equilibrium of the protocol game if and only if:

a) $EP_A(\alpha_0,\beta^*,\delta^*) \geqslant EP_A(\alpha_0,\beta^*,\delta)$ for all probability distribution δ; and

b) $EP_B(\alpha_0,\beta^*,\delta^*) \geqslant EP_B(\alpha_0,\beta,\delta^*)$ for all probability distribution β.

Note that, the equilibrium just defined represents a Nash equilibrium *perfect in subgames*, (see Section A.4.1) which is a refinement for the Nash equilibria in dynamic games.

According to this definition, in tuple $(\alpha_0,\beta^*,\delta^*)$, δ^* represents the best response *Alice* can give to $Bob's$ strategy β^*; likewise, β^* represents the best response *Bob* can give to $Alice's$ strategy δ^*. We will now enunciate and formally proof the following theorem:

Theorem 3.3.1. *When parameter $\alpha_0 > 0$, the strategy $[(quit_A,\cdot), quit_B]$ constitutes the only Nash Equilibrium of Syverson's protocol game.*

Proof. According to definition 3.3.8, for a tuple $(\alpha_0,\beta^*,\delta^*)$, where $\alpha_0 > 0$, to represent a Nash Equilibrium of Syverson's protocol game, it would have to satisfy requirements 3.3.8.a) and 3.3.8.b).

- *Requirement 3.3.8.a):*

  We consider the following equation:

$$\begin{aligned} EP_A(\alpha_0,\beta^*,\delta) = \quad & -\alpha_0 * F_A + \alpha_0 * \beta^* * F_A + \\ & \beta^* * u_A^+ - \alpha_0 * \beta^* * \delta * F_A - u_A^- + \alpha_0 * u_A^- \end{aligned} \tag{3.16}$$

  where $\alpha_0 > 0$.

Note that, the partial derivative of $EP_A(\alpha_0, \beta^*, \delta)$ with respect to δ is zero or negative:

$$\frac{\partial EP_A(\alpha_0, \beta^*, \delta)}{\partial \delta} = (-\alpha_0 * \beta^* * F_A) \tag{3.17}$$

This allows us to establish the following results:

- When equation 3.17 is negative, the maximum payoff value for entity A is obtained when $\delta = 0$.
- When equation 3.17 is zero, the payoff obtained by entity A is the same as when $\delta = 0$.

Therefore, a dominant strategy for entity A is to follow the distribution $\delta^* = 0$. Note that parameter δ^* represents the probability of merchant *Alice* sending message m_3 at the last step of the protocol. The value $\delta^* = 0$ represents the strategy $quit_A$ at step three of Syverson's protocol. Hence, for a rational merchant *Alice* the best response to any strategy buyer *Bob* might follow, is to *quit* at step three not sending message m_3.

- *Requirement 3.3.8.b):*

Likewise, to satisfy requirement 3.3.8.b we consider the following relation:

$$EP_B(\alpha_0, \beta, \delta^*) = \beta * [u_B^+(3) * \delta^* * (1 - \alpha_0) - u_B^-] \tag{3.18}$$

where $\alpha_0 > 0$.

Next equation is the partial derivative of $EP_B(\alpha_0, \beta, \delta^*)$ with respect to β:

$$\frac{\partial EP_B(\alpha_0, \beta, \delta^*)}{\partial \beta} = [u_B^+(3) * \delta^* * (1 - \alpha_0) - u_B^-] \tag{3.19}$$

Hence, Entity B knows that for entity *Alice* the dominant strategy is $\delta^* = 0$. Equation 3.19 allows to establish the following statement:

- The maximum payoff value for entity B in Syverson's protocol game is reached when $\beta^* = 0$.
- The payoff B obtains is $EP_B(\alpha_0, \beta^* = 0, \delta^*) = 0$.

Parameter β^* represents the probability of buyer *Bob* sending message m_2 at step two of the protocol. The value, $\beta^* = 0$ represents the strategy $quit_B$ at step two of Syverson's protocol. Therefore, for a rational entity B, the best response to the dominant strategy from merchant A is to *quit* at step two, not sending message m_2.

However, A carries out the following further calculations. For $\beta = 0$:

$$EP_A(\alpha_0, \beta^* = 0, \delta^* = 0) = -\alpha_0 * F_A + u_A^- * (\alpha_0 - 1) < 0 \tag{3.20}$$

Whereas:

$$EP_A(quit_A, \cdot) = 0 \tag{3.21}$$

Therefore, there is no tuple $(\alpha_0, \beta^*, \delta^*)$ where $\alpha_0 > 0$ that constitutes a Nash equilibrium of Syverson's protocol game. The strategy $[(quit_A, \cdot), quit_B]$ constitutes the only Nash Equilibrium, when parameter $\alpha_0 > 0$.

□

When $\alpha_0 > 0$, although there are many other probabilistic strategy profiles by which Syverson's participants can gain higher benefits, none of them constitute a Nash equilibrium. In particular, if $\alpha_0 > 0$ for all $\beta > 0$, entity A would be inclined to deviate from the protocol description to gain greater profit forcing entity B to reconsider their position and change their behavior.

The following corollary can be directly derived from Theorem 3.3.1.

Corollary 3.3.1. *The following probabilistic strategy profiles represent different Nash equilibria in Syverson's protocol game.*

$$\begin{aligned} N_1 \equiv &\quad (p = 0, q = 0, \beta^* = 0, \delta^* = 0) \\ N_2 \equiv &\quad (p = 0, q = 1, \beta^* = u_A^-/u_A^+, \delta^* = u_B^-/u_B^+(3)) \\ N_3 \equiv &\quad (p = 0, q = 1, \beta^* = 1, \delta^* = 1) \end{aligned} \tag{3.22}$$

Note $\alpha = p/(p+q) = 0$ in all three equilibria.

Proof. The proof will consist of graphically calculating all Nash equilibria in the game.

When parameter α is equal to zero, the protocol game gets reduced to the game analyzed by Buttyán et al., shown in Fig. (2.2). However, participant's reputation, past experience or network reliability are still valuable parameters to be considered in the analysis. We will use a graph to proof that apart from the Nash equilibrium identified by Buttyán et al.'s analysis, there are two other equilibria besides that one. These two new equilibria stem from the ability within our extended model, to represent the different reactions entities have when confronting uncertainty and distrust at each step in the protocol execution.
The following expressions are evaluated:

$$\begin{aligned} EP_A(m_1,\cdot) = &\ \beta * (u_A^+ - u_A^-) + (1-\beta) * (-u_A^-) \\ = &\ \beta * (u_A^+) - u_A^- \\ EP_A(quit_A,\cdot) = &\ 0 \end{aligned} \tag{3.23}$$

So:

$$EP_A(m_1,\cdot) = \beta * (u_A^+) - u_A^- \geqslant 0 \Leftrightarrow \beta \geqslant u_A^-/u_A^+ \tag{3.24}$$

That implies that for values of β below u_A^-/u_A^+, entity A's *best response* to B is to quit the protocol at round one. See Figure Fig. 3.8(top–left) for a graphical representation of entity's A best–response function.
Likewise, the following expressions are also evaluated:

$$\begin{aligned} EP_B(m_2) = &\ \delta * (u_B^+(3) - u_B^-) + (1-\delta) * (-u_B^-) \\ = &\ \delta * (u_B^+(3)) - u_B^- \\ EP_B(quit_B) = &\ 0 \end{aligned} \tag{3.25}$$

So:

$$EP_B(m_2) = \delta * (u_B^+(3)) - u_B^- \geqslant 0 \Leftrightarrow \delta \geqslant u_B^-/u_B^+(3) \tag{3.26}$$

That implies that for values of δ below $u_B^-/u_B^+(3)$, entity B's *best response* to A is to quit the protocol. See Figure Fig. 3.8 (top–right) for a graphical representation of entity's B best–response function.

In Fig. 3.8 (bottom) we represent the two best–response functions for entities A and B and the points of intersection. Such points represent all possible equilibria in the game of imperfect information described in Fig. 2.2.

The points circled in Fig. 3.8(bottom) correspond to the following probabilistic strategy profiles:

$N_1 \equiv (p = 0,\, q = 0,\, \beta^* = 0,\, \delta^* = 0)$
$N_2 \equiv (p = 0,\, q = 1,\, \beta^* = u_A^-/u_A^+,\, \delta^* = u_B^-/u_B^+(3))$
$N_3 \equiv (p = 0,\, q = 1,\, \beta^* = 1,\, \delta^* = 1)$

Where:

N_1 represents *pure* strategies $(quit_A,\cdot)$ and $(quit_B)$ for entities A and B respectively.

N_2 represents *mixed* strategies $\left(m_1, m_3, 1, u_B^-/u_B^+(3)\right)$ and $(m_2, u_A^-/u_A^+)$ for entities A and B respectively.

N_3 represents *pure* strategies (m_1, m_3) and (m_2) for entities A and B respectively. □

Equilibrium N_3 had already been identified by Buttyán et al.'s analysis.

However, we find their analysis to be both simplistic and unrealistic. The new equilibria found, N_1 and N_2, stem from a more complex and complete analysis. Considering factors such as trust amongst participants or network reliability results in two new equilibria. In N_1, entities simply do not find a way to interact in equilibrium and both decide to quit. On the other hand, in N_2, the equilibrium depends on the values of $item_A$ and $item_B$ and the cost of the exchange. We believe this to be a more accurate result, as entities, in real scenarios, would indeed behave differently depending on the value of the items to be exchanged.

3.4 Conclusions

A number of conclusions have risen at the end of this work. We present them from a series of different angles:

A more flexible and complete analytical framework:

Syverson's rational–exchange protocol was first analyzed by Buttyán et al. in [Buttyán and Hubaux, 2004], using the definition of rationality within a game–theoretical model. The model proposed was based on the representation of rational–exchange protocols as dynamic games of *perfect* information. In Section 2.6 we identified a few drawbacks and limitations, propelling an extension to the model using further and more advanced concepts within Game Theory. In particular, using games of *imperfect* information, we have formalized and analyzed the effect that factors such as participant reputation, protocol robustness and unpredictable participant behavior, have on the outcome of the protocol execution.

To the best of our knowledge, it is the first time that such parameters have been formalized and considered when analyzing a security protocol. In particular, the conjecture about the protocol's robustness is specially important, as what we have presented in this chapter is a novel way to express participant unpredictable behavior.

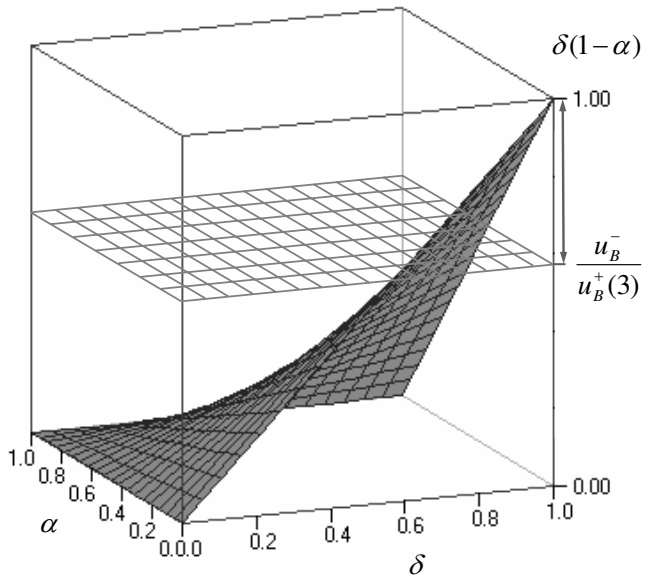
Moreover, our new extended model presents high levels of flexibility to formalize other factors apart from the ones analyzed in this chapter, for example parameters to do with network reliability.

Individually rational outcomes: From a *Decision Theory* point of view (Section 3.3.3), the premise of individual rationality is based on the assumption that each individual is committed to achieving the best outcome for itself, regardless of the effect doing so has on others. From this angle, we have obtained interesting results such as for example, in some instances, a participant B

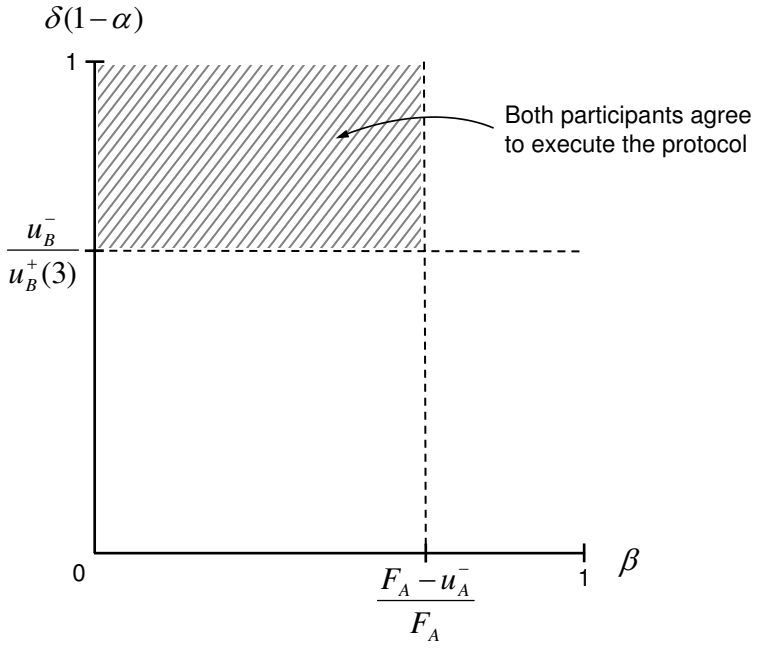
would be inclined to participate of Syverson's protocol even it is very likely that the protocol presents flaws that the other participant could take advantage of.

Nash Equilibria: From a *Game Theory* point of view, the concept of Nash Equilibrium had to be extended to evaluate *probabilistic strategy profiles*. The conclusions derived when computing all Nash Equilibria in the new protocol game are very determinant: When the protocol is not 100% flawless, although the exchange can take place, such an outcome does not constitute a Nash Equilibrium, which means that entities are motivated to deviate from the protocol description and it is beyond the formal model to asses the reasons why they would not do so. By contrast, if the protocol is considered to be completely safe, three points constitute Nash equilibria in the protocol game: when both entities quit the protocol without carrying the exchange, when both entities behave correctly, or when the conjectures about each other's behavior take a certain value (u_A^-/u_A^+, $u_B^-/u_B^+(3)$). These values depend on the ratio between the cost and the profit entities expect to gain when exchanging items $item_A$ and $item_B$.

Repeated Scenarios: Finally, although the current model described in this chapter has only served to analyze single instances of a rational exchange protocol, we believe this is only part of its significance, as it is necessary to analyze single executions before studying player's strategies in iterated scenarios. The work in this chapter represents the basis for any further analysis in that direction.



(a)



(b)

Figure 3.2: Feasible outcomes for Syverson's protocol game.

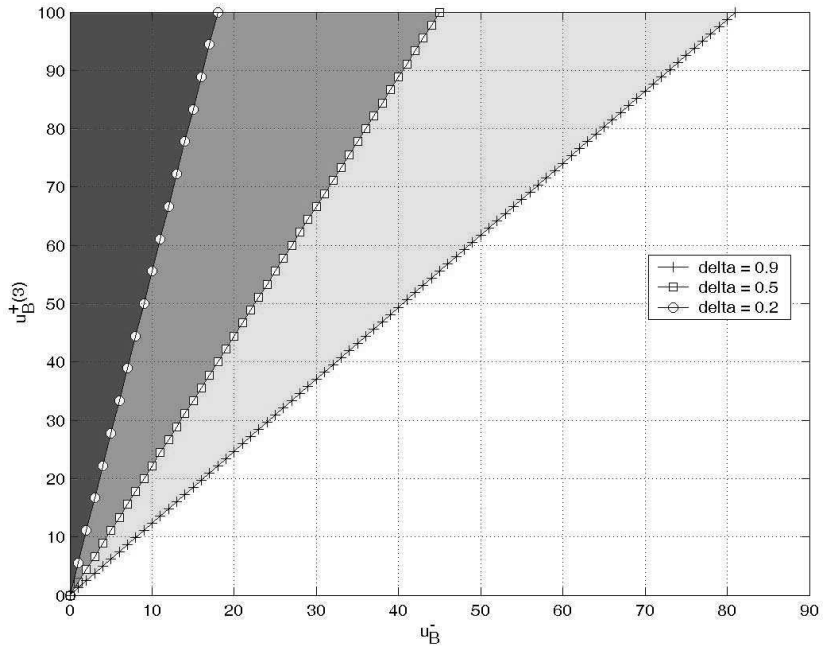


Figure 3.3: For different values of δ (level of trust on merchant *Alice* to deliver message 3), the shadowed area gives the set of values for which *Bob* would rationally choose to participate in the protocol as the profit is sufficiently higher than the cost.

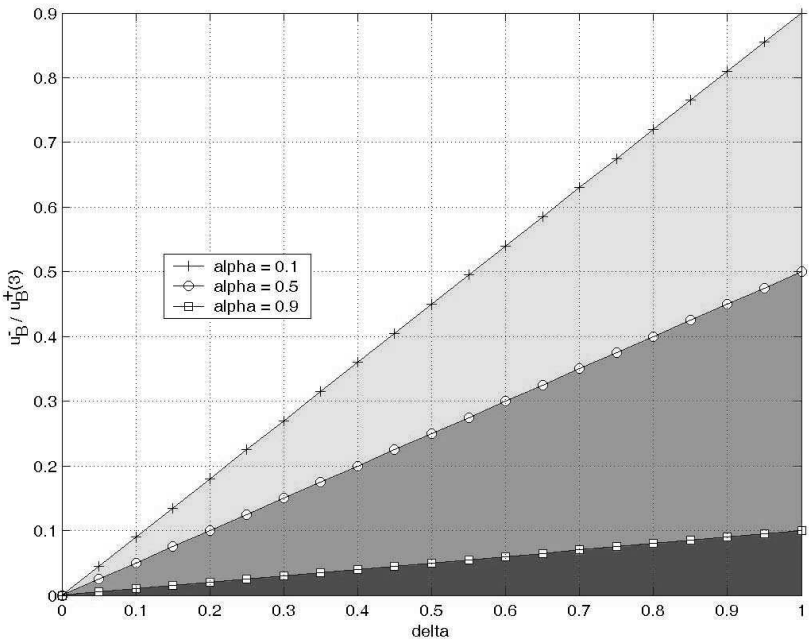


Figure 3.4: The level of trust on the protocol design, coupled with the level of trust on merchant *Alice* determines the threshold for the ratio cost/profit. For a poor level of trust on the protocol design, only a high profit and a high level of trust on *Alice* will motivate B not to quit the protocol.

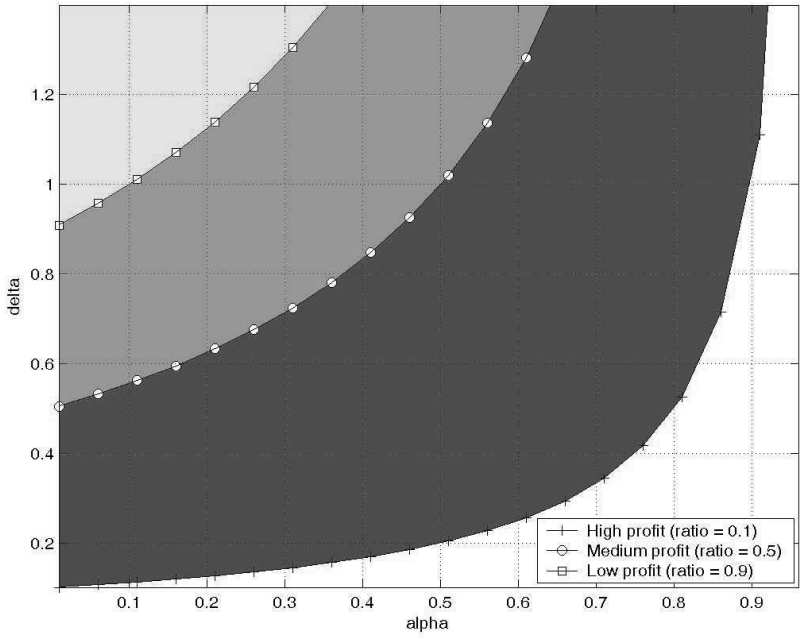


Figure 3.5: When the benefit expected is high, *Bob* will be part of the protocol even when it is quite probable that the protocol presents vulnerabilities. When the cost is 90% of the profit, both the protocol and the merchant must offer the maximum guarantee. The shadowed areas represent the values for which *Bob* will rationally choose to be part of the protocol.

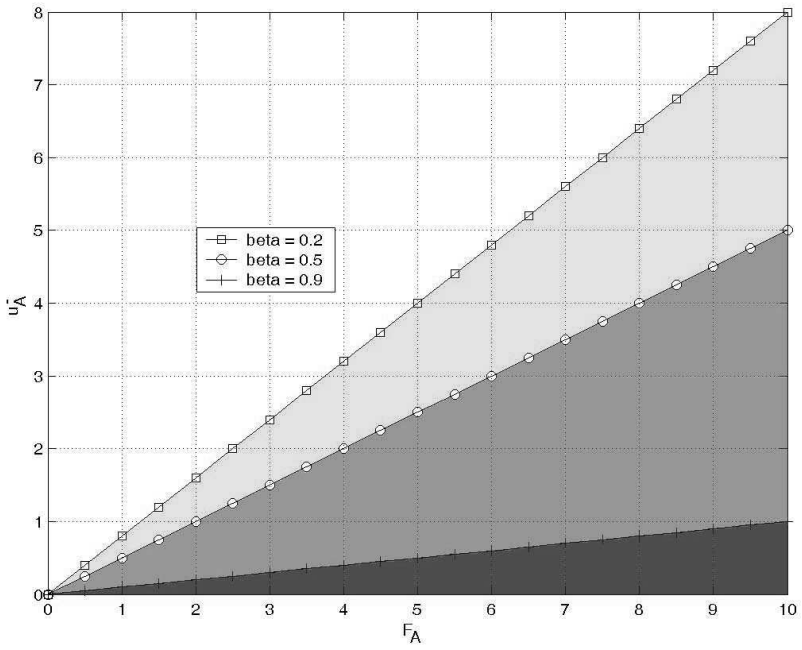


Figure 3.6: The limit for the ratio between cost and penalty decreases as β increases. In other words, committing fraud is not attractive when the cost of the penalty is much greater than the cost of sending $item_A$ and if B is increasingly more likely to detect the fraud.

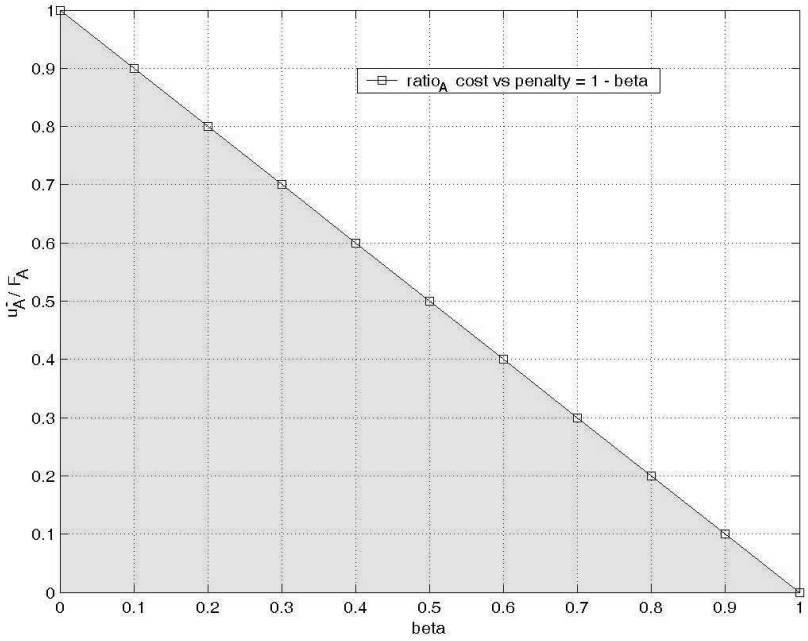


Figure 3.7: Note how when $\beta = 0$ (B is definitely quitting at stage two), *Alice* is rationally forced to behave honestly unless the ratio is 1 (cost is equal to the value of the penalty). Similarly, when $\beta = 1$ (B is definitely sending m_2 at stage two), *Alice* would only misbehave if the cost exceeds the value of the penalty.

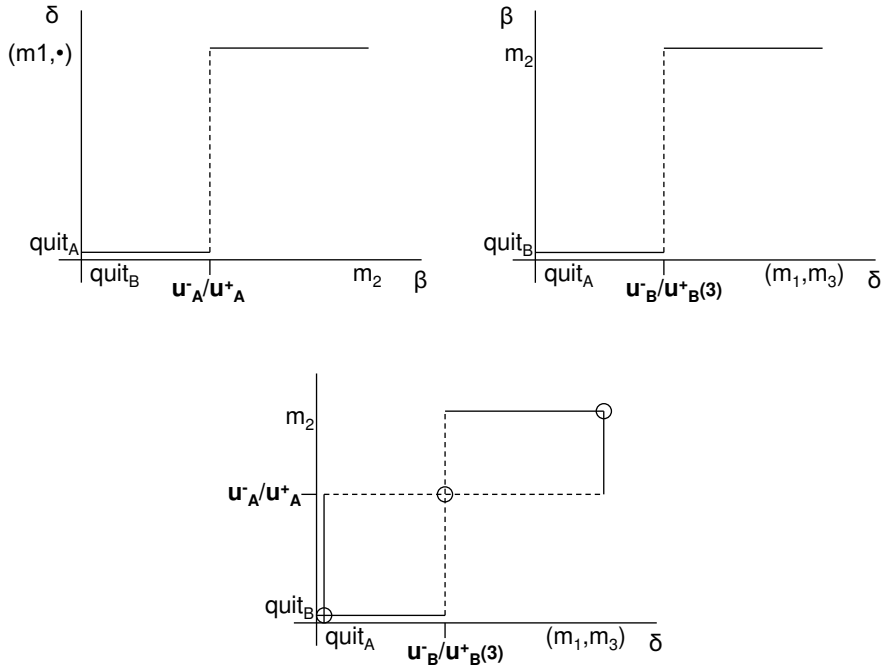


Figure 3.8: Best–response functions for entity A (top–left), B (top–right), and the intersection of both when interacting during Syverson protocol (bottom).

Chapter 4

A Model based on Bayesian Games

4.1 Introduction

In previous chapters we described how some aspects of Buttyán et al.'s model seemed too restrictive and unrealistic. Moreover, we extended such a formalism to capture issues of a rational protocol execution never considered before in formal analysis. In this chapter we intend to further elaborate on the extensions of the model proposed by Buttyán et al.

4.1.1 Chapter Overview

In particular, we will propose and extension to the model based on representing rational–exchange protocols as games of *incomplete information or Bayesian* games. In the following sections, we will analyze Syverson's protocol in order to illustrate our proposal.

The so–called games of incomplete information or Bayesian games are those in which players do not have all the information about their opponents; for example, other players' payoff functions. In such a case, player's conjectures over other participants' payoff values can be taken into account when making the optimal decision at any given point during the protocol execution.

Several Game Theory results allow us to predict the outcome of such a game and therefore the outcome of the protocol execution it represents. Despite the analysis becoming more complex than by using basic Game Theory, we find it more realistic and more informative.

A rational–exchange protocol can be defined in terms of the perfect Bayesian equilibrium of a dynamic game of incomplete information. Our model enables us to

consider different classes of protocol parties within the protocol game (e.g. honest and dishonest parties) as well as modeling attributes such as reputation or any other belief that could have an effect on the protocol outcome. The resulting new model let us reason about exchange protocols from the point of view of Bayesian rationality, a notion that may be in some scenarios more appropriate than that defined in terms of Nash equilibrium.

4.1.2 Chapter Organization

The chapter is organized as follows. In Section 4.1.1, we give an overview of our approach. Section 4.2 is devoted to formally present the new model. Section 4.3 serves to illustrate how this framework can be used to analyze Syverson's protocol and finally, in Section 4.4 we describe the conclusions to this work.

4.2 Extended Model Based on Dynamic Games of Incomplete Information

Please refer to Section A.5 for a detailed description of the most relevant concepts in Bayesian games: player's type, player's beliefs and the notion of perfect Bayesian equilibrium –a generalization of Nash equilibrium for this kind of games. In this section, we will give an informal introduction to some of these concepts.

In a Bayesian game, each player is allowed to have some private information that affects the overall game play but which is not known by others. This information is usually related to their payoff values (what players receive at the end of the game, depending on what strategies all players play). Players have initial beliefs about the type of their opponents and can update their beliefs on the basis of the actions they have played. Both concepts are informally defined in the next paragraphs:

- *Player's type.* The type of a player univocally determines that player's payoff function, so that different types will be associated with different payoff functions. A Bayesian game is modeled by introducing Nature as a player in the game. Nature randomly chooses a type for each player according to the probability distribution across each player's type space.

- *Player's beliefs.* Every player defines a belief system over the type of player their opponents are. Player's set of beliefs will assist them in the process of choosing the *best–response* strategies to confront other participants.

The main advantage of this formalism is that it allows us to introduce diverse forms of uncertainty in the analysis. For instance, in the simplest case, we can

consider that players can be either *honest* or *dishonest*, thus having a type space with just two elements. Of course, executing a protocol against a dishonest party will surely be different that dealing with an honest one. This fact is modeled by means of different payoff functions for each type, in such a way that the strategies to follow during the protocol run will be different in each case.

Furthermore, types can also be useful when considering features regarding network behavior. In Buttyán et al.'s model, the network is considered reliable, so it has a fixed strategy: To deliver messages. However, it is not difficult to conceive a more realistic scenario in which messages are not always properly delivered. In this case, the effects of an unreliable network can be easily incorporated into the analysis by considering different types of network.

In the next section, we will discuss how this extension of basic Game Theory can be incorporated into Buttyán et al.'s formalism to analyze rational–exchange protocols in a more powerful manner.

4.2.1 Bayesian Extended Model

We will extend the definition of Buttyán et al.'s model by the use of:

(I) *Incomplete information in protocol games:* Represented by a type space for every player in the game.

(II) *Belief System:* Strategies based on probability distribution functions over players's type space.

The concept of *protocol game* was introduced by Buttyán et al. as a way to represent an exchange protocol (see Section 2.3.1). The following definitions will serve to describe the way in which a two–entity exchange protocol Π can be represented as a *protocol game of incomplete information.*

Definition 4.2.1 (Bayesian protocol game)**.** *Given a two–entity exchange protocol* $\Pi = \langle P, \mathcal{O}, T \rangle$*, the following protocol game of incomplete information, denoted as* G^B_Π*, is defined to represent such a protocol:*

$$G^B_\Pi = \langle P, A, Q, p, (\mathcal{I}_i)_{i \in P}, (\preceq_i)_{i \in P}, \mathcal{T} \rangle \tag{4.1}$$

where:

- $P = \{P_1, P_2\}$ *is a set of players.*
- A *and* Q *are the set of actions and set of action sequences respectively, satisfying:*

a1. $\epsilon \in Q$, *where* ϵ *is the empty sequence.*

a2. if $q = (a_k)_{k=1}^{t} \in Q$ *and* $0 < w < t$, *then* $q = (a_k)_{k=1}^{w} \in Q$

a3. if $q = (a_k)_{k=1}^{t} \in Q$, $0 < t < n$ *and* $a \in A$ *then* $q \cdot a$ *denotes the action composed by* q *followed by* a.

a4. A finite sequence of actions $q \in Q$ *is said to be terminal if there is no* $a \in A$, *such that* $q \cdot a \in Q$. *The set of terminal sequences of actions is denoted by* Z.

a5. $A(q) = \{a \in A : q \cdot a \in Q\}$ *denotes the set of available actions after action sequence* $q \in Q \backslash Z$.

a6. In particular, for every $q = (a_k)_{k=1}^{t-1} \in Q \backslash Z$, $A(q) = \{quit,\ m_t,\ gm_t\}$ *where,*

- m_t *represents action send message* m_t, *where* m_t *is as described in the protocol.*
- gm_t *represents action send message* gm_t, *where* gm_t *represents a deviation from message* m_t.

a7. p *is the player function. It assigns a player* $p(q) \in P$ *to every non–terminal sequence* $q \in Q \backslash Z$. *The interpretation is that player* $p(q)$ *has the turn after the sequence of actions* q.

- $\mathcal{I}_i$ *is an information partition for player* $P_i \in P$. *It is a partition of the set* $\{q \in Q \backslash Z : p(q) = i\}$ *satisfying:*

  b1. If sequences q *and* q' *are in the same information set* $I_i \in \mathcal{I}_i$, *then* $A(q) = A(q')$.

  b2. If sequences q *and* q' *are in the same information set* $I_i \in \mathcal{I}_i$, *then player* P_i *is forced to define a probability distribution* α_i *over every action sequence in* I_i, *so* $\sum_{q \in I_i} \alpha_i(q) = 1$.

- *A preference relation* $\preceq_i$ *is defined for each player* $P_i \in P$ *over set* Z.

- $\mathcal{T}_i$ *denotes the type space for each player* P_i. *It is assumed that each player* $P_i \in P$ *has a type* $T_i \in \mathcal{T}_i$. $\mathcal{T}$ *represents a type–profile space defined as* $\mathcal{T} = \mathcal{T}_1 \times \mathcal{T}_2$. *A type profile* $T \in \mathcal{T}$ *is a tuple of types* (T_1, T_2), *one for each player, which univocally determines the type of every player involved in the protocol game.*

- *Finally, a probability distribution function* θ_i *is defined over each type space* $\mathcal{T}_i$, $i \in \{1, 2\}$.

4.3 A Bayesian Analysis of Syverson's Protocol

To illustrate the proposed Bayesian model, we will analyze Syverson's protocol (Fig. 2.1), although assuming a richer set of environmental hypotheses.

We note that in such a protocol entity B always plays a more risky strategy than player A. Entity B is the first in sending her item, $item_B$, hoping that A would not misbehave and that the exchange would take place successfully. Moreover, it is very difficult for B to be able to misbehave, as m_2 is always a *fresh* message (contains m_1) and it is also a token for non–repudiation of origin, so B will always be proven responsible for any malicious action. By contrast, A could not be held responsible for any malicious action until the end of protocol and, in most cases, provided B gains access to the proof issued by A. Furthermore, A gains access to $item_B$ before B gains access to $item_A$. It is B then who has to carefully analyze A's reputation, credibility, surroundings, and her current and real intentions.

In such a context, we will only consider the simplest scenario, that in which participant A can be either an honest or a dishonest protocol party, while B is always honest.

As in a regular Syverson's protocol game, the network is considered to be reliable, so it plays following a fixed strategy delivering all messages.

Throughout this chapter we will refer to Syverson's Bayesian protocol game derived from Syverson's protocol description as G^B_{Sy}. We will develop the appropriate specifications for players A and B in G^B_{Sy}.

4.3.1 Player Types

Definition 4.3.1 (Syverson's type–profile space)**.** *Let $\mathcal{T} = \mathcal{T}_A \times \mathcal{T}_B$ be the type–profile space in G^B_{Sy}, where $\mathcal{T}_A = \{A_h, A_d\}$ and $\mathcal{T}_B = \{B_h\}$ are the type spaces for players A and B, respectively.*

By convention, subscript h denotes an honest participant, while d represents a dishonest one.

The following are the probability distributions of the different types of the different entities in relation with each other.

Definition 4.3.2 (Syverson's type–profile probability distribution)**.** *We define the following probability distribution θ over $\mathcal{T}_A$:*

$$\begin{aligned} \theta_h = &\ \text{Prob}(A_h|B) \\ \theta_d = &\ \text{Prob}(A_d|B) \\ &\ s.t.\ \theta_h + \theta_d = 1 \end{aligned} \tag{4.2}$$

Note that $\text{Prob}(B|A_h) = \text{Prob}(B|A_d) = 1$, since $\mathcal{T}_B$ has only one element.

4.3.2 Player Strategies

Since each type is associated to a (possibly different) payoff function, uncertainty about the opponent's type has severe implications in the decision-making process, particularly in computing the best strategy during the protocol execution.

Definition 4.3.3 (Syverson's player set of actions)**.** *The set of actions available to players A and B in G^B_{Sy} is:*

$$A = A_A \cup A_B$$

where:

$$\begin{aligned} A_A = &\ \{m_1, m_3, gm_1, quit_A\} \\ A_B = &\ \{m_2, quit_B\} \end{aligned} \tag{4.3}$$

and,

- *$quit_X$ represents action quit the protocol for an entity X.*
- *m_k represents action send message m_k.*
- *gm_k represents action send garbage gm_k.*

Definition 4.3.4 (Player A pure strategies)**.** *A pure strategy for player A is a tuple:*

$$s_A = \big((s_1, s_3)_d, (s_1, s_3)_h\big) \tag{4.4}$$

where:

- *$s_1 \in A_A$ defines an action at round 1 of the protocol, and*
- *$s_3 \in A_A$ defines an action at round 3.*
- *The first component in s_A represents a strategy for type A honest and the second one for A dishonest.*

Alternatively, player B has two possible pure strategies.

Definition 4.3.5 (Player's B pure strategies)**.** *A pure strategy for player B is:*

$$s_B = \{quit_B, m_2\} \tag{4.5}$$

Definition 4.3.6 (Syverson's strategy profile)**.** *A strategy profile in G^B_{Sy} is a vector $s = (s_A, s_B)$ of individual strategies, one for each player.*

Note that specifying a strategy profile determines univocally the outcome of the game.

4.3.3 Player's Beliefs

The following probability distributions represent the set of beliefs each entity holds over their opponent's type and the set of actions at each particular stage of the protocol.

Definition 4.3.7 (Player A belief system)**.** *Let σ_B be a probability distribution over the set of actions A_B defined as:*

$$\sigma_B \in \Delta(A_B) \text{ satisfying: } \sigma_B(m_2) + \sigma_B(quit_B) = 1 \tag{4.6}$$

Note that, ($\Delta(A_B)$ denotes the space of probability distributions over the set A_B).

By contrast, B's attempt to anticipate A's behavior in the game is represented by the following functions:

Definition 4.3.8 (Player B belief system)**.** *At stage two of the protocol, B's beliefs are represented by the following probability distribution functions over A's set of actions:*

$$\alpha_h, \alpha_d \in \Delta(A_A) \tag{4.7}$$

satisfying:

$$\begin{aligned} \alpha_h(gm_1) + \alpha_h(m_1) &= 1 \\ \alpha_d(gm_1) + \alpha_d(m_1) &= 1 \end{aligned} \tag{4.8}$$

and,

$$\beta_h, \beta_d \in \Delta(A_A) \tag{4.9}$$

satisfying:

$$\begin{aligned} \beta_h(quit_A|m_1) + \beta_h(m_3|m_1) &= 1 \\ \beta_d(quit_A|m_1) + \beta_d(m_3|m_1) &= 1 \end{aligned} \tag{4.10}$$

We can assume that B also holds the following beliefs representing the fact that, when A has cheated, A will never sign the token of non–repudiation of origin to claim responsibility for such misbehavior; instead, A will always quit the protocol. Therefore:

$$\begin{aligned} \text{Prob}[quit_A|gm_1 \wedge A_h] &= 1 \\ \text{Prob}[m_3|gm_1 \wedge A_h] &= 0 \\ \text{Prob}[quit_A|gm_1 \wedge A_d] &= 1 \\ \text{Prob}[m_3|gm_1 \wedge A_d] &= 0 \end{aligned} \tag{4.11}$$

4.3.4 Payoff Functions

As stated before, one of the key points of Bayesian games is the fact that each type of player is associated with a possibly different payoff function. Syverson's payoff functions for entities A and B in G^B_{Sy} are denoted as:

$$U_A, U_B : \mathcal{T}_A \times A_A \times A_A \times A_B \to \mathbb{R}$$

Fig. 4.1 represents Syverson's Bayesian protocol game G^B_{Sy} in extensive–form. For each branch in the tree, a payoff value is defined representing the total outcome that players A and B obtain when taking such a path. Payoff values are defined as in Section 2.3.3. Note that for entities type A_d, completing the protocol produces a lower income than uncomplete runs.

As an example, $U_A(A_d, m_1, m_3, m_2)$ represents the payoff obtained by a participant A, type *dishonest*, when A takes actions m_1 and m_3 at steps 1 and 3 of the protocol game G^B_{Sy} and B takes action m_2 at stage two.

$$U_A(A_d, m_1, m_3, m_2) = u^+_{A_d} - u^-_{A_d} \tag{4.12}$$

By contrast, $U_A(A_d, m_1, quit_A, m_2)$ represents the payoff obtained by a participant A, type *dishonest*, when A takes actions m_1 and $quit_A$ at steps 1 and 3 of the protocol game and B takes action m_2 at stage two.

$$U_A(A_d, m_1, quit_A, m_2) = u^+_{A_d} \tag{4.13}$$

4.3.5 Expected Payoffs

Definition 4.3.9 (Syverson's expected payoff values)**.** *We denote by $EP(i, s_i)$ the expected payoff for player i when following strategy s_i in G^B_{Sy}.*

Entity A's expected payoff

We first consider the expected payoffs when players follow pure strategies. For every strategy profile $s_A = \Big((s_1, s_3)_d, (s_1, s_3)_h\Big)$ for player A, the expected payoff value is:

$$\begin{aligned} EP(A_h, s_A) = \; & \sigma_B(m_2) * U_A(A_h, (s_1, s_3)_h, m_2) + \\ & \Big(1 - \sigma_B(m_2)\Big) * U_A(A_h, (s_1, s_3)_h, quit_B) \end{aligned} \tag{4.14}$$

$$\begin{aligned} EP(A_d, s_A) = \; & \sigma_B(m_2) * U_A(A_d, (s_1, s_3)_d, m_2) + \\ & \Big(1 - \sigma_B(m_2)\Big) * U_A(A_d, (s_1, s_3)_d, quit_B) \end{aligned} \tag{4.15}$$

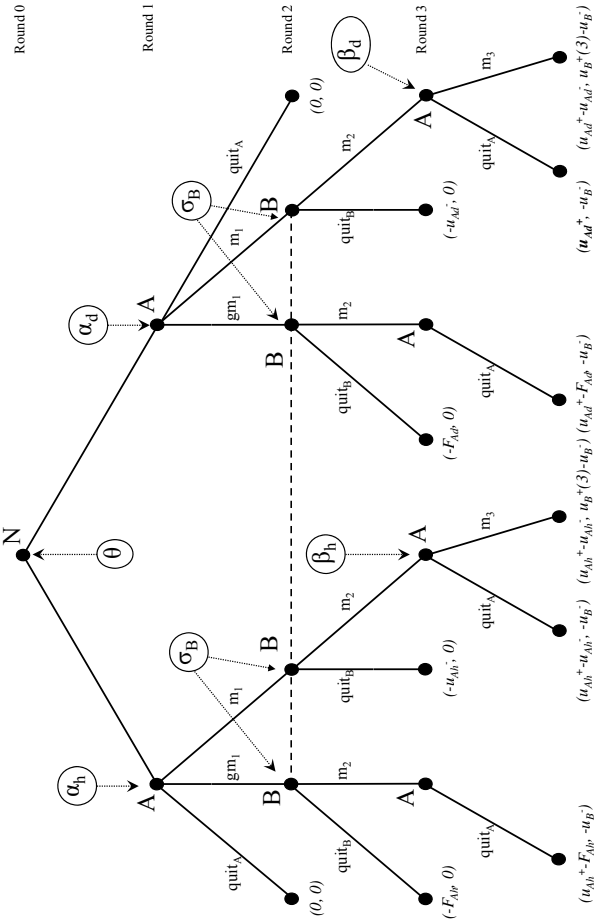


Figure 4.1: Syverson's Bayesian game in extensive–form.

Entity B's Expected Payoff

In the case of player B, we have:

$$EP(B, quit_B) = \quad 0 \tag{4.16}$$

$$\begin{aligned} EP(B, m_2) = \quad & \theta_h * \Big[\alpha_h(gm_1) * (-u_b^-) + \alpha_h(m_1) * \\ & \Big(\beta_h(quit_A|m1) * (-u_B^-) + \\ & (1 - \beta_h(quit_A|m_1)) * (u_B^+(3) - u_B^-)\Big)\Big] + \\ & \theta_d * \Big[\alpha_d(gm_1) * (-u_b^-) + \alpha_d(m_1) * \\ & \Big(\beta_d(quit_A|m1) * (-u_B^-) + \\ & (1 - \beta_d(quit_A|m_1)) * (u_B^+(3) - u_B^-)\Big)\Big] \\ = \quad & -u_B^- + u_B^+(3) * \Big(\theta_h * \alpha_h(m_1) * \\ & \beta_h(m_3|m_1) + \theta_d * \alpha_d(m_1) * \beta_d(m_3|m_1)\Big) \end{aligned} \tag{4.17}$$

()

Now, we will denote:

$$L_B = \theta_h * \alpha_h(m_1) * \beta_h(m_3|m_1) + \theta_d * \alpha_d(m_1) * \beta_d(m_3|m_1) \tag{4.18}$$

Note that we can conclude that:

$$EP(B, m_2) \geqslant EP(B, quit_B) \Leftrightarrow -u_B^- + u_B^+(3) * L_B \geqslant 0 \tag{4.19}$$

or, equivalently:

$$EP(B, m_2) \geqslant EP(B, quit_B) \Leftrightarrow L_B \geqslant u_B^-/u_B^+(3) \tag{4.20}$$

Therefore, B would play action send message m_2 at round 2 of the protocol, instead of action $quit_B$, iff a linear combination, denoted as L_B, of her set of beliefs verifies the relation given by expression (4.20).

4.3.6 Perfect Bayesian Equilibrium

Drew Fudenberg and Jean Tirole formally defined the perfect Bayesian equilibrium (PBE) for extensive Bayesian games in [Fudenberg and Tirole, 1991b]. PBE adds to Nash equilibrium the requirement that players choose optimally given their beliefs about the rest of the game. In extensive Bayesian games of incomplete information, each player is not only aware of the informational uncertainties over the other participants, but also analyzes their implications. Thus, each player looks for the best response, anticipating other party's reaction. Please refer to Section A.5.3 for

an extended exposition on the topic.

Bayes Requirements

We will formally define candidates to be a PBE in the extensive–form game G^B_{Sy} depicted in Fig. 4.1:

Definition 4.3.10 (PBE candidate)**.** *Candidates to be PBE in G^B_{Sy} will be of the form strategy-belief profile, denoted as $(S;\rho)$ where:*

$$S = (s_A, s_B) \quad \text{with} \quad s_A \in S_A, s_B \in S_B.$$

and

$$\rho = (\alpha_h, \alpha_d, \beta_h, \beta_d, \sigma_B, \theta)$$

is a tuple containing both participant's belief systems.

Definition 4.3.11 (PBE)**.** *A given profile $(S^*;\rho^*)$ represents a Perfect Bayesian equilibrium in G^B_{Sy} if it defines a set of strategies such that, for every player i and every information set I_i, player's i strategy is her best response to the opponent's strategy, given her beliefs at set $\mathcal{I}_i$.*

In other words, to be a PBE, the strategies and beliefs defined by $(S^*;\rho^*)$ must satisfy what it is called *Bayes requirements 1 to 4*. Please find formal description of these requirements in Appendix A.

PBE Candidates in Syverson's Bayesian Protocol Game

To assist us in finding the appropriate PBE candidates in our particular case, we will formulate the following set of requirements:

To be a PBE candidate for Syverson's Bayesian protocol game, the strategies and beliefs defined by $(S^*;\rho^*) = (s^*_A, s^*_B; \alpha^*_h, \alpha^*_d, \beta^*_h, \beta^*_d, \sigma^*_B, \theta^*)$ must satisfy the following formulæ:

- For every strategy s_A of A_d and for all α_d and β_d probability distributions:

$$EP(A_d, s^*_A) * \alpha^*_d(s^*_1) * \beta^*_d(s^*_3) \geqslant EP(A_d, s_A) * \alpha_d(s_1) * \beta_d(s_3) \tag{4.21}$$

- For every strategy s_A of A_h and for all α_h and β_h probability distributions:

$$EP(A_h, s^*_A) * \alpha^*_h(s^*_1) * \beta^*_h(s^*_3) \geqslant EP(A_h, s_A) * \alpha_h(s_1) * \beta_h(s_3) \tag{4.22}$$

- And finally, for all σ_B probability distribution:

$$\begin{aligned} &EP(B, quit_B) * \sigma_B^*(quit_B) + EP(B, m_2) * \sigma_B^*(m_2) \geqslant \\ &EP(B, quit_B) * \sigma_B(quit_B) + EP(B, m_2) * \sigma_B(m_2) \end{aligned} \tag{4.23}$$

Since $EP(B, quit_B) = 0$, expression (4.23) can be reduced to:

$$EP(B, m_2) * \sigma_B^*(m_2) \geqslant EP(B, m_2) * \sigma_B(m_2) \tag{4.24}$$

for all σ_B probability distribution.

Note that, when $EP(B, m_2) > 0$, expression (4.24) is satisfied iff $\sigma_B^*(m_2) = 1$. Likewise, when $EP(B, m_2) < 0$, expression (4.24) is satisfied iff $\sigma_B^*(m_2) = 0$.

In previous calculations –see expressions (4.18) and (4.20)–, we have computed a value L_B which represents a threshold to determine a value for $EP(B, m_2)$. B's best response to all possible A's strategies will be determined by the value of the linear combination L_B. Observe that, this is a stronger result than equation (4.20) as we are now considering all possible mixed strategies for A and B.

Therefore, we will present the following strategy-belief profile $(S^*; \rho^*)$ as our first candidate to PBE for the game in Fig. 4.1:

$$(S^*; \rho^*) = \Big((m_1, m_3)_h, (m_1, m_3)_d, m_2; \alpha_h^*, \alpha_d^*, \beta_h^*, \beta_d^*, \sigma_B^*, \theta^*\Big) \tag{4.25}$$

with $\sigma_B^*(m_2) = 1$ and $L_B^* \geqslant u_B^-/u_B^+(3)$, being L_B^* the linear combination defined in equation (4.18).

Note that the presented candidate expresses the participant B's intention to succeed in the exchange of $item_A$ and $item_B$.

The next PBE candidate to be considered represents the set of strategies for A and B when A believes that B is likely to leave the protocol at round 2. Then, A's best response is to quit too:

$$(S^0; \rho^0) = \Big((quit_A, quit_A)_h, (quit_A, quit_A)_d, quit_B; \alpha_h^0, \alpha_d^0, \beta_h^0, \beta_d^0, \sigma_B^0, \theta^0\Big) \tag{4.26}$$

with $\sigma_B^0(m_2) = 0$ and $L_B^0 < u_B^-/u_B^+(3)$, being L_B^0 the linear combination defined in equation (4.18).

PBE in Syverson's Bayesian Protocol Game

In order to prove that each one of the previous profiles conforms a PBE, they must satisfy Bayesian requirements 1 to 4. The reader will find formal description of these requirements in Section A.5.3 of the Appendix.

We will commence with the case of $(S^*;\rho^*)$ defined in equation (4.25), as $(S^0;\rho^0)$ (equation (4.26)) can be trivially derived from the following steps.

Lemma 4.3.1. *The strategy-believe profile $(S^*;\rho^*)$ in Syverson's Bayesian game satisfies Bayes requirement 1.*

Proof. It is required that each player A and B assigns a probability distribution over the nodes in each information set $I_i \in \mathcal{I}_i$.

The set $I_B \in \mathcal{I}_B$ identified at round 2 of the protocol is the only information set with more than one element. Indeed, the requirement 1 is satisfied as, considering the belief profile ρ^*, B defines the distributions α_d^* and α_h^* to satisfy $\alpha_h^*(gm_1)+\alpha_h^*(m_1)=1$ and $\alpha_d^*(gm_1)+\alpha_d^*(m_1)=1$. Since $\theta_h^*+\theta_d^*=1$ (see equation (4.2)), then

$$\begin{array}{l}\theta_h^* * \alpha_h^*(gm_1)+\theta_h^* * \alpha_h^*(m_1)+\\ \theta_d^* * \alpha_d^*(gm_1)+\theta_d^* * \alpha_d^*(m_1)=1\end{array} \tag{4.27}$$

□

Lemma 4.3.2. *The strategy-belief profile $(S^*;\rho^*)$ in Syverson's Bayesian game satisfies Bayes requirement 2.*

Proof. Requirement 2 forces B to be rational and to behave accordingly to its beliefs once B has reached the information set I_B and for the rest of the game.

Let CG_{Sy}^B be the *continuation game* starting at the information set $I_B \in \mathcal{I}_B$, and let $\rho^*(I_B)$ be the conditional beliefs at I_B. Then, the strategy-belief profile $(S^*;\rho^*(I_B))$ must be a Nash equilibrium of the continuation game CG_{Sy}^B. Evaluating the payoff vectors we obtain:

$$EP(B, quit_B, CG_{Sy}^B)=0 \tag{4.28}$$

$$\begin{array}{rcl}EP(B,m_2,CG_{Sy}^B)\geqslant 0 & \Leftrightarrow & \theta_h * \alpha_h^*(m_1) * \beta_h^*(m_3|m_1)+\\ & & \theta_d * \alpha_d^*(m_1) * \beta_d^*(m_3|m_1)\geqslant\\ & & u_B^-/u_B^+(3)\end{array} \tag{4.29}$$

which it is, in fact, the value of L_B described in the candidate's definition (equation (4.17)).

Therefore, the profile strategy given by $(S^*;\rho^*(I_B))$ constitutes a Nash equilibrium of the continuation game CG_{Sy}^B. □

Lemma 4.3.3. *The strategy-believe profile $(S^*;\rho^*)$ in Syverson's Bayesian game satisfies Bayes requirement 3.*

Proof. Requirement 3 forces B to establish *sensible* beliefs at the on–equilibrium–path information set I_B. This set of beliefs must be determined from the strategy profile according to Bayes' rule.

Therefore, B has to establish probability distributions α_d^* and α_h^* in terms of the different actions that A can take at round 1 of the protocol. This is, if B believes that A_d would choose the action of sending m_1 with probability q_{d1}, the action of sending gm_1 with probability q_{d2} and action $quit_A$ with probability $1 - q_{d1} - q_{d2}$, then $\alpha_d^*(m1)$ and $\alpha_d^*(gm_1)$ must take the following values:

$$\alpha_d^*(m_1) = \frac{q_{d1}}{(q_{d1} + q_{d2})} \qquad \alpha_d^*(gm_1) = \frac{q_{d2}}{(q_{d1} + q_{d2})} \tag{4.30}$$

Likewise, B is forced to define:

$$\alpha_h^*(m_1) = \frac{q_{h1}}{(q_{h1} + q_{h2})} \qquad \alpha_h^*(gm_1) = \frac{q_{h2}}{(q_{h1} + q_{h2})} \tag{4.31}$$

□

Lemma 4.3.4. *The strategy-believe profile $(S^*; \rho^*)$ in Syverson's Bayesian game satisfies Bayes requirement 4.*

Proof. Requirement 4 forces B to establish *sensible* beliefs at any off–equilibrium–path information set.

There are no information sets off the equilibrium path, so requirement 4 is trivially satisfied. □

Theorem 4.3.1. *The strategy-believe profile $(S^*; \rho^*)$ (equation (4.25)) is a perfect Bayesian equilibrium of Syverson's Bayesian protocol game.*

Proof. Immediate by Definition A.5.9 and Lemmas 1 to 4. □

A similar rationale can be applied to prove the strategy-belief profile $(S_0; \rho_0)$ (equation (4.26)) does also constitute a PBE in the protocol game. In this case, however, the gains obtained are less than that obtained following $(S^*; \rho^*)$. As a result, the strategy to be followed will depend on the specific values of the items and the beliefs that player B holds about the behavior of the other party.

4.3.7 Discussion

What Theorem 4.3.1 tell us is that the strategy S^* is an equilibrium (i.e. will constitute a successful exchange for both parties), but it depends on the specific values taken by the beliefs of each participant, ρ^*. To be precise, the series of $B's$ beliefs would have to form a linear combination which would determine the best

response participant B can give to a player A. At the same time, A would create its own set of beliefs to confront participant B. There could be more than one equilibrium, as many as linear combinations L_B there are, such that $L_B \geqslant u_B^-/u_B^+(3)$.

Next we provide an example with a set of values for which an equilibrium is reached when both entities, behaving rationally, carry out a successful exchange (see Fig. 4.1). We also present a scenario where B does not see justification in exchanging items with A, as the price to pay for $item_A$ is too high to justify the risk.

Let us suppose that entity B has reasons (past experience, reputation factor, etc.) to believe that entity A is not always honest. B estimates A to be honest with probability $\theta_h = 0.85$. Let us suppose that entity B does also hold enough evidence to estimate that, when A is honest, the probability of A misbehaving at step one of the protocol is very low, i.e. the probability of sending the correct message m_1 at step one is very high, $\alpha_h(m_1) = 0.9$. Let B suppose that, by contrast, when A is dishonest, the probability of sending the correct message m_1 is also high, $\alpha_d(m_1) = 0.7$. Given the previous set of values, B computes L_B and establishes a decision-making criteria. For instance, when the value $u_B^- = 3.5$, L_B is not big enough to encourage B to follow an exchanging strategy; instead, B would be better off quitting the protocol without sending payment. A, aware of B calculations, would execute the strategy which best responds to B, this is, to also quit (see the analysis for cases 1 and 2 in Fig. 4.1). By contrast, when $u_B^- = 2.5$, the payment B is asked to pay for $item_A$ is lower and satisfies the required criteria given by expression (4.20). Entity B would then participate in the protocol following a strategy to complete the exchange successfully.

4.4 Conclusions

It has been established in previous chapters that the formal framework provided by basic (i.e. extensive–form) games and related concepts (Nash equilibrium) is somewhat narrow to capture some of the complexities that could be relevant for an in-depth analysis of rational exchange protocols.

In this chapter we provided an extended framework based on the notion of dynamic games of incomplete information and the associated PBE, that can be easily adapted to a variety of more realistic and more complex scenarios.

We introduced the notion of player type which it is a novel way in which represent participant dishonest behavior, radically different from the way it was represented in previous model based on games of imperfect information (further details are given in the section below). In this case, a dishonest player will behave rationally according

Payoffs		
$u^{+}_{Ah} = 3$	$u^{-}_{Ah} = 2$	$F_{Ah} = 6$
$u^{+}_{Ad} = 3$	$u^{-}_{Ad} = 1$	$F_{Ad} = 6$

Beliefs			
$\theta_h = 0.85$	$\alpha_h(m_1) = 0.9$	$\beta_h(m_3\|m_1) = 0.9$	$\Rightarrow$ $\mathbf{L_B = 0.76}$
$\theta_d = 0.15$	$\alpha_d(m_1) = 0.7$	$\beta_d(m_3\|m_1) = 0.7$	

Case 1:	$u^{+}_{B}(3) = 4$ $u^{-}_{B} = 3.5$	$\Rightarrow$	$\mathbf{u^{-}_{B}/u^{+}_{B}(3) = 0.875}$ Since $L_B < 0.875 \Rightarrow B$ would not participate.

Case 2:	$u^{+}_{B}(3) = 4$ $u^{-}_{B} = 2.5$	$\Rightarrow$	$\mathbf{u^{-}_{B}/u^{+}_{B}(3) = 0.625}$ Since $L_B > 0.625 \Rightarrow B$ would participate.

Table 4.1: A numerical example of a Bayesian analysis.

to a set of fixed goals (set of payoff values) different from the set of goals of an honest participant. A dishonest participant has no interest in behaving correctly as it will never represent an increase in its payoff value. In a similar way, honest participants will not misbehave, as misbehaving will not result beneficial. However, the difficulty and the level of uncertainty is placed in not knowing which type of player our opponent is and whether his behavior is driven by what we consider honest or dishonest motives.

Our new model was applied to the formal analysis of Syverson's rational exchange protocol. Few interesting results were encountered and a practical example served to illustrate possible real outcomes.

Our model is not exempted from inconveniences. In practice, dealing with incomplete information implies that participants have to take an active role when applying and implemented the model and each participant has to individually decide whether there exits a secure context in which the rational exchange could be successfully completed.

4.4.1 Differences Between the Two Proposed Models

Finally, we state the differences between the formalism described in previous Chapter 3, based on games of imperfect information, and the current Bayesian model.

On the one hand, the model described in Chapter 3, based on games of imperfect information, is used for the formal analysis of rational schemes when executed in environments of the following characteristics:

- There exists an unique set of protocol payoff values, which is the same for all

instances of the protocol. This is, for every protocol run, both entities are informed of the utility values each obtains, for every possible outcome of the scheme and, this is the same in all different executions.

- To predict the outcome of the protocol, the model will analyze participant actions at every step of the scheme, subject to environmental factors such as trust, reputation, network reliability, protocol robustness, etc.)

On the other hand, the model based on games of incomplete information, is used for the formal analysis of rational schemes when executed in environments of the following characteristics:

- There exists different sets of protocol payoff values and participant entities cannot be sure which particular set is used in each run of the protocol. In other words, entities are only aware of the benefits or loses they obtain for every possible protocol outcome, but they do not know what other entities will obtain.

- To analyze the outcome of the protocol, the model will analyze participant actions at every step of the scheme, subject to a conjecture about what set of payoff values is part of the game and, environmental factors such as trust, reputation, network reliability, protocol robustness, etc.)

The formal analysis of a given protocol will be carried out with one model or another, depending on the characteristics of the particular execution environment.

In the next chapter, we will use this proposed Bayesian model to analyze a secure P2P file sharing system, in which peers can be of two different types.

Chapter 5

Bayesian Analysis of a Secure P2P Content Distribution Protocol

5.1 Introduction

In recent years, Game Theory has in other instances been used to model nodes' behavior in P2P systems (see e.g. [Buragohain et al., 2003] and [Golle et al., 2001]). In the latter, Golle et al. analyze free-riding situations using a model based on basic Game Theory. However, they find perturbations –e.g. users joining and leaving the system– when reaching for a Nash Equilibrium. Other Game–theoretical models have also been introduced to formalize trust and reputation in secure P2P systems ([Gupta and Somani, 2005] and [Nurmi, 2006]).

5.1.1 Chapter Overview

In this chapter we show how our model, based on Bayesian games, can be an useful tool to analyze in a formal manner a P2P system. We will apply our formalism to the analysis of a secure P2P content distribution protocol ([Palomar et al., 2006b]), showing how nodes can dynamically adapt their strategies to highly transient communities and how some security aspects rest on the formal proof of notions such as rationality or best–response strategies.

The scheme we intent to analyze was first introduced by Palomar et al. in [Palomar et al., 2006b]. In this work, the authors presented a two–phase protocol ensuring file content integrity and content access control in a P2P file sharing system. We will formally analyze the second part of the scheme based on rational content access control.

5.1.2 Chapter Organization

The chapter is organized as follows. In Section 5.2 we give a brief resume of the terms most commonly used in P2P systems and particular notation used in Palomar et al.'s work. In Section 5.3 we describe the scheme focus of our analysis. Section 5.4 is devoted to the formal analysis of the protocol described and, finally, in Section 5.6 we discuss the main conclusions to this work.

5.2 P2P Terms and Specific Notation

- N is the number of nodes (also denoted peers) in the P2P network. Each node is denoted by n_i.
- Each n_i has a pair of public and private keys, denoted by K_{n_i} and $K_{n_i}^{-1}$, respectively.
- m denotes the content that a specific node wishes to publish.
- $h(m)$ represents a cryptographic hash function applied to content m.
- $E_{K_{n_i}^{-1}}(m)$ is the asymmetric encryption of content m using K_{n_i} as key. Similarly, we denote by $E_{K_S}(m)$ the symmetric encryption of message m using a secret key K_S.
- $s_{n_i}(m)$ represents n_i's signature over m, i.e.:

$$s_{n_i}(m) = E_{K_{n_i}^{-1}}(h(m)) \tag{5.1}$$

- Finally, we denote $s_{n_i}^{n_j}(m)$, n_i's signature on m concatenated with n_j's identity, i.e.:

$$s_{n_i}^{n_j}(m) = E_{K_{n_i}^{-1}}(n_j||h(m)) \tag{5.2}$$

5.3 P2P File Sharing Protocol Description

Palomar et al.'s protocol offered content integrity based on the collaboration among a fraction of peers in the system. Moreover, the model establishes a rational content access control by means of a challenge–response mechanism, whereby nodes may achieve good reputation and privileges. Contrary to classic trust systems where trust decisions are directly or indirectly given by nodes' past behavior, the scheme uses cryptographic proofs of work to discourage selfish behavior and to reward cooperation.

5.3.1 Content Access Protocol

Although as mentioned before the scheme is structured in two main phases (content authentication and content access), we will only illustrate the latter, as that will be the focus of our analysis. For further and more detailed description of the whole scheme please refer to [Palomar et al., 2006a] and [Palomar et al., 2006b].

Collaborative and Non–Collaborative Peers

Let us consider a P2P file sharing system in which participant nodes are typified to be *collaborative* and *non–collaborative* and which are supposed to interact following a specific interacting protocol.

A collaborative node is one which is not expected to deviate from the protocol specified for the interaction between peers. In other words, a collaborative node does not find reward in misbehaving. By contrast, a non–collaborative node will be expected to deviate from the specified file sharing protocol as in doing so the node presupposes to obtain bigger payoff value.

Access to File Content

Let R be a requester node who requires a specific file m. At this point, node R would launch a search query across the community which would lead to a list of sources (provider nodes) that keep an encrypted replica of the desired content. Each node in the list would also publish proof of the required content's integrity. Requester R must select a source P according to some criterion and then ask P for a *trapdoor* to access content m. For this, R must initiate a four-step protocol in which P must also participate.

Asking for a Trapdoor

Briefly explained, before R can reach the trapdoor (l bits out of the secret key K_S used to encrypt the required content m), she has to solve a challenge issued by P. The challenge represents a proof-of-work for granting permission to access the file content and its complexity depends on the content security level and a conjecture over the community's collaboration nature.

We further elaborate on the protocol steps in the best possible case, i.e. in which both main parties are motivated to behave correctly and to follow the protocol faithfully (both nodes are collaborative). The scheme is illustrated in Fig. 5.1.

The requester R contacts the provider P using the last part of the content's integrity proof been published, $E_{K_P}\big(h(m)\big)$, and signs it (Fig. 5.1 –message m_1). With this message m_1, P can check R's identity (implicit in the notation used) and

$$
\begin{array}{rl}
1.\ R \rightarrow P: & m_1 = E_{K_P}\big(h(m)\big), \sigma_1 \\
2.\ P \rightarrow R: & m_2 = E_{K_R}(\varsigma_j), \sigma_2 \\
3.\ R \rightarrow P: & m_3 = E_{K_P}(\tau_j), \sigma_3 \\
4.\ P \rightarrow R: & m_4 = E_{K_R}(\omega(K_S), \sigma_4
\end{array}
$$

$$
\begin{array}{rl}
\text{where} & \sigma_1 = s_R^P\big(E_{K_P}\big(h(m)\big)\big) \\
 & \sigma_2 = s_P^R\big(E_{K_R}(\varsigma_j)\big) \\
 & \sigma_3 = s_R^P\big(E_{K_P}(\tau_j)\big) \\
\text{and} & \sigma_4 = s_P^R\big(E_{K_R}(\omega(K_S))\big)
\end{array}
$$

Figure 5.1: Content access scheme: Asking for a trapdoor.

the required content's hash. After this, P elaborates a conjecture θ over requester R's real nature. The estimation will be based on:

1. R's collaborative attitude (i.e. if P believes that R is collaborative or non–collaborative).

2. The results of past interactions: past experience within the same community will allow P to adjust new estimates.

This conjecture can take the form of a numerical value, so if the computed value is higher than a given threshold P decides to continue with the protocol otherwise, P would ignore R's request. Note that the decision of interacting with R depends on time-varying factors, such as the accumulated experience of P within this community. As a consequence, it seems reasonable to update these beliefs regularly. The underlying idea is that P will try to reduce the cost when she estimates it is highly possible to interact with non–collaborative peers.

If R is estimated to be collaborative, P computes a challenge ς and sends it to R (message m_2). Upon receiving it, R sends back the corresponding response τ (message m_3) to P. If P considers τ correct, she sends R the trapdoor $\omega(K_S)$ necessary to recover the key K_S (message m_4).

Concerning the challenge itself, there exist a number of primitives which can be used for this purpose. The basic idea is that the verification by the challenger should be fast, but the computation by the requester has to be fairly slow.

5.4 Formal Analysis: Bayesian Framework

Our analysis of the scheme introduced in the previous section is an application of Chapter 4's formal model based on Bayesian games. We will only reproduce those aspects of the formal model which are essential to the goals and scope of this chapter using the same notation whenever possible. The main goals of this

analysis are based on supporting rationality proof of the protocol and also studying the dynamics created when nodes interact in a file sharing P2P system using the proposed scheme.

5.4.1 Players and Types

The following definitions formalize some of the Bayesian concepts for our specific P2P system.

Each protocol participant becomes a player in the corresponding protocol game. Let provider P and requester R be the players of the protocol game, denoted as G^B_{RP} and created from the protocol description given in Fig. 5.1. We consider $\{P, R\}$ as the complete set of players.

Definition 5.4.1 (Players type profiles)**.** *Let $\mathcal{T} = \mathcal{T}_P \times \mathcal{T}_R$ be the type–profile space in G^B_{RP}, where $\mathcal{T}_p = \{C\}$ and $\mathcal{T}_R = \{C, NC\}$ are the type spaces for players P and R, respectively. A type C denotes a collaborative node, while NC denotes a non–collaborative one.*

In other words, in our particular instance, a provider node P has a single type, *collaborative*. By contrast, requester nodes R have two different types. We will denote by P a collaborative provider, and by R_C and R_{NC} a collaborative and a non–collaborative requester, respectively.

We consider the following probability distribution over the space $\mathcal{T}$:

$$\begin{aligned} &\theta_C = \text{Prob}(R_C|P) \\ &\theta_{NC} = \text{Prob}(R_{NC}|P) \\ &\text{s.t. } \theta_C + \theta_{NC} = 1 \end{aligned} \tag{5.3}$$

Note that $\text{Prob}(P|R_C) = \text{Prob}(P|R_{NC}) = 1$.

5.4.2 Strategies and Beliefs

As previously explained, a *pure strategy* for a player E in a game G is a complete contingency plan which describes the series of *actions* that player E would take at each possible decision point in the game G. For our specific instance we define:

Definition 5.4.2 (Players set of actions)**.** *Let $A_P = \{m_2, m_4, quit_P\}$ and $A_R = \{m_1, m_3, quit_R\}$ be the sets of actions for players P and R, respectively.*

Definition 5.4.3 (Pure strategies for player P)**.** *In G^B_{RP}, the complete set of* pure *strategies for player P, denoted as S_P, is defined as $S_P = \{s^P_1, s^P_2, s^P_3\}$, where: $s^P_1 = (m_2, quit_P)$, $s^P_2 = (m_2, m_4)$ and $s^P_3 = (quit_P, \cdot)$.*

The first component of each tuple represents the action taken by player P at round 2 of the protocol (P's first turn to move). In a similar way, the second component represents the action taken by P at round 4 of the protocol (player P's second chance to make a move).

Definition 5.4.4 (Pure strategies for player R)**.** *In G^B_{RP}, a pure strategy for player R is represented by a tuple $s^R = (s^{R_C}, s^{R_{NC}})$ where:*

- *s^{R_C} represents the strategy to follow by a node R of type* collaborative*,*
- *$s^{R_{NC}}$ represents the strategy to follow by a node R of type* non–collaborative*,*

Each $(s^{R_C}, s^{R_{NC}}) \in S^R \times S^R$ where:

- *$S^R = \{s^R_1, s^R_2\}$*
- *$s^R_1 = (m_1, quit_R)$ and $s^R_2 = (m_1, m_3)$.*

Then, the complete set of pure strategies for player R is then described as:

$$\{(s^R_1, s^R_1), (s^R_2, s^R_2), (s^R_1, s^R_2), (s^R_2, s^R_1)\} \tag{5.4}$$

In this case, the first component of each tuple represents the action player R takes at stage one in the protocol game, and the second describes the action at stage three (first and second turns for R to move).

Definition 5.4.5 (Strategy profile)**.** *A strategy profile in the G^B_{RP} game is a vector $s = (s^R, s^P)$ of individual strategies, one for each player, where $s^R \in S^R \times S^R$ and $s^P \in S^P$.*

Note that, specifying a strategy profile univocally determines the outcome of the game.

The following probability distributions represent the set of beliefs each entity holds over the opponent's type and actions at each particular stage of the protocol.

Definition 5.4.6 (Node P belief system)**.** *At stage two of the protocol, P's conjecture over peer R's real nature (requester R could be collaborative or non–collaborative) is represented by the following probability distribution function over $\mathcal{T}_R$:*

$$\theta = \text{Prob}(R_{NC}|m_1) \qquad 1 - \theta = \text{Prob}(R_C|m_1) \tag{5.5}$$

p^-	Cost for node P to elaborate a puzzle to include in m_2.
p^+	Profit for node P when completing the protocol sending the trapdoor included in m_4. This value represents the potential new business generated by well behaved providers ensuring the continuation of the current system.
r^t	Value it has for requester R to have received a puzzle. When R is non–collaborative, this value represents the reward to having misled provider P to enter the protocol, when R had no intention to send back a response.
r^+	When the requester is collaborative, this value represents the gain after receiving the trapdoor included in m_4.
r^-	Cost for player R to elaborate an answer to the P's challenge.

Table 5.1: Payoffs in the G^B_{RP} game.

Likewise, requester nodes are able to form conjectures over the provider's intention to quit the protocol at round two of the protocol. We define the following probability distribution function to represent such a belief:

Definition 5.4.7 (Node R belief system)**.** *At stage two of the protocol, R's conjecture over peer P's actions is represented by the following probability distribution function:*

$$\alpha = \text{Prob}(quit_P|P) \qquad 1-\alpha = \text{Prob}(m_2|P) \tag{5.6}$$

Note that, at stages three and four of the protocol, entities are *rationally* forced to follow the protocol description as they obtain a better payoff value by doing so. Hence, there is no need for opponent's nodes to form conjectures over any other kind of behavior at those steps. Therefore, rationality is therefore forcing nodes to take specific actions. We will give formal proof of this statement in further sections.

5.4.3 Payoff Functions

As stated before, one of the key points of Bayesian games is the fact that each type of player is associated with a different payoff function. We define the following payoff functions:

$$U_R, U_P : \mathcal{T}_R \times S_R \times S_P \to \mathbb{R}$$

Table 5.1 relates all possible payoff values obtained by players in the G^B_{RP} game. In addition, we impose the following constraints:

$$\begin{gathered} r^- > r^t \\ p^+ > p^- > 0 \\ 0 \leqslant \omega, \theta, \alpha \leqslant 1 \end{gathered} \tag{5.7}$$

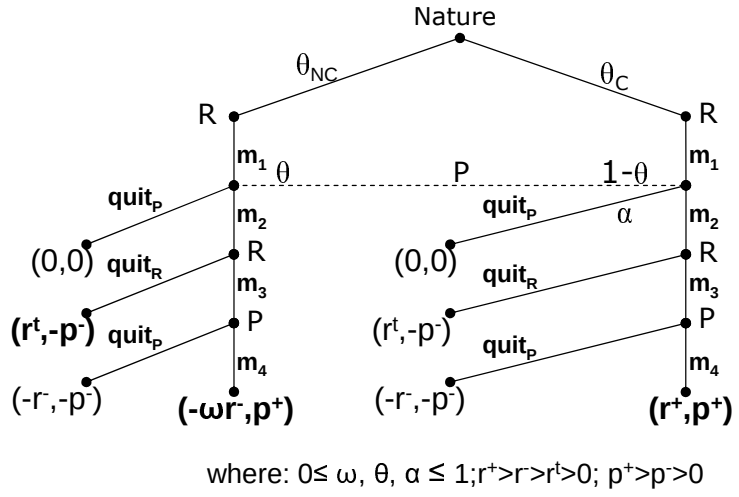


Figure 5.2: G^B_{RP} in extensive–form.

	$s^R_1 = (m_1, quit_R)$	$s^R_2 = (m_1, m_3)$
R_C	*dominated* $EG(R_C, s^R_1, \alpha) < EG(R_C, s^R_2, \alpha)$	*dominant* $EG(R_C, s^R_1, \alpha) < EG(R_C, s^R_2, \alpha)$
R_{NC}	*dominant* $EG(R_{NC}, s^R_1, \alpha) > 0$	*dominated* $EG(R_{NC}, s^R_2, \alpha) < 0$

Table 5.2: Dominant and dominated strategies for player R.

Moreover, a detailed representation in extensive–form of the protocol game G^B_{RP} is provided in Fig. 5.2. Briefly, the common interpretation of an extensive–form game is the following: The game can be thought of as a tree, where the edges and the vertices are associated to actions and sequences of actions, respectively. Terminal vertices are those that cannot be followed by any other actions. When a sequence of actions reaches a terminal vertex, the game ends. For each branch in the tree, the payoff value associated to its final node represents the total outcome that players R and P obtain when following such a path. The reader is referred to Appendix A for further details on Games in extensive–form.

5.4.4 Dominated Strategies and Expected Gains

In this section, we will compute the gains each player expects to obtain when following a specific strategy.

There are cases when it is possible to anticipate the moves that rational players

	s^P_1= (m_2, $quit_P$)	s^P_2 = (m_2, m_4)	s^P_3 = ($quit_P$,•)
$\theta < p^+/(p^+ + p^-)$	*dominated* EG(P, s, θ)<0	*dominant* EG(P, s, θ)>0	*dominated* EG(P, s, θ)=0
$\theta = p^+/(p^+ + p^-)$	*dominated* EG(P, s, θ)<0	EG(P,s, θ)=0	EG(P, s, θ)=0
$\theta > p^+/(p^+ + p^-)$	*dominated* EG(P, s, θ)<0	*dominated* EG(P, s, θ)<0	*dominant* EG(P, s, θ)>0

Table 5.3: Dominant and dominated strategies for player P.

will or will not take during the protocol game execution. All those actions for which the expected payoff is lower than the one obtained following other options are called *dominated strategies*. By contrast, a *dominant strategy* is such that, a rational player will always choose to follow it, as the expected gain by doing so is greater than by taking a different move (see Section A.1.2 for further details).

Dominated strategies can be eliminated from the formal analysis as self-interested rational players will never follow them. In our specific analysis of the G^B_{RP} protocol game, we can clearly identify one dominated strategy for player P at the final stage of the protocol game:

- Action *send* m_4 *dominates the last round of the protocol game.* Every rational provider P, having reached stage four in the protocol game, will always choose to send message m_4 as the expected payoff is greater by doing so than by quitting the game. Both a reputation system and the prospect of future profitable interactions are represented by the positive value p^+.

We compute the *Expected payoff* values $EP(\cdot)$ for each player involved and the remaining set of moves, by multiplying the probability of following a specific branch of the tree and corresponding payoff expected at the final node.

The following are the expected payoff values from entity R's point of view. Table 5.2 summarizes the following results:

$$\begin{aligned} EP(R_C, s^R_1, \alpha) &= (1-\alpha)\cdot(r^t) \\ EP(R_C, s^R_2, \alpha) &= (1-\alpha)\cdot r^+ \\ EP(R_{NC}, s^R_1, \alpha) &= (1-\alpha)\cdot r^t \\ EP(R_{NC}, s^R_2, \alpha) &= (1-\alpha)\cdot(-\omega r^-) \end{aligned} \tag{5.8}$$

Equations (5.8) let us formally reason and establish the following statements:

- Action $quit_R$ is dominated by action *send* m_3 at stage three of the protocol game, when R type is collaborative. At this stage in the protocol game, R

is sure of P's latest move (participant rationality is public information) so choosing to send m_3 offers R a greater payoff value. Note $EP(R_C, s_1^R, \alpha) < EP(R_C, s_2^R, \alpha)$. Strategy s_2^R is therefore a dominant strategy for R_C.

- By contrast, action $quit_R$ dominates strategy *send* m_3 at stage three of the protocol game and for player R, type non–collaborative. Choosing $quit_R$ offers R_{NC} a positive payoff value of $(1-\alpha) * r^t \ \forall\ 0 < \alpha < 1$, whereas choosing m_3 will only get a negative payoff value. Strategy s_1^R is therefore a dominant strategy for player R_{NC}.

Similar calculations can be carried out from player P's point of view. For this we have:

$$\begin{aligned} EP(P, s_1^P, \theta) = &\ -p^- \\ EP(P, s_2^P, \theta) = &\ \theta \cdot (-p^-) + (1-\theta) \cdot p^+ = \\ &\ p^+ - \theta \cdot (p^+ + p^-) \\ EP(P, s_3^P, \theta) = &\ 0 \end{aligned} \tag{5.9}$$

Table 5.3 summarizes these results.

Equations (5.9) let us formally reasoning and establishing the following statements:

- Strategy s_1^P is clearly a dominated strategy for player P, as the expected payoff is negative $\forall\ 0 \leqslant \theta \leqslant 1$.

- Strategy S_2^P is a dominant strategy over S_3^P if and only if $EP(P, s_2^P, \theta) > 0 \Leftrightarrow \theta < p^+/(p^+ + p^-)$.

5.5 Evaluation of Rationality

As described above, the formal model will serve to formally prove that Palomar et al.'s scheme is rational, that is, rational (self-interested) entities will always follow the steps described by the protocol.

5.5.1 Nash Equilibrium.

An equilibrium of the system will be represented by an equilibrium of the game. An equilibrium of the system will be a certain state which self-interested parties will not want to unilaterally move from. An equilibrium of the game is a set of strategies from which players would not want to individually deviate to obtain better payoff values.

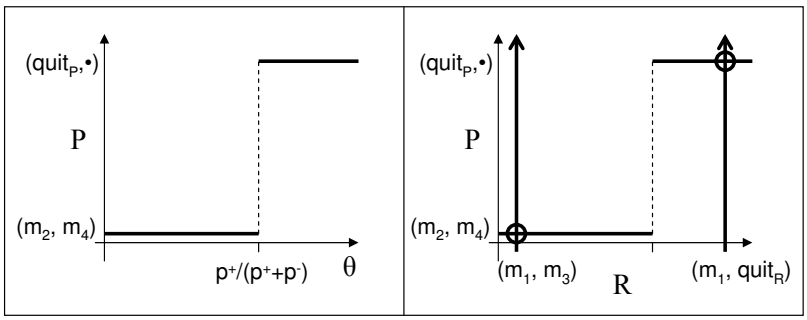


Figure 5.3: (Left) Best–response function for P. (Right) Intersection of best–response functions for P and R (both R_C and R_{NC}).

We will consider a best–response function for each player and type. The best–response function offers players the best strategies when responding to all possible types of an opponent and all their possible strategies.

Figure Fig. 5.3 (left) depicts graphically the best–response function for player P, according to the expected payoff values calculated in equations (5.9).

Best–response function for peer R can also be defined from equations (5.8). Figure Fig. 5.3(right) shows the intersection with player P's best–response function. The left vertical line correspond to requester type C, while the right vertical line correspond to requester type NC. It is precisely in these intersection points, where both best–response functions cross each other that the equilibrium is reached. Neither the provider nor the requester would want to modify their strategies unilaterally, as by doing so they would not obtain better results. Note that one of the equilibrium points is reached when all players complete all steps in the protocol. This serves to formally prove that our scheme is a rational one ([Buttyán and Hubaux, 2001]).

Furthermore, note that the reasoning in Section 5.4.4 allows us to conclude that such equilibria represent Perfect Bayesian Equilibrium points as defined by Bayesian requirements in Section A.5.3.

5.5.2 Impact of Non–collaboration.

Additionally, the formal model let us formally identify and measure two main factors of the proposed content access scheme.

- Firstly, the system dynamics depends upon the conjecture θ that player P makes on the type of community in which it is immersed. This conjecture could vary and it can be dynamically adjusted while the system is operative.

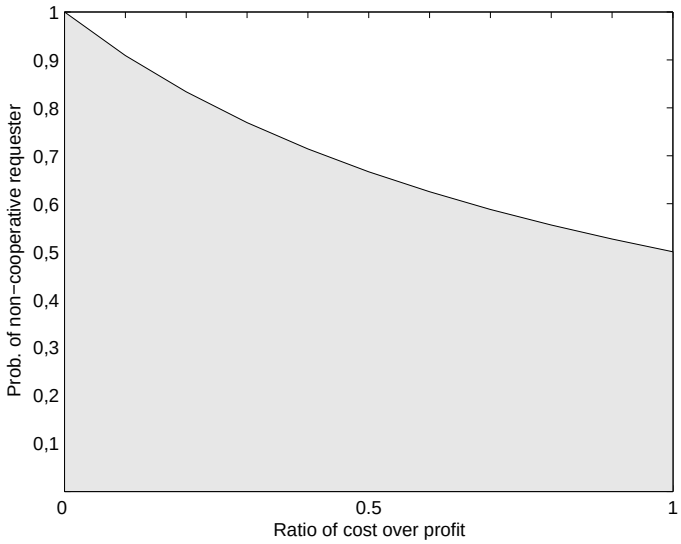


Figure 5.4: Relationship between θ, p^+ and p^-.

- Secondly, the ratio p^-/p^+ (cost over profit) does also influence the system behavior. Both parameters are used as control parameters for the dynamics of the system.

- Figure Fig. 5.4 shows the relationship between the ratio cost over profit (p^-/p^+) and the threshold computed by P $(\theta < p^+/(p^+ + p^-))$ to accept the request and enter the protocol. Note that P's conjecture over the proportion of non–collaborative nodes within the community θ = Prob("R being no-collaborative") must always be lower than 0.5. P uses these calculations as a defense mechanism against communities where the number of non–collaborative nodes is greater that the number of collaborative ones.

5.6 Conclusions

Infrastructure-less networks on which in general, one cannot assume the existence of centralized services such as those provided by TTPs, present a challenge in terms of formalizing collaboration-based security protocols.

In this paper we have analyzed the protocol game of a rational content sharing scheme modeling all possible interactions of the protocol participants and establishing formal proof of its rationality.

Although we are assuming participant rational behavior, we are able to consider non–collaborative players and to measure the effect they might have on the overall

system performance.

Part II

Automated Design of Multi–party Rational–Exchange Security (M–RES) Protocols

Chapter 6

Introduction to the Automated Synthesis of Cryptographic Protocols

6.1 Introduction

Evolutionary Computation is the general term for several computational techniques which are based, to some extent, on natural phenomena from the real world and which are most commonly deployed for the purpose of heuristic search.

Heuristic search is concerned with the development and application of general purpose optimization techniques in search of *optimal* solutions to *very difficult* problems. In the absence of the best possible answer to a question, an heuristic technique is sometimes able to produce optimal solutions within feasible and tractable computational settings.

Heuristic techniques have been successfully applied across many scientific and engineering domains. In particular, they have been very successful when applied to cryptology. Hao, Clark and Jacob were the first ones to show in a series of works ([Clark and Jacob, 2000], [Clark and Jacob, 2001], [Chen et al., 2004] and [Chen et al., 2005]), how heuristic search can be used to automatically synthesize cryptographic protocols that are demonstrably correct and satisfy various security and efficiency criteria. Moreover, in [Hernández-Castro et al., 2006] the authors presented a general scheme for the design of block ciphers by means of genetic programming. Also, applied to cryptanalysis, in [Estévez-Tapiador et al., 2007a] and [Estévez-Tapiador et al., 2007b] the authors applied heuristic techniques for the cryptanalysis of S-boxes, hash functions and stream ciphers.

The aim of this chapter is to give the reader a basic understanding of what an

heuristic search technique consists of, its applications and most common types. We will also introduce several related concepts, which will be used in future chapters for the presentation of our proposed synthesis model. In particular, we will present an introduction to fitness landscapes and random walks. Both are techniques, within the Evolutionary Computation area, closely linked to any heuristic approach. The use of these techniques provides us with valuable information to determine the main components of any heuristic algorithm to be applied.

6.1.1 Chapter Organization

This chapter is organized as follows. In Section 6.2 we give a brief introduction to the most commonly used heuristic search algorithms. In Section 6.3 we present fitness landscapes and random walks. These two concepts together with the notion of fitness autocorrelation provide the basics for the applied techniques. Finally, in Section 6.4 we will offer a survey on the existing work on applying heuristic search to the automated synthesis of security protocols.

6.2 Heuristic Search

Formally, the general aim of an heuristic technique is to find optimal solutions to problems that are structured as a function of some variables, in the presence of some constraints. These can be formulated as:

$$\text{Maximize } F(x) \text{ subject to}$$
$$x \in C \subset X$$

where:

- X represents the set of all possible vectors $x = (x_1, \ldots, x_n)$ of decision variables, usually referred to as the *solution space*.
- Set C represents the set of imposed constraints and,
- Function F is defined such that it computes the *fitness*[1] of a given vector x. The fitness function is usually defined to measure how good a vector x is and whether it represents an optimal solution to the maximization problem in hand. In other cases, it is defined so it provides guidance to the search.

In other words, given a space X of vectors and some measurable characteristic value $F(x)$ associated to each vector, the problem is to find a solution with the maximum possible $F(x)$ value, subject to a set of constraints.

Different approaches can be taken when confronting such a maximization problem. In the next section we analyze the most common.

[1] For minimization problems this function is denoted as *cost* function.

6.2.1 Search Methods for Optimal Solutions

Brute force is the least sophisticated of search methods. It consists in evaluating $F(x)$ for every vector $x \in X$ in turn. Either a solution with the sought $F(x)$ value is found or there is no solution to the problem. The method presents clear disadvantages. In some instances, one simply cannot evaluate every single possible point in the solution space as these are too numerous. In other instances, it is not even possible to enumerate every solution. In both cases, a statistical search could be more effective.

A *statistical search* consists in randomly selecting a sample of the solution space evaluating each one of the sampling vectors. This type of search is usually applied when the fitness value $F(x)$ of any vector x, provides little exploitable information about which vector x', from the solution space, should be evaluated next. This has been the standard method for many years and it is still commonly used, in particular for the design of security protocols as protocol designers do usually take a proposal from the space of protocols, and then check for the required security properties.

Finally, a completely different approach denoted as *guided search* consists in selecting an individual from the solution space and, via a neighboring operator, generating a series of consecutive neighbor vectors in such a way that, for two neighbor points in the solution space, we expect to obtain similar fitness values. Note that the way in which the neighboring operator is defined will determine the outcome of the search. Likewise, the fitness function chosen must show some degree of linearity and continuity over the space of possible solutions. The search process progresses moving from one solution to a neighbor solution trying to improve the fitness value.

An heuristic search is a guided search. Examples of heuristic techniques for the search of optimal solutions are:

Hill Climbing: Techniques based on Hill Climbing evaluate the fitness values in the neighborhood of a current solution and make the search move to a neighbor solution if and only if, the move improves the current fitness value. The problem with this technique and its derivatives is quite obvious: if the search starts in the wrong place the result might end up being a local rather than a global optimum.

Simulated Annealing: Simulated
Annealing ([Kirkpatrick et al., 1983]), is inspired by the cooling processes of molten metals. It merges a basic Hill Climbing technique with a probabilistic acceptance of non–improving solutions, which allows the search to escape from local optima. We introduce the basic scheme in Fig. 6.1 and detail the process

```
S ← S_0
T ← T_0
repeat until stopping criterion is met
        repeat MIL times
                Pick C ∈ N(S) with uniform probability
                if F(C) > F(S) then
                        S ← C
                else
                        Pick U ∈ (0, 1) with uniform probability
                        if F(C) > F(S) + T ln U then
                                S ← C
                T ← αT
```

where:
S_0 represents the initial individual
T_0 is the initial temperature
MIL defines the number of moves in the inner loop.
$N(S)$ represents S's neighborhood
$F(\cdot)$ represents the fitness function
$0 < \alpha < 1$ is the cooling factor

Figure 6.1: Basic Simulated Annealing algorithm.

in the following paragraphs.

Roughly, this technique requires a first individual S_0 which is randomly generated and presented for evaluation. Each individual's evaluation consists of computing its fitness value $F(\cdot)$. There is also a control parameter $T \in \mathbb{R}^+$ known as temperature, which takes an initial value T_0 and which dynamically decreases its value during the annealing process. The first randomly generated S_0 is evolved to a different solution C in the neighborhood of S_0. The new scheme is also evaluated. If the new individual C reaches a higher level of overall fitness than the original one, then it is accepted as a new valid scheme from which to generate the next one. If the new individual represents a solution worse than the previous one, the new scheme could only be accepted if the control parameter T is above a specific value. In other words, better solutions are always accepted and worse solutions are accepted when the temperature is still above a certain threshold. The process is repeated a fixed number of times depending on the initial temperature and the number of solutions being accepted.

Tabu search: Tabu search [Glover, 1987], is a widely used modern search technique. If merges a best improvement local search algorithm with some

$S \leftarrow$ Generate_Initial_Solution()
$TL \leftarrow$ Initialize_Tabu_list
repeat until stopping criterion is met
  Compute $AllowedSet(S) = \{C \in N(S) | C \notin TL\}$
  Select best C in $AllowedSet(S)$
  Update TL
  $S \leftarrow C$

where:
$N(S)$ represents S's neighborhood and,
TL is the Tabu list.

Figure 6.2: Basic Tabu algorithm.

historical search information, that is, it uses some form of memory of the evolution of the search. If a particular solution S is reached, then it becomes tabu for a number T_s of transactions, generally referred to as tabu tenure. If a solution is tabu, then the search is prevented from moving to that solution. This way, the local space of a given solution is thoroughly explored before making a move avoiding cycles. The basic scheme is formally described in Figure 6.2.1.

Generally, some aspiration criterion is also introduced to allow tabu solutions to abandon the tabu list in less than T_s moves. Also, in practise, maintaining lists of tabu solutions is very inefficient and this is usually overcome by keeping lists of attributes, for example, the fitness value of a solution. In this case, the search will be prevented from visiting neighboring solutions with the same fitness for a fixed number of moves. Many different variants of a general tabu procedure exist ([Glover, 1990], [de Werra et al., 1995]).

Iterated local search : Iterated local search can be considered one of many hybrid techniques aimed to search over the space of local optima, before moving on to the search of global optima. A random solution S_0 is generated and local search is applied to reach a local optimum S^*. This local optimum is perturbed in some way to obtain S' and local search applied again to reach another local optimum $S^{*'}$. Then, some criterion is applied to determine if the search should move from S^* to $S^{*'}$. A variety of moving criteria can be adopted (always accept, only accept improving moves, probabilistic acceptance in a annealing–like manner, etc.) Figure 6.2.1 explicitly describes the general process.

Genetic algorithms: Genetic algorithms [Goldberg, 1989], are heuristic search techniques based on natural selection. Figure 6.4 describes the pseudo–code

```
S_0 ← Generate_Initial_Solution()
S* ← Local_Search(S_0)
repeat until stopping criterion is met
    S' ← Perturbation(S*)
    S*' ← Local_Search(S')
    S* ← Apply_acceptance_criterion(S*,S*')
```

Figure 6.3: Basic ILS algorithm.

```
Choose initial population
Evaluate the fitness of each individual in the population
repeat until stopping criterion is met
  Select best fitness individuals to reproduce
  Breed new generation through crossover and mutation and
    give birth to offspring
  Evaluate the fitness of each offspring
  Replace worst ranked part of population with offspring
```

Figure 6.4: Basic Genetic algorithm.

of a basic genetic algorithm.

Most frequently, solutions are represented as binary strings, but other encodings are also possible. The process usually starts from a randomly generated population of individuals and successive generations are produced using a combination of evolutionary operators (crossover and mutation). The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions.

In each successive generation, the fitness of every individual in the population is evaluated. A proportion of the population is then selected in a way such that, fitter solutions (as measured by a fitness function) are typically more likely to be selected. Each pair of selected individuals produces a child solution using methods of crossover and mutation, creating new solutions which share many of the characteristics of its parents. A new generation of individuals is created which is different from the previous one. Generally, the average fitness of the new population will have increased since only the best organisms are selected for breeding, along with a small proportion of less fit solutions. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached amongst the population.

Combined techniques: Techniques like the ones described in the previous

paragraphs are often used in combination. It is common to apply a specific heuristic algorithm to find the start point for a different technique to carry on with the search. It is also usual to finish the last moves of a search with a variant over the technique applied for the search up to that point.

6.3 Fitness Landscape Analysis

Taken from biology, the notion of *fitness landscape* has become an important concept in Evolutionary Computation. Recently, the landscapes of a range of problems have been analyzed in an attempt to determine the relationship between fitness landscape structure and the performance of a particular heuristic technique. In other words, the analysis of landscape structures could allow us to determine the difficulty of a task and hence apply the most appropriate heuristic search algorithm.

To study the fitness landscape properties of a given problem the following components have to be defined:

1. A neighboring operator which, given a vector in the solution space, generates a neighbor. The neighboring operator will have a large effect on determining the fitness landscape for a particular problem.

2. A technique to produce a number of consecutive neighboring vectors within the solution space. (For example, the random walk technique).

3. A fitness function which assigns each generated vector a fitness value.

The mapping provided by the fitness function over neighbor points of the solution space constitutes the landscape.

6.3.1 The Random Walk Technique and the Autocorrelation Function

The random walk technique produces a number of consecutive neighboring vectors within the solution space. In the initial phase of a random walk a random solution is produced and evaluated. Then, using the neighbor operator a neighbor individual is generated and calculated its fitness. The same step is repeated k times, this representing the length of the walk.

A number of techniques for the analysis of landscape structures exist. In particular, Weinberger in [Wenberger, 1990] investigated how the *autocorrelation function* of the fitness values along the steps of a *random walk* relates to the ruggedness of the examined landscape.

In the coming paragraphs we will provide details on the definition of a fitness autocorrelation function and its most representative shapes resulting in significant landscape properties.

Fitness Autocorrelation Function

Definition 6.3.1. *The autocorrelation of a random walk relates the fitness of any two individuals which are s steps apart. We denote:*

$$R(t,s) = \frac{E(f_t * f_{t+s}) - E(f_t) * E(f_{t+s})}{var(f)} \tag{6.1}$$

where

- f_t *is the fitness of the t–th individual along the walk,*
- f_{t+s} *is the fitness of the individual s steps further along the walk and,*
- $var(f)$ *represents the variance of the entire series.*

This value is calculated for all possible pairs (t, s) in each walk.

Some characteristic shapes for the autocorrelation function can be interpreted as described below:

- Most correlation values are close to zero. This is characteristic of a random landscape where there is no correlation between fitness values. It is usually representative of extremely rugged fitness landscapes wherein any guided search is likely to operate as a random search.
- A second characteristic form is the fast decaying form. It is usually representative of moderately *rugged* fitness landscapes in which guided search algorithms can often obtain good solutions in less time than a classic random search.
- The third characteristic form of an autocorrelation function is the slow decaying form. This corresponds to highly correlated fitness values and it is usually representative of *even* fitness landscapes. In other words, neighbor solution points offer very similar fitness which will probably make any guided search quite slow.
- Finally, a constant or periodic autocorrelation function often indicates a badly defined neighboring operator which would periodically generate individuals which are too similar to one another.

A desirable fitness autocorrelation function lies somewhere in between the second and third type. At the same time, one also has to take into account that the study of fitness autocorrelation is not determinant of the difficulty of the task to be solved by any heuristic search algorithm.

6.4 Heuristic Search Applied to the Synthesis of Security Protocols

Finally, we will also provide a brief survey on some of the work already developed in the area of automated design of security protocols.

It is clear that the number of possible protocols achieving a set of goals from a set of initial assumptions grows exponentially as the number of goals or the number of participant entities rise. Therefore, for a protocol designing technique to be scalable and feasible it cannot be based on simple enumeration. In this context, a methodology based on an heuristic search represents a good compromise between optimal solutions and computational tractability.

As mentioned before, Hao, Clark and Jacob were the first ones to show how heuristic search can be used to automatically synthesize cryptographic protocols, that are provably secure and satisfy various other efficiency and effectiveness criteria. Their approach is novel in the design of security protocols, providing an excellent framework for the automated exploration of very large design spaces, far greater than could be considered using manual design.

In their first work ([Clark and Jacob, 2000]), the authors show how evolutionary search, in the form of a Genetic algorithm, can be used to generate correct and efficient protocols. They describe and implement an automated process by which, given a set of assumptions and goals, security protocols achieving those goals are automatically synthesized. In their model, they use BAN logic or logic of beliefs (Section 2.5.1 and [Burrows et al., 1990]), to represent protocol assumptions, as well as participants' individual goals and other security requirements. Protocols are represented by bit strings. These strings are subject to the genetic evolutionary mechanisms (crossover and mutation) over different populations, so protocols satisfying the required goals, emerge through evolution. The resulting protocols are evaluated according to two different properties:

- *Correctness*, in terms of how many of the security goals a protocol achieves and,
- *Efficiency*, in terms of how early in the protocol the goals are attained. In this regard, different evaluation strategies are adopted to guide the search. Some

of these are detailed in what follows:

- Early Credit (EC): This strategy assigns better fitness to protocols which achieve more security goals at early stages of the protocol.
- Uniform Credit (UC): It applies an uniform weight to all security goals, regardless at what stage they are achieved during a protocol run.
- Delayed Gratification (DG): This strategy captures the idea that early gratification of goals may not necessarily be a good thing.
- Destination Judgement (DJ): It does not matter when a protocol satisfies the set of stated goals, the important thing is how many it satisfies in the end.

Clark et al. apply the previously described formalism for the synthesis of a three entity symmetric key distribution protocol, with six or fewer messages.

Further refinements to this initial model are provided in subsequent works. In [Clark and Jacob, 2001] the authors present a process for the automated synthesis of symmetric key distribution protocols based on heuristic search algorithm Simulated Annealing (SA). A protocol is represented by a list of integers representing the messages being exchanged between two of the participant entities. Each message contains a sequence of beliefs that one entity sends to another (by construction, senders only send beliefs they actually hold). Associated with every entity is a vector of its current beliefs. After a message is sent, the beliefs vector for the current receiver is updated and entities check whether their goals have been satisfied. This representation allows a very simple move strategy for a SA search which randomly changes any of the beliefs involved in any of the messages. Initially, the search algorithm applies perturbations to a randomly generated protocol to generate new schemes. Sequentially, similar changes are applied to each intermediate protocol measuring their fitness.

Furthermore, in [Chen et al., 2004], Chen, Clark and Jacob use SA for the synthesis of provably secure protocols in which principals are able to use asymmetric encryption.

Finally, in [Chen et al., 2005] authors extend previous works in several ways:

(1) They use a subset of the SVO logic to represent protocol assumptions, participants' individual goals and, as a proof system. SVO logic ([Syverson and van Oorschot, 1994a]) is an extension to BAN logic presenting some additional features.

(2) Additional factors are considered for the evaluation of synthesized protocols:

- *Security*, in terms of how many security goals a protocol achieves and how early in the protocol. Two strategies are evaluated: early credit (EC) and uniform credit (UC).
- *Efficiency*, in terms of number of messages, number of encryptions and the number of interactions with particular principals (servers, TTP, etc.)

In this case, the fitness function is defined as the sum of a *security fitness* value and an *efficiency fitness* value. The first will reward protocols that achieve greater proportion of stated goals, whilst the second will punish protocols with many messages, protocols with more encryption and protocols with higher number of interactions with particular protocol participants.

A similar technique based on Genetic algorithms is also applied by Park et al. in [Park and Hong, 2005], to the synthesis of cryptographic protocols for a fault-tolerant agent replication system.

Finally, some of the protocols synthesized by these techniques are already well known schemes in their fields.

6.5 Conclusions

The above work indicates that the mechanisms of Evolutionary Computation and other heuristic algorithms can provide a plausibly tool for the automated generation of secure and efficient protocols. Although most of the schemes and solutions generated by these tools need further refinements and, these final touches might need additional security analysis, the new methodology can be used to search the design space and provide input to human designers, who would then derive concrete refinements of the abstract protocols.

In the next chapters we will define a methodology based on Simulated Annealing, for the automated synthesis of rational–exchange cryptographic protocols.

Chapter 7

Foundations for the Automated Synthesis of M–RES Protocols

7.1 Introduction

In Part I of this thesis we provided a formalism by which rational cryptographic protocols could be *analyzed*. The model was based on Game Theory; in particular, protocols were represented as games in extensive–form (trees) and rationality was described in terms of the Nash equilibria in the game.

In Part II of the document we will be focusing on the automated *design* of cryptographic rational schemes. For this, new tools will be defined to represent exchange problems and rational protocols, which will be complementary to the extensive–form games used in Part I. This new representation will allow us to:

- Facilitate the automated synthesis of cryptographic rational protocols by means of heuristic search techniques.
- Formally analyze new schemes using Part I analytical model. We will refer to this as the *proof system* of our design methodology.
- Define a taxonomy aimed at classifying rational exchange protocols and problems, according to different environmental factors and participant's nature.

As it is only recently that cryptographic protocols have been designed under the sole assumption that the parties are rational – an introduction to a new adversarial model from a Rational Cryptography point of view is given in Section 1.1– very few rational solutions exist. Furthermore, these have not been designed to solve exchange problems but other types of questions, such as secret sharing or multi–party computation ([Nielsen et al., 2007] provides an excellent survey). In this part

of our work new rational exchange schemes will be automatically designed under the sole assumption of participant rationality.

7.1.1 Chapter Overview

In this chapter, we describe in detail the formal foundations for the automated synthesis of multi–party rational–exchange security protocols (M–RES protocols). In particular, the formalism here described satisfies the following main properties:

- It is highly flexible and scalable in multi–party environments as it will serve to represent any given multi–party exchange scenario, with any number of participant agents.
- Only rational (self-interested) entities are considered in the formalism.
- The undercurrent proof system is based on Game Theoretical results, in particular, on previous models described in Part I. This way the schemes synthesized by the proposed methodology can be *provably rational*. (Existing works by other authors described in Section 6.4 based their proof system on logics of beliefs.)
- Given an exchange problem, the model defines a framework suitable for a heuristic technique to search for an optimal rational solution.
- Simple linear structures such as vectors and matrices will be used to represent all aspects of a multi–party exchange problem.
- Finally, it allows us to define a taxonomy to classify M–RES protocols considering criteria such as incentives and coalitions. This taxonomy will be based on:
  (a) Identifying whether protocol participants are part of any incentive scheme which would make them behave in a certain, predetermined way; and,
  (b) Identifying whether participants are members of a coalition.

7.1.2 Chapter Organization

The chapter is organized as follows. Section 7.2 will describe the way in which candidate solutions (exchange protocols) are represented. In Section 7.3, we will define a utility function to be able to evaluate and assess how good a protocol is in solving a particular multi–party exchange problem. In Section 7.4 we formally express the goals of any heuristic algorithm used in the search and estimate the size of the solution space. Sections 7.5 and 7.6 depict a taxonomy for M–RES protocols

based on the formalism previously defined and propose a classification of M–RES protocols. Finally, Section 7.7 presents the main conclusions of this work.

7.1.3 Preliminaries

Prior to describing the model in detail, the following definition extends Definition 3.2.1 to multi-party environments and serves to unify notation throughout this chapter:

Definition 7.1.1 (Exchange protocol)**.** *We notate a multi–party exchange protocol as a tuple* $\Pi = \langle P, \mathcal{O}, T \rangle$ *where:*

- $P = \{P_0, \ldots, P_{v-1}\}$ *is the set of protocol participants,*
- $\mathcal{O} = \{o_0, \ldots, o_{m-1}\}$ *is the set of all items/tokens being exchanged during the protocol execution; and*
- T *is an ordered collection of* n *protocol steps describing the scheme, each of the form:*

$$(t) \quad P_i \rightarrow P_j : o_{t_1}, \ldots, o_{t_{k_t}} \tag{7.1}$$

  where:

  - $t = 0, \ldots, n-1$ *is the step number.*
  - P_i, $P_j \in P$, $i \neq j$, *are the sender and receiver of the message, respectively.*
  - $\{o_{t_1}, \ldots, o_{t_{k_t}}\} \subseteq \mathcal{O}$ *are the items* P_i *sends to* P_j*, subject to* P_i *owning those items at step* t *of the protocol.*

Note that this definition does not mention rationality, fairness or feasibility of the exchange. It merely describes a series of messages being exchanged between participants. At the end of the protocol execution some entities would have lost control over some of their items as well as having gained access over new ones.

7.2 Representation of Candidate Solutions

Informally, a candidate solution to a given multi–party exchange problem is a multi–party exchange protocol dictating the steps that each participant has to follow to interchange their commodities in a certain way. Further along, the utility function will decide how good a candidate protocol is in giving a solution to the specific exchange problem, or how close the scheme is to an optimal solution. A solution will be considered optimal when (1) it is rational and, (2) it satisfies every participant's set of requirements.

Initially, we will describe the way to represent protocols within the model.

$$S^{\Pi} = \begin{pmatrix} sender & receiver & o_0 & o_1 & o_2 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Figure 7.1: Example of a protocol matrix.

7.2.1 Protocol Matrix

A protocol Π, defined as in 7.1.1, is represented by a matrix $S^{\Pi} \in \mathcal{M}_{n\times(m+2)} = [s^{\Pi}_{t,j}]$ of integers, where each row s_t ($t \in \{0,\ldots,n-1\}$) is interpreted as a message in which the first two components identify the sender and the receiver of the message, respectively, and the rest of the row components represent the items being sent. Formally:

- $\forall\ t \in \{0,\ldots,n-1\}$,

$$\begin{array}{l} s^{\Pi}_{t,0} \text{ represents the sender entity of message } s_t, \\ s^{\Pi}_{t,1} \text{ represents the receiver entity of message } s_t \text{ and,} \end{array} \tag{7.2}$$

- $\forall j \in \{0,\ldots,m-1\}$,

$$s^{\Pi}_{t,(2+j)} = \begin{cases} 1 & \text{iff entity } s^{\Pi}_{t,0} \text{ sends entity} \\ & \quad s^{\Pi}_{t,1} \text{ item } o_j \text{ at step } t \text{ in the protocol} \\ 0 & \text{otherwise} \end{cases} \tag{7.3}$$

As an example, the matrix shown in Fig. 7.1 represents a three step exchange protocol involving three entities. At step zero, entity P_0 sends entity P_1 item o_0 then, entity P_1 sends entity P_2 item o_1 and finally, at step two, entity P_2 sends entity P_0 items o_1 and o_2.

Matrix S^{Π} serves to represent the series of steps that participant entities have to take along a protocol execution. However, the actual real message content being sent at each step in the protocol is subject to: (1) the sender entity holding the referred items $s^{\Pi}_{t,(2+j)}$ at that point in the protocol run and (2) the items being accessible to that entity. During the protocol execution this information will be captured in a new matrix $H(t)$ denoted *state matrix*.

7.2.2 State Matrix

The protocol is executed in a fixed number of n steps. Matrix $H(t) = [h_{i,j}(t)] \in \mathcal{M}_{v\times m}$ captures the possessions of each party at the end of step $t-1$ of the protocol

with $1 \leqslant t \leqslant n$, where:

$$h_{i,j}(t) = \begin{cases} \texttt{ACC} & \text{iff } P_i \text{ holds and has complete access to item } o_j \\ \texttt{NO\_ACC} & \text{iff } P_i \text{ holds item } o_j \text{ but has no access to it} \\ \texttt{LOST} & \text{iff } P_i \text{ has lost control over item } o_j \\ \texttt{UNKNO} & \text{iff item } o_j \text{ is unknown to entity } P_i \end{cases} \tag{7.4}$$

As an example: $H(1)$ represents the possessions of each participant entity after step zero of the protocol. A matrix denoted $H(0)$ will represent the possessions of each different entity at the initial point, this is, prior to the exchange. At this stage, no $h_{i,j}(0)$ value could be set to `LOST` as no lost could have taken place at $t = 0$.

Moreover, the following are the numerical values for each one of the possible elements in the state matrix. These values will allow us to compute partial and final benefits for each entity with a simple scalar product of vectors. The values are:

$$\begin{array}{lll} \texttt{ACC} = & 1 & Multipliedbyapayoffvaluewillincrementthetotalutility. \\ \texttt{UNKNO} = & 0 & Multipliedbyapayoffvaluewillannullthetotalutility. \\ \texttt{LOST} = & -1 & Multipliedbyapayoffvaluewilldecrementthetotalutility. \end{array} \tag{7.5}$$

The state `NO_ACC` could take any value different from the previous ones. This state will serve to annull the utility until a certain event takes place and it is never multiplied by the corresponding utility value. Different situations could derive in a non–accessible status of an item o_j for a particular entity P_i (i.e. $h_{i,j}(t) = \texttt{NO\_ACC}$). Some of these are:

1. If item o_j is encrypted and entity P_i holds o_j but it does not hold the decryption key.

2. If entity P_i is able to generate item o_j but it needs to gain access to other items in order to do so. In this case, item o_j must remain non–accessible until gaining control over the rest of the required tokens.

3. If item o_j is a type of e-good such that it only becomes available when other events take place, so entity P_i will only gain access to item o_j when such events have occurred.

As an example, matrix $H(0)$ shown in Fig. 7.2 represents an initial state matrix in which an entity P_0 holds and has access to item o_0. By contrast, entity P_1 holds item o_1 but has no access to it, and P_2 holds and has access to an item o_2.

The non–accessibility indicates that there might be dependency relationships between different possessed items as well as between the items that different entities

$$H(0) = \begin{pmatrix} o_0 & o_1 & o_2 \\ \texttt{ACC} & \texttt{UNKNO} & \texttt{UNKNO} \\ \texttt{UNKNO} & \texttt{NO\_ACC} & \texttt{UNKNO} \\ \texttt{UNKNO} & \texttt{UNKNO} & \texttt{ACC} \end{pmatrix}$$

Figure 7.2: Example of a state matrix.

hold. A matrix R will capture the dependency relationships for every element in the state matrix H and for the particular exchange problem in hand.

7.2.3 Dependency Matrix

A matrix $R = [r_{i,j}] \in \mathcal{M}_{(v \cdot m) \times (v \cdot m)}$ will capture the dependency relations for each $h_{i,j} \in H$ for a given exchange problem.

The following function will assist in representing such links. We notate $\varsigma(i, m)$ a function which returns a tuple of two components: the first component corresponds to the quotient of the integer division between i and m and, the second component to the remainder of that division.

$$\varsigma(i, m) = (\lfloor i/m \rfloor \ , \ mod(i, m)) \tag{7.6}$$

Three different types of dependency can be expressed within the model: *positive* (`POS_DR`), *negative* (`NEG_DR`), and non–existent (`NO_DR`). They are defined as follows:

- $r_{i,j} = \texttt{POS\_DR}$ if and only if:

$$(h_{\varsigma(i,m)} = \texttt{ACC}) \quad \wedge \quad (h_{\varsigma(j,m)} = \texttt{NO\_ACC}) \Rightarrow \quad h_{\varsigma(j,m)} = \texttt{ACC} \tag{7.7}$$

  In other words, item $o_{mod(j,m)}$ becomes accessible to entity $P_{\lfloor j/m \rfloor}$ once entity $P_{\lfloor i/m \rfloor}$ has gained access to item $o_{mod(i,m)}$. We refer to this as a positive relationship for $P_{\lfloor i/m \rfloor}$.

- $r_{i,j} = \texttt{NEG\_DR}$ if and only if:

$$\left.\begin{array}{c} (h_{\varsigma(i,m)} = \texttt{ACC}) \quad \wedge \\ \big((h_{\varsigma(j,m)} = \texttt{NO\_ACC}) \vee (h_{\varsigma(j,m)} = \texttt{UNKNO})\big) \end{array}\right\} \Rightarrow \quad h_{\varsigma(i,m)} = \texttt{NO\_ACC} \tag{7.8}$$

  In other words, item $o_{mod(i,m)}$ cannot be accessible to entity $P_{\lfloor i/m \rfloor}$ until entity $P_{\lfloor j/m \rfloor}$ has gained access to item $o_{mod(j,m)}$. We denote this a negative relationship for $P_{\lfloor i/m \rfloor}$.

- Additionally, $r_{i,j} = \texttt{NO\_DR}$ will represent the absence of any dependency relationships. Note that, $\forall i, 0 \leqslant i < v \cdot m$, $r_{i,i} = \texttt{NO\_DR}$ as no item can be neither positive nor negative related to itself.

Matrix R shown in Fig. 7.3 can serve as an example of the possible dependency relationships when considering matrix H from previous example 7.2. This matrix represents a positive relationship ($r_{3,4} = \texttt{POS\_DR}$) for entity $P_{\lfloor 3/3 \rfloor} = P_1$ with regard to items $o_{mod(3,3)} = o_0$ and $o_{mod(4,3)} = o_1$. Informally, this dependency connection expresses the fact that if entity P_1 gains access to item o_0 then item o_1 becomes accessible to P_1. Formally, as defined in (7.7), we can write:

$$(h_{\varsigma(3,3)} = \texttt{ACC}) \quad \wedge \quad (h_{\varsigma(4,3)} = \texttt{NO\_ACC}) \Rightarrow \quad h_{\varsigma(4,3)} = \texttt{ACC} \tag{7.9}$$

Which is equivalent to:

$$(h_{1,0} = \texttt{ACC}) \quad \wedge \quad (h_{1,1} = \texttt{NO\_ACC}) \Rightarrow \quad h_{1,1} = \texttt{ACC} \tag{7.10}$$

Likewise, matrix R shown in Fig. 7.3 indicates that there exists a negative dependency relationship ($r_{7,2} = \texttt{NEG\_DR}$) for entity $P_{\lfloor 7/3 \rfloor} = P_2$ with regard to items $o_{mod(7,3)} = o_1$ and $o_{mod(2,3)} = o_2$. Informally, this indicates that upon receiving item o_1, this will only become accessible to entity $P_{\lfloor 7/3 \rfloor} = P_2$ when item o_2 is available to entity $P_{\lfloor 2/3 \rfloor} = P_0$. Formally, as defined in (7.8), we can write:

$$\left.\begin{array}{c}(h_{\varsigma(7,3)} = \texttt{ACC}) \quad \wedge \\ \big((h_{\varsigma(2,3)} = \texttt{NO\_ACC}) \vee (h_{\varsigma(2,3)} = \texttt{UNKNO})\big)\end{array}\right\} \Rightarrow \quad h_{\varsigma(7,3)} = \texttt{NO\_ACC} \tag{7.11}$$

Which is equivalent to:

$$\left.\begin{array}{c}(h_{2,1} = \texttt{ACC}) \quad \wedge \\ \big((h_{0,2} = \texttt{NO\_ACC}) \vee (h_{0,2} = \texttt{UNKNO})\big)\end{array}\right\} \Rightarrow \quad h_{2,1} = \texttt{NO\_ACC} \tag{7.12}$$

7.2.4 Expressing More Complex Dependency Relationships and Applying Matrix R

Further and more complex dependency links may exist involving several items. However, in that case, important considerations must be taken into account.

If a dependency relationship involves more than one item:

1. All dependencies must be grouped by sign so that all negative relationships are considered independently from the positive ones.

$$R = \begin{pmatrix} \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} \\ \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} \\ \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{POS\_DR} & \texttt{NO\_DR} \\ \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{POS\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} \\ \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} \\ \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} \\ \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} \\ \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NEG\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} \\ \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} & \texttt{NO\_DR} \end{pmatrix}$$

Figure 7.3: Example of an dependency matrix.

2. All negative relationships with regard to one element of matrix $H(t)$ must be combined to satisfy one single negative constraint. In other words, if an item is only accessible when other events take place, the absence of only one of these events will make the item non–accessible. This is formally captured by the following expression:

$$\left.\begin{array}{c} \text{if } (h_{\varsigma(i,m)} = \texttt{ACC}) \quad \wedge \\ \big(\exists j \text{ such that:} \\ r_{ij} = \texttt{NEG\_DR} \quad \wedge \\ ((h_{\varsigma(j,m)} = \texttt{NO\_ACC}) \vee (h_{\varsigma(j,m)} = \texttt{UNKNO}))\big) \end{array}\right\} \Rightarrow \quad h_{\varsigma(i,m)} = \texttt{NO\_ACC} \tag{7.13}$$

3. If for a given negative dependency relationship ($r_{ij} = \texttt{NEG\_DR}$) the corresponding opposite positive dependency ($r_{ji} = \texttt{POS\_DR}$) is not defined, then a certain sequence of events is being forced in the protocol and particular messages could invalidate certain tokens for ever. In other words, if a token is made invalid because a previous event has not taken place and there is no rule to make that token available again, then the item will stay non-accessible until the end of the protocol execution. Hence, a negative dependency link without the corresponding positive rule forces a certain sequence of steps in any valid protocol solution.

4. All positive dependencies must be evaluated recursively, as positive changes in an item status can trigger other changes across the state matrix.

5. Negative relationships must always be evaluated before the positive ones.

Note that, the only restriction imposed by this representation is that negative and positive dependencies between any two given elements cannot be expressed simultaneously. By contrast, the representation provides a powerful method to

For each step denoted $s_i^\pi\langle H(i)\rangle \rightarrow H(i+1)$ do:

(1) Extract *sender* (read $s_{i,0}^\pi$)
(2) Extract *receiver* (read $s_{i,1}^\pi$)
(3) **For** $k = 0, \ldots, m-1$
  if $\left(s_{i,2+k}^\pi = 1 \text{ and } h_{sender,2+k}(i) = \texttt{ACC}\right)$ **then**
    $h_{sender,2+k}(i+1) = \texttt{LOST}$
    $h_{receiver,2+k}(i+1) = \texttt{ACC}$
    apply_dependencies to $h_{receiver,2+k}(i+1)$

Figure 7.4: Algorithm for the transference of items applying dependency relationships expressed in matrix R, as described in Section 7.2.4.

represent complex dependency relationships between all tokens involved in the exchange.

7.2.5 Updating the State Matrix

Initially, a candidate solution consists of a protocol matrix S^Π, a state matrix $H(0)$ and a dependency matrix R specific to the exchanging environment. As the protocol execution progresses, the state matrix $H(0)$ is updated according to the instructions given in each row of the protocol matrix and the positive and negative restrictions imposed by matrix R. The last of the sequence of matrices H, i.e. $H(n)$ will reflect the possessions that each entity holds at the end of the protocol and also those items that each entity has lost control over.

The following expression (7.14) serves to formally denote the updating of the state matrix $H(0)$ as indicated by the first row s_0^π of the protocol matrix S^Π. $H(1)$ is the result of this operation representing the possessions of each participant entity after step zero is executed.

$$s_0^\pi\langle H(0)\rangle \rightarrow H(1) \tag{7.14}$$

Expression (7.15) shows the consecutive updates being carried until the last row of the protocol matrix s_{n-1}^π is interpreted:

$$s_{n-1}^\pi\langle s_{n-2}^\pi\langle \ldots s_0^\pi\langle H(0)\rangle\rangle\rangle = s_{n-1}^\pi\langle H(n-1)\rangle \rightarrow H(n) \tag{7.15}$$

Finally, each step denoted $s_i^\pi\langle H(i)\rangle \rightarrow H(i+1)$ can be detailed in the algorithm for the transference of items shown in Figure 7.4.

Protocol step t	**State matrix** $H(t+1)$ **after step** t			**State matrix** $H(t+1)$ **after applying dependency matrix**		
$t=0$ $P_0 \rightarrow P_1 : o_0$	LOST ACC UNKNO	UNKNO NO_ACC UNKNO	UNKNO UNKNO ACC	LOST ACC UNKNO	UNKNO ACC UNKNO	UNKNO UNKNO ACC
$t=1$ $P_1 \rightarrow P_2 : o_1$	LOST ACC UNKNO	UNKNO LOST ACC	UNKNO UNKNO ACC	LOST ACC UNKNO	UNKNO LOST NO_ACC	UNKNO UNKNO ACC
$t=2$ $P_2 \rightarrow P_0 : o_1, o_2$	LOST ACC UNKNO	UNKNO LOST NO_ACC	ACC UNKNO LOST	LOST ACC UNKNO	UNKNO LOST ACC	ACC UNKNO LOST

Figure 7.5: Algorithm for the transference of items applied to protocol matrix S^{Π} given in Fig. 7.1, initial state matrix $H(0)$ described in Fig. 7.2 and dependency matrix R shown in Fig. 7.3.

Example

As an example, Figure 7.5 illustrates the application of the algorithm shown in Fig. 7.4 to the protocol matrix S^{Π} given in Fig. 7.1, the initial state matrix $H(0)$ described in Fig. 7.2 and the dependency matrix R shown in Fig. 7.3.

In the example note that the protocol dictates a series of instructions for all participant entities but, at each step, the real message content is subject to other factors specified in the state matrix and in the dependency matrix. For instance, at step three in the protocol, entity P_2 is supposed to send entity P_0 items o_1 and o_2. However, entity P_2 has no access to item o_1 at that point in the protocol execution so message 3 only contains item o_2.

How good the protocol S^{Π} (example Fig. 7.1) is in giving solution to a particular exchange problem, or whether the protocol is rational will be decided when defining a benefit matrix and a utility function. These two concepts are described in the following section.

7.3 Utility Function

A utility function is individually defined for each participant of the protocol to evaluate the gains obtained at each step of the scheme.

7.3.1 Benefit Matrix

In our model, all participants assign every item involved in the exchange a particular value depending on whether that entity is interested in gaining access to that item

$$B = \begin{pmatrix} o_0 & o_1 & o_2 \\ \texttt{COST} & \texttt{NO\_COST} & 5 \\ 3 & \texttt{COST} & \texttt{NO\_COST} \\ \texttt{NO\_COST} & 3 & \texttt{BENEF} \end{pmatrix}$$

Figure 7.6: Example of a benefit matrix.

and, furthermore, whether the entity increases cost by sending that item, or by keeping it. Those values also serve to represent each individual's set of requirements and are captured in the following matrix.

Matrix $B = [b_{i,j}] \in \mathcal{M}_{v \times m}$ is defined as:

$$b_{i,j} = \begin{cases} \texttt{COST} & \text{iff } P_i \text{ incurs cost when losing control over } o_j \\ \texttt{NO\_COST} & \text{iff item } o_j \text{ is of no value to participant } P_i \\ \texttt{BENEF} & \text{iff } P_i \text{ obtains benefit when losing control over } o_j \\ > 1 & \text{iff item } o_j \text{ is required by participant } P_i \end{cases} \tag{7.16}$$

In the case of b_{ij} being greater than one, b_{ij} also represents the value that item o_j is worth to entity P_i if and only if o_j becomes accessible to P_i.

The following numerical values assigned to each element in matrix B will allow us to compute entities' partial and final benefits, by multiplying each row of matrices H and B. The values are:

$$\begin{aligned} \texttt{COST} &= \quad 1 \\ \texttt{NO\_COST} &= \quad 0 \\ \texttt{BENEF} &= -1 \end{aligned} \tag{7.17}$$

As an example, matrix B shown in Fig. 7.6 represents a benefit matrix in which entity P_0 increases cost when losing control over item o_0 and item o_2 is worth five units to P_0. In a similar way, item o_0 is worth three units to entity P_1 and losing control over item o_1 decreases P_1's payoff. Finally, entity P_2 values item o_1 with three units while holding onto item o_2 does not result profitable.

7.3.2 Maximum and Minimum Benefit Values

The following criteria will serve: (1) to compute the maximum benefit that an entity can obtain in a single protocol run; and (2) to compute the minimum benefit that each entity P_i will obtain, which satisfies its requirements.

- The maximum benefit $\hat{b}_i$ represents the payoff obtained when the outcome of the protocol run is the most favorable for entity P_i. It is computed considering that the entity has gained access to all its required items, plus it has sent all items for which losing control over is beneficial, and has kept all items for

which sending represents a cost.

$$\hat{b}_i = \sum_{j=0}^{m-1} \mid b_{ij} \mid \tag{7.18}$$

- A minimum benefit value $\bar{b}_i$ represents the minimum payoff that entity P_i would expect to obtain with the exchange. The minimum that a rational entity will consider as satisfactory is that in which: the entity has gained access to all required items, has had to lose control over items for which sending represents a cost, and it does not possess any of the items for which replaying is beneficial.

$$\bar{b}_i = \sum_{b_{ij}>1} b_{ij} - \sum_{b_{ij}=\texttt{COST}} b_{ij} \tag{7.19}$$

  Furthermore, we denote $\bar{b} = \left(\bar{b}_0, \ldots, \bar{b}_{v-1}\right)$, the vector of minimum payoff values for each entity P_i, $i \in \{0, \ldots, v-1\}$.

We will now define a utility function to compute participants' payoff values after each step in the protocol, as well as global protocol payoff value at the end of a protocol run.

7.3.3 Participant Payoff and Differential Payoff

At each step of the protocol (after updating the state matrix according to the algorithm for the transference of items), we can compute the gains achieved by a player so far and refer to those as "utility" or "payoff" values.

Definition 7.3.1 (Participant payoff)**.** *Given a protocol matrix of the form S^{Π}, the payoff value for participant P_i after step $t-1$, $(1 \leqslant t \leqslant n)$, can be computed as:*

$$u_i(t) = \sum_{\substack{j=0 \\ h_{i,j} \neq \texttt{NO\_ACC}}}^{m-1} b_{ij} h_{ij}(t), \quad i \in \{0, \ldots, v-1\} \tag{7.20}$$

As an example, $u_i(1)$ represents participant P_i's payoff after step zero of the protocol and $u_i(n)$ is the payoff attained by P_i after the last step $t-1$.

Also note that:

- $u_i(0)$ denotes the initial utility for entity P_i
- non–accessible items do not increase the overall utility.

Furthermore, we denote $u(t) = \big(u_0(t), \ldots, u_{v-1}(t)\big)$, the vector of payoff values for each entity P_i, $i \in \{0, \ldots, v-1\}$, at step t.

Definition 7.3.2 (Participant differential payoff)**.** *Given a protocol matrix of the form S^{Π}, the differential payoff value for a player P_i between steps t_1 and t_2, with $0 \leqslant t_1 \leqslant t_2 \leqslant n$, is defined as:*

$$du_i(t_1, t_2) \quad = u_i(t_2) - u_i(t_1) \tag{7.21}$$

During a protocol execution, there may be steps at which players go into a temporarily "worst" state (i.e. $du_i(t, t+1) \leqslant 0$). The relevant fact, however, is whether at the end of the protocol run P_i gets enough differential utility:

- If $du_i(0, n) > 0$, the exchange is profitable to P_i,
- If $du_i(0, n) < 0$, the exchange is non–profitable to P_i,
- If $du_i(0, n) = 0$, the exchange is of no use to P_i,

7.3.4 Protocol Global Payoff and Protocol Differential Payoff

Additionally, a global protocol utility function will be defined to describe the overall payoff of a protocol solution S^{Π}.

Definition 7.3.3 (Protocol global payoff)**.** *Given the space of all protocol matrices of the form S^{Π}, we define a function $U : S^{\Pi} \to \mathbb{R}$ which assigns a utility value to every given protocol, such that:*

$$U(S^{\Pi}) \quad = \sum_{i=0}^{v-1} u_i(n) = \sum_{i=0}^{v-1} \sum_{\substack{j=0 \\ h_{i,j} \neq \texttt{NO\_ACC}}}^{m-1} b_{ij} h_{ij}(n) \tag{7.22}$$

Likewise, an overall differential utility is defined.

Definition 7.3.4 (Protocol global differential payoff)**.** *Given a protocol matrix of the form S^{Π}, the protocol global differential utility is defined as:*

$$dU(S^{\Pi}) \quad = \sum_{i=0}^{v-1} du_i(0, n) \tag{7.23}$$

7.3.5 Protocol Matrix Concatenation

Given two exchange protocols Π_1 and Π_2 with equal number of participants and number of tokens, these can be concatenated by adding the set of instructions of

protocol Π_2 to the end of protocol Π_1. Similarly, two protocol matrices can be concatenated to describe a new protocol.

The following operator formally expresses protocol matrix concatenation.

Definition 7.3.5 (Protocol matrix concatenation)**.** *Let P be a set of entities and $\mathcal{O}$ a set of exchangeable tokens. Given two protocols $\Pi_1 = \langle P, \mathcal{O}, T_1 \rangle$ and $\Pi_2 = \langle P, \mathcal{O}, T_2 \rangle$, the protocol resulting of concatenating Π_1 and Π_2 is denoted as:*

$$\Pi_1 \circ \Pi_2 = \langle P, \mathcal{O}, T_1 \circ T_2 \rangle \tag{7.24}$$

where $T_1 \circ T_2$ denotes the concatenation of sequences T_1 and T_2 of protocol steps, from protocols Π_1 and Π_2 respectively.

The protocol matrix resulting from concatenating protocols Π_1 and Π_2 is denoted as: $S^{\Pi_1 \circ \Pi_2}$.

The following theorem serves to prove the non–linearity of utility function U defined in Definition 7.3.3.

Theorem 7.3.1. *Given two protocol matrices S^{Π_1} and S^{Π_2} with the same number v of entities and the same number m of tokens, the utility function U is non–linear with respect to protocol concatenation:*

$$U(S^{\Pi_1 \circ \Pi_2}) \neq U(S^{\Pi_1}) + U(S^{\Pi_2}) \tag{7.25}$$

Proof. Let p and q be the number of steps of protocols Π_1 and Π_2 respectively. The process of executing protocol Π_1 over an initial state described by a matrix $H(0)$, can be expressed using the notation from Section 7.2.5. This is:

$$s_{p-1}^{\pi_1} \langle s_{p-2}^{\pi_1} \langle \dots s_0^{\pi_1} \langle H(0) \rangle \rangle \rangle \rightarrow H(p) \tag{7.26}$$

Utility value $U(S^{\Pi_1})$ will then be computed with elements from matrix $H(p)$.

In a similar way, the process of executing protocol Π_2 over an initial state described by matrix $H(0)$ can be expressed as:

$$s_{q-1}^{\pi_2} \langle s_{q-2}^{\pi_2} \langle \dots s_0^{\pi_2} \langle H(0) \rangle \rangle \rangle \rightarrow H(q) \tag{7.27}$$

and the corresponding utility value $U(S^{\Pi_2})$ will be computed with elements from matrix $H(q)$.

By contrast, the process of executing protocol $\Pi_1 \circ \Pi_2$ can be expressed as:

$$\begin{aligned} &s_{q-1}^{\pi_2} \langle s_{q-2}^{\pi_2} \langle \dots \langle s_0^{\pi_2} \langle s_{p-1}^{\pi_1} \langle s_{p-2}^{\pi_1} \langle \dots s_0^{\pi_1} \langle H(0) \rangle \rangle \rangle \rangle \rangle \rangle \rangle = \\ &s_{q-1}^{\pi_2} \langle s_{q-2}^{\pi_2} \langle \dots \langle s_0^{\pi_2} \langle H(p) \rangle \rangle \rangle \end{aligned} \tag{7.28}$$

□

Note that, the execution of protocol Π_2 starts over an initial matrix $H(p)$ which will, in most cases, be different from the original $H(0)$. Hence, applying the utility function to evaluate the protocol matrix $S^{\Pi_1 \circ \Pi_2}$ will produce a different result than evaluating S^{Π_1} and S^{Π_2} separately.

7.4 Solution Space – Goals and Dimension

In simple terms, our goal will be to explore the space of all exchange protocols to find rational schemes for which all participants' utility values are maximum or the nearest possible, and above the minimum required.

We formalize the previous statement in the following definition.

Definition 7.4.1 (Search goal)**.** *Given the space of all protocol matrices of the form S^{Π}, the goal of the search is to find S^{Π} such that $U(S^{\Pi})$ is maximum, subject to S^{Π} satisfying the following conditions:*

(1) The final payoff value must be above the minimum required by each entity P_i. This is:

$$\forall i \in \{0, \ldots, v-1\}, \quad \bar{b_i} \leqslant u_i(n) \tag{7.29}$$

(2) Any entity that has obtained its minimum payoff required, would only maintain an active role in the protocol (being sender or recipient of messages) when the benefit attained for that action directly increases its individual payoff. This is:

$$\begin{array}{l} \forall i \in \{0, \ldots, v-1\}, \textit{ if } \exists t < n \textit{ such that } \bar{b_i} \leqslant u_i(t) \textit{ then} \\ \forall k, t < k \leqslant n\textit{: } (s^{\Pi}_{k,0} = i \vee s^{\Pi}_{k,1} = i) \Rightarrow u_i(t) < u_i(k) \end{array} \tag{7.30}$$

As previously stated, our proof system will rely on the model described in part I of this thesis. That formalism, based on Game Theory, will allow us to ensure that the schemes satisfying the search goals are in fact *rational* solutions to the exchange problem in hand.

7.4.1 Rationality Proof

The basic idea is to represent synthesized exchange protocols satisfying conditions *(1)* and *(2)* as dynamic games of perfect information.

Informally, a given exchange protocol represents a *feasible* solution to an exchange problem when the protocol describes a series of steps allowing entities to exchange their commodities. However, most of these feasible solutions can only

occur when pre-execution agreements take place between agents, typically in private environments.

On the other hand, a given exchange protocol represents a *rational* solution to an exchange problem when the protocol allows entities to exchange the desired commodities and the protocol outcome constitutes a Nash equilibrium (*perfect in sub–games*) of the corresponding protocol game. Ensuring that the protocol outcome is a Nash equilibrium of the protocol game will ensure that entities do not deviate from such an outcome, as unilaterally changing strategy does not result in a higher payoff value.

Next we will proof that the goals of the search render rational schemes for any given multi–party exchange problem.

Theorem 7.4.1. *Any given exchange protocol* Π*, represented by protocol matrix* S^{Π} *and satisfying conditions (1) and (2) in Definition 7.4.1, is a rational–exchange protocol.*

Proof. Let $G_{S^{\Pi}}$ represent the protocol game derived from the description of protocol S^{Π}. The Nash equilibrium points of the game $G_{S^{\Pi}}$ can be computed applying a very simple *backward induction algorithm*. The process would proceed by first considering the last actions of the final participant of the protocol. It determines which action the final entity would take to maximize its utility. Using this information and taking the induction one step backward, one can then determine what the second to last participant will do to maximize its own utility function too. This process continues until one reaches the first participant of the protocol, hence determining the actions of all consecutive participants.

Formally, the process can be sketched as follows:

- **At the last step in the protocol:**

  Let $\left[P_{s_n} \rightarrow P_{r_n} : m_n \text{ where } s_n \neq r_n \text{ with } s_n, r_n \in \{0, \ldots, v-1\}\ \right]$ be the last step in the protocol Π. At that point, the sender P_{s_n} has to choose between two possible strategies: *quit* or *send_*m_n. The following are P_{s_n}'s final payoffs:

Strategy	Final Payoff
P_{s_n} sends message m_n	$u_{s_n}(n)$
P_{s_n} quits	$u_{s_n}(n-1)$

  Let us suppose that:

$$u_{s_n}(n-1) \geqslant u_{s_n}(n) \tag{7.31}$$

Note that in this case, the protocol could not be called rational as it would be dictating P_{s_n} to take an irrational action which would go against its own self-interest.

However, satisfying conditions (1) and (2) will imply that:

$$u_{s_n}(n-1) \geqslant \bar{b_{s_n}} \Rightarrow u_{s_n}(n-1) < u_{s_n}(n) \tag{7.32}$$

This clearly contradicts assumption 7.31. So rational entity P_{s_n} is forced to choose strategy $send\_m_n$, as in doing so the payoff value obtained is greater.

- **At the step before last in the protocol:**

  Let $\left[P_{s_{n-1}} \rightarrow P_{r_{n-1}} : m_{n-1} \text{ where } s_{n-1} \neq r_{n-1} \text{ with } s_{n-1}, r_{n-1} \in \{0, \ldots, v-1\}\right]$ be the step before last in the protocol. Then, $P_{s_{n-1}}$ will have to choose between two possible strategies: {quit, $send\_m_{n-1}$}.

  As $P_{s_{n-1}}$ knows that, given the opportunity, rational entity P_{s_n} will choose action $send\_m_n$ at the last step of the protocol. The following are the final payoffs obtained by following each of the strategies:

Strategy	Final Payoff
$P_{s_{n-1}}$ sends message m_{n-1}	$u_{s_{n-1}}(n)$
$P_{s_{n-1}}$ quits	$u_{s_{n-1}}(n-2)$

  Let us suppose that:

$$u_{s_{n-1}}(n-2) \geqslant u_{s_{n-1}}(n) \tag{7.33}$$

  Then, satisfying conditions *(1)* and *(2)* imply that:

$$\Rightarrow u_{s_{n-1}}(n-2) < u_{s_{n-1}}(n-1) \tag{7.34}$$

  Furthermore, if $P_{s_{n-1}}$ is the sender at the last step of the protocol then:

$$u_{s_{n-1}}(n-1) < u_{s_{n-1}}(n) \Rightarrow u_{s_{n-1}}(n-2) < u_{s_{n-1}}(n) \tag{7.35}$$

  and, if $P_{s_{n-1}}$ is not sender again then:

$$u_{s_{n-1}}(n-1) = u_{s_{n-1}}(n) \Rightarrow u_{s_{n-1}}(n-2) < u_{s_{n-1}}(n) \tag{7.36}$$

Both results contradict initial assumption 7.33. So, entity $P_{s_{n-1}}$ is forced to choose strategy $send\_m_{n-1}$, as in doing so the expected payoff value obtained at the end of the protocol is greater.

- **In general, at any intermediate step in the protocol:**

  Let $\left[P_{s_k} \rightarrow P_{r_k} : m_k \text{ where } s_k \neq r_k \text{ with } s_k, r_k \in \{0, \ldots, v-1\}\ \right]$ be the last step in the protocol. At that point, P_{s_k} will have to choose between two possible strategies: *quit* or $send\_m_k$.

  As P_{s_k} knows how rational participants will behave in future steps, the following are the final payoffs obtained by following each of the two possible strategies:

Strategy	Final Payoff
P_{s_k} sends message m_k	$u_{s_k}(n)$
P_{s_k} quits	$u_{s_k}(k-1)$

  Suppose that:

$$u_{s_k}(k-1) \geqslant u_{s_k}(n) \tag{7.37}$$

  Then, applying conditions *(1)* and *(2)* we have that:

$$u_{s_k}(k-1) \geqslant \bar{b_{s_k}} \Rightarrow u_{s_k}(k-1) < u_{s_k}(k) \tag{7.38}$$

  a. If P_{s_k} is sender again in future steps then:

$$u_{s_k}(k) < u_{s_k}(n) \Rightarrow u_{s_k}(k-1) < u_{s_k}(n) \tag{7.39}$$

     This clearly contradicts initial assumption 7.37.

  b. If P_{s_k} is not sender again then:

$$u_{s_k}(k) = u_{s_k}(n) \Rightarrow u_{s_k}(k-1) < u_{s_k}(n) \tag{7.40}$$

     Contradicting our initial assumption 7.37.

  So, at any intermediate step in the protocol any entity P_{s_k} is forced to choose strategy $send\_m_k$ as in doing so, the expected payoff value obtained at the end of the protocol is greater.

- **At the first step in the protocol:** The same reasoning applies to all consecutive steps until one reaches the first step in the protocol. At step one, the sender entity can predict every other participant future action so, no

entity would deviate from the protocol description, as the highest payoff values are obtained when participants choose strategies according to the protocol instructions.

This way, by definition, the protocol outcome attained when following the steps of the protocol constitutes a Nash equilibrium perfect in sub-games for the corresponding protocol game G_{S^Π}. This being the formal proof that any synthesized protocol, result of our proposed design technique, is rational. □

7.4.2 How Many Exchange Protocols Exist?

As described in Section 7.2, a protocol is represented by a matrix $S^\Pi \in \mathcal{M}_{n\times(m+2)}$, where n is the number of protocol steps and m is the number of tokens involved in the exchange. Each row represents a message in the protocol such that, the first two components of each row describe the sender and receiver of that message respectively, so the number of possible combinations sender–recipient amongst v entities is $\frac{v!}{(v-2)!}$. Furthermore, in each row, elements three to m are in {0,1} and represent the items being sent. In this case, there are 2^{nm} possible combinations. We can then compute an estimate of the total number of exchange protocols subject to evaluation as:

$$\mathcal{O}\big(\frac{v!}{(v-2)!}2^{nm}\big) = \mathcal{O}\big(v(v-1)2^{nm}\big) = \mathcal{O}(v^2 2^{nm}) \tag{7.41}$$

For example, for a 3 entity scenario, a maximum of 10 messages exchanged and a total of 6 items in each message, the search space has an estimated complexity of 2^{63} possible protocols.

Although, the aforementioned expression gives an estimate of how many exchange protocols there are, it is difficult to determine how many of these protocols represent *feasible* solutions to the specific exchange problem, and even more challenging is to estimate how many of those feasible solutions represent a rational exchange.

7.4.3 Finding a Solution is Hard

Definition 7.4.1 describes a optimization problem of the form introduced in Section 6.2. However, two main aspects make the problem of finding the maximum solution hard. These are:

1. *The size of the solution space.* As the number of entities or the number of exchangeable items grows, this makes it impossible to evaluate every possible

candidate solution, in search of a protocol matrix with maximum global payoff value.

2. *The problem is a non–linear programming problem.* Indeed, Theorem 7.3.1 proved utility function U to be a non–linear function with respect to protocol concatenation. In this case then, well established linear programming algorithms (simplex-based methods or interior point methods) cannot be applied to resolve this particular optimization problem.

Our approach (further detailed in subsequent chapters) will be to apply optimization techniques based on heuristic search algorithms.

7.5 A Taxonomy for M–RES Protocols

In this section, we use the formal model just described, in particular the combination of values in state matrix H and benefit matrix B, to classify M–RES protocols.

Traditionally, in any given exchange scheme, once the exchanging objectives have been achieved, motivation is derived from incentive schemes of many kinds (reputation factors, loyalty schemes, etc.) and/or, the presence of coalitions between participant entities.

Informally, an incentive scheme is an external, artificial reinforcement to make entities behave in a certain way which a priori would not appear to be rational. In an environment where entities are self-interested and aimed at maximizing their own utility values, an incentive scheme must always represent a bonus on entities' payoff values.

In a similar way, when an entity is part of a coalition, helping other members of the same coalition to achieve their goals must also report an increase in the payoff values of the coalition members.

In this section we will give formal definitions of these concepts (incentives and coalitions) and we will classify protocol participants according to whether entities are incentivized and/or members of a coalition.

7.5.1 Incentive Schemes

Within our taxonomy, an incentive scheme is considered to be a mechanism by which entities are motivated to exchange their own items.

Usually, participants of an exchange protocol are driven by the desire of gaining access to their required items and will be eager to receive those without sending anything in return. An incentive scheme will motivate participants to lose control

over the items they own, so other entities can gain access to them. The incentive represents a bonus over their payoff if their own items are sent to other participants.

For example, a reputable e-merchant will send the corresponding e-goods after receiving payment from a buyer, as its future business might depend on the outcome of the current transaction. In this case, an incentive scheme could be an external reputation system forcing the merchant to behave honestly in each transaction.

Usually, participants of a protocol are not all motivated by the same incentive factor. Some protocol participants might be motivated by a reputation factor while others might respond to some kind of legal enforcement. In other words, for a given protocol, it could be that an incentive program is applicable to only one portion of the participant entities leaving the rest out of any incentive schemes.

Furthermore, an entity might be incentivized towards the exchange of a particular item whereas there could be no incentive in losing control over a different one.

With regard to this definition of incentive scheme we can identify and classify two types of entity: *Incentivized* and *Non–incentivized*.

Definition 7.5.1 (Incentivized entity)**.** *Given an entity P_i and an item o_j which P_i initially owns or is capable to generate, entity P_i is incentivized towards the exchange of item o_j, if P_i can increase its payoff value by losing control over o_j.*

In other words, sending item o_j represents a benefit for P_i.

Definition 7.5.2 (Non–incentivized entity)**.** *Given an entity P_i and an item o_j which P_i initially owns or is capable to generate, entity P_i is non–incentivized towards the exchange of item o_j, if P_i's payoff value decreases by losing control over o_j.*

In other words, sending item o_j represents a cost for P_i. A non–incentivized entity towards and item o_j is motivated to keep control over that item.

7.5.2 Coalition Schemes

Within our taxonomy, a coalition scheme is considered a mechanism by which entities are rationally forced to help allied participants to achieve their goals. This is done by forwarding other participant's items when these are required by allied members.

Typically, there could be different coalitions formed amongst several participants of a given protocol while others entities might stay single. Furthermore, these coalitions might intersect (one entity belongs to more than one coalition) or they might form disjointed groups.

With regard to this definition of coalition and in relation to protocol participants we can identify and classify two types of entities: *Collaborative* and *Non–collaborative*.

Definition 7.5.3 (Collaborative entity)**.** *An entity P_i is collaborative towards another entity P_j, if P_i can increase its payoff by forwarding those items required P_j, subject to having received them and having got access to them at some earlier point in the protocol.*

Note that this is only related to items which P_i does not previously own or is not able to generate, but items which are sent to P_i by other entities. A collaborative entity P_i towards another entity P_j represents a coalition between entities P_i and P_j with regard all items required by each one of them.

Definition 7.5.4 (Non–Collaborative entity)**.** *An entity P_i is non–collaborative towards another entity P_j, if P_i's payoff value decreases by forwarding those items required P_j subject to having received them and having got access to them at some earlier point in the protocol.*

In other words, if entities P_i and P_j are not part of a coalition then relying the items required by each other, represents a decrease in their payoff values.

7.5.3 Formal Representation of Incentives and Coalitions

The combination of the values from initial state matrix $H(0)$ (Equation (7.4)) and benefit matrix B (Equation (7.16)) are used to represent and identify different types of entity and therefore different types of exchanging scenarios. Equations 7.4 and 7.16 gave the semantics for the different values that these matrices could hold during the protocol execution. In this section we will add significance to all possible combinations of these values as follows:

- If $h_{i,j}(0)$=`ACC` or $h_{i,j}(0)$=`NO_ACC` (i.e. item o_j belongs to entity P_i at the initial state of the protocol or P_i will be able to generate o_j at some stage along the protocol execution).

  In either case, the corresponding $b_{i,j}$ value will have the following semantics:

  - If $b_{i,j}$ = `BENEF`, entity P_i is incentivized to carry the exchange of item o_j. This is, sending item o_j represents an increase in P_i's payoff value.
  - If $b_{i,j}$ = `COST`, entity P_i is not incentivized to carry the exchange of item o_j. This is, losing control over item o_j represents a decrease on P_i's payoff value.

	$b_{i,j} = \texttt{NO\_COST}$	$b_{i,j} = \texttt{COST}$	$b_{i,j} = \texttt{BENEF}$	$b_{i,j} > 1$
$h_{i,j}(0) = \texttt{ACC} \vee$ $h_{i,j}(0) = \texttt{NO\_ACC}$	P_i indifferent to the exchange	P_i non–incentivized to exchange o_j	P_i incentivized to exchange o_j	Non–applicable
$h_{i,j}(0) = \texttt{UNKNO}$	P_i indifferent towards forwarding o_j	P_i not allied with P_k if P_k requires o_j	P_i allied with P_k if P_k requires o_j	Item o_j required by P_i

Table 7.1: Incentives and coalitions table.

  - If $b_{i,j} = \texttt{NO\_COST}$, entity P_i is indifferent towards item o_j.

  Note that P_i losing control over an item o_j at step t in the protocol execution decreases its utility value defined in equation (7.20) by two units. By contrast, if $b_{i,j} = \texttt{BENEF}$, the payoff value is increased by two.

- If $h_{i,j}(0) = \texttt{UNKNO}$ (i.e. item o_j is unknown to entity P_i at the initial state of the protocol).

  The corresponding $b_{i,j}$ value will have the following semantics:

  - If $b_{i,j} > 1$, item o_j is one of the required items by entity P_i and $b_{i,j}$ represents how much item o_j is worth to entity P_i.
  - If $b_{i,j} = \texttt{BENEF}$, entity P_i is part of a coalition with whatever entity is requiring item o_j.
  - If $b_{i,j} = \texttt{COST}$, entity P_i is not part of any coalition with whatever entity is requiring item o_j.
  - If $b_{i,j} = \texttt{NO\_COST}$, entity P_i is indifferent towards forwarding item o_j.

Note that if P_i is not part of a coalition with entity P_k requiring o_j, then relaying item o_j represents a cost for entity P_i according to equation (7.20). By contrast, if P_i and P_k are allies, and P_i has got the chance to forward item o_j required by entity P_k then, P_i immediate utility value is increased.

Table 7.1 summarizes every possible combination of values.

7.6 Protocol Classification

Rational–exchange protocols can be typified attending the different types of entity involved in the exchange. Our taxonomy is based on identifying two main aspects:

(1) whether entities are incentivized and (2) whether they are part of any coalition. These criteria define different types of environments in which a protocol is executed.

7.6.1 Classification Attending Incentives

Symmetrically Incentivized. All participants are rational and part of an incentive scheme of one kind or another. That is, all entities derive a positive payoff when losing control over each one of their own items.

We can formally represent this type of environment as follows:

$$\begin{gathered}\forall i \in \{0,\ldots,v-1\} \quad and \quad \forall j \in \{0,\ldots,m-1\} \\ (h_{i,j}(0) = \texttt{ACC} \vee h_{i,j}(0) = \texttt{NO\_ACC}) \Rightarrow b_{i,j} = \texttt{BENEF}\end{gathered} \tag{7.42}$$

Symmetrically Non–Incentivized. All entities are rational and increase their costs when losing control over their own items.

We can formally represent this type of environment as follows:

$$\begin{gathered}\forall i \in \{0,\ldots,v-1\} \quad and \quad \forall j \in \{0,\ldots,m-1\} \\ (h_{i,j}(0) = \texttt{ACC} \vee h_{i,j}(0) = \texttt{NO\_ACC}) \Rightarrow b_{i,j} = \texttt{COST}\end{gathered} \tag{7.43}$$

Mixed with respect to incentives. All entities are rational but some entities could be incentivized with respect to one or more of their items as well as there will be entities which are not incentivized at all.

7.6.2 Examples

To illustrate the previous classification, we will consider the following example: a two–entity scenario, each entity in possession of one single item which is required by the other participant. Additionally, an arbitrary value $val_1 = 3$ is chosen to represent what item o_1 is worth to entity P_0 and, a value of $val_2 = 5$ will correspond to what item o_0 is worth to entity P_1.

The following are the initial state and the benefit matrices representing the exchange problem described:

$$H(0) = \begin{pmatrix} \texttt{ACC} & \texttt{UNKNO} \\ \texttt{UNKNO} & \texttt{ACC} \end{pmatrix} \quad B = \begin{pmatrix} - & val_1 \\ val_2 & - \end{pmatrix} \tag{7.44}$$

Finally, the following are the sequence of matrices $H(t)$, when executing a protocol in which P_0 sends P_1 item o_0 and then P_1 sends P_0 item o_1:

$$H(0) = \begin{pmatrix} \texttt{LOST} & \texttt{UNKNO} \\ \texttt{ACC} & \texttt{ACC} \end{pmatrix} \quad H(1) = \begin{pmatrix} \texttt{LOST} & \texttt{ACC} \\ \texttt{ACC} & \texttt{LOST} \end{pmatrix} \tag{7.45}$$

- *Example of a symmetrically incentivized exchange.* All participants are considered to be part of an incentive scheme, i.e. all entities derive a positive payoff when losing control over each one of their owned items. The diagonal of the benefit matrix B will be set to -1.

$$\text{Benefit matrix } B = \begin{pmatrix} \texttt{BENEF} & val_1 \\ val_2 & \texttt{BENEF} \end{pmatrix}$$

The following are the differential utilities for each participant entity when executing the protocol defined by matrices (7.45):

$$\begin{aligned} &u_0(0) = -1 \\ &u_0(1) = 1 \\ &u_0(2) = 4 \\ &du_0(0,2) = u_0(2) - u_0(0) = 5 \end{aligned} \tag{7.46}$$

$$\begin{aligned} &u_1(0) = -1 \\ &u_1(1) = 4 \\ &u_1(2) = 6 \\ &du_1(0,2) = u_1(2) - u_1(0) = 7 \end{aligned} \tag{7.47}$$

Note that the protocol defined by matrices (7.45), when executed by the type of entity defined in this example, results beneficial for both participants.

- *Example of a symmetrically non–incentivized exchange.* All entities incur cost when sending their items. In this case, the diagonal of the benefit matrix will be set to 1.

$$\text{Benefit matrix } B = \begin{pmatrix} \texttt{COST} & val_1 \\ val_2 & \texttt{COST} \end{pmatrix}$$

The following are the differential utilities for each participant entity when executing the protocol defined by matrices (7.45):

$$\begin{aligned} &u_0(0) = 1 \\ &u_0(1) = -1 \\ &u_0(2) = 2 \\ &du_0(0,2) = u_0(2) - u_0(0) = 1 \end{aligned} \tag{7.48}$$

$$\begin{aligned} &u_1(0) = 1 \\ &u_1(1) = 6 \\ &u_1(2) = 4 \\ &du_1(0,2) = u_1(2) - u_1(0) = 3 \end{aligned} \tag{7.49}$$

By contrast, in the type of exchange scenario described in this example, the exchange does not result in any benefit for participant P_0 whereas, for entity P_1, the exchange is profitable. However, also note that P_1 is not motivated to execute the last step of the protocol as the payoff value encountered at step one is greater than the final. The scheme does not result *rational* from entity's P_1 point of view.

- *Example of a mixed exchange.* Mixed environment for the exchange, there is no symmetry with regard incentives. The diagonal of the benefit matrix will contain -1 and 1 values.

$$\text{Benefit matrix } B = \begin{pmatrix} \texttt{COST} & val_1 \\ val_2 & \texttt{BENEF} \end{pmatrix}$$

The following are the differential utilities for each participant entity, when executing the protocol defined by matrices (7.45):

$$\begin{aligned} &u_0(0) = 1 \\ &u_0(1) = -1 \\ &u_0(2) = 2 \\ &du_0(0,2) = u_0(2) - u_0(0) = 1 \end{aligned} \tag{7.50}$$

$$\begin{aligned} &u_1(0) = -1 \\ &u_1(1) = 4 \\ &u_1(2) = 6 \\ &du_1(0,2) = u_1(2) - u_1(0) = 7 \end{aligned} \tag{7.51}$$

Note that, for entity P_0, the protocol defined by matrices (7.45), in a mixed exchange scenario of the type described in this example, does not result in any benefit. By contrast, for entity P_1, the exchange reports a substantial profit.

7.6.3 Classification Attending Coalitions

Attending to whether there exist coalitions amongst participants, we propose the following classification to distinguish all possible exchanging scenarios:

Coalition free. All participants are rational and none of them are part of any coalitions.

We can formally represent this type of environment as follows:

$$\begin{gathered}\forall i \in \{0,\dots,v-1\} \quad and \quad \forall j \in \{0,\dots,m-1\} \\ (h_{i,j}(0) = \texttt{UNKNO}) \wedge \neg(b_{i,j} > 1) \Rightarrow (b_{i,j} = \texttt{COST}) \vee (b_{i,j} = \texttt{NO\_COST})\end{gathered} \tag{7.52}$$

In other words, an entity P_i cannot increase its payoff by forwarding other participant's items which are not required by P_i.

With coalitions. We can formally represent this type of environment as follows:

$$\begin{gathered}\exists i \in \{0,\dots,v-1\} \text{ and } \exists j \in \{0,\dots,m-1\} \text{ s.t.} \\ (h_{i,j}(0) = \texttt{UNKNO}) \wedge (b_{i,j} = \texttt{BENEF})\end{gathered} \tag{7.53}$$

In other words, at least one entity P_i is able to increase its payoff by forwarding other participant's item o_j. In this case, P_i is said to form coalition with every P_k such that o_j is an item required by P_k (i.e. $b_{k,j} > 1$).

Note that for some matrices B we will be able to quickly identify the existence of intersections between different coalitions. If there exists a column j in B and two indices $i, i' \in \{0,\dots,v-1\}$, $i \neq i'$, such that:

$$(h_{i,j}(0) = \texttt{UNKNO}) \quad \wedge \quad (h_{i',j}(0) = \texttt{UNKNO}) \quad \wedge \quad (b_{i,j} = \texttt{BENEF}) \quad \wedge \quad (b_{i',j} = \texttt{BENEF}) \tag{7.54}$$

Then, P_i and P_i' will form two different coalitions with P_k, if entity P_k requires item o_j.

Note that, although the total number of possible coalitions is 2^v, an exchanging scenario could be characterized by more than one alliance, and these could easily intersect in zero, one or more participants increasing the total number of possible different scenarios to 2^{2^v}.

7.6.4 Examples

To illustrate the previous classification, we will consider the following example: a four–entity scenario, each entity in possession of one single item to be exchanged,

such that: P_0 requires o_3, P_1 requires o_0, P_2 requires o_1 and P_3 requires o_2. The same arbitrary value *val* is chosen to represent what each the required items is worth to each requiring entity. Also, we consider a symmetrically incentivized exchange, i.e. participants are incentivized towards the exchange of their items.

The following matrices define such a scenario:

$$\text{Initial state matrix } H(0) = \begin{pmatrix} \texttt{ACC} & \texttt{UNKNO} & \texttt{UNKNO} & \texttt{UNKNO} \\ \texttt{UNKNO} & \texttt{ACC} & \texttt{UNKNO} & \texttt{UNKNO} \\ \texttt{UNKNO} & \texttt{UNKNO} & \texttt{ACC} & \texttt{UNKNO} \\ \texttt{UNKNO} & \texttt{UNKNO} & \texttt{UNKNO} & \texttt{ACC} \end{pmatrix}$$

$$\text{Benefit matrix } B = \begin{pmatrix} \texttt{BENEF} & - & - & val \\ val & \texttt{BENEF} & - & - \\ - & val & \texttt{BENEF} & - \\ - & - & val & \texttt{BENEF} \end{pmatrix}$$

- *Example coalition free.* No participant is member of a coalition with any other entity. Therefore, the benefit matrix B will be such that there will be no `BENEF` values outside the main diagonal.

$$\text{Benefit matrix } B = \begin{pmatrix} \texttt{BENEF} & \texttt{NO\_COST} & \texttt{COST} & val \\ val & \texttt{BENEF} & \texttt{COST} & \texttt{COST} \\ \texttt{NO\_COST} & val & \texttt{BENEF} & \texttt{COST} \\ \texttt{COST} & \texttt{COST} & val & \texttt{BENEF} \end{pmatrix}$$

- *Example with coalitions.* For example two intersecting coalitions are part of the exchange: P_0 with P_2 and P_0 with P_3; P_1 stays single; and P_2 and P_3 are not allied. Figure 7.7 is a graphical visualization of these intersections. Next is their representation in the benefit matrix B.

$$\text{Benefit matrix } B = \begin{pmatrix} \texttt{BENEF} & \texttt{BENEF} & \texttt{BENEF} & val \\ val & \texttt{BENEF} & \texttt{COST} & \texttt{COST} \\ \texttt{NO\_COST} & val & \texttt{BENEF} & \texttt{BENEF} \\ \texttt{NO\_COST} & \texttt{COST} & val & \texttt{BENEF} \end{pmatrix}$$

7.7 Conclusions

The formalism here described is a novel approach to the way in which exchange security protocols are described and represented. Traditionally, automated tools have always been applied to the analysis and verification of existing security

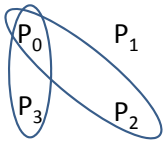


Figure 7.7: Example with coalitions.

State matrix $H(t)$	Matrix of initial and consecutive possessions. It captures what each entity holds at each step t in the protocol.
Dependency matrix R	Matrix of dependency relationships. It contains what relationships exist between the different items and how these affect other holders.
Benefit matrix B	Matrix of benefit values. For each entity it captures: the set of required items, the set of items for which relaying is beneficial and, the set of items for which not relaying is beneficial.
Protocol matrix S^{Π}	Matrix representing the protocol steps. For each step of the protocol it describes the sender, the receiver and the message content.

Table 7.2: Main components of formal model for the representation of multi–party exchange problems and candidate solutions.

protocols. In this chapter we have adopted a new approach, ensuring rationality as part of the design of an exchange scheme.

Protocols and some surrounding execution considerations have been structured and formatted in very simple linear structures. Besides the simplicity of the model, it allows us to formally represent and parameterize any given multi–party exchange scenario. The main components of the model are summarized in the Table 7.2.

Moreover, two relevant theoretical results ensure the soundness of our approach:

- Theorem 7.4.1 ensures that all synthesized protocols are demonstrably rational. A formal proof, based on backward induction is used to prove that protocols, satisfying the goals of the search, are all rational.

- Theorem 7.3.1 formally establishes the non–linearity of the chosen utility function. This makes the usage of heuristic optimization techniques totally appropriate for the search problem in hand.

Finally, this formalism sets the basis to consider further and traditionally very complex factors when designing a solution to an exchange problem. Issues such as entity coalitions or incentive schemes are easily defined and encountered within

the described model. In this sense, the work provides us with a simple taxonomy according to which synthesized rational–exchange protocols can be easily classified. In the next chapter we will present our experimental work and we will observe how, in some of the instances, the presence of certain coalitions will determine the success of a particular search, whereas in other cases the absence of these coalitions or the presence of what we can denote as *bad coalitions* will prevent the search from succeeding. Therefore, being able to represent and identify such factors in a given exchanging scenario will help in understanding the outcome of a synthesis process. In this sense, the taxonomy represents an extra tool when studying complex entity relations.

Chapter 8

Heuristic Synthesis of v–RES Protocols

8.1 Introduction

The formalism described in previous chapters allows for the application of a general purpose optimization technique to explore the space of protocols and, to potentially find rational solutions to a given multi–party exchange problem.

In this chapter we present the first results obtained when automatically synthesizing a three–party rational–exchange protocol, by applying a meta–heuristic search algorithm based on Simulated Annealing. These, we have called 3–RES protocol. Such protocol can be easily extended to a family of v–RES protocols for any number v of protocol participants.

8.1.1 Chapter Overview

The focus of our experimental work will be placed on a particular three–entity exchange problem. For every participant we will be giving a series of initial assumptions and requirements, which will be represented using the matrices described in Chapter 7. A heuristic search algorithm will then try to find a rational scheme as defined by goals in Definition 7.4.1, to solve the specific exchange problem. As a result, an automatically synthesized 3–RES protocol will be produced.

Furthermore, a whole family of multi–party security protocols can be derived from the 3–RES protocol just synthesized and, a formal analysis based on Game Theory, will serve to prove rationality of every protocol in the family.

8.1.2 Chapter Organization

The chapter is organized as follows. In Section 8.2, we apply the formalism described in Chapter 7 to the parametrization of a particular three–entity exchange problem. We also propose an heuristic search technique, based on Simulated Annealing, for the synthesis of a rational–exchange protocol to give solution to the specific exchange problem in hand. In Section 8.4 we use Theory of Landscapes to determine the level of difficulty of the task. Sections 8.5 and 8.6 serve to describe the result obtained by the proposed search technique and, to compare these with results attained using other search algorithms. Section 8.7 presents the resulting synthesized 3–RES protocol. Furthermore, in Section 8.8 the protocol is extended to a family of multi–party rational protocols for which a formal proof of rationality is given in Section 8.8.5. Finally, in Section 8.9 we present the main conclusions of this experimental work.

8.2 3–RES Problem Description

We will first give an informal description of a series of initial assumptions and other aspects of the three–entity exchanging problem in hand.

1. The specific exchange problem will consist of an entity P_0 which aims to collect a series of electronic items from entities P_1 and P_2, delivering the appropriate tokens in return. All entities, P_0, P_1 and P_2, are considered to be rational. The following items are involved in the scheme:

   - o_0: Request token issued by P_0 containing a description of the item that P_0 requires from P_1.
   - o_1: Request token issued by P_0 containing a description of the item that P_0 requires from P_2.
   - o_2: Return token issued by P_0 for P_1 in return for o_4.
   - o_3: Return token issued by P_0 for P_2 in return for o_5.
   - o_4: Customized item issued by entity P_1 as specified by P_0 in o_0.
   - o_5: Customized item issued by entity P_2 as specified by P_0 in o_1.

2. None of the collected items in isolation is of any value to entity P_0. In other words, P_0 is interested in collecting all (i.e. o_4 and o_5) or none of these items.

3. No entity is part of any incentive scheme, i.e. the scenario is symmetrically non–incentivized.

4. All messages sent by P_1 and P_2 must be signed with their corresponding private keys and all messages received by these entities must be encrypted with their corresponding public keys.

5. Additionally, each entity assigns a value to each required item:

   - Entity P_0 requires items o_4 and o_5 each worth an arbitrary value `I_VAL`.
   - Entity P_1 requires items o_0 to be able to generate o_4 and it also requires o_2 in return for issuing o_4. Thus, item o_0 will be worth `REQ_VAL` and o_2 will be worth `I_VAL` to P_1.
   - Entity P_2 requires items o_1 to be able to generate o_5 and it also requires o_3 in return for issuing o_5. Thus, item o_1 will be worth `REQ_VAL` and o_3 will be worth `I_VAL` to P_2.

6. Finally, the nature of these items is such that their utilities only become available when the corresponding tokens are delivered in return. Although this restriction seems hard and unrealistic, there are a few real life examples where items are of this nature. For example, P_0 could be an user trying to book a holiday package consisting of accommodation, flights and tickets for several local tourist attractions. User P_0 needs either all or none of the required items and, additionally no item becomes available unless the providers of the required services have received payment.

8.3 3–RES Data Representation

This section will be used to parameterize the information given in Section 8.2 describing the particular exchanging problem. Of course, other type of problem or scenario could have also been formatted rendering a completely different result.

8.3.1 Entities and Items

The following are the main parameters of the exchange problem:

- $v = 3$ represents the number of entities,
- $m = 6$ represents the number of items involved in the exchange and,
- $n = 10$ is the maximum number of messages in the protocol solution.

Matrices $H(t)$, B and R will then be of the following dimensions:

- $H(t)$, $B \in \mathcal{M}_{3\times 6}$ and,

- $R \in \mathcal{M}_{18\times 18}$

Note that, according to expression (7.41), with these parameters the search space has a complexity of $O(2^{63})$.

8.3.2 Initial State Matrix

At a initial state (t=0), P_0 holds request tokens (o_0 and o_1) for the required items (o_4 and o_5). Additionally, P_0 does also hold return tokens o_2 and o_3 at this stage.

Since items o_4 and o_5 must be individually tailored by P_1 and P_2 to satisfy P_0's requirements, a *non–accessible* status is assigned to them until the request tokens are received by the appropriate entities.

Matrix $H(0)$ defines this initial state of possessions:

$$H(0) = \begin{pmatrix} \texttt{ACC} & \texttt{ACC} & \texttt{ACC} & \texttt{ACC} & \texttt{UNKNO} & \texttt{UNKNO} \\ \texttt{UNKNO} & \texttt{UNKNO} & \texttt{UNKNO} & \texttt{UNKNO} & \texttt{NO\_ACC} & \texttt{UNKNO} \\ \texttt{UNKNO} & \texttt{UNKNO} & \texttt{UNKNO} & \texttt{UNKNO} & \texttt{UNKNO} & \texttt{NO\_ACC} \end{pmatrix}$$

8.3.3 Dependency Matrix

Section 8.2 describes a specific scenario and the nature of the e-items involved in such a particular exchange problem. The following list enumerates the dependencies inferred from that description. Further below, Table 8.1 expresses the logical formulae for each one of them.

1. The utility value for item o_4 cannot be made accessible to entity P_0 if entity P_1 has not received item o_2.

2. The utility value for item o_5 cannot be made accessible to entity P_0 if entity P_2 has not received item o_3.

3. If entity P_1 receives item o_0 then entity P_1 can access item o_4 (once the request is received, P_1 can generate item o_4 according to the instructions described in item o_0).

4. If entity P_1 receives item o_2 then entity P_0 can access item o_4.

5. If entity P_2 receives item o_1 then entity P_2 can access item o_5 (once the request is received, P_2 can generate item o_5 according to the instructions described in item o_1).

6. If entity P_2 receives item o_3 then entity P_0 can access item o_5.

$r_{i,j}$	Sign of the relation	Formula
$r_{4,8} = \texttt{NEG\_DR}$	Negative	$(h_{0,4} = \texttt{ACC}) \wedge \big((h_{1,2} = \texttt{NO\_ACC}) \vee (h_{1,2} = \texttt{UNKNO})\big) \Rightarrow h_{0,4} = \texttt{NO\_ACC}$
$r_{5,15} = \texttt{NEG\_DR}$	Negative	$(h_{0,5} = \texttt{ACC}) \wedge \big((h_{2,3} = \texttt{NO\_ACC}) \vee (h_{2,3} = \texttt{UNKNO})\big) \Rightarrow h_{0,5} = \texttt{NO\_ACC}$
$r_{6,10} = \texttt{POS\_DR}$	Positive	$(h_{1,0} = \texttt{ACC}) \wedge (h_{1,4} = \texttt{NO\_ACC}) \Rightarrow h_{1,4} = \texttt{ACC}$
$r_{8,4} = \texttt{POS\_DR}$	Positive	$(h_{1,2} = \texttt{ACC}) \wedge (h_{0,4} = \texttt{NO\_ACC}) \Rightarrow h_{0,4} = \texttt{ACC}$
$r_{13,17} = \texttt{POS\_DR}$	Positive	$(h_{2,1} = \texttt{ACC}) \wedge (h_{2,5} = \texttt{NO\_ACC}) \Rightarrow h_{2,5} = \texttt{ACC}$
$r_{15,5} = \texttt{POS\_DR}$	Positive	$(h_{2,3} = \texttt{ACC}) \wedge (h_{0,5} = \texttt{NO\_ACC}) \Rightarrow h_{0,5} = \texttt{ACC}$

Table 8.1: Logical formula for 3–RES dependency relationships.

Matrix R for the dependency relations described in Section 8.2 will be of this form:

$$r_{i,j}(t) = \begin{cases} \texttt{NEG\_DR} & \text{iff } (i = 4 \wedge j = 8) \vee (i = 5 \wedge j = 15) \\ \texttt{POS\_DR} & \text{iff } (i = 6 \wedge j = 10) \vee (i = 8 \wedge j = 4) \vee \\ & (i = 13 \wedge j = 17) \vee (i = 15 \wedge j = 5) \vee \\ \texttt{NO\_DR} & \text{Otherwise} \end{cases} \tag{8.1}$$

8.3.4 Benefit Matrix

The values that each entity assigns to each item involved in the exchange, together with information about incentive schemes, have been defined when describing the problem. All that information determines the benefit matrix B for a given synthesis process.

For the purpose of exploring the space of 3–RES solutions, we will consider four different scenarios (so four different B matrices), each one of them described below and summarized in Figure 8.1.

Types of scenario considered in the search are:

(A). *There exist three intersecting coalitions amongst participants:* P_0 and P_1, P_1 and P_2 and, P_2 with P_0. The benefit matrix which represents such a scenario is:

$$B = \begin{pmatrix} \texttt{NO\_COST} & \texttt{NO\_COST} & \texttt{COST} & \texttt{COST} & \texttt{I\_VAL} & \texttt{I\_VAL} \\ \texttt{REQ\_VAL} & \texttt{BENEF} & \texttt{I\_VAL} & \texttt{BENEF} & \texttt{COST} & \texttt{BENEF} \\ \texttt{BENEF} & \texttt{REQ\_VAL} & \texttt{BENEF} & \texttt{I\_VAL} & \texttt{BENEF} & \texttt{COST} \end{pmatrix}$$

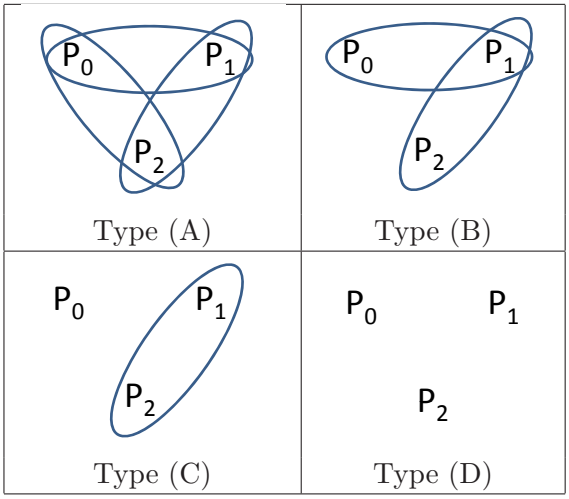


Figure 8.1: Four different scenarios considered in the process of synthesis.

Note that the values of NO_COST, COST and BENEF were defined in Definition 7.16 and also, values REQ_VAL and I_VAL were introduced in Section 8.2.

(B). *There exist two intersecting coalitions:* P_0 with P_1 and P_1 with P_2. The benefit matrix which represents such a scenario is:

$$B = \begin{pmatrix} \texttt{NO\_COST} & \texttt{NO\_COST} & \texttt{COST} & \texttt{COST} & \texttt{I\_VAL} & \texttt{I\_VAL} \\ \texttt{REQ\_VAL} & \texttt{BENEF} & \texttt{I\_VAL} & \texttt{BENEF} & \texttt{COST} & \texttt{BENEF} \\ \texttt{BENEF} & \texttt{REQ\_VAL} & \texttt{BENEF} & \texttt{I\_VAL} & \texttt{COST} & \texttt{COST} \end{pmatrix}$$

(C). *There exists only one coalition:* P_1 with P_2. The benefit matrix which represents such a scenario is:

$$B = \begin{pmatrix} \texttt{NO\_COST} & \texttt{NO\_COST} & \texttt{COST} & \texttt{COST} & \texttt{I\_VAL} & \texttt{I\_VAL} \\ \texttt{REQ\_VAL} & \texttt{BENEF} & \texttt{I\_VAL} & \texttt{BENEF} & \texttt{COST} & \texttt{COST} \\ \texttt{BENEF} & \texttt{REQ\_VAL} & \texttt{BENEF} & \texttt{I\_VAL} & \texttt{COST} & \texttt{COST} \end{pmatrix}$$

(D). *There are no coalitions between the participant entities.* The benefit matrix which represents such a scenario is:

$$B = \begin{pmatrix} \texttt{NO\_COST} & \texttt{NO\_COST} & \texttt{COST} & \texttt{COST} & \texttt{I\_VAL} & \texttt{I\_VAL} \\ \texttt{REQ\_VAL} & \texttt{COST} & \texttt{I\_VAL} & \texttt{COST} & \texttt{COST} & \texttt{COST} \\ \texttt{COST} & \texttt{REQ\_VAL} & \texttt{COST} & \texttt{I\_VAL} & \texttt{COST} & \texttt{COST} \end{pmatrix}$$

Entity	$\hat{b}_i$	$\bar{b}_i$
P_0	12	8
P_1	11	6
P_2	11	6

Type (A)

Entity	$\hat{b}_i$	$\bar{b}_i$
P_0	12	8
P_1	11	6
P_2	11	5

Type (B)

Entity	$\hat{b}_i$	$\bar{b}_i$
P_0	12	8
P_1	11	5
P_2	11	5

Type (C)

Entity	$\hat{b}_i$	$\bar{b}_i$
P_0	12	8
P_1	11	3
P_2	11	3

Type (D)

Figure 8.2: Minimum and maximum expected payoffs for each type of scenario and values `REQ_VAL` $= 2$ and `I_VAL` $= 5$.

8.3.5 Computing Entities' Minimum Requirements

For each entity P_i, values $\hat{b}_i$ and $\bar{b}_i$ (Definitions 7.18 and 7.19) represent the maximum and minimum payoff values respectively, that entity P_i would expect to obtain with the exchange.

Considering each one of the four different scenarios previously described, and the following arbitrary values (`REQ_VAL` $= 2$ and `I_VAL` $= 5$), we obtain the set of values shown in Fig. 8.2.

8.3.6 Computing Fitness

Utility values are calculated as defined in equation (7.20) at each step in the protocol and for every participant entity. In a similar way, the overall protocol fitness value is computed as defined in equation (7.22) at the end of the protocol run.

However, Definition 7.4.1 indirectly forces the fitness function to adopt the following approach: when evaluating a given protocol, the fitness taken will be the maximum utility value obtained along the whole execution. That is, if along a protocol execution the maximum global fitness was reached at a step before the final, then the protocol will only be considered a possible rational solution up to that step. For example, if along a protocol execution a maximum global fitness value was reached at step 4. Even though the protocol might consist of 20 instructions, the protocol could only be considered a possible rational solution up to step 4, and its global fitness value will be the fitness value attained at that stage.

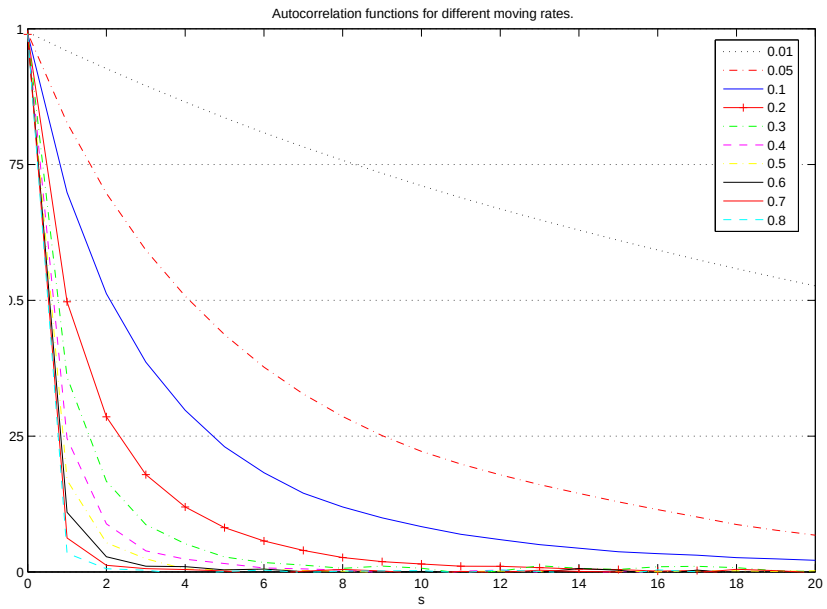


Figure 8.3: Autocorrelation functions of the fitness landscape of different moving rates.

8.4 3–RES Fitness Landscape

Before presenting our final results we will make use of Theory of Landscapes to proof the efficiency of our global formalism and in particular for the synthesis of 3–RES protocols.

8.4.1 Move Operator

The following routine is our candidate neighboring operator:

Random Mutation: A random mutation is a modification of a fixed number (determined by a *moving rate*) of elements in the protocol matrix S^{Π}.

Different *moving rates* will be considered in the experimental work. These are: 1%, 5%, 10%, 20%, 30%, 40%, 50%, 60%, 80%.

8.4.2 Fitness Autocorrelation.

The function used to evaluate autocorrelation of a set of fitness values was defined by equation (6.1). For the purposes of our current work we are presenting the autocorrelation functions obtained when evaluating the different neighboring

operators previously described. In all cases, the autocorrelation is computed by averaging 200 random walks of length 2000 individuals each. Figure 8.4.2 shows the plots of each of these autocorrelation functions. See Section 6.3 for a more in–depth discussion on autocorrelation curves.

Moving rates of 10% and 20% seem to offer higher guarantee of success for any given guided search. For lower rates, the autocorrelation functions represent landscapes too flat. By contrast, autocorrelation values for moving rates higher than 20% start representing rough fitness landscapes, where any guided search could easily degenerate into a random algorithm.

Finally note that although the autocorrelation functions shown in the graph correspond to an scenario type (A) (scenario with three intersecting coalitions), the fitness function is the same in every type of exchange, so it projects the same kind of landscape. The same results are therefore obtained in every other type of scenario.

8.5 Search Technique and Parametrization

Simulated Annealing (SA) shown in Figure 6.1 will be used as search technique. The basic algorithm has been slightly modified to stop when the first rational–exchange protocol which satisfies the requirements is found. (This can be done by previously computing the minimum required global fitness).

As described in Section 8.4, given a candidate solution (specified by a protocol matrix S^{Π}), a neighbor $S^{\Pi'}$ is obtained by randomly modifying a percentage (moving rate) of its elements.

The acceptance criterion in SA is given by:

$$S^{\Pi'} \text{ is accepted if } U(S^{\Pi'}) - U(S^{\Pi}) > T_i \ln u \tag{8.2}$$

where:

- S^{Π} and $S^{\Pi'}$ are the current and mutated solutions respectively,
- $U(\cdot)$ is the global protocol fitness function as defined in equation (7.22),
- T_i is the current temperature and,
- u is a random number uniformly generated in $[0, 1]$.

At each cycle, the temperature is geometrically decreased by:

$$T_{i+1} = \alpha T_i \tag{8.3}$$

$0 < \alpha < 1$ being the cooling factor.

After m cycles the temperature is $T_m = \alpha^m T_0$, where T_0 is the initial temperature. For T_m to be very close to 0 (say $\epsilon = 10^{-6}$) after m cycles, a cooling rate of:

$$\alpha = \left(\frac{\epsilon}{T_0}\right)^{\frac{1}{m}} \tag{8.4}$$

is needed.

The experimental work has been carried out adjusting those SA parameters (T_0, m and α) according to the definition of two different profiles:

I. In **profile (I)**, an initial temperature of $T_0 = 1.44$ is decreased by a cooling factor of 0.9705 satisfying the following properties:

   (a) In the first cycle, the probability of accepting a bad move which decreases the global protocol fitness value by just one unit is approximately 0.5.

   (b) By half the total number of cycles, the probability of accepting a bad move which decreases the global protocol fitness value by more than one unit is almost zero. So from exactly half the total number of cycles onwards, the search behaves as a pure Hill Climbing (HC) algorithm.

II. In **profile (II)**, an initial temperature of $T_0 = 1.44$ is decreased by a cooling factor of 0.9419 satisfying the following properties:

   (a) In the first cycle, the probability of accepting a bad move which decreases the global protocol fitness value by just one unit is approximately 0.5.

   (b) By one quarter of the total number of cycles, the probability of accepting a bad move which decreases the global protocol fitness value by more than one unit is almost zero. So from exactly one forth of the total number of cycles onwards, the search behaves as a pure Hill Climbing (HC) algorithm.

Extensive experimentation has demonstrated that around 200 cycles with 1000 moves in the SA inner loop are sufficient to reach solutions in reasonable time. Table 8.2 summarizes all experimental parameters.

8.6 Results

Experimental work has been carried out for each one of the scenarios described in Section 8.3.4 and Figure 8.1.

SA Parameter	Value
Number of cycles	210
Number attempts	1000
Initial temperature	1.44
PRNG	Mersenne Twister

SA Parameter	Profile (I)	Profile (II)
Cooling rate	0.9705	0.9419

(i)

3–RES Parameter	Value
No. parties	3
Max. No. of messages per protocol	10
Max. No. of items per message	6
Total No. of items to exchange	6

(ii)

Table 8.2: (i) General SA parameters. (ii) 3–RES parameters.

8.6.1 With Three Intersecting Coalitions and No Incentives

Figure 8.4(a) shows the results obtained for the two SA profiles and different moving rates (column MR), in an scenario type (A). Additionally, both SA profiles (I and II) are compared with the results obtained when applying a classic Hill Climbing algorithm (HC). The success rate (column SR) represents the percentage of executions attaining a feasible rational protocol over 500 trials. The average number of protocols evaluated in each of those 500 trials is indicated in column Avg.NPE.

In both SA profiles, the best results are obtained with a moving rate of 0.1, attaining more than 99% of success (i.e. almost every execution produces a valid solution) by evaluating approximately only $24,500$ protocols. These numbers imply synthesizing a protocol for this scenario in less than 1 minute in a common laptop.

Success rates for slightly lower or higher mutation rates are similar, though the number of total candidates evaluated before reaching a solution grows considerably, thus resulting in a more inefficient search. As expected, higher mutation rates transform the search in an almost random procedure with fewer chances to succeed.

Fig. 8.4(b) serves to compare the data shown in Fig. 8.4(a). An efficiency parameter considering the ratio between success rate and number of protocols evaluated has been defined to measure and compare the three different search modes. The curves show how for each search mode the efficiency decreases as the moving rate increases and how applying simulated annealing profile I renders better results

than any other approach.

Finally, further comparatives are shown in Fig. 8.4(c) where a random search is applied to resolve the same exchange problem. The table shows how for a random process evaluating ten times more protocols than in a guided search, the rate of success is still below 25%.

8.6.2 With Two Intersecting Coalitions and No Incentives

Figure 8.5(a) shows the results obtained for the two SA profiles and HC, for different moving rates (column MR) and in an scenario type (B). Also as previously, the success rate (column SR) represents the percentage of executions attaining a feasible rational protocol over 500 trials. The average number of protocols evaluated in each trial is indicated in column Avg. NPE.

Again, shown in Fig. 8.5(b), a moving rate of 0.1 seems to perform better than any other and, although success rates do not reach such high values as in previous experiments, still the tool is able to synthesize a rational scheme in less than 1 minute in almost all instances.

Finally, further comparatives are shown in Fig. 8.4(c) where a random search is applied to resolve the same problem and the same type of scenario.

8.6.3 With One Coalition and No Incentives

Figure 8.6(a) shows the results obtained for the two SA profiles and HC in an scenario type (C). Likewise, Fig.8.6(b) shows the results on a graph with three curves. Again in this case, SA out performs HC (0.1% is the best moving rate) and any guided search is overwhelmingly more efficient than a classic random approach (see Fig. 8.6(c)).

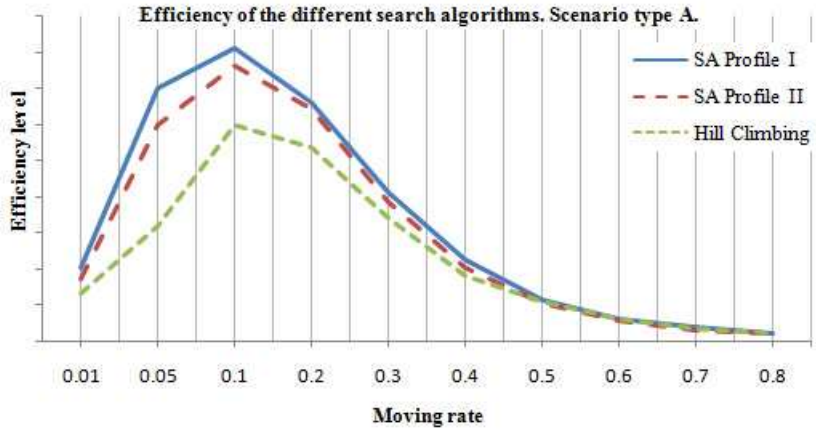
8.6.4 With No Coalitions and No Incentives

Surprisingly, no schemes are found in this type of scenario. However, intuitively one can argue that if entities are neither incentivized to exchange their items, nor motivated to help others in the exchange, then how can rationality act to enforce participants to follow the steps of any exchange scheme?

Although a formal proof of this result is out of the scope of this experimental work, a sketch can be easily outlined if we considered the protocol game representing any exchange protocol in this type of scenario. In such a game, for at least one of the players (usually the first to play) its final payoff will be less than the utility they hold at the start of the game. Under these circumstances, such a player will not enter the game. Consequently, by applying backward induction, we can conclude

Results with Three Coalitions and No Incentives						
	HC		SA (Profile I)		SA (Profile II)	
MR	SR	Avg. NPE	SR	Avg. NPE	SR	Avg. NPE
0.01	67.4%	100,640	81.2%	80,294	76.0%	87,952
0.05	89.8%	56,678	98.6%	28,197	97.4%	32,571
0.1	97.6%	32,641	99.6%	24,531	99.2%	26,019
0.2	97.6%	36,412	100.0%	30,305	98.6%	30,737
0.3	92.8%	54,678	98.2%	47,542	95.4%	49,198
0.4	80.4%	87,988	91.2%	81,342	84.8%	83,644
0.5	66.4%	119,520	71.0%	125,366	67.4%	126,732
0.6	47.4%	151,292	48.8%	155,280	45.4%	157,794
0.7	31.8%	172,398	33.2%	175,532	28.4%	180,882
0.8	20.6%	187,248	20.4%	188,870	22.0%	187,372

(a)



(b)

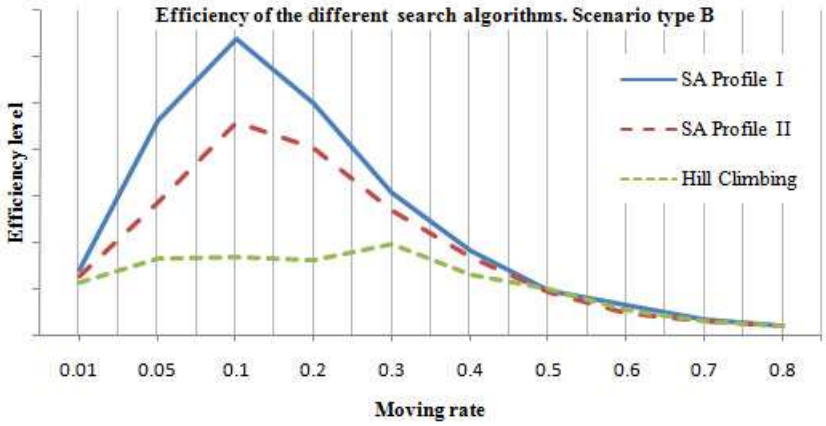
Random Search	
SR	Avg. NPE
2.0%	19,823
6.4%	96,975
12.4%	186,519
23.0%	355,611

(c)

Figure 8.4: Results on scenario type *A*. Success rate of success (SR) and average number of protocols evaluated (Avg. NPE) per trial. Results estimated over 500 trials comparing three different search methods: SA profile I, SA profile II and Hill Climbing.

Results with Two Coalitions and No Incentives						
	HC		SA (Profile I)		SA (Profile II)	
MR	SR	Avg. NPE	SR	Avg. NPE	SR	Avg. NPE
0.01	62.6%	110,272	71.4%	100,990	67.6%	105,360
0.05	66.8%	81,114	95.0%	41,132	81.6%	57,010
0.1	67.2%	80,226	97.6%	30,651	90.0%	39,182
0.2	69.2%	85,208	98.0%	39,296	90.4%	44,838
0.3	77.4%	79,106	94.0%	61,246	87.0%	64,600
0.4	68.2%	125,022	85.0%	94,202	79.2%	92,940
0.5	62.8%	125,022	64.6%	134,358	62.2%	131,916
0.6	44.0%	156,738	50.0%	157,916	41.8%	165,162
0.7	29.0%	179,750	29.2%	179,336	28.2%	178,968
0.8	19.2%	187,082	18.4%	193,296	20.4%	189,398

(a)



(b)

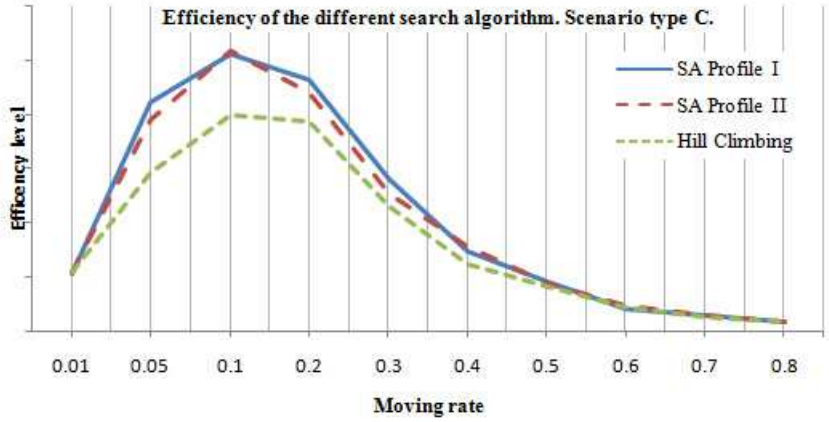
Random Search	
SR	Avg. NPE
22%	352,950

(c)

Figure 8.5: Results on scenario type B. Success rate (SR) and average number of protocols evaluated (Avg. NPE) per trial. Results estimated over 500 trials comparing three different search methods: SA profile I, SA profile II and Hill Climbing.

Results with One Coalition and No Incentives						
	HC		SA (Profile I)		SA (Profile II)	
MR	SR	Avg. NPE	SR	Avg. NPE	SR	Avg. NPE
0.01	64.2%	117,714	65.8%	121,808	64.4%	119,580
0.05	90.4%	61,402	97.6%	46,198	95.0%	48,604
0.1	97.4%	48,708	99.0%	38,694	97.4%	37,606
0.2	96.8%	49,748	99.2%	42,676	97.8%	44,346
0.3	87.8%	76,196	95.2%	68,060	89.0%	70,186
0.4	70.0%	111,474	79.8%	108,774	78.2%	99,718
0.5	56.2%	137,452	64.0%	138,704	60.2%	135,204
0.6	37.2%	165,374	36.8%	172,078	39.0%	167,848
0.7	23.6%	183,298	27.2%	180,896	25.0%	183,390
0.8	16.8%	192,470	17.4%	192,674	16.0%	194,440

(a)



(b)

Random Search	
SR	Avg. NPE
20.2%	358,869

(b)

Figure 8.6: Results on scenario type C. Success rate (SR) and average number of protocols evaluated (Avg. NPE) per trial. Results estimated over 500 trials comparing three different search methods: SA profile I, SA profile II and Hill Climbing.

that the Nash equilibrium of the game will be represented by the strategy profile in which every player plays action "quit".

8.6.5 Discussion

All in all, the best rates of success are systematically achieved by SA. Even though a simple HC technique attains very good solutions too, the average number of protocols evaluated per trial serves as an experimental proof of efficiency, in favor of a more sophisticated heuristic based on SA. Furthermore, our preliminary experimentation indicates that this is certainly the case in more complex exchange scenarios in which the number of entities and the number of items are considerable larger.

Finally, as for a pure random search, the numbers are several orders of magnitude below the results obtained by any of the other two techniques.

8.7 Automatically Synthesized 3–RES protocols

In this section we will give details of a particular automatically synthesized protocol and furthermore, we will sketch a formal rationality proof of the scheme. Describing in detail this particular protocol will serve to:

1. Illustrate our methodology (further three–entity solutions will be presented in subsequent Section 8.7.2) and,

2. To present a whole family of multi-party rational exchange security protocols based on this particular solution (Section 8.8).

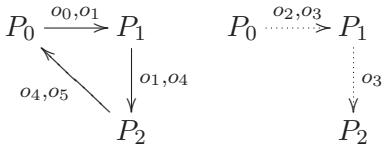
8.7.1 A Two Phase 3–RES Protocol

The following protocol is synthesized by the heuristic technique in an scenario type (A) previously described in Figure 8.1. The steps dictated by the scheme are:

(1) Entity P_0 sends entity P_1, a message including o_0 and o_1, descriptions of the required items.

(2) Entity P_1 produces, according to the appropriate description, a customized o_4 destined to P_0. Entity P_1 sends P_2 a message containing o_4 and the description token o_1.

(3) Entity P_2 produces, according to the appropriate description, a customized o_5 destined to P_0. Entity P_2 sends P_0 items o_4 and o_5.



(a)

(1)	$P_0 \rightarrow P_1$:	$\{o_0, o_1\}_{K_{P_1}}$
(2)	$P_1 \rightarrow P_2$:	$\{\{o_1, o_4\}_{K_{P_1}^{-1}}\}_{K_{P_2}}$
(3)	$P_2 \rightarrow P_0$:	$\{o_4, o_5\}_{K_{P_2}^{-1}}$
(4)	$P_0 \rightarrow P_1$:	$\{o_2, o_3\}_{K_{P_1}}$
(5)	$P_1 \rightarrow P_2$:	$\{\{o_3\}_{K_{P_1}^{-1}}\}_{K_{P_2}}$

(b)

Figure 8.7: A synthesized 3–entity rational–exchange protocol. The protocol runs in the two phases illustrated at the top (a). Further security refinements are applied to the scheme as shown in (b).

(4) Participant P_0 sends P_1 a message including o_2 and o_3, the return-tokens for the items received.

(5) P_1 receives a message with two return-tokens. It takes o_2 and sends o_3 to entity P_2.

Figure 8.7(a) shows an example of a synthesized protocol for the problem described in Section 8.2.

As established in the initial assumptions of the problem, all messages sent by entities P_1 and P_2 must be signed with the corresponding private keys ($K_{P_1}^{-1}$ and $K_{P_2}^{-1}$), as well as all messages being received by these entities must be encrypted with the appropriate public keys (K_{P_1} and K_{P_2}). Applying these further security refinements to each message of the original scheme results in the protocol shown in Figure 8.7(b).

Rationality

Rationality of the scheme previously described can be directly inferred by the methodology used in the synthesis of the protocol. Informally, the following list describes those aspects of the scheme which ensure rationality and feasibility of the solution:

- **From entity's P_0 point of view.** As stated in the initial assumptions, items o_4 and o_5 are of no use to entity P_0 until the corresponding return items o_2 and

o_3 have reached entities P_1 and P_2 respectively. To this regard, since entity P_0 requires either all or none of these items, entity P_0 is *rationally* forced to send message 4.

- **From entity's P_1 point of view.** As previously mentioned, entity P_0 requires either all or none of these items. Again, this assumption forces entity P_1 to send P_2 messages 2 and 5.

- **From entity's P_2 point of view.** Similar rationale will force entity P_2 to send message 3 to P_0.

Therefore, no entity would unilaterally deviate from the 3–RES protocol as they could not obtain better utility value in doing so. The scheme is then a rational solution.

8.7.2 Other Automatically Synthesized Solutions

We will now present other three entity rational exchange protocols, also automatically synthesized using the described methodology and, which will serve to resolve the same exchange problem described in Section 8.2. Formal analysis of these schemes will follow similar rationale as for the protocol in the previous section.

Other solutions for scenario type (A)

Different solutions for scenario type (A) are listed in Table 8.3.

Scheme (a) in Table 8.3 is completely equivalent to the solution described in Fig. 8.7 but in this case, it is entity P_2 who first receives the tokens from P_0.

Scheme (b) represents a five message protocol in which entity P_0 decides to send P_2's payment before having received the required token and using P_1 as intermediary. By contrast, P_1's payment is sent after the required token is received and without intermediaries.

Scheme (c) is an interesting solution in which only the first message of the exchange contains more than one item, the other six messages are composed of only one item.

Finally, schemes (d) and (e) represent variants of the protocol in (c).

Other solutions scenario type (B)

Other solutions for scenario type (B) are listed in Table 8.7.2.

In this type of scenario the solution described in Fig. 8.7 does also provide a rational exchange. Furthermore, other schemes of different characteristics and length are presented in Table 8.7.2 (a) to (f). All were automatically designed to

(1)	$P_0 \to P_2:$	$\{o_0, o_1\}$
(2)	$P_2 \to P_1:$	$\{o_0, o_5\}$
(3)	$P_1 \to P_0:$	$\{o_4, o_5\}$
(4)	$P_0 \to P_2:$	$\{o_2, o_3\}$
(5)	$P_2 \to P_1:$	$\{o_2\}$

(a)

(1)	$P_0 \to P_1:$	$\{o_0, o_1, o_3\}$
(2)	$P_1 \to P_2:$	$\{o_1, o_4\}$
(3)	$P_2 \to P_0:$	$\{o_4, o_5\}$
(4)	$P_1 \to P_2:$	$\{o_3\}$
(5)	$P_0 \to P_1:$	$\{o_2\}$

(b)

(1)	$P_0 \to P_2:$	$\{o_0, o_1, o_2\}$
(2)	$P_2 \to P_1:$	$\{o_0\}$
(3)	$P_2 \to P_0:$	$\{o_5\}$
(4)	$P_1 \to P_0:$	$\{o_4\}$
(5)	$P_2 \to P_1:$	$\{o_2\}$
(6)	$P_0 \to P_2:$	$\{o_3\}$

(c)

(1)	$P_0 \to P_1:$	$\{o_1\}$
(2)	$P_0 \to P_2:$	$\{o_0\}$
(3)	$P_2 \to P_1:$	$\{o_0\}$
(4)	$P_1 \to P_2:$	$\{o_1, o_4\}$
(5)	$P_2 \to P_0:$	$\{o_4, o_5\}$
(6)	$P_0 \to P_1:$	$\{o_2\}$
(7)	$P_0 \to P_2:$	$\{o_3\}$

(d)

(1)	$P_0 \to P_1:$	$\{o_0, o_1\}$
(2)	$P_1 \to P_2:$	$\{o_1, o_4\}$
(3)	$P_2 \to P_0:$	$\{o_4\}$
(4)	$P_0 \to P_2:$	$\{o_2\}$
(5)	$P_2 \to P_1:$	$\{o_2\}$
(6)	$P_2 \to P_0:$	$\{o_5\}$
(7)	$P_0 \to P_2:$	$\{o_3\}$

(e)

Table 8.3: Synthesized 3-entity rational protocols for scenario type (A).

resolve, in a rational way, the exchange problem presented in Section 8.2 in an scenario with two coalitions and no incentives.

Finally, it is noticed that no protocol ends with P_2 sending P_0 a message. As P_2 is not an allied of P_0, P_2 has no incentive to forward the tokens that P_0 requires. However, P_1 and P_2 serve as intermediaries for P_0 to forward each other tokens, as they are both in coalition.

Other solutions scenario type (C)

In a similar way, the solution described in Fig. 8.7 is also synthesized in this type of scenario. Furthermore, other solutions are listed in Table 8.7.2 (a) to (c). In this case, no rational solution ends with entities P_1 or P_2 sending P_0 the corresponding tokens.

8.8 v–RES Protocol Family

The protocol previously described in Figures 8.7(a) and (b) can be easily extended to any given $v \geqslant 3$ parties, defining a new family of v–RES protocols.

In general terms, v–RES is a family of multi–party rational–exchange security protocols by which an entity P_0 aims to collect a series of items from other participant entities $(P_1, \ldots, P_{v-1})$, delivering the appropriate tokens in return and considering that no item in isolation would be of any use to P_0, as this needs all or none of the required items.

Note that no restrictions such as simultaneous broadcasting or synchronizing mechanisms are imposed on the system, so we believe there are many other problems (secret sharing, multiparty function computation, multiple access control, etc.) for which v–RES provides a framework for a rational solution to the problem.

As v–RES protocols will resolve a similar problem to the one described in Section 8.2, the next sections will serve to present a more general version of the problem and to describe in detail each message of the scheme. Finally, a proof of rationality will be provided using the formalism based on Game Theory from Part I of this thesis.

8.8.1 v–RES Initial Assumptions and Formal Notation

Several aspects of the exchange problem extended to any given v parties,are informally described in what follows:

- *Electronic items exchanged:* The nature of these items must be such that their utility only become available when the corresponding token is delivered

(1)	$P_0 \to P_2:$	$\{o_0, o_1\}$
(2)	$P_2 \to P_1:$	$\{o_0, o_5\}$
(3)	$P_1 \to P_0:$	$\{o_4, o_5\}$
(4)	$P_0 \to P_2:$	$\{o_2, o_3\}$
(5)	$P_2 \to P_1:$	$\{o_2\}$

(a)

(1)	$P_0 \to P_1:$	$\{o_0, o_1\}$
(2)	$P_1 \to P_2:$	$\{o_1, o_4\}$
(3)	$P_2 \to P_0:$	$\{o_4, o_5\}$
(4)	$P_0 \to P_2:$	$\{o_2, o_3\}$
(5)	$P_2 \to P_1:$	$\{o_2\}$

(b)

(1)	$P_0 \to P_1:$	$\{o_0, o_1\}$
(2)	$P_1 \to P_0:$	$\{o_4\}$
(3)	$P_1 \to P_2:$	$\{o_1\}$
(4)	$P_0 \to P_1:$	$\{o_2\}$
(5)	$P_2 \to P_0:$	$\{o_5\}$
(6)	$P_0 \to P_2:$	$\{o_3\}$

(c)

(1)	$P_0 \to P_1:$	$\{o_0\}$
(2)	$P_0 \to P_2:$	$\{o_1, o_2\}$
(3)	$P_1 \to P_0:$	$\{o_4\}$
(4)	$P_2 \to P_0:$	$\{o_5\}$
(5)	$P_2 \to P_1:$	$\{o_2\}$
(6)	$P_0 \to P_2:$	$\{o_3\}$

(d)

(1)	$P_0 \to P_1:$	$\{o_0, o_1, o_3\}$
(2)	$P_1 \to P_2:$	$\{o_1\}$
(3)	$P_2 \to P_0:$	$\{o_4, o_5\}$
(4)	$P_1 \to P_2:$	$\{o_3\}$
(5)	$P_0 \to P_1:$	$\{o_2\}$

(e)

(1)	$P_0 \to P_2:$	$\{o_0, o_1\}$
(2)	$P_2 \to P_1:$	$\{o_0, o_5\}$
(3)	$P_1 \to P_0:$	$\{o_4, o_5\}$
(4)	$P_0 \to P_1:$	$\{o_3\}$
(5)	$P_1 \to P_2:$	$\{o_3\}$
(6)	$P_0 \to P_1:$	$\{o_2\}$

(f)

Table 8.4: Synthesized 3-entity rational protocols for scenario type (B).

(1)	$P_0 \rightarrow P_1:$	$\{o_3\}$
(2)	$P_0 \rightarrow P_2:$	$\{o_0, o_1, o_2\}$
(3)	$P_2 \rightarrow P_1:$	$\{o_0, o_5\}$
(4)	$P_2 \rightarrow P_1:$	$\{o_2\}$
(5)	$P_1 \rightarrow P_2:$	$\{o_3\}$

(a)

(1)	$P_0 \rightarrow P_1:$	$\{o_0, o_1, o_3\}$
(2)	$P_1 \rightarrow P_2:$	$\{o_1, o_4\}$
(3)	$P_2 \rightarrow P_0:$	$\{o_4, o_5\}$
(4)	$P_0 \rightarrow P_2:$	$\{o_2\}$
(5)	$P_1 \rightarrow P_2:$	$\{o_3\}$
(6)	$P_2 \rightarrow P_1:$	$\{o_2\}$

(b)

(1)	$P_0 \rightarrow P_2:$	$\{o_0, o_1\}$
(2)	$P_2 \rightarrow P_0:$	$\{o_5\}$
(3)	$P_0 \rightarrow P_1:$	$\{o_3\}$
(4)	$P_2 \rightarrow P_1:$	$\{o_0\}$
(5)	$P_1 \rightarrow P_2:$	$\{o_3\}$
(6)	$P_1 \rightarrow P_0:$	$\{o_4\}$
(7)	$P_0 \rightarrow P_1:$	$\{o_2\}$

(c)

Table 8.5: Synthesized 3-entity rational protocols for scenario type (C).

in return. Additionally, no item in isolation is of any value to entity P_0. In other words, P_0 is interested in collecting all or none of these items.

- *Providers of e-items:* Participant entities P_i providing with the electronic items, must be part of a visible and recognizable PKI (Public Key Infrastructure). No other trusted or semi-trusted parties are involved in the scheme. Note that this is not a restriction on entity P_0 who can maintain anonymous his/her real identity.
- *Coalitions and Incentives:* As we have seen in previous section, either by incentives (the potential to gain further business, reputation factors, etc.) or by coalitions (helping others to achieve better payoff), v–RES participants must be rewarded when behaving according the protocol description.

We will consider the following notation to represent the previously described v–party scenario:

- Let $P = \{P_0, \ldots, P_{v-1}\}$ be the set of participant entities.
- Let $D = \{desc\_item_i\}_{i=1,\ldots,v-1}$ represent a list of *description-tokens* which user P_0 composes, with details on each of the requested items from each entity P_i, $i = 1, \ldots, v-1$.
- We assume that in each case, entity P_i will be able to produce a token $item_i$, tailored to satisfy user P_0's request, described by $desc\_item_i$.

- Let $Y = \{pay_i\}_{i=1,\ldots,v-1}$ be a set of *payment-tokens* produced by user P_0. Each pay_i is destined to entity P_i in return for $item_i$. Each entity P_i must specify how token pay_i should be constructed (it could be of general knowledge or $item_i$ could include this information for each particular instance) to ensure immediate or future payment.

8.8.2 v–RES Two–Phase Protocol

The protocol is executed in two main phases informally described in the next paragraphs.

- Phase I:
  - Customer P_0 sends entity P_1, a message including set D with descriptions for all the required items.
  - Entity P_1 produces, according to token $desc\_item_1$, a customized $item_1$ destined to P_0. It also deletes $desc\_item_1$ from set D and it establishes who would be the next entity to satisfy the requirement described by $desc\_item_2$.
  - Entity P_1 sends P_2 a message containing $item_1$ and the set D with the remaining description-tokens.
  - The process is repeated from any P_i to P_{i+1} until all requirements are satisfied and set D is empty.
  - Finally, the last entity P_{v-1} sends P_0 all items $\{item_i\}_{1,\ldots,v-1}$ completing the first phase of the exchange.

- Phase II:
  - User P_0 produces $n-1$ payment-tokens, one for each participant entity. P_0 sends entity P_1 the set of payments P.
  - When a participant P_i, $i = 1 \ldots v-2$, receives a message with a set of payments P, it takes the appropriate token, deletes it from the list and forwards the message to the next entity.

The topology to represent a v–RES is a ring, although logically the scheme resolves a more complex interaction matrix.

Figure Fig. 8.8 depicts the dynamics of a general v–RES protocol.

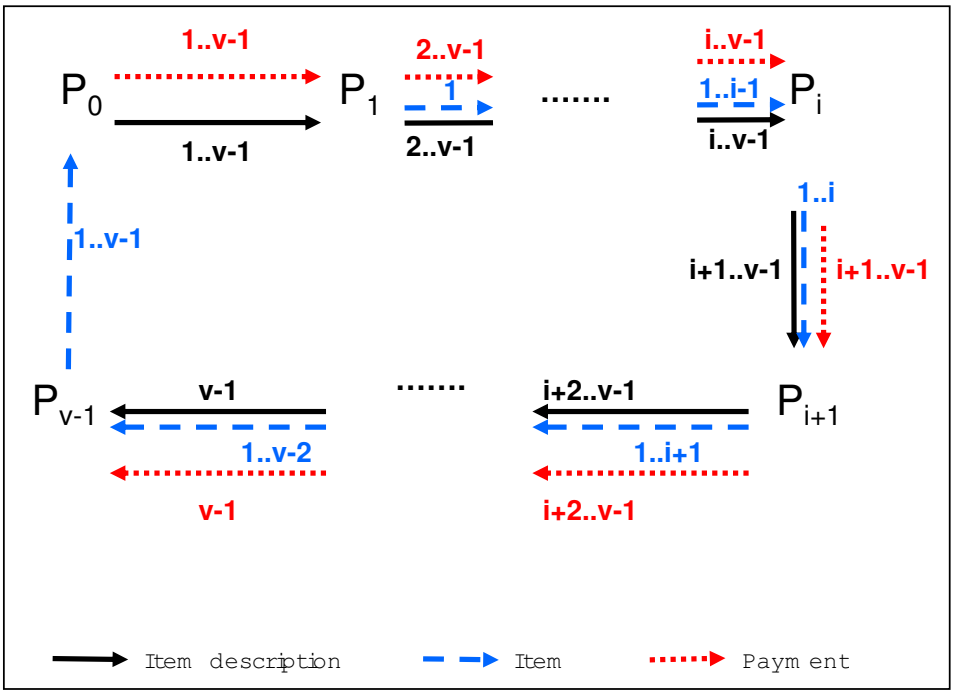


Figure 8.8: Sketch of the v–RES protocol. Different arrows represent different message content: bold lines represent the description of the requested items, dashed lines represent tailored $item_i$ destined to P_0 and, dotted lines represent payment messages.

8.8.3 Detailed Message Content

What follows is a detailed description of every message exchanged using the aforementioned notation.

- Phase I of the v–RES protocol:

  In phase I, entity P_0 issues a list D with descriptions for all required items. It also creates a *fresh* identifier id to uniquely reference the exchange. Finally, P_0 locates the entity she trusts the most, and initiates the protocol by sending the first message, encrypted with entity P_1's public key K_{P_1}.

  At each step in this phase, an entity P_i receives an encrypted message containing a list of previous participants, the list of items and signatures already collected and the set of pending requests D. P_i then composes the corresponding $item_i$ according to $desc\_item_i$, signs the item, removes $desc\_item_i$ from set D and sends message m_i^I, encrypted with the next entity's public key. At the end of phase I, the last entity P_{v-1} sends P_0 a single signed message, including all requested items. The following Figure 8.9 details

$$
\begin{array}{lll}
P_0 \rightarrow P_1: & m_0^I = & \{id, P_0, P_1, D\}_{K_{P_1}} \\
P_i \rightarrow P_{i+1}: & m_i^I = & \{id, P_0, P_1, \ldots, P_i, item_1, \ldots, item_i, D, \sigma_1, \ldots, \sigma_i\}_{K_{P_{i+1}}} \\
 & \text{with} & D = D - \{desc\_item_j\},\ j = 1, \ldots, i \\
P_{v-1} \rightarrow P_0: & m_{v-1}^I = & \{id, P_0, P_1, \ldots, P_{v-1}, item_1, \ldots, item_{v-1}, \sigma_1, \ldots, \sigma_{v-1}\}_{K_{P_{v-1}}^{-1}}
\end{array}
$$

where $\sigma_k = sig(\{id, P_0, P_k, item_k\}, K_{P_k}^{-1}),\ k = 1, \ldots, v-1$

Figure 8.9: Message content. Phase I of the v–RES protocol.

$$
\begin{array}{lll}
P_0 \rightarrow P_1: & m_0^{II} = & \{id, P_0, E, P\}_{K_{P_1}} \\
P_{i-1} \rightarrow P_i: & m_{i-1}^{II} = & \{id, P_0, E, P\}_{K_{P_i}} \\
 & \text{with } E = E - P_{i-1}\ ,\ P = P - \{pay_{i-1}\},\ i = 2, \ldots, v-1 &
\end{array}
$$

Figure 8.10: Message content. Phase II of the v–RES protocol.

message content for phase I of the scheme.

- Phase II of the v–RES protocol:

  In phase II, entity P_0 sends a list P with all payment-tokens and a list E of participant entities, so each participant entity P_i receives a payment-token in return for $item_i$.

  In particular, at each step in phase II, an entity P_i receives an encrypted message containing a list of P of payment–tokens. P_i must collect the corresponding payment and forward the rest of the tokens to the next entity in the list.

  Figure 8.10 details the message content for phase II of the scheme.

8.8.4 v–RES Protocol Game

In this section we will apply the formalism described in Chapter 3 to the formal analysis of the family of v–RES protocols. As stated in the formal model, the initial description of a rational–exchange protocol can be used to construct the corresponding *protocol game* (2.3.1.) The following game can be derived from any v party protocol instance of the family of v–RES protocols.

Definition 8.8.1 (v–RES protocol game)**.** *Let $G_{v--RES} = \{P, S, \overrightarrow{u}\}$ be the protocol game derived from the v–RES protocol description in Section 8.8, where:*

P = $\{P_0, \ldots, P_{v-1}\}$, *a set of participant entities. Each* P_i, $i \in \{1, \ldots, v-1\}$ *with a pair of keys* $(K_{P_i}, K_{P_i}^{-1})$.

S = $S_0 \times S_{v-1}$, *a set of strategy profiles, where:*

- $S_0 = \{(send\_m_0^I, send\_m_0^{II}), (send\_m_0^I, quit^{II})\}$ *is the set of tuples representing all possible strategies for entity* P_0.

- $S_i = \{(send\_m_i^I, send\_m_i^{II}), (send\_m_i^I, quit^{II}), (quit^I, send\_m_i^{II}), (quit^I, quit^{II})\}$, *is the set of tuples representing all possible strategies for entities* P_i, $\forall i \in \{1, \ldots, v-2\}$.

- $S_{v-1} = \{send\_m_i^I, quit^I\}$ *is the set of actions representing all possible strategies for entity* P_{v-1}.

$\overrightarrow{u}$ = $(u_0, \ldots, u_{v-1})$ *is a vector of* v *utility functions with domain over the set* S *of strategy profiles and range over* $\mathbb{R}$.

Note that, superscripts I and II denote phases I and II of the protocol respectively. Also, $send\_m_i^j$ represents action *send message* m_i *at phase* j *of the protocol game* and $quit^j$ denotes action *quit the protocol game at phase* j.

Furthermore, the first component of every possible strategy vector $s_i \in S_i$ $\forall i \in \{0, \ldots, v-1\}$, refers to action taken at phase I of the protocol while component two of any strategy tuple is referring to action taken at phase II in the protocol game.

Finally, any given strategy profile $s = (s_0, \cdots, s_{v-1}) \in S$ may be represented by the tuple (s_i, s_{-i}) in the protocol game.

Definition 8.8.2. *The utility function* $u_0(\cdot)$, *for user* P_0 *in* G_{v--RES} *is defined as follows:*

$\forall s \in S$ *where* $s = (s_0, \cdots, s_{v-1})$*:*

$$u_0(s) = \begin{cases} 0 & \textit{if} \quad \big[s_0 = (send\_m_0^I, quit^{II})\big] \vee \\ & \quad \big[\exists k \in \{1, \ldots, v-1\} : s_k = (quit^I, \cdot)\big] \\ -\sum_{j=1}^{i} pay_j & \textit{if} \quad \big[s_0 = (send\_m_0^I, send\_m_0^{II})\big] \wedge \\ & \quad \big[s_{v-1} = (send\_m_{v-1}^I)\big] \wedge \\ & \quad \big[\exists i \in \{1, \ldots, v-2\} : \big[[s_k = (send\_m_i^I, send\_m_i^{II}) \\ & \quad \forall k < i] \wedge [s_i = (send\_m_i^I, quit^{II})]\big]\big] \\ 1 & \textit{if} \quad \big[s_0 = (send\_m_0^I, send\_m_0^{II})\big] \wedge \\ & \quad \big[s_i = (send\_m_i^I, send\_m_i^{II}), \forall i \in \{1, \ldots, v-2\}\big] \wedge \\ & \quad \big[s_{v-1} = (send\_m_{v-1}^I)\big] \end{cases}$$

Note that value 1 is taken arbitrarily as the first positive integer greater than zero.

Next, $u_i : S \to \mathbb{R}$ represents the utility function for entity P_i, with $i \in \{1, \ldots, v-2\}$.

Note that, as mentioned before, either by incentives (the potential to gain further business, reputation factors, etc.) or by coalitions (helping others to achieve better payoff), entities must be rewarded when behaving according the protocol description. In this case, we have adopted what we consider to be the most realistic approach, in which entities gain further potential businesses every time they successfully complete a protocol run. This is represented by a factor $0 < \delta_i \leqslant 1$ defined as the percentage of future payments that entities obtain every time a protocol run is successfully completed. The following definition formalizes such a concept.

Definition 8.8.3. *We define,* $\forall s = (s_0, \ldots, s_{v-1}) \in S$ *and* $\forall i \in \{1, \ldots, v-2\}$ *the following utility function:*

$$u_i(s) = \begin{cases} 0 & \text{if } \big[s_0 = (send\_m_0^I, quit^{II})\big] \vee \\ & \quad \big[\exists k \in \{1, \ldots, v-1\} : s_k = (quit^I, \cdot)\big] \vee \\ & \quad \big[\exists k < i : s_k = (send\_m_k^I, quit^{II})\big] \\ pay_i & \text{if } \big[s_0 = (send\_m_0^I, send\_m_0^{II})\big] \wedge \\ & \quad \big[s_{v-1} = (send\_m_n^I)\big] \wedge \\ & \quad \big[s_k = (send\_m_k^I, send\_m_k^{II}) \forall k < i\big] \wedge \\ & \quad \big[s_i = (send\_m_i^I, quit^{II})\big] \\ (\delta_i + 1) * pay_i & \text{if } \big[s_0 = (send\_m_0^I, send\_m_0^{II})\big] \wedge \\ & \quad \big[s_{v-1} = (send\_m_n^I)\big] \wedge \\ & \quad \big[s_k = (send\_m_k^I, send\_m_k^{II}) \forall k \leqslant i\big] \end{cases}$$

Finally, we will define an utility function for entity P_{v-1}.

Definition 8.8.4. *The following $u_{v-1} : S \to \mathbb{R}$ represents the utility function for entity P_{v-1}, $\forall s = (s_0, \ldots, s_{v-1}) \in S$:*

$$u_{v-1}(s) = \begin{cases} 0 & \text{if } \big[s_0 = (send\_m_0^I, quit^{II})\big] \vee \\ & \quad \big[\exists k \in \{1, \ldots, v-1\} : s_k = (quit^I, \cdot)\big] \vee \\ & \quad \big[\exists k \in \{1, \ldots, v-2\} : s_k = (send\_m_k^I, quit^{II})\big] \\ pay_n & \text{if } \big[s_0 = (send\_m_0^I, send\_m_0^{II})\big] \wedge \\ & \quad \big[s_{v-1} = (send\_m_{v-1}^I)\big] \wedge \\ & \quad \big[s_k = (send\_m_i^I, send\_m_i^{II}) \forall k < v-1\big] \end{cases}$$

Figures 8.11 and 8.12 represent in extensive–form both phases of the G_{v--RES} game. The tree represents the different moves each participant can make and all the possible outcomes. The vectors assigned to each terminal node represent the values of the payoff function. The first value corresponds to participant P_0 and the rest are the payoff values for entities P_1 to P_{v-1}.

8.8.5 Rationality by Backward Induction

As in Section 7.4, our formal analysis of the v–RES family will be based on applying backward induction to the protocol game G_{v--RES}.

The following theorem serves to formulate the main result of this analysis.

Theorem 8.8.1. *The strategy profile $s^* \in S$ defined as $s^* = (s_0^*, \ldots, s_{v-1}^*)$ where:*

$$\begin{aligned} &s_i^* = (send\_m_i^I, send\_m_i^{II}) \qquad \forall i \in \{0, \ldots, v-2\} \text{ and,} \\ &s_{v-1}^* = (send\_m_{v-1}^I) \end{aligned} \tag{8.5}$$

represents a Nash equilibrium perfect in sub–games for the G_{v--RES} game.

Proof. The proof is based on applying the backward induction algorithm to G_{v--RES} defined in Section 8.8.

Entity P_{v-2} is the last player to move in the last phase of the protocol game. Entity P_{v-2} has to chose between quitting the protocol $quit^{II}$ or sending message m_{v-2}^{II}. Each option will render P_{v-2} the following different payoff values:

$$u_{v-2}(s_{v-2}, s_{-(v-2)}) = \begin{cases} pay_{v-2} & \text{if} \quad \left[s_{v-2} = (\cdot, quit^{II})\right] \\ (\delta_{v-2} + 1) * pay_{v-2} & \text{if} \quad \left[s_{v-2} = (\cdot, send\_m_{v-2}^{II})\right] \end{cases} \tag{8.6}$$

All participant entities are considered to be rational, so all entities play to maximize their payoffs. Since $\delta_{v-2} > 0$, strategy $(\cdot, send\_m_{v-2}^{II})$ is a dominant[1] strategy for P_{v-2} in phase II of the protocol. When it is entity P_{v-3}'s turn to play, P_{v-3} is aware of entity P_{v-2}'s dominant strategy and behaves accordingly to maximize her payoff. In general, the backward induction process forces every entity P_i in phase II of the protocol to choose between the following two strategies with the following different payoffs:

$$u_i(s_i, s_{-i}) = \begin{cases} pay_i & \text{if} \quad \left[s_i = (\cdot, quit^{II})\right] \\ (\delta_i + 1) * pay_i & \text{if} \quad \left[s_i = (\cdot, send\_m_i^{II})\right] \end{cases} \tag{8.7}$$

Therefore, in phase II every entity P_i $i \in \{1, \ldots, v-2\}$ is *rationally* forced to play $send\_m_i^{II}$ instead of $quit^{II}$.

When it is P_0's turn to initiate phase II, P_0 knows the strategies that entities P_1 to P_{v-2} are going to play and behaves accordingly, maximizing her payoff. What follows are the values for entity P_0's payoff function at the beginning of phase II.

[1] See Section A.1.2 for formal descriptions of dominant and dominated strategies of a game.

$$u(s_0, s_{-0}) = \begin{cases} 0 & \text{if} \quad \left[s_0 = (\cdot, quit^{II})\right] \\ 1 & \text{if} \quad \left[s_0 = (\cdot, send\_m_0^{II})\right] \end{cases} \tag{8.8}$$

Entity P_0 is then *rationally* forced to follow the protocol description and initiate phase II choosing strategy $s_0 = (\cdot, send\_m_0^{II})$. Considering the space of strategies S_0 defined in Definition 8.8.1, in order to maximize her payoff value, entity P_0 is forced to choose strategy $s_0 = (send\_m_0^{I}, send\_m_0^{II})$.

Following the backward induction process onto phase I, entity P_{v-1} is the last mover of the first phase. At this point, P_{v-1} knows how rational entities P_0 and P_1 to P_{v-2} are going to play in phase II. Next, are entity P_{v-1}'s two possible strategies and the corresponding payoffs at the end of phase I:

$$u_{v-1}(s_{v-1}, s_{-\{v-1\}}) = \begin{cases} 0 & \text{if} \quad \left[s_{v-1} = (quit^{I})\right] \\ pay_{v-1} & \text{if} \quad \left[s_{v-1} = (send\_m_{v-1}^{I})\right] \end{cases} \tag{8.9}$$

Strategy $s_{v-1} = (send\_m_{v-1}^{I})$ is a dominant strategy since entity P_{v-1} would not choose a different action under any circumstances .

Backing the process to any step in phase I of the protocol, for all $i \in \{0, \ldots, v-2\}$, we obtain the following results:

$$u_i(s_i, s_{-i}) = \begin{cases} 0 & \text{if} \quad \left[s_i = (quit^{I}, send\_m_i^{II})\right] \\ (\delta_i + 1) * pay_i & \text{if} \quad \left[s_i = (send\_m_i^{I}, send\_m_i^{II})\right] \end{cases} \tag{8.10}$$

Strategy $(send\_m_i^{I}, send\_m_i^{II})$ is therefore a dominant strategy for all P_i, with $i \in \{0, \ldots, v-2\}$.

Summarizing, by applying backward induction to the v–RES protocol game, we have stated the following results:

a. Strategy s^* defined in Equation (8.5) is a dominant strategy for each participant entity.

b. In all sub–games considered during the induction process, the strategy profile s^* represents a Nash Equilibrium as no player has anything to gain by changing her strategy unilaterally.

Therefore, strategy s^* defined in Equation (8.5) is a Nash equilibrium perfect in sub–games as defined in Section A.4.1. □

Corollary 8.8.1. *For any $v \geqslant 3$, the v–RES protocol, as defined in Section 8.8, is a multi–party rational–exchange security protocol.*

Proof. Theorem 8.8.1 defines a strategy profile representing an unique solution for the G_{v--RES} game. Such a strategy profile corresponds exactly to the v–RES protocol specification given in Section 8.8.

Such a solution is also a sub–game perfect equilibrium, so no other strategies result in higher benefits when any of the entities unilaterally change their behavior. Therefore, deviating from the protocol description does not represent a profitable option. This being the actual definition of a rational–exchange protocol given in by Definition 1.3.2, is a conclusive result. □

8.9 Conclusions

Although, the formal foundations of our methodology ensure high levels of flexibility and scalability for any multi–party rational exchange problem, for the purpose of this chapter, we have designed and implemented a three–entity search algorithm based on Simulated Annealing.

The methodology has been proven to be extremely efficient (high success rates in very low running times) for the synthesis of 3–RES protocols.

Furthermore, an automatically discovered three–entity solution has been extended to a family of v–party protocols for which rationality has been formally proven. To the best of our knowledge, the protocols of this family are the first multi–party rational-exchange schemes proposed so far in the literature.

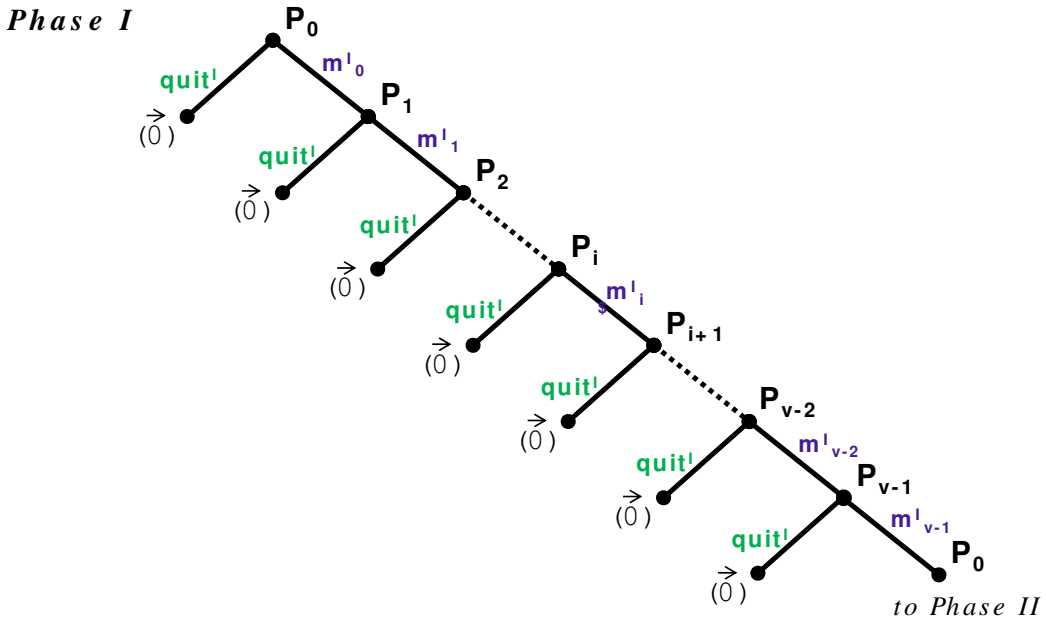


Figure 8.11: Formal representation of the v–RES protocol game, phase I.

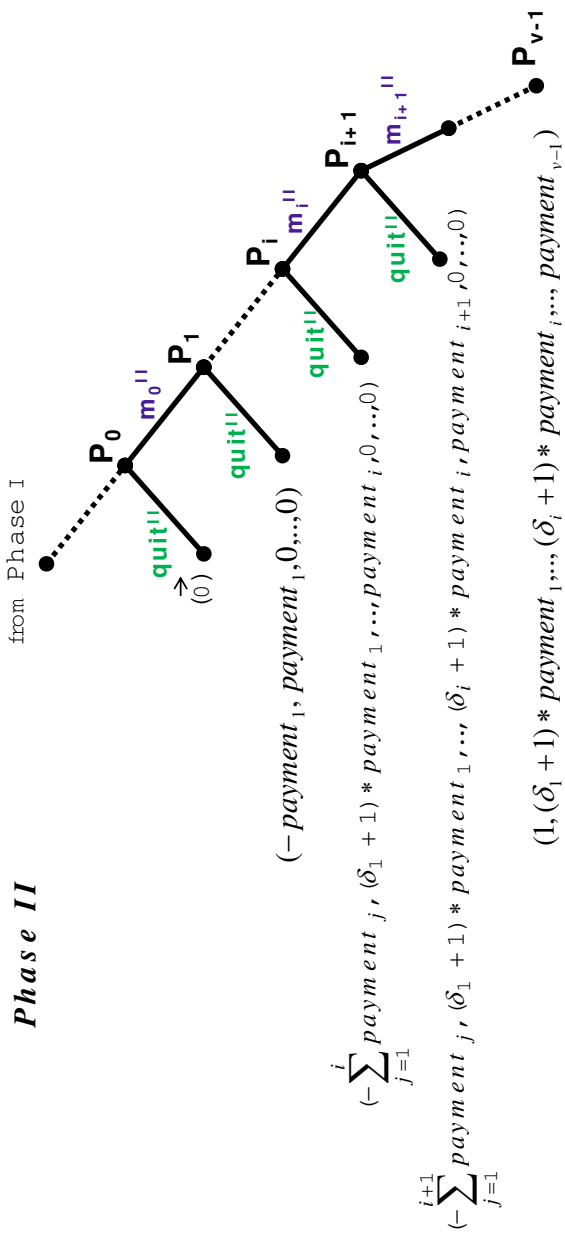


Figure 8.12: Formal representation of the v–RES protocol game, phase II.

Chapter 9

Solving More Complex Problems

9.1 Introduction

The formalism described in Chapter 7 allows us to transform the problem of how to design a multi–party rational exchange protocol into an optimization problem. Representing multi-parte rational exchange problems and protocols using a set of matrices is the first step of such a transformation. Additionally, the definition of several utility functions make it possible to carry out an algebraical evaluation of all candidate solutions.

In Chapter 8 two heuristic techniques, Hill Climbing and Simulated Annealing, were successfully applied to the automated synthesis of 3–entity rational exchange protocols.

In this chapter we present the results when the same approach is adopted for the synthesis of rational protocols in more complex exchanging scenarios where, not only the number of entities increases, but also the number of tokens and the possible dependencies that there might exists amongst these items.

9.1.1 Chapter Overview

Unfortunately, there is no such a thing as a bench marking set of rational exchange problems. Beyond standard parameters such as number of entities and number of exchangeable tokens, multi-party rational exchange problems can come in any form or shape. Each exchanging scenario has got its own peculiarities which may depend on a variety of factors:

- Value of the items to be exchanged.

- Level of dependency amongst these items.
- Asymmetrical or non-asymmetrically incentivized environments and,
- Coalition free vs. jointed or disjointed coalited instances.

Therefore, increasing the global scale of a multi–party exchange problem causes an explosion on the number of variants of the given problem.

In particular, increasing the scale of a problem increases the size of the matrices of the model (state matrix H, benefit matrix B and dependency matrix R) in which all environmental factors must be defined and represented by their elements.

In this chapter we have focused on two sets of experimental work:

1. We experiment with the synthesis of v–RES protocols ($v > 3$) for the problem described in Section 8.2 and,
2. We describe a tool which will assist us in the automated definition and generation of multi-party exchange problems. Given a set of input parameters, the tool produces a series of exchanging problems of certain characteristics which are represented using matrices H, B and R. Amongst all valid problems generated and, attending certain criteria, the most difficult ones are selected. For these the same methodology based on heuristic algorithms is applied in search of rational solutions.

As in previous chapters, we will also provide the corresponding statistical tables describing performance rates and, we will present some synthesized multi–party rational exchange protocols resulting of the experimentation.

9.1.2 Chapter Organization

The chapter is organized as follows. In Section 9.2 we present the results for the synthesis of v–RES protocols ($v > 3$) in the particular exchanging scenario described in previous Section 8.2. In Section 9.3 we describe the necessary parameters and the main properties of randomly generated multi–party exchange problems. We also define the criteria followed to select the set of problems which are going to be resolved using heuristic search techniques and give a graphical representation of these. In Section 9.4 we apply a heuristic technique, based on Simulated Annealing, for the synthesis of rational solutions to the previously selected problems and we present the performance rates attained. Some of the rational solutions synthesized in previous section are described in Section 9.5. Finally, in Section 9.6 we present the final conclusions to this experimental work.

v	SR	Avg. NPE
3	99.6%	24,531
4	100.0%	35,848
5	100.0%	780,811
6	90.0%	10,213,297

Table 9.1: Automatically synthesized v–RES solutions.

9.2 Automated Generation of v–RES Solutions

In Chapter 8 two heuristic techniques were successfully applied for the automated synthesis of 3–entity rational exchange protocols. These protocols were constructed to solve the particular problem described in Section 8.2 referred to as the 3–RES problem. Furthermore, four different scenarios were considered for the synthesis of candidate solutions to such a problem.

In this section we present the results obtained when applying a heuristic technique, based on Simulated Annealing, in search of solutions for v–RES problems when $v > 3$. As it was formally established in Section 8.8, for each v–RES problem there exists a rational solution. These protocols constitute the so called v–RES protocol family.

Table 9.1 shows, for different values of v (number of entities), the success rate (column SR) and average number of protocols evaluated in each search (column Av. NPE.). The experimentation was carried out over 500 trials. Note that the technique is successful in all instances. For a bigger number of entities, the average number of protocols evaluated will simply be larger.

9.3 Randomized Multi–party Rational Exchange Problems

In this section a tool has been implemented which, given a set of parameters it automatically generates valid exchange problems represented in matrices $H(0)$ (initial state) , B (benefit matrix) and R (dependency matrix).

We will now describe the set of input parameters as well as the main characteristics of the problems automatically generated by this tool.

9.3.1 Problem Generation Parameters and Characteristics

When automatically generating a multi–party exchange problem the following parameters must be determined.

Input Parameters

1. Number of entities,
2. Number of tokens and,
3. A factor defined as $R\_DENSITY$ which establishes the density of the dependency matrix R. More on this parameter will come in the following paragraphs.

For such a set of input parameters, a series of exchange problems are randomly generated with the following characteristics.

Characteristics

- **Entities' initial possessions:** An initial state matrix $H(0)$ is randomly generated such that:
  - For each item, only one entity is the initial owner of that token.
  - Entities can posses zero, one or more than one items.
  - Also, all items are *accessible* to their corresponding owners and no item can be marked as *lost*.
- **Entities' final requirements:** Entities' final requirements are randomly generated and represented in a benefit matrix B such that:
  - For each token, only one entity (different from the owner) is randomly chosen to require this item.
  - A positive integer greater than one is randomly generated, representing how much this item is worth to the requiring entity.
  - Entities could require zero, one or more than one items.
- **Coalitions:** Coalitions between participants are randomly generated and represented in benefit matrix B.
  - Coalitions are such that, sending messages containing tokens required by an allied entity increases the sender's utility.
  - By contrast, forwarding tokens required by non–allied participants could be either cost–free or, it could represent a cost for the sender entity.
  - Of every possible pair of participant entities, 1/3 are considered to be in coalition.

  - Of every possible pair of participant entities, 1/3 are considered to be in a non–coalition form and to have no cost in relaying each other tokens and,
  - Finally, 1/3 of pairs of participants are considered to be in a non–coalition form, and to incur cost when forwarding each other required–tokens.

- **Incentives:** Incentives are also represented in benefit matrix B.
  - It is randomly chosen whether the owner of a token is incentivized to the exchange of that item. Each entity will be incentivized to the exchange of 1/2 of its own tokens.

- **Items inter–dependencies:** All dependency relationships amongst exchangeable items of a given problem will also be randomly generated and represented in matrix R. The density of this matrix will be determined by the input parameter $R\_DENSITY$. $R\_DENSITY$ will be the percentage of values in matrix R different from zero. Although other distributions can apply, the problems generated by this tool will be such that:
  - $(1/3) * R\_DENSITY$ will be negative relationships and $(2/3) * R\_DENSITY$ are positive ones. Positive and negative dependency relationships are defined in Section 7.2.3.
  - Negative relationships are not necessarily followed by positive ones. This means that a certain sequence of events in a protocol solution may be forced by the fact that items could become *non-accessible* forever when they are sent in the wrong order.

9.3.2 Difficult Problems

For each set of input parameters, a series of problems are randomly generated of the previously discussed characteristics. Amongst all these problems only the most difficult will be selected and an heuristic algorithm will be applied in search of a rational solution. All rational solutions will be represented by a protocol matrix S as defined in Section 7.2.1.

The same technique denominated Random Walk technique (see Section 6.3.1) will be used to determine the difficulty of a randomized multi–party exchange problem. A value λ will serve to measure difficulty in relation to the fitness landscape of a given problem. The parameter λ is computed as $\sum_i (L/\rho_i)$ where L is the random walk length and ρ_i are the autocorrelation values for each shift value in the fitness of neighboring protocols.

Input Parameters
No. of entities
No. of exchangeable tokens
R_DENSITY factor
Output Parameters
Matrix H(0)
Entities initial possessions
Matrix B
Entities final requirements
Coalitions
Incentives
Payoff values
Matrix R
Items dependency relationships
Tuning Parameters
No. of problems generated
Random walk length
No. of random walks

Table 9.2: Randomized problem generation parameters.

Small autocorrelation values will represent abrupt fitness landscapes which will be identified as difficult problems and will be represented by high λ values.

Finally, table 9.2 serves to summarize all exposed parameters involved in the generation of randomized multi–party exchange problems.

9.3.3 Graphical Representation of Problems

A graphical representation of multi–party rational exchange problems will assist in identifying the type of problem generated. Part of the information about a given exchanging scenario, in particular information contained in matrices $H(0)$ and B, can be represented in a graph of the following properties:

- Each entity is represented in a circle.
- The set of items required by each one of the entities is grouped in a triangle figure attached to the corresponding entity.
- For each item that en entity possesses at the initial state, there will be an arrow pointing to the owner, labeled with the name of the possessed token. Dashed arrows describe non–incentivized entities towards the exchange of such an item, whereas solid lines indicate that the owner entity is part of an incentive scheme for the exchange of that item.
- Finally, solid lines connecting entities represent alliances or coalitions between entities, whereas dashed crossing lines indicate that relaying each other

required–tokens represents a cost in their utilities. The absence of links indicates cost free relays, this is, entities being indifferent towards each others required–items.

Figure 9.1 serves as an example of a exchanging scenario of five entities and six tokens, represented as a graph. The figure represents:

- The coalitions $P_0 \leftrightarrow P_1$, $P_0 \leftrightarrow P_4$ and, $P_1 \leftrightarrow P_2$.
- The non–coalition forms of relationship $P_1 \leftrightarrow P_3$ and $P_3 \leftrightarrow P_4$.
- Entity P_0 initially possesses item o_5 but it is not incentivized to exchange it.
- P_0 also requires item o_3.
- Entity P_1 initially possesses item o_0 and it is not incentivized to exchange it.
- P_1 requires item o_2.
- Entity P_2 has no initial possessions and requires items o_1 and o_5.
- Entity P_3 initially possesses items o_2 and o_4 and it is incentivized to lose control over both of them.
- P_3 requires item o_0.
- Finally, entity P_4 initially possesses item o_1 and it is incentivized for the exchange of this item and, it also possesses o_3 but it is not incentivized to lose control over it.
- P_4 requires item o_4.

9.4 Automated Generation of M–RES Solutions to Randomized Problems

Similar optimization technique, based on Simulated Annealing, as the one described in Chapter 7 will be used to explore the space of protocols and to find rational solutions for any given randomized multi–party exchange problem.

Several rounds of preliminary experimentation gave us the indication that the performance rate of the applied search technique was highly influenced by the $R\_DENSITY$ value of the given problem. For this reason, our experimental work has focused on five types of randomized problems:

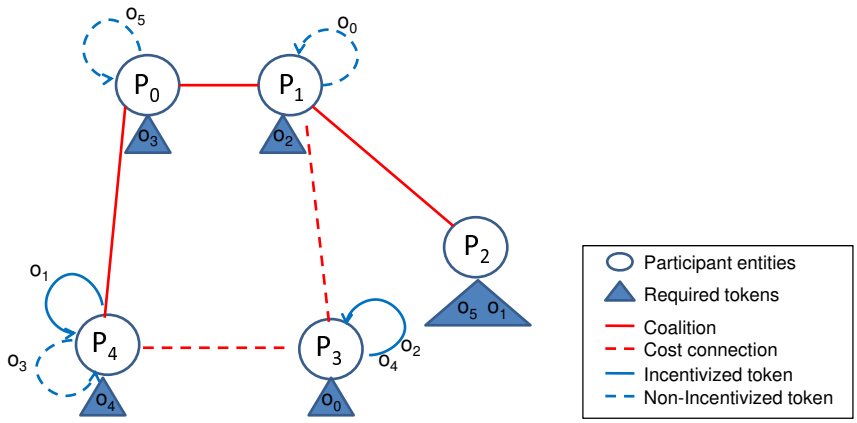


Figure 9.1: Example of a exchanging scenario of five entities and six tokens.

1. Multi–party exchange problems with zero $R\_DENSITY$. These are problems in which there is no dependency between the items to be exchanged.

2. Multi–party exchange problems with a low density value; these are problems where 1.0% of values in matrix R are non zero values ($R\_DENSITY = 0.010$).

3. Multi–party exchange problems with medium $R\_DENSITY$ value of 0.018; these are problems where 1.8% of values in matrix R are non zero values. Note that this is the density value for any of the v–RES family problems.

4. Multi–party exchange problems with medium $R\_DENSITY$ value of 0.02; problems where 2.0% of values in matrix R are non zero values.

5. Multi–party exchange problems with high $R\_DENSITY$; these are problems where 5.0% of values in matrix R are non zero values.

Several trials of experimentation have shown that the search technique fails when applied to problems with higher $R\_DENSITY$ values than 0.05. Highly dependency rates amongst exchangeable items make the exchange problem a difficult one to resolve or even impossible.

The algorithm used in the search is Simulated Annealing. Table 9.2(a) shows the parameters of the experiment. Furthermore, Table 9.2(b) shows the performance rates obtained when exploring the space of candidate solutions for different problems. Varying the number of entities, number of tokens and number of protocol steps, column (SR) shows the success rate of the search technique used, and column (Av. NPE.) shows the average number of protocols evaluated in each trial search.

SA Parameter	**Value**
Number of cycles	210
Number attempts	100000
Initial temperature	0.43
Cooling Factor	0.8837
PRNG	Mersenne Twister

(a)

			R_DENSITY									
			0.0		1.0		1.8		2.0		5.0	
No. of Entities	No. of Tokens	No. of Messages	SR	Av. NPE.	SR	Av. NPE.	SR	Av. NPE.	SR	Av. NPE.	SR	Av. NPE.
4	6	7	100.0%	71,039	100.0%	110,203	73.0%	850,836	30.0%	1,872,032	50.0%	1,080,918
6	9	11	80.0%	77,234	90.0%	62,595	40.0%	135,219	80.0%	74,119	0.0%	$21 \cdot 10^6$
6	15	11	80.0%	1,448,821	0.0%	$21 \cdot 10^6$	0.0%	$21 \cdot 10^6$	0.0%	$21 \cdot 10^6$	"	"
8	15	15	60.0%	1,131,533	0.0%	$21 \cdot 10^6$	0.0%	$21 \cdot 10^6$	"	"	"	"
10	15	15	0.0%	$21 \cdot 10^6$	0.0%	$21 \cdot 10^6$	"	"	"	"	"	"

(b)

Figure 9.2: Results on the synthesis of multi–party rational exchange problems.

9.5 Automatically Synthesized Protocols

Next we present some of the multi–party problems automatically generated, their graphical representation and several rational protocols synthesized to solve such problems.

We focus on describing four and six entity protocols to illustrate the type of solution generating using our methodology.

9.5.1 Examples with Four Entities and Zero R_DENSITY

We will now describe a four–entity randomized problem with zero $R\_DENSITY$. Matrices $H(0)$, B will define every parameter of the problem and matrix R will contain all zero values. The number of exchangeable items will be six.

Matrix $H(0)$ defines the initial state of possessions and benefit matrix B defines coalitions, incentives and payoff values. Both are represented in Figure 9.3(a). Furthermore, Figure 9.3(a) also represents the information described in previous matrices as a graph. The figure represents the coalition between $P_2 \leftrightarrow P_3$ and the non–coalition form of relationship between $P_0 \leftrightarrow P_2$ and $P_1 \leftrightarrow P_2$. Entity P_0 initially possesses items o_0 and o_4 and it is incentivized to exchange any of them. P_0 requires item o_3. Entity P_1 initially possesses items o_3 and o_5 and it is not incentivized to exchange any of them. P_1 requires items o_0 and o_4. Entity P_2 has item o_1 as the only initial possession and requires items o_2 and o_5. Finally, entity P_3 initially possesses item o_2 and it is incentivized to lose control over it. P_3 requires item o_1.

In Figure 9.3(b) we present three possible rational protocols which are automatically synthesized and give solution to the exchange problem just described.

Figure 9.3(b.I) shows a synthesized solution in which the number of messages is five. Entity P_1 starts by splitting its tokens between P_0 and P_3, one to each recipient. P_3 sends the token just received from P_0 together with its own item o_2 to P_2. Entity P_0 keeps the token received from P_1 and sends its own tokens to P_1. Finally, P_2 sends P_3 its own item o_1. Figures 9.3(b.II) and (b.III) show two more solutions to the previously described problem. In this case, the protocols consist of six messages.

9.5.2 Examples with Six Entities and Zero R_DENSITY

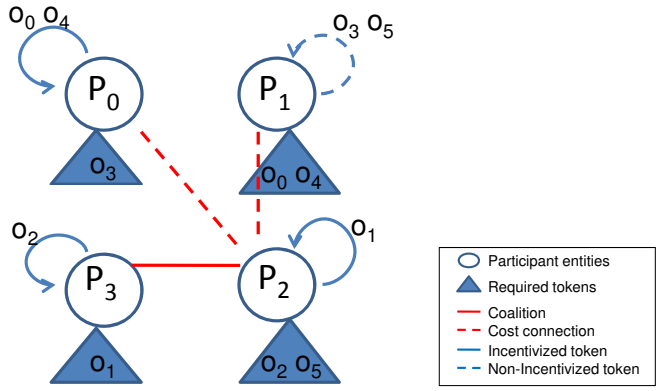
A randomized six–entity problem is described in Figure 9.4(a). Matrix $H(0)$ defines the initial state of possessions, benefit matrix B defines coalitions, incentives and payoff values, and matrix R contains all zero values. The number of exchangeable items is eight.

Matrix H(0)

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Matrix B

$$\begin{pmatrix} -1 & 0 & 1 & 15 & -1 & 1 \\ 51 & 0 & 1 & 1 & 51 & 1 \\ 1 & -1 & 27 & 1 & 1 & 168 \\ 0 & 108 & 1 & 0 & 0 & -1 \end{pmatrix}$$



(a)

(1)	$P_1 \to P_0:$	$\{o_3\}$
(2)	$P_1 \to P_3:$	$\{o_5\}$
(3)	$P_3 \to P_2:$	$\{o_2, o_5\}$
(4)	$P_0 \to P_1:$	$\{o_0, o_4\}$
(5)	$P_2 \to P_3:$	$\{o_1\}$

(b.I)

(1)	$P_2 \to P_1:$	$\{o_1\}$
(2)	$P_1 \to P_3:$	$\{o_5\}$
(3)	$P_1 \to P_0:$	$\{o_3\}$
(4)	$P_3 \to P_2:$	$\{o_2, o_5\}$
(5)	$P_1 \to P_3:$	$\{o_1\}$
(6)	$P_0 \to P_1:$	$\{o_0, o_4\}$

(b.II)

(1)	$P_3 \to P_2:$	$\{o_2\}$
(2)	$P_1 \to P_3:$	$\{o_5\}$
(3)	$P_1 \to P_0:$	$\{o_3\}$
(4)	$P_0 \to P_1:$	$\{o_0, o_4\}$
(5)	$P_3 \to P_2:$	$\{o_5\}$
(6)	$P_2 \to P_3:$	$\{o_1\}$

(b.III)

(b)

Figure 9.3: (a) Four–entity randomized problem. (b) Four–entity automatically synthesized solutions.

Furthermore, Figure 9.4(b.I) and (b.II) show two different solutions to the previously described problem. The protocols consist of seven messages each.

9.5.3 Examples with Four Entities and Medium $R\_DENSITY$

Figure 9.5(a) represents a randomized four–entity problem with nine exchangeable items. Matrix $H(0)$ defines the initial state of possessions and benefit matrix B defines coalitions, incentives and payoff values. The value of $R\_DENSITY$ for this problem is 0.018, this is, 1.8% of the elements of matrix R are non–zero values. Table Figure 9.5(b) represents the dependency relationships between several of the exchangeable items.

A rational solution automatically synthesized for this problem is shown in Figure 9.6.

9.5.4 Examples with Six Entities and Medium $R\_DENSITY$

Figure 9.7(a) represents a randomized six–entity problem with nine exchangeable items. Matrix $H(0)$ defines the initial state of possessions and benefit matrix B defines coalitions, incentives and payoff values. The value of $R\_DENSITY$ for this problem is 0.018. This is, 1.8% of the elements of matrix R are non–zero values and serve to represent the dependency relationships between several of the exchangeable items. Values of matrix R are defined in Table (b) in Figure 9.7.

Rational solutions automatically synthesized for this problem are shown in Figure 9.8(a) and (b). Both protocols consisting on nine messages each.

9.6 Conclusions

In this chapter we have carried out further experimentation applying a heuristic search algorithm (Simulated Annealing) for the synthesis of rational protocols in more complex randomized environments.

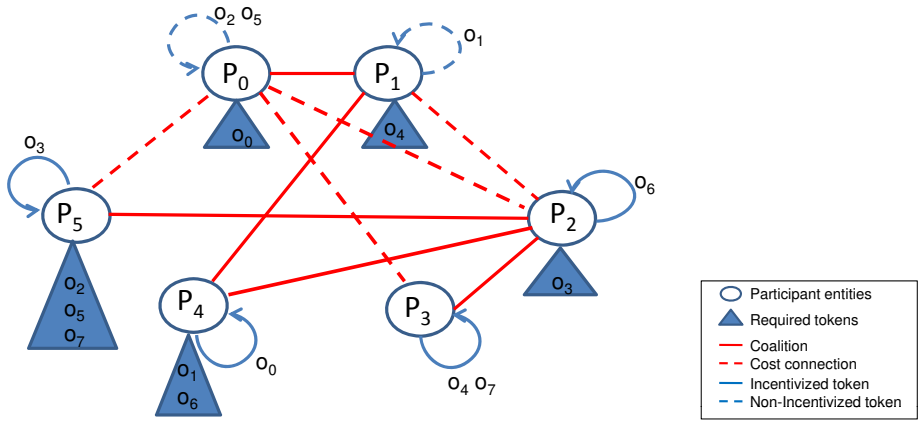
The resulting data indicates that the technique is highly successful and that it can resolve exchange problems in reasonable bounds of time. Furthermore, for every protocol synthesized using this methodology there exists a formal proof of rationality. All protocols can be represented as protocol–games within the framework described in Part I of this thesis. The Nash equilibrium (NE) of these games can be computed applying backward induction (see Theorem 7.4.1), being able to specify that the NE corresponds to the sequence of steps described in the protocol.

Matrix H(0)

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Matrix B

$$\begin{pmatrix} 202 & 0 & 1 & 1 & -1 & 1 & 0 & 1 \\ -1 & 1 & 0 & 1 & 133 & 0 & -1 & 0 \\ 1 & -1 & -1 & 291 & 1 & -1 & -1 & -1 \\ 1 & 0 & -1 & 0 & -1 & -1 & 0 & -1 \\ -1 & 82 & 0 & -1 & -1 & 0 & 91 & 0 \\ 1 & 0 & 156 & -1 & 0 & 142 & 0 & 278 \end{pmatrix}$$



(a)

(1)	$P_0 \rightarrow P_3$:	$\{o_2, o_5\}$
(2)	$P_1 \rightarrow P_2$:	$\{o_1\}$
(3)	$P_4 \rightarrow P_0$:	$\{o_0\}$
(4)	$P_2 \rightarrow P_4$:	$\{o_1, o_6\}$
(5)	$P_3 \rightarrow P_1$:	$\{o_4\}$
(6)	$P_3 \rightarrow P_5$:	$\{o_2, o_5, o_7\}$
(7)	$P_5 \rightarrow P_2$:	$\{o_3\}$

(b.I)

(1)	$P_5 \rightarrow P_2$:	$\{o_3\}$
(2)	$P_0 \rightarrow P_3$:	$\{o_2, o_5\}$
(3)	$P_1 \rightarrow P_4$:	$\{o_1\}$
(4)	$P_4 \rightarrow P_0$:	$\{o_0\}$
(5)	$P_2 \rightarrow P_4$:	$\{o_6\}$
(6)	$P_3 \rightarrow P_5$:	$\{o_2, o_5, o_7\}$
(7)	$P_3 \rightarrow P_1$:	$\{o_4\}$

(b.II)

(b)

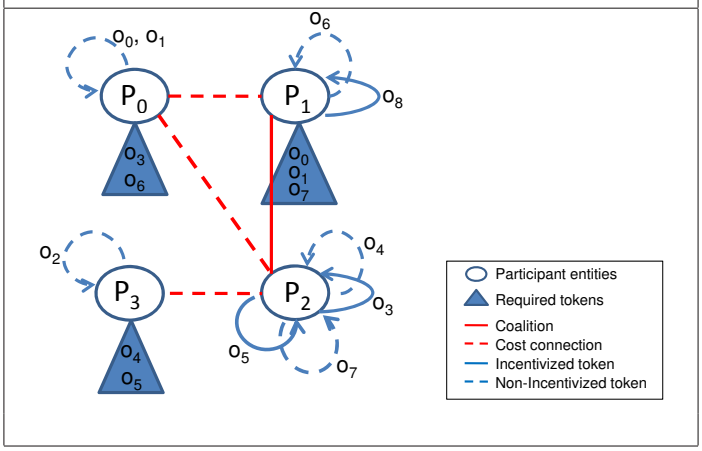
Figure 9.4: (a) Six–entity randomized problem. (b) Six–entity automatically synthesized solutions.

Matrix H(0)

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Matrix B

$$\begin{pmatrix} 1 & 1 & 1 & 224 & 0 & 0 & 44 & 1 & 1 \\ 72 & 252 & -1 & 1 & 0 & 0 & 1 & 157 & -1 \\ -1 & -1 & 99 & 1 & -1 & 1 & 1 & -1 & 162 \\ 0 & 0 & 1 & 0 & 225 & 254 & 0 & 0 & -1 \end{pmatrix}$$



(a)

Matrix R		
$r_{0,12}$ = POS_DR	$r_{4,10}$ = POS_DR	$r_{2,17}$ = NEG_DR
$r_{4,34}$ = POS_DR	$r_{8,21}$ = POS_DR	$r_{6,33}$ = NEG_DR
$r_{8,30}$ = POS_DR	$r_{12,24}$ = POS_DR	$r_{21,10}$ = NEG_DR
$r_{13,7}$ = POS_DR	$r_{15,30}$ = POS_DR	$r_{25,18}$ = NEG_DR
$r_{18,21}$ = POS_DR	$r_{18,35}$ = POS_DR	$r_{33,18}$ = NEG_DR
$r_{19,16}$ = POS_DR	$r_{27,18}$ = POS_DR	$r_{34,25}$ = NEG_DR
$r_{27,34}$ = POS_DR	$r_{28,4}$ = POS_DR	
$r_{28,11}$ = POS_DR	$r_{29,15}$ = POS_DR	
$r_{32,14}$ = POS_DR	$r_{32,28}$ = POS_DR	
$r_{33,28}$ = POS_DR	$r_{35,10}$ = POS_DR	

(b)

Figure 9.5: Four–entity randomized problem ($R\_DENSITY = 1.8\%$).

(1)	$P_0 \rightarrow P_2$:	$\{o_0, o_1\}$
(2)	$P_1 \rightarrow P_3$:	$\{o_6\}$
(3)	$P_2 \rightarrow P_3$:	$\{o_0, o_1, o_3, o_5, o_7\}$
(4)	$P_3 \rightarrow P_1$:	$\{o_0, o_1, o_2, o_7\}$
(5)	$P_3 \rightarrow P_0$:	$\{o_3, o_6\}$
(6)	$P_2 \rightarrow P_3$:	$\{o_4\}$
(5)	$P_1 \rightarrow P_2$:	$\{o_2, o_8\}$

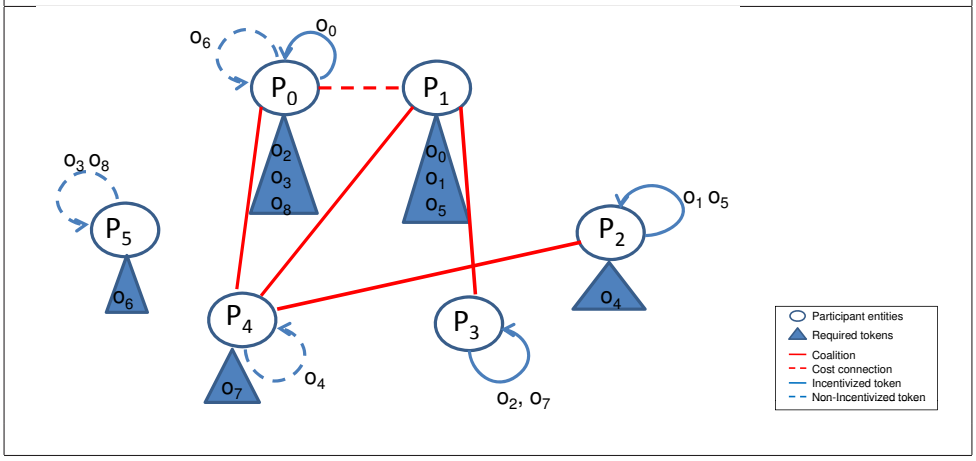
Figure 9.6: Four–entity synthesized solution ($R\_DENSITY = 1.8\%$).

Matrix H(0)

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Matrix B

$$\begin{pmatrix} -1 & 1 & 196 & 244 & 0 & 1 & 1 & -1 & 136 \\ 69 & 91 & 1 & 1 & 0 & 259 & 0 & -1 & 1 \\ 0 & -1 & 0 & 0 & 79 & -1 & 0 & -1 & 0 \\ -1 & -1 & -1 & 0 & 0 & -1 & 0 & -1 & 0 \\ -1 & -1 & -1 & -1 & 1 & -1 & 0 & 6 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 216 & 0 & 1 \end{pmatrix}$$



(a)

Matrix R			
$r_{2,18} = \texttt{POS\_DR}$	$r_{3,17} = \texttt{POS\_DR}$	$r_{5,7} = \texttt{POS\_DR}$	$r_{5,35} = \texttt{NEG\_DR}$
$r_{6,43} = \texttt{POS\_DR}$	$r_{10,8} = \texttt{NEG\_DR}$	$r_{10,40} = \texttt{POS\_DR}$	$r_{12,41} = \texttt{NEG\_DR}$
$r_{15,24} = \texttt{NEG\_DR}$	$r_{18,45} = \texttt{NEG\_DR}$	$r_{19,16} = \texttt{POS\_DR}$	$r_{19,28} = \texttt{POS\_DR}$
$r_{21,7} = \texttt{POS\_DR}$	$r_{21,17} = \texttt{POS\_DR}$	$r_{22,3} = \texttt{POS\_DR}$	$r_{22,10} = \texttt{POS\_DR}$
$r_{24,5} = \texttt{POS\_DR}$	$r_{24,38} = \texttt{POS\_DR}$	$r_{24,52} = \texttt{POS\_DR}$	$r_{25,53} = \texttt{POS\_DR}$
$r_{26,12} = \texttt{POS\_DR}$	$r_{29,8} = \texttt{NEG\_DR}$	$r_{29,36} = \texttt{POS\_DR}$	$r_{30,1} = \texttt{POS\_DR}$
$r_{30,19} = \texttt{NEG\_DR}$	$r_{32,1} = \texttt{POS\_DR}$	$r_{32,3} = \texttt{POS\_DR}$	$r_{32,15} = \texttt{POS\_DR}$
$r_{33,19} = \texttt{POS\_DR}$	$r_{33,39} = \texttt{NEG\_DR}$	$r_{33,50} = \texttt{NEG\_DR}$	$r_{35,20} = \texttt{NEG\_DR}$
$r_{35,24} = \texttt{NEG\_DR}$	$r_{35,26} = \texttt{POS\_DR}$	$r_{35,43} = \texttt{NEG\_DR}$	$r_{36,15} = \texttt{POS\_DR}$
$r_{39,46} = \texttt{POS\_DR}$	$r_{40,37} = \texttt{NEG\_DR}$	$r_{41,8} = \texttt{NEG\_DR}$	$r_{41,46} = \texttt{NEG\_DR}$
$r_{43,9} = \texttt{POS\_DR}$	$r_{44,17} = \texttt{NEG\_DR}$	$r_{45,37} = \texttt{NEG\_DR}$	$r_{46,48} = \texttt{NEG\_DR}$
$r_{47,3} = \texttt{POS\_DR}$	$r_{47,38} = \texttt{POS\_DR}$	$r_{49,24} = \texttt{NEG\_DR}$	$r_{51,14} = \texttt{POS\_DR}$
$r_{51,27} = \texttt{NEG\_DR}$	$r_{51,36} = \texttt{POS\_DR}$	$r_{51,43} = \texttt{POS\_DR}$	$r_{52,27} = \texttt{NEG\_DR}$
$r_{53,11} = \texttt{POS\_DR}$			

(b)

Figure 9.7: Six–entity randomized problem ($R\_DENSITY = 1.8\%$).

(1)	$P_5 \rightarrow P_0:$	$\{o_8\}$
(2)	$P_4 \rightarrow P_2:$	$\{o_4\}$
(3)	$P_0 \rightarrow P_3:$	$\{o_0\}$
(4)	$P_5 \rightarrow P_3:$	$\{o_3\}$
(5)	$P_3 \rightarrow P_4:$	$\{o_2, o_3, o_7\}$
(6)	$P_0 \rightarrow P_5:$	$\{o_6\}$
(7)	$P_4 \rightarrow P_0:$	$\{o_2, o_3\}$
(8)	$P_2 \rightarrow P_1:$	$\{o_1, o_5\}$
(9)	$P_3 \rightarrow P_1:$	$\{o_0\}$

(a)

(1)	$P_5 \rightarrow P_0:$	$\{o_8\}$
(2)	$P_4 \rightarrow P_2:$	$\{o_4\}$
(3)	$P_0 \rightarrow P_3:$	$\{o_0\}$
(4)	$P_5 \rightarrow P_3:$	$\{o_3\}$
(5)	$P_3 \rightarrow P_4:$	$\{o_2, o_3, o_7\}$
(6)	$P_0 \rightarrow P_5:$	$\{o_6\}$
(7)	$P_4 \rightarrow P_0:$	$\{o_2, o_3\}$
(8)	$P_2 \rightarrow P_1:$	$\{o_1, o_5\}$
(9)	$P_3 \rightarrow P_1:$	$\{o_0\}$

(b)

Figure 9.8: Six–entity rational solutions ($R\_DENSITY = 1.8\%$).

Part III

List of Contributions, Conclusions and Future Work

Chapter 10

Conclusions and List of Contributions

10.1 Summary

The work in this thesis has been divided into two different parts: *Part (I) Game Theoretical Analysis of Rational–Exchange Protocols* and *Part (II) Automated Design of Multi–party Rational–Exchange Security (M–RES) Protocols.*

The first part was devoted to the presentation of two different Game–theoretical models for the formal analysis of rational–exchange security protocols. The aim of these formalisms is to formally establish whether a given exchange scheme is or not a rational protocol.

In our opinion, various aspects of the two models represent a significant contribution to the area of protocol formal validation and analysis. The following relation highlights the most relevant:

1. **In-depth analytical framework:** Both of our formalisms extend an existing paradigm which already applies Game Theory to formally represent and verify rational schemes. However, in that previous work, only basic Game Theoretical results were applied making the analysis too simplistic and unrealistic.

   By contrast, we make use of more advanced concepts in Game Theory for the representation and analysis of exchange protocols, hence providing a formal framework for more rigorous in-depth analysis. Despite the analytical process becoming more complex than by using basic Game Theory, we find it more realistic and informative.

2. **Environmental factors:** The capability to include contextual factors into

the analysis of rational–exchange schemes, such as trust amongst participants, reputation, robustness of the scheme, reliability of the network, etc., is also a relevant aspect of our global approach.

Both models are highly flexible in terms of the number and type of environmental factors taken into account in the analytical process. Any contextual aspect can be parameterized and its impact on the outcome of a given rational scheme evaluated. For instance, trust might not play a role in a particular context while entities' past experience, or being part of an incentive scheme could have an effect on the overall process.

3. **Participant unpredictable behavior:** The way in which participant unpredictable behavior is defined within both formalisms also represents a significant break–through within the area.

   Traditionally, protocol participant misbehavior has always being approached from a restrictive point of view. This is, entities' capability to misbehave or deviate from a given scheme had always followed a predefined structure. By contrast, unpredictable behavior is in no way restricted in either of our two proposals. The way to model the unpredictable is solely based on assumptions over the probability of such events taking place rather than on their nature.

4. **Scalability:** Scalability is an important feature of any analytical model. In this case, our models set the basis for any multi–party rational exchange scheme to be represented and analyzed using the proposed methodology.

5. **Two models:** We have presented two models with many common attributes as well as significant dissimilarities. Informally, differences between the two models stem from different levels of uncertainty that participant entities hold about each other (further details were given in Section 4.4.1). In each formalism a different type of game is used to represent a given protocol.

   The application of both paradigms to the formal analysis of Syverson's rational exchange protocol serves to illustrate the two models. Moreover, the formal validation of a content distributing protocol in a pure peer to peer system, further illustrates the versatility of our approach.

5. **Significance for any further analysis:** Finally, the models described only served to analyze single instances of a rational exchange protocol, however, it is necessary to analyze single executions before studying player's strategies in iterated scenarios and this work represents the basis for any further analysis in that direction (in Section 10.3.1 we further elaborate on this topic).

In the second part of this thesis, Game Theory and heuristic search are applied to the automated design of multi–party rational–exchange protocols. Automated protocol synthesis represents a significant contribution to the design of rational–exchange schemes in which the number of existing proposals was so far minimal. Furthermore, several other aspects of this new approach are interesting and relevant within the field. The following relation highlights the most significant:

1. **Integral methodology:** The proposed methodology serves to integrate into the design process of a multi–party rational–exchange protocol the following:
   - Aspects, specific to any exchanging scenario, such as incentive schemes and coalitions amongst participants.
   - Items dependency relationships are also easily specifiable within the design phase. In many scenarios, there may exist strong dependency links between two or more of the exchangeable items, for example, an encrypted text and the decryption key.
   - A system proof based on Game Theory, for the automated design of provable multi–party rational–exchange protocols.
2. **Heuristic search:** The usage of non-standard computation techniques applied to the automated synthesis of rational protocols allows us to explore vast solution spaces, far greater than those possible through manual design.
4. **Experimental results:** Some experimental works guarantee the effectiveness and efficiency of our approach. Several multi–party rational protocols are depicted as a result of the automated synthesis.
5. **Family of M-RES protocols:** A family of rational–exchange protocols is defined and a formal proof of rationality is depicted, based on Game Theory and backward induction.

10.2 Publications

Some of the contributions presented in this thesis have already been published in various peer–reviewed conference proceedings and journals. Below we list them:

Conference and Workshop Publications

1. **"An Extended Model of Rational Exchange Based on Dynamic Games of Imperfect Information"**.

Alcaide, A., Estévez–Tapiador, J.M., Hernández–Castro, J.C. and Ribagorda, A. In *Proceedings of the International Conference on Emerging Trends in Information and Communication Security, ETRICS 2006.* LNCS Vol. 3995/2006, pp. 396-408.

2. **"Towards Automated Design of Multi–party Rational Exchange Security Protocols"**.

   Alcaide, A., Estévez–Tapiador, J.M., Hernández–Castro, J.C. and Ribagorda, A. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IATW '07.* IEEE Computer Society, pp. 387–390.

3. **"A Multi–party Rational Exchange Protocol"**.

   Alcaide, A., Estévez–Tapiador, J.M., Hernández–Castro, J.C. and Ribagorda, A. In *Proceedings of the Conference On the Move to Meaningful Internet Systems 2007, OTM 2007 Workshops.* Vol. 4805/2007, pp. 42–43.

4. **"Bayesian Analysis of Secure P2P Sharing Protocols"**.

   Palomar, E., Alcaide, A., Estévez–Tapiador, J.M. and Hernández–Castro, J.C. In *Proceedings of the Conference On the Move to Meaningful Internet Systems 2007, OTM 2007 Workshops.* Vol. 4805/2007, pp. 1701–1717.

5. **"Nature-Inspired Synthesis of Rational Protocols"**.

   Alcaide, A., Estévez–Tapiador, J.M., Hernández–Castro, J.C. and Ribagorda, A. In *Proceedings of 10th International Conference On Parallel Problem Solving from Nature - PPSN X.* LNCS Vol. 5199/2008, pp. 981–990.

Journal Articles

1. **"Bayesian rational exchange"**.

   Alcaide, A., Estévez–Tapiador, J.M., Hernández–Castro, J.C. and Ribagorda, A. *International Journal of Information Security*, Vol.7, No.1, pp. 85–100, Jan. 2008.

2. **"Cryptanalysis of Syverson's Rational Exchange Protocol"**.

   Alcaide, A., Estévez–Tapiador, J.M., Hernández–Castro, J.C. and Ribagorda, A. *International Journal of Network Security*, Vol.7, No.2, pp. 179-184, Sep. 2008.

3. **"Automated Synthesis of Multiparty Rational Exchange Security Protocols"**. Alcaide, A., Estévez–Tapiador, J.M., Hernández–Castro, J.C.

and Ribagorda, A. To appear in *International Transactions on Systems Science and Applications*. 2008

Book Chapters

1. "**Game Theory and Cooperation Analysis**" In "Cooperative Wireless Communications". (To be published by Auerbach Publications, Taylor&Francis Group, 2008.)

10.3 Open Issues and Future Work

In this section we discuss some open issues and ideas for possible enhancements to our research work, and potential future lines of investigation.

10.3.1 Repeated Scenarios for Rational–Exchange Protocols

We belief this is the most interesting future line of research to culminate the formal analysis of rational–exchange protocols.

In many cases, a more realistic scenario for the execution of a rational–exchange protocol will be that in which the same protocol is repeatedly executed for an indefinite number of times. By their very nature, repeated games are complex objects for players to play and for theorists to analyze. There is a huge number of possible strategies, even when the game is repeated just a few times. If k is the number of strategy profiles and n the number of iterations, k^n will be the number of all possible strategy profiles for the n-repeated game (not counting possibly randomized solutions). This makes the equilibrium analysis appear, at first, unmanageable.

With regard to rational–exchange protocols we can describe two possible iterated scenarios: (1) That in which the same game[1] is repeatedly played by the same set of players; and, (2) when the same game is repeatedly played but the participants are not necessarily the same ones in every iteration. Examples of these two environments could be: (1) A fixed set of entities which interact amongst each other by repeatedly executing the same rational–exchange protocol. (2) A variable community of peers in a file exchanging peer to peer system, a set of possible nodes in an ad-hoc wireless network or a market place in which different customers exchange electronic items with specific fixed providers.

For either type of environment, the following main components can be identified for any formal analysis:

[1]The game is understood as the protocol game linked to the protocol description as specified within the formal model. Player and protocol participant are terms used indistinctly.

- Protocol participants will react to past experience and will also take into account the future impact of their current actions.
- Along the different executions, patterns of behavior such as rewarding and punishment or cooperation and threats will emerge.
- Cooperative outcomes on one-shot games usually require of external enforcement mechanisms not always available in long term interactions.

Fortunately, there are various Game Theory results which make a connection between outcomes of a *one-shot game* (game played only once) with the corresponding *repeated game* (the same game played repeatedly). We now sketch two main areas within Game Theory which will help to analyze the two possible scenarios described above and their main components. One is due to the Nobel Prize winner Robert J. Aumann[2] and it is related to *Repeated Games* in the first type of environments. The other area is due to John Maynard-Smith and George Price and it is related to *Evolutionary Stable Strategies* which can be applied to the analysis of scenarios described as type (2).

Aumann's Theory on Repeated Games

Given an n–person game G of complete information, the supergame of G denoted as G^*, represents the same game played indefinitely by the same set of players. The important questions arising are:

1. What are the Nash equilibria of G^*?,
2. What are the resulting outcomes? and,
3. Is there a connection between the one-shot game G and the corresponding supergame G^*?

Aumann ([Aumann, 1959],[Aumann, 1960],[Aumann, 1961]) was the first to provide an extensive analysis of infinitely repeated games. He formally and rigourously proved the following well known theorem Folk Theorem[3], which establishes a major and fundamental connection between cooperative behavior in the one-shot game and non–cooperative strategies in the corresponding supergame.

Theorem 10.3.1 (The Folk Theorem)**.** *The set of Nash equilibrium outcomes of the repeated game G^* is precisely the set of feasible and individual rational outcomes of the one-shot game.*

[2]Aumann was awarded the 2005 Nobel Memorial Prize in Economic Sciences.

[3]More relevant for our future analysis will be the Perfect and Strong versions of the Folk Theorem.

In Game Theory terms, an outcome is *non–cooperative* if it does not require any external enforcement mechanism to make it happen. A Nash equilibrium point is self–enforcing as it is not worth for any players to deviate from it. Thus, it does not require of any outside enforcement scheme, and so it represents non–cooperative behavior. By contrast, general feasible outcomes need of an external enforcement so they represent the *cooperative* approach. In Aumann words: "*In a sense, the repetition itself, with its possibilities of retaliation, becomes the enforcement mechanism.*"
In simple terms, Aumann shows that:

- One can succinctly analyze complex repeated scenarios by examining one-shot instances.
- That simple, natural and familiar behaviors emerge in supergames.
- How cooperation can emerge from a non–cooperative setup.

For future work and in the case of Syverson's protocol, the set of feasible and individually rational outcomes could be easily derived from Figures 3.2 and 3.8. Furthermore, the formal model will need to be extended to formalize concepts such as those just described in the previous paragraphs. The objective would be to formally prove rationality of Syverson's protocol in repeated scenarios as this would have significant implications from a security point of view.

Evolutionary Stable Strategies

Evolutionary Game Theory (EGT) is a different approach to the classic analysis of games. Instead of directly calculating properties of a game, populations of players using different strategies are simulated and a process similar to natural selection is used to determine how the population evolves. Concepts such as Evolutionary Stable Strategies (ESS) introduced by John Maynard-Smith and George R. Price ([Maynard-Smith and Price, 1973]) are the proposed refinement of the Nash equilibrium for populations of players which seem to evolve in an stable manner when adopting such strategies. All ESS represent a subset of the Nash equilibria.

Formally, a strategy is called evolutionary stable if a population of individuals homogenously playing this strategy is able to outperform and eliminate a small amount of any mutant strategy introduced into the population.

To be exact, consider an n–player game where each player P_i has a strategy space denoted by S_i. An EGT approach would be to model each P_i (now renamed agent) by a population of players. The population for agent P_i would then be

partitioned into groups $E_{i_1}, \cdots, E_{i_k}$. Individuals in group E_{i_j} would all play the same (possibly mixed) strategy from S_i. The next step, then, would be to randomly make members of all different populations play against each other. The sub–populations that perform the best would grow, and those that did not perform well would shrink. The process of randomly playing members of all populations and refining the populations based on performance would be repeated indefinitely. Ideally the evolution would converge to some stable state for each population, which would represent a (possibly mixed) best–response strategy for each agent.

For future work, and in relation to Syverson's protocol, the question to be answered would be: How will a population of individuals that repeatedly play Syverson's protocol-game evolve? Again, the answer to this question could have significant implications from a security point of view in real implementations of Syverson's scheme.

Finally, although in recent years there have been various reports and experiments on deciding best strategies in iterated scenarios when playing *static* games ([Axelrod and Hamilton, 1981]), there have not been yet the same level of deep analysis when dealing with *dynamic* games (on imperfect or incomplete information) in indefinitely repeated scenarios.

10.3.2 Rational Protocol Synthesis: Further Analytical Work

In the area of the automated design of rational protocols further analytical work is needed. Here we present several lines for research which could be directed to:

- Examine and analyze multi–party exchange problems in terms of the different topologies of the graph representing the exchanging scenarios.
- Extend the synthesis methodology to allow for the synthesis of specific protocols in terms of:
  - Number of times a token can be relayed during the protocol execution.
  - Number of times a particular entity has an active role in the protocol.
- Further transformations of other protocol design problems (rational shared secret distribution or rational multi–party computation), into optimization problems, using simple linear structures as matrices and vectors.

Part IV

Appendices

Appendix A

Principles on Game Theory

The basic theory came into being in 1944, with the classic *Theory of Games and Economic Behavior* by John von Neumann and Oskar Morgenstern [von Neumann and Morgenstern, 1944]. In general, Game Theory provides a formal modeling approach to situations in which decision makers interact with other players. It analyzes and represents such situations as games, where players choose different actions in an attempt to maximize their returns. Although some Game Theoretic analysis appears similar to Decision Theory, Game Theory studies the decisions made in environments where players interact. In other words, Game Theory studies choice of optimal behavior when costs and benefits of each option depend upon the choices of other individuals.

We will now introduce a few basic related concepts and some mathematical notation. Since we cannot cover at length any of the notions here presented, the reader is referred to [Gibbons, 1992b] and [Fudenberg and Tirole, 1991a] for excellent Game Theory tutorial books.

A.1 Basic Concepts

Informally, a game is considered to be a collection of players who play different moves (*legal* moves will be those which comply with the game rules), aiming at maximizing their individual payoff obtained when the game is ended. Usually, players are able to conform different game strategies and following one or another strategy will dictate which move to make at each given turn in the game.

In a *static* game, players make their moves simultaneously. By contrast, in a *dynamic* game, players take it in turns to move, so their actions may depend on what actions other players have taken in previous turns.

The following definitions compose the basic Game Theory.

Definition A.1.1 (Pure strategy)**.** *In a game, a pure strategy for player P_i, denoted by s_i, is a complete contingency plan for player P_i. It describes the series of actions that this player would take at each possible decision point in the game. We consider $s_i \in S_i$, where S_i represents the set of all possible strategies for player P_i.*

Definition A.1.2 (Strategy profile)**.** *A strategy profile is a vector of strategies $(s_1, \ldots, s_n)$, one for each player P_i of a game. The set of strategy profiles, denoted by S, is the Cartesian product of the strategy spaces of all players:*

$$S = S_1 \times \cdots \times S_n \tag{A.1}$$

A convention is to describe as s_{-i} the strategies chosen by all other players except for a given player P_i. Any given strategy profile in a game may then be represented by the tuple (s_i, s_{-i}).

Note that, specifying a strategy profile univocally determines the outcome of a game.

Definition A.1.3 (Payoff function)**.** *In a game, a payoff function u_i (also called* utility function*) is defined for each player P_i.*

$$u_i : S \to \mathbb{R} \tag{A.2}$$

So that, for each strategy profile $(s_i, s_{-i}) \in S$, $u_i(s_i, s_{-i})$ represents the player P_i's payoff when P_i plays strategy s_i and the other players follow strategies s_{-i}.

Definition A.1.4 (Game in normal form)**.** *We denote a game in normal form as the tuple $G = \langle P, S, \overrightarrow{u} \rangle$ where:*

- *$P = \{P_1, \ldots, P_n\}$ is a set of n players,*
- *$S = S_1 \times \cdots \times S_n$ is the corresponding strategy profile space and*
- *$\overrightarrow{u} = (u_1, \ldots, u_n)$ is a vector of payoff functions, one for each player P_i.*

A.1.1 Game Equilibrium

A Nash equilibrium, named after John Nash[1], is a strategy profile $s^* = (s_i^*, s_{-i}^*)$ in which no player has an incentive to unilaterally modify her strategy. In other words, given the other players' strategies s_{-i}^*, player P_i cannot increase her payoff by choosing a strategy different from s_i^*. Next is the formal definition for such a concept:

[1] In 1978 John Forbes Nash was awarded the John Von Neumann Theory Prize for his invention of non–cooperative equilibria, now called Nash equilibria. In 1994 he received the Nobel Prize in Economics as a result of his work in Game Theory as a Princeton graduate student.

Definition A.1.5 (Nash equilibrium)**.** *Given a game* $G = \langle P, S, \overrightarrow{u} \rangle$ *, a strategy profile* $s^* \in S$*, represents a Nash equilibrium if and only if, for every player* P_i*,* $i \in \{1, \ldots, n\}$*:*

$$u_i(s_i^*, s_{-i}^*) \geqslant u_i(s_i, s_{-i}^*) \quad \forall s_i \in S_i. \tag{A.3}$$

A Nash equilibrium is said to represent a solution for a given game. Moreover, once such a state is identified, rational (self-interested) players are *forced* to follow the set of strategies to reach such an outcome as by definition, changing their strategy unilaterally will not result in better payoffs.

A.1.2 Dominated and Dominant Strategies

There are cases when it is possible to anticipate the strategy that a rational player will follow during a protocol game execution. By contrast, in other instances, we might be able to identify those strategies which will never be followed by any rational participant. For either case we present the following concepts:

Definition A.1.6 (Dominated strategies)**.** *A strategy for which the expected payoff is lower than the one obtained following any other option is called dominated strategy.*

Definition A.1.7 (Dominant strategies)**.** *A dominant strategy is such that, a rational player will always choose to follow it, as the expected payoff by doing so is greater than by taking a different move.*

Note that, dominated strategies can be eliminated when computing a game Nash equilibrium as rational (self-interested) players will never follow them. By contrast, dominant strategies are key when trying to establish the equilibrium of any given game.

A.1.3 Mixed Strategies

A randomized or mixed strategy for a player P_i is a strategy which chooses randomly between different strategies at each step of the game.

Definition A.1.8 (Mixed strategy)**.** *Given a game* $G = \langle P, S, \overrightarrow{u} \rangle$ *where* $S_i = \{s_{i1}, \ldots, s_{iK_i}\}$ *is the space of strategies for player* P_i *then, a mixed strategy for player* P_i *is a probability distribution function over* S_i *denoted as* $p_i = (p_{i1}, \ldots, p_{iK_i})$*, where* $0 \leqslant p_{ik} \leqslant 1 \ \forall k = 1, \ldots, K_i$ *and* $\sum_{k=1}^{K_i} p_{ik} = 1$*.*

Note that, as described in A.1.1, a pure strategy is a mixed strategy in which the probability distribution function p_i assigns probability one to one of the strategies in the strategy space and zero to every other option.

Definition A.1.9 (Probabilistic strategy profile)**.** *A probabilistic strategy profile is a vector* $(p_1^*, \ldots, p_n^*)$ *where for each player* P_i*,* p_i *is a probability distribution over the space of strategies* S_i*.*

By definition A.1.5, a Nash equilibrium represented by a strategy profile $s^* \in S$ guarantees that, for every player P_i, $i \in \{1, \ldots, n\}$, the pure strategy s_i^* is the best response from player P_i to all other player's pure strategies s_{-i}^*. Likewise, a Nash equilibrium on mixed strategies guarantees that a mixed strategy p_i^* is the best response from player P_i to all other player's mixed strategies p_{-i}^*. This is illustrated in the next definition:

Definition A.1.10 (Nash equilibrium on mixed strategies)**.** *Given a game* $G = \langle P, S, \overrightarrow{u} \rangle$*, a probabilistic strategy profile* $(p_1^*, \ldots, p_n^*)$ *represents a Nash equilibrium if and only if, for every player* P_i*,* $i \in \{1, \ldots, n\}$*,*

$$v_i(p_i^*, p_{-i}^*) \geqslant v_i(p_i, p_{-i}^*) \qquad \forall p_i \in \Delta(S_i). \tag{A.4}$$

where:

$$v_i(p_i, p_{-i}) = \sum_{j_1=1}^{K_1} * \cdots * \sum_{j_n=1}^{K_n} p_{1j_1} * \cdots * p_{nj_n} * u_i(s_{1j_1}, \ldots, s_{nj_n}) \tag{A.5}$$

and,
$\Delta(S_i)$ *denotes the space of probability distributions over the set* S_i*.*

We will conclude this section with an seminal theorem by John Nash ([Nash, 1950]) which ensures the existence of at least one Nash equilibrium point on possibly randomized strategies.

Theorem A.1.1 (Nash Equilibrium existence)**.** *In every finite normal form game* $G = \langle P, S, \overrightarrow{u} \rangle$ *with n players, where* S_i *is finite for each player* P_i*,* $i = 1, \ldots, n$ *there exists at least one mixed strategy equilibrium.*

In general, it is possible for a game to have multiple Nash equilibria.

To illustrate some of the definitions described in this section and before introducing any further concepts, in the next section of this Appendix we will present two classic games in Game Theory: *The Prisoners Dilemma* and *The Battle of Sexes.*

A.2 The Prisoner's Dilemma

The Prisoners Dilemma is a classic game studied in Game Theory since its origin in 1950 when Albert W. Tucker formalized the game with prison sentence payoffs and gave it the name of "Prisoner's Dilemma" ([Poundstone, 1992]).

	Cooperate	Defect
Cooperate	(R, R)	(S, T)
Defect	(T, S)	(P, P)

Table A.1: First component of each tuple corresponds to the sentence payoff for prisoner A and the second component for prisoner B. The values satisfy the following inequalities: $S < P < R < T$ and $R > (T + S)/2$.

The Prisoners Dilemma shows that, in certain circumstances, if the members of a group trust each other, they can choose a course of actions that will bring the best possible outcome for each one of them. By contrast, without trust, each individual will aim at maximizing their individual outcome which can lead them to suboptimal solutions.

In the Prisoner's Dilemma two players act as prisoners who have been jointly charged of a crime (which they did commit) but questioned separately. The police only have enough evidence to be sure of a conviction for a minor offence, but not enough for the more serious crime. The criminals had previously agreed to never betray each other in the case of being arrested. The following prison sentence table (Table A.1) is presented to both criminals who then have to decide what to do:

- If one of the criminals testifies for the prosecution of the other and the other remains silent, the betrayer serves a small sentence (represented by value T, *Temptation to defect payoff*) and the silent accomplice receives a full sentence for the crime committed (represented by value S, *Sucker's payoff*).
- If both stay silent, both prisoners are sentenced but only for the minor offence (represented by value R, *Reward for mutual cooperation payoff*).
- If they both betray each other, each one receives an equal portion of the sentence (represented by value P, *Punishment for mutual defection payoff*).

A.2.1 Nash Equilibrium of the Prisoner's Dilemma Game

A Nash Equilibrium implies that neither player has an incentive to alter their strategy *unilaterally*, since all other actions will decrease their expected payoff value.

a) One-shot Prisoners Dilemma

In the standard one-shot prisoner's dilemma, the game is played only once and players have to simultaneously choose between *cooperation* or *defection*. Playing defection is a dominant strategy for both players (it reports better payoff when

players do not know what the other player is going to choose) so they are expected to play such an action and never choose cooperation. The "defect-defect" outcome is a Nash Equilibrium because neither player has an incentive to unilaterally perform cooperation; unilateral cooperation would shift the player from the third worst payoff (value P) to the very worst payoff (value S).

b)Iterated Prisoners Dilemma

If two players play Prisoner's Dilemma more than once in succession (that is, having memory of at least the previous game), this is called iterated Prisoner's Dilemma.

N-Iterated Prisoner's Dilemma: If a PD is going to be iterated exactly N times for some known constant N, then in all rounds, the optimal strategy is to defect. Therefore, $always\_defect$ represents the only possible Nash equilibrium for the N-iterated PD game.

An intuitive proof of this, based on backward induction, can be sketched as follows: players might as well defect on the last iteration since the opponent will not have a chance to punish them in consecutive runs. Besides, $defect$ is a dominant strategy when there are no more rounds to play (similar to a single–play game). Therefore, both will defect on the last turn. Thus, each player might as well defect on the second-to-last turn, since the opponent will defect on the last no matter what is done, so both will defect on the second-to-last iteration and so on. The same reasoning will take players to $defect$ in all iterations.

For cooperation to emerge in N-iterated PD games, the total number of rounds must be random, or at least unknown to the players. However, in this case although $always\_defect$ is no longer a strictly dominant strategy, it still represents a Nash equilibrium of the N-iterated game.

Indefinitely Iterated Prisoner's Dilemma: Nobel Prize winner Robert J. Aumann was the first to provide an extensive analysis of infinitely repeated games. Amongst results shown by Aumann in his 1959 paper [Aumann, 1959] is the formal proof of the well known Folk Theorem which establishes that, rational players repeatedly interacting for indefinitely long games can yield cooperative outcomes.

In 1979, Robert Axelrod (University of Michigan) hosted a tournament to see what kind of strategies would perform best over the IPD game ([Axelrod and Hamilton, 1981]). He invited a number of well-known game theorists to submit strategies to be run by computers. In the tournament,

Husband \ Wife	Opera	Football
Opera	$(1, 2)$	$(0, 0)$
Football	$(0, 0)$	$(2, 1)$

Table A.2: First component of each tuple corresponds to the husband's payoff and the second component to the wife.

programs played the PD game against each other and themselves repeatedly. At each iteration, each program would specify whether to cooperate or defect based on its opponent's previous moves and the predefined strategy. Some of the strategies submitted were: $always\_defect$, $always\_cooperate$ and $random\_defect$ (this strategy defects 50% of the time).

The winner of Axelrod's tournament was the *TIT FOR TAT* strategy. The strategy cooperates on the first move, and then does whatever its opponent has done on the previous move. Thus, when matched against the all-defect strategy, TIT FOR TAT strategy always defects after the first move. When matched against the all-cooperate strategy, TIT FOR TAT always cooperates. This strategy has the benefit of both cooperating with a friendly opponent, getting the full benefits of cooperation, and of defecting when matched against an opponent who defects. When matched against itself, the TIT FOR TAT strategy always cooperates.

A.3 The Battle of Sexes

We have just described a game (The Prisoner's Dilemma Game) for which a Nash equilibrium is found when entities play only pure strategies (see Definition A.1.1). We will now describe another simple game in which the Nash equilibrium computed is reached when players play mixed or randomized strategies (see Definition A.1.8).

The Battle of the Sexes is a two player coordination game played by a couple, husband and wife. The husband would most prefer going to a football match whereas the wife would like to go to the opera. Both prefer attending together to one of the events rather than going to different ones. If they cannot communicate, where should they go?

The payoff matrix (Table A.2) shows the corresponding payoff values for each possible outcome.

A.3.1 Nash Equilibrium of the Battle of Sexes

In this game, neither player has a dominant strategy. Husband and wife have no other option than to form conjectures about what the other will choose. Let p be the probability that the wife assigns to the event of him, the husband, choosing *opera* and $(1-p)$ the probability assigned to the likelihood of the husband choosing $football$. Likewise, the husband also conjectures about his wife's most probable action and assigns probability q to the event of his wife choosing *opera* and $(1-q)$ to the event of his wife choosing $football$.

Both espouses compute their expected payoff values taken into account the aforementioned conjectures. This way, the husband performs the following calculations, where EP stands for $Expected\ Payoff$:

$$\begin{aligned} EP(husband, opera) &= q \\ EP(husband, football) &= 2 * (1-q) \end{aligned} \tag{A.6}$$

Then:

$$\begin{aligned} EP(husband, opera) > EP(husband, football) \Leftrightarrow \\ q > 2 * (1-q) \Leftrightarrow q > (2/3) \end{aligned} \tag{A.7}$$

Likewise, the wife would compute the following $Expected\ Payoff$ values:

$$\begin{aligned} EP(wife, opera) &= 2 * p \\ EP(wife, football) &= (1-p) \end{aligned} \tag{A.8}$$

Then:

$$\begin{aligned} EP(wife, opera) > EP(wife, football) \Leftrightarrow \\ 2 * p > (1-p) \Leftrightarrow p > (1/3) \end{aligned} \tag{A.9}$$

Graphical representation of inequations A.7 and A.9 enables us to compute the Nash equilibrium points as the intersection points between the two functions. See Figure Fig. A.1.

This game has two pure strategy Nash equilibria, one where both go to the opera and another where both go to the football game. There is also a Nash equilibrium in mixed strategies, where the players go to their preferred event more often than to the other (each player attends their preferred event with probability 2/3). This presents an interesting case since the two pure strategy Nash equilibria are unfair, one player consistently does better than the other, the mixed strategy Nash equilibrium represents the most preferable solution for the game.

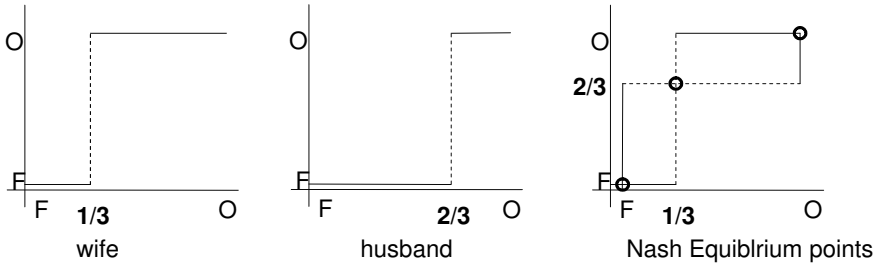


Figure A.1: Best–response function for the wife (left), husband (center), and the intersection of both (right).

A.4 Dynamic Games of Imperfect Information

In this section we will formalize the concept of dynamic games of imperfect information. In simple terms, a dynamic sequential game is one of imperfect information if a player does not know exactly what actions other players took up to that point in the game. Intuitively, if it is my turn to move, I may not know what every other player has done up to the current point.

Note that, some of the following definitions are only and extension of those presented in Section A.1.

Definition A.4.1 (Game in extensive–form)**.** *An* extensive–form *of a dynamic game of imperfect information is defined by the tuple:*

$$\langle P, A, Q, p, (\mathcal{I}_i)_{i\in P}, (\preceq_i)_{i\in P}\rangle \tag{A.10}$$

where:

- P *is a set of players.*
- A *is a set of actions.*
- Q *is a set of action sequences, satisfying:*
  - $\epsilon \in Q$*, where* ϵ *is the empty sequence.*
  - *if* $(a_k)_{k=1}^{w} \in Q$ *and* $0 < v < w$*, then* $(a_k)_{k=1}^{v} \in Q$
  - *if* $(a_k)_{k=1}^{v} \in Q\ \forall v \geqslant 1$*, then* $(a_k)_{k=1}^{\infty} \in Q$

  If q *is a sequence of actions and* a *is an action, then* $q \cdot a$ *denotes the action composed by* q *followed by* a*. A finite sequence of actions* $q \in Q$ *is said to be terminal if there is no* a *such that* $q \cdot a \in Q$*. The set of terminal sequences of*

actions is denoted by Z. Finally, $A(q) = \{a \in A : q \cdot a \in Q\}$ denotes the set of available actions after $q \in Q \backslash Z$.

- *p is the player function. It assigns a player $p(q) \in P$ to every non–terminal sequence $q \in Q \backslash Z$. The interpretation is that player $p(q)$ has the turn after the sequence of actions q.*
- *$\mathcal{I}_i$ is an information partition for player $P_i \in P$. It is a partition of the set $\{q \in Q \backslash Z : p(q) = i\}$ preserving the property that if the sequences q and q' are in the same information set $I_i \in \mathcal{I}_i$, then $A(q) = A(q')$.*
- *$\preceq_i$ is a preference relation of player $P_i \in P$ on Z.*

The common interpretation of an extensive–form game is the following. The game can be thought of as a tree, where the edges and the vertices are associated to actions and sequences of actions, respectively. The empty sequence ϵ represents the root of the tree. The game begins at ϵ and ends at a terminal node. After any non–terminal sequence of actions $q \in Q \backslash Z$, the player given by $p(q)$ chooses an available action from the set $A(q)$. Next, q is extended with a, and the current history of the game becomes $q \cdot a$. Terminal vertices are those that cannot be followed by any other action. When a sequence of actions q reaches a terminal vertex, the game ends.

The sequences $q \in Z$ are the possible outcomes of the game. The preference relations $\preceq_i$ establishes which outcomes are preferred by player P_i. Thus, if $q, q' \in Z$ and $q \preceq_i q'$, then player P_i prefers q' to q.

The most usual form of representing preference relations are payoffs. A vector $y(q) = (y_i(q))_{i \in P}$ of real numbers is assigned to every terminal sequence of actions $q \in Z$, in such a way that $q \preceq_i q' \Leftrightarrow y_i(q) \leqslant y_i(q')$. The value $y_i(q)$ can be interpreted as a measure of how much player P_i gains when the game is developed as described by q.

Information sets for players are defined in terms of their local state. Formally, $\Sigma_i(q)$ denotes the information that player P_i has obtained after the sequence of actions q.

Information sets represent the information available to players at every stage of the game. When an information set covers several nodes, then the player does not know in which node of the information set she is –or, equivalently, she does not know the last action of her rival. Usually, nodes belonging to the same information set are graphically represented by a dashed line that links them together. When an information set is not a singleton (i.e. it has more than one node), it is necessary to specify the player beliefs. Formally, beliefs are represented by a probability distribution over the nodes belonging to the information sets.

If there exists at least one information set $I_i \in \mathcal{I}_i$ such that $|I_i| > 1$, then the game is called a game of *imperfect* information. On the contrary, if for all players every information set is a singleton, then the game is called a game of *perfect* information.

A.4.1 Nash Equilibrium in Games of Imperfect Information

Definitions for player *Strategy*, *Strategy Profile* and *Probabilistic Strategy Profile* described in Section A.1 are still applicable to this type of games. As for the notion of Nash equilibrium, a new definition is provided which is a refinement of the concept previously described (see Definition A.1.10) to be used in dynamic games. The new equilibrium is referred to as a Nash equilibrium *perfect in subgames*.

Definition A.4.2 (Subgames)**.** *Given G a dynamic game of imperfect information, a subgame of G is any part of the game satisfying:*

- *The initial node is in a singleton information set.*
- *The subgame contains all the nodes that are successors of the initial node.*
- *It contains all the nodes that are successors of any node it contains.*
- *If a node of a particular information set is in the subgame then, all members of that information set belong to the subgame.*

Definition A.4.3 (Nash equilibrium perfect in subgames)**.** *Given G a dynamic game of imperfect information, a probabilistic strategy profile is a subgame perfect Nash equilibrium if it represents a Nash equilibrium of every subgame of the original game G.*

More informally, this means that if players played any smaller game that consisted of only one part of the larger game and their behavior represents a Nash equilibrium of that smaller game.

A.5 Dynamic Games of Incomplete Information or Bayesian Games

In a Bayesian game, each player is allowed to have some private information that affects the overall game but which is not known by others. This information is usually related to their payoff values (what players receive at the end of the game, depending on what strategies all players play). In the following, we briefly introduce the two most relevant concepts in Bayesian games: player's types and player's beliefs.

Subsequently, we discuss the notion of perfect Bayesian equilibrium –a generalization of Nash equilibrium for this kind of games.

A.5.1 Player's Type

Following John C. Harsanyi's framework, a way of modeling uncertainty in a game is by introducing the notion of a player's type [Gibbons, 1992a]. The type of a player determines univocally that player's payoff function, being perfectly possible that different types will be associated with different payoff functions.

The following definitions formalize this concept:

Definition A.5.1 (Player's type and type space)**.** *We will assume that each player $P_i \in P$ has a type $T_i \in \mathcal{T}_i$, where $\mathcal{T}_i$ is the type space for player P_i.*

Definition A.5.2 (Type profile)**.** *A type profile is a tuple of types $T = (T_1, \ldots, T_n)$, one for each player, which univocally determines the type of every player involved in a specific game. We denote by $\mathcal{T} = \mathcal{T}_1 \times \cdots \times \mathcal{T}_n$ the type–profile space.*

Note that games of incomplete information are not limited to a discrete type-space profile. In fact, we can conceive continuous type spaces of the general form $\mathcal{T}_i \subseteq \mathbb{R}$. We can for example assign a type $T_i \in [0, 1]$ to each player P_i, which can be viewed as her *reputation* factor.

A.5.2 Player's Belief System

In a Bayesian game, the incompleteness of information means that each player does not know the type of the rest of the players with complete certainty. As a result, players have initial beliefs about the type of each player, and can update them according to Bayes' Rule as the game advances and more information is available (e.g. beliefs about a player can change on the basis of the actions she have played).

The next definition formalizes this notion.

Definition A.5.3 (Belief system)**.** *The belief that a player has about the type of player $P_j \in P$ is represented by a probability distribution over P_j's type-space $\Delta(\mathcal{T}_j)$. In general, we will denote each belief by a Greek letter $\alpha(\cdot), \beta(\cdot), \ldots$ The set of all beliefs will be termed ρ.*

($\Delta(X)$ denotes the space of probability distributions over the set X).

In practice, assignment of types to players is carried out by introducing a fictitious player: Nature. In the course of the game, nature randomly chooses a type for each player according to a probability distribution over each player's type space. A typical scenario in a dynamic game of incomplete information is

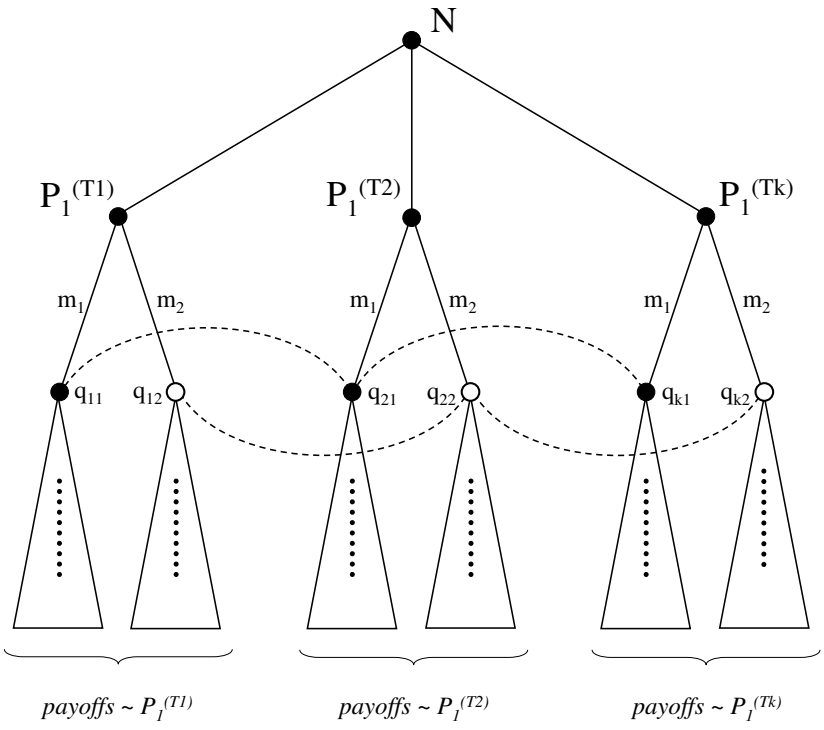


Figure A.2: Illustration of a game of incomplete information.

graphically illustrated by Fig. A.2, in which $P_i^{T_j}$ means that player P_i has type T_j. Initially, nature (N) "moves", thus selecting a type for player P_1, who moves either m_1 or m_2. Next, it is the turn for player P_2. Despite P_2 has observed the move performed by P_1, she is not sure about P_1's type. Formally, $I_1 = \{q_{11}, q_{21}, \ldots, q_{k1}\}$ and $I_2 = \{q_{12}, q_{22}, \ldots, q_{k2}\}$ form disjoint information sets for P_2, since she knows whether she is in I_1 or I_2, but she does not know the specific node.

From this point on, P_2 will have to include into her analysis a belief system, i.e. the probability of P_1 being of type T_1, T_2, etc. Therefore, strategies must take into account not only the several ways in which other players can play, but also the probability for each player to be of a specific type.

A.5.3 Perfect Bayesian Equilibrium

In 1991, Drew Fudenberg and Jean Tirole formally defined the perfect Bayesian equilibrium (PBE) for extensive Bayesian games [Fudenberg and Tirole, 1991b]. PBE adds to Nash equilibrium the requirement that players choose optimally given their beliefs about the rest of the game. In extensive Bayesian games of incomplete information, each player is not only aware of the informational uncertainties over

the other participants, but also analyzes their implications. Thus, each player looks for the best response, anticipating other party's reaction.

Perfect Bayesian Equilibrium Candidates

The following will formally define candidates to be a PBE.

Definition A.5.4 (Perfect bayesian equilibrium candidates)**.** *In an extensive–form Bayesian game the tuple strategy-believe profile* $(s;\rho)$ *constitutes a PBE candidate if:*

- *The profile* $(s;\rho)$ *does not only represent a Bayesian equilibrium of the whole game, but also in each of the continuation subgames. This is, from each information set, the moving player's strategy maximizes its expected payoff for the remainder of the game, considering its beliefs and all player's strategies.*

- *On the equilibrium path, Bayes' rule and equilibrium strategies determine believes. An information set is on-the-equilibrium path if, it is reached with positive probability iff the game is played according to the equilibrium strategies.*

- *Off-the-equilibrium path, Bayes' rule and equilibrium strategies determine beliefs where possible. A defection from the equilibrium path does not imply that other players can increase their payoffs by unilaterally changing their strategy.*

The stated $(s;\rho)$ strategy-belief profile would describe a vector of strategies such that for every player $P_i \in P$ and every information set $I_i \in \mathcal{I}_i$, player's P_i strategy is her best response, given her believes at set $\mathcal{I}_i$.

Before formally defining the concept of Bayesian perfect equilibrium, a series of requirements are necessary [Gibbons, 1992a].

Perfect Bayesian Equilibrium Requirements

Definition A.5.5 (Bayes requirement 1)**.** *Given a strategy profile* s*, it is required that, for each player* $P_i \in P$*, and at each of her information sets* $I_i \in \mathcal{I}_i$*, player* P_i *has beliefs* $\rho(I_i) \in \Delta(I_i)$ *about the node at which she is located, conditional upon being informed that play has reached the information set* I_i*.*

This requirement establishes that at every node of any information set, a player should have some beliefs about the node at which she is located, given that she has reached that information set. In fact, the beliefs $\rho(I_i) \in \Delta(I_i)$ are no more than a probability distribution over the nodes in I_i.

Definition A.5.6 (Bayes requirement 2)**.** *Let us suppose the continuation game defined by the information set $I_i \in \mathcal{I}_i$ of some player P_i, and the conditional beliefs $\rho_i(I_i)$. The restriction of the strategy-belief profile $(s;\rho)$ to this game must be a Nash equilibrium of the continuation game starting at information set I_i.*

The concept of perfect equilibrium of the continuation game adds to the Nash concept the requirement that players choose optimally in continuation games. More generally, Bayes requirement 2 rejects all strategy profiles which specify at any information set an action which is dominated at that information set.

Definition A.5.7 (Bayes requirement 3)**.** *Beliefs at any on the equilibrium path information sets must be determined from the strategy profile according to Bayes' rule. This is, if $I_i \in \mathcal{I}_i$ is an information set of player P_i reached with positive probability when players follow strategy profile s, then $\rho(I_i) \in \Delta(I_i)$ must be computed from S according to Bayes' rule.*

Definition A.5.8 (Bayes requirement 4)**.** *The beliefs at any off the equilibrium path information set must be determined from the strategy profile according to Bayes rule whenever possible.*

This requirement establishes that a defection from the equilibrium path does not increase the chance that others will play irrationally. In a PBE, players cannot threaten to play strategies that are strictly dominated beginning at any information set, off the equilibrium path.

Perfect Bayesian Equilibrium

Definition A.5.9 (Perfect bayesian equilibrium)**.** *Given a strategy profile s and a set of beliefs ρ, then the strategy-belief profile $(s;\rho)$ forms a perfect Bayesian equilibrium iff it satisfies Bayes requirements 1 to 4.*

Bibliography

[Abadi et al., 2002] Abadi, M., Glew, N., Horne, B., and Pinkas, B. (2002). Certified email with a light on-line trusted third party: Design and implementation. In *Proceedings of 2002 International World Wide Web Conference*, pages 387–395.

[Abadi and Gordon, 1997] Abadi, M. and Gordon, A. D. (1997). A calculus for cryptographic protocols: the spi salculus. Technical Report Technical report 414, University of Cambridge, Computer Laboratory, Cambridge, UK.

[Abraham et al., 2006] Abraham, I., Dolev, D., Gonen, R., and Halpern, J. (2006). Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In *PODC06: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 53–62.

[Alcaide et al., 2005] Alcaide, A., Estévez-Tapiador, J., Izquierdo, A., and Sierra, J. (2005). A formal analysis of fairness and non–repudiation in the rsa-cegd protocol. In *Proceedings of the Computational Science and Its Applications ICCSA 2005*, pages 1309–1318.

[Amadio et al., 2002] Amadio, R., Lugiez, D., and Vanackere, V. (2002). On name generation and set-based analysis in the dolev-yao model. Technical Report 4379, RR-INRIA.

[Ambainis et al., 2004] Ambainis, A., Jakobsson, M., and Lipmaa, H. (2004). Cryptographic randomized response techniques. In *DIMACS/PORTIA Workshop on Privacy-Preserving Data Mining*.

[Asokan et al., 1997] Asokan, N., Schunter, M., and Waidner, M. (1997). Optimistic protocols for fair exchange. In *4th ACM Conference on Computer and Communications Security*, pages 8–17.

[Asokan et al., 1998] Asokan, N., Shoup, V., and Waidner, M. (1998). Asynchronous protocols for optimistic fair exchange. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 86–99.

[Atallah et al., 2004] Atallah, M. J., Bykova, M. V., Li, J., Frikken, K. B., and Topkara, M. (2004). Private collaborative forecasting and benchmarking. In *3rd annual Workshop on Privacy in the Electronic Society (WPES 04)*.

[Aumann, 1959] Aumann, R. J. (1959). Acceptable points in general cooperative n-person games. *Contributions to the Theory of Games IV, Annals of Mathematics Study*, 40:287–324. Princeton University Press, Princeton, NJ.

[Aumann, 1960] Aumann, R. J. (1960). Acceptable points in games of perfect information. *Pacific Journal of Mathematics*, 10:381–417.

[Aumann, 1961] Aumann, R. J. (1961). The core of a cooperative game without side payments. *Transactions of the American Mathematical Society*, 98:539–552.

[Axelrod and Hamilton, 1981] Axelrod, R. and Hamilton, W. D. (1981). The evolution of cooperation. *Science*, 27(211):1390–1396.

[Bahreman and Tygar, 1994] Bahreman, A. and Tygar, J. (1994). Certified electronic mail. In *Proceedings of 1994 Symposium on Network and Distributed System Security*, pages 3–19.

[Bezáková and Dani, 2005] Bezáková, I. and Dani, V. (2005). Allocating indivisible goods. *ACM SIGecom Exchanges*, 5(3):11–18.

[Bicakci and Baykal, 2003] Bicakci, K. and Baykal, N. (2003). One-time passwords: Security analysis using ban logic and integrating with smartcard authentication. In *Proceedings Computer and Information Sciences - (ISCIS 2003)*, pages 794–801.

[Bogetoft et al., 2006] Bogetoft, P., Damgard, I., Jakobsen, T., Nielsen, K., Pagter, J., and Toft, T. (2006). A practical implementation of secure auctions based on multiparty integer computation. In *Proceedings Financial Cryptography*. LNCS Vol. 4107, pp. 142-147.

[Buragohain et al., 2003] Buragohain, C., Agrawal, D., and Suri, S. (2003). A game theoretic framework for incentives in p2p systems. In *Proceedings of the 3rd Int. Conf. on Peer-to-Peer Computing*. Linkping, Sweden, IEEE Computer Society (2003) 48–56.

[Burrows et al., 1990] Burrows, M., Abadi, M., and Needham, R. (1990). A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36.

[Buttyán, 2001] Buttyán, L. (2001). Building blocks for secure services: Authenticated key transport and rational exchange protocols. Technical report, Swiss Federal Institute of Technology. Lausanne (EPFL). Ph.D. Thesis No. 2511.

[Buttyán and Hubaux, 2001] Buttyán, L. and Hubaux, J. (2001). Rational exchange– a formal model based on game theory. In *Proceedings 2nd International Workshop on Electronic Commerce*. Springer-Verlag. LNCS Vol. 2232, pp. 114.

[Buttyán and Hubaux, 2004] Buttyán, L. and Hubaux, J. (2004). A formal model of rational exchange and its application to the analysis of syverson's protocol. *Journal of Computer Security*, 12(3/4):551–588.

[Buttyán et al., 2002] Buttyán, L., Hubaux, J., and Capkun, S. (2002). A formal analysis of syverson's rational exchange protocol. In *Proceedings 15th IEEE Computer Security Foundations Workshop (CSFW15)*.

[Chen et al., 2004] Chen, H., Clark, J., and Jacob, J. (2004). Automatic design of security protocols. *Computational Intelligence*, 20(3):503–516. Special Issue on Evolutionary Computing in Cryptography and Security.

[Chen et al., 2005] Chen, H., Clark, J., and Jacob, J. (2005). Synthesising efficient and effective security protocols. *Electronic Notes in Theoretical Computer Science*, pages 25–41.

[Chevalier et al., 2003] Chevalier, Y., Kuester, R., Rusinowitch, M., and Turuani, M. (2003). An np decision procedure for protocol insecurity with xor. In *Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science*.

[Clark and Jacob, 2000] Clark, J. and Jacob, J. (2000). Searching for a solution: engineering tradeoffs and the evolution of provably secure protocols. In *Proceedings IEEE Symposium on Security and Privacy*, pages 82–95.

[Clark and Jacob, 2001] Clark, J. and Jacob, J. (2001). Protocols are programs too: the meta-heuristic search for security protocols. *Information and Software Technology*, 43(14):891–904.

[Conitzer and Sandholm, 2004] Conitzer, V. and Sandholm, T. (2004). New complexity results about nash equilibria. Technical report, Game Theory Society (GAMES-04). Oral presentation.

[Cramer et al., 2001] Cramer, R., Damgard, I., and Nielsen, J. (2001). Multiparty computation from threshold homomorphic encryption. In *Advances in CryptologyEurocrypt 2001*, volume 2045, pages 280–300.

[de Werra et al., 1995] de Werra, D., Hertz, A., and Taillard, E. (1995). A tutorial on tabu search. In *Proc. of Giornate fi Lavaro AIRO'95*, pages 13–24.

[Delaune, 2006] Delaune, S. (2006). An undecidability result for agh. *Theoretical computer science*, 368(1):161–167.

[Dennin, 1999] Dennin, D. (1999). The limits of formal security models. National Computer System Security Award Acceptance Speech.

[Dolev and Yao, 1983] Dolev, D. and Yao, A. (1983). On the security of public-key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208.

[Durgin et al., 1999] Durgin, N., Lincoln, P., Mitchell, J., and Scedrov, A. (1999). Undecidability of bounded security protocols. In *Proc. Workshop on Formal Methods and Security Protocols (FMSE99)*.

[Estévez-Tapiador et al., 2007a] Estévez-Tapiador, J.M., Clark, J., and Hernández Castro, J. (2007a). Non-linear cryptanalysis revisited: Heuristic search for approximations to s-boxes. In *Proceedings of the Eleventh IMA International Conference on Cryptography and Coding*, pages 99–117.

[Estévez-Tapiador et al., 2007b] Estévez-Tapiador, J., Hernández Castro, J., and Clark, J. (2007b). Heuristic search for non-linear cryptanalytic approximations. In *IEEE Congress on Evolutionary Computation, CEC 2007.*, pages 3561–3568.

[Even and Goldreich, 1983] Even, S. and Goldreich, O. (1983). On the security of multi-party ping-pong protocols. In *Proc. 24th Annual Symposium on Foundations of Computer Science*, pages 34–39.

[Franklin and Tsudik, 1998] Franklin, M. K. and Tsudik, G. (1998). Secure group barter: Multi-party fair exchange with semi-trusted neutral parties. In *Financial Cryptography*, pages 90–102.

[Fudenberg and Tirole, 1991a] Fudenberg, D. and Tirole, J. (1991a). *Game Theory*. MIT Press. ISBN 13: 978-0-262-06141-4.

[Fudenberg and Tirole, 1991b] Fudenberg, D. and Tirole, J. (1991b). Perfect bayesian equilibrium and sequential equilibrium. *Journal of Economic Theory*, 53(2):236–260.

[Gibbons, 1992a] Gibbons, R. (1992a). *Game Theory for Applied Economists*. Princeton University Press.

[Gibbons, 1992b] Gibbons, R. (1992b). *A Primer in Game Theory*. Harvester-Wheatsheaf. ISBN 13: 978-0-74501-159-2.

[Gilboa and Zemel, 1989] Gilboa, I. and Zemel, E. (1989). Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1:80–93.

[Gilpin et al., 2007] Gilpin, A., Hoda, S., Pea, J., and Sandholm, T. (2007). Gradient based algorithms for finding nash equilibria in extensive form games. In *In Workshop on Internet and Network Economics (WINE)*.

[Gilpin and Sandholm, 2007] Gilpin, A. and Sandholm, T. (2007). Lossless abstraction of imperfect information games. *Journal of the ACM*, 54(5).

[Glover, 1987] Glover, F. (1987). Tabu search methods in artificial intelligence and operations research. *ORSA Artificial Intelligence*, 1(2).

[Glover, 1990] Glover, F. (1990). Tabu search: A tutorial. *Special Issue on the Practice of Mathematical Programming*, 20(1):74–94.

[Goldberg, 1989] Goldberg, D. (1989). *Genetic Algorithms in Search, optimization and Machine Learning*. Addison-Wesley.

[Golle et al., 2001] Golle, P., Leyton-Brown, K., and Mironov, I. (2001). Incentives for sharing in peer-to-peer networks. In *Proceedings of the Conference on Electronic Commerce*, pages 14–17. Tampa, USA, ACM Press (2001).

[Gong, 1993] Gong, L. (1993). Variations on the themes of message freshness and replay. In *Proc. of the IEEE Computer Security Foundations Workshop VI*, pages 131–136. Franconia, New Hampshire.

[Gordon and Katz, 2006] Gordon, S. D. and Katz, J. (2006). Rational secret sharing revisited. In *Security and Cryptography for Networks*, pages 229–241. Volume 4116/2006.

[Gupta and Somani, 2005] Gupta, R. and Somani, A. (2005). Game theory as a tool to strategize as well as predict nodes behavior in peer-to-peer networks. In *Proceedings of the 11th Int. Conf. on Parallel and Distributed*, pages 244–249. Systems, Fukuoka, Japan, IEEE Computer Society (2005).

[Halpern and Teague, 2004] Halpern, J. and Teague, V. (2004). Rational secret sharing and multiparty computation: Extended abstract. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing STOC '04*. ACM 1-58113-852-0/04/0006.

[Hernández-Castro et al., 2006] Hernández-Castro, J., Estévez-Tapiador, J., Ribagorda-Garnacho, A., and Ramos-Alvarez, B. (2006). Wheedham: An

automatically designed block cipher by means of genetic programming. In *CEC 2006. IEEE Congress on Evolutionary Computation*, pages 192 – 199.

[Husdal, 2004] Husdal, J. (2004). Flexibility and robustness as options in reducing risks and uncertainties. Molde University College, Molde, Norway.

[Izmalkov et al., 2005] Izmalkov, H., Micali, S., and Lepinski, M. (2005). Rational secure computation and ideal mechanism design. In *46th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 585–595.

[Khetawat et al., 1997] Khetawat, A., Lavana, H., and Brglez, F. (1997). Collaborative workflows: A paradigm for distributed benchmarking and design on the internet. Technical Report 97–TR@CBL–02, CS Dept., North Carolina State University, Raleigh.

[Kirkpatrick et al., 1983] Kirkpatrick, S., C., G., and M., V. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

[Koller and Megiddo, 1992] Koller, D. and Megiddo, N. (1992). The complexity of two-person zero-sum games in extensive form. *Games and Economic Behavior*, 4(4):528–552.

[Kremer, 2003] Kremer, S. (2003). Formal analysis of optimistic fair exchange protocols. Technical report, Universit Libre de Bruxelles. Facult de Sciences. Ph.D. Thesis.

[Kremer et al., 2002] Kremer, S., Markowitch, O., and Zhou, J. (2002). An intensive survey of fair non-repudiation protocols. *Computer Communications*, 25(17):1606–1621.

[Kremer and Raskin, 2000] Kremer, S. and Raskin, J. (2000). A game approach to the verification of exchange protocols. In *Proceedings of the 1st Workshop on Issues in the Theory of Security*. LNCS Vol. 921, pp. 220. Springer-Verlag.

[Lamport, 1977] Lamport, L. (1977). Proving the correctness of multiprocess programs. *IEEE Trans. on Software Engineering*, 3(2).

[Lemke and Howson, 1964] Lemke, C. and Howson, J. (1964). Equilibrium points of bimatrix games. *Journal of the Society of Industrial and Applied Mathematics*, 12(413-423).

[Lipton et al., 2004] Lipton, R., Markakis, E., Mossel, E., and Saberi, A. (2004). On approximately fair allocations of indivisible goods. In *5th ACM Conference on Electronic Commerce (EC)*.

[Lowe, 1997] Lowe, G. (1997). Casper: a compiler for the analysis of security protocols. In *Simon Foley, editor, 10th Computer Security Foundations Workshop*, pages 18–30. Rockport, Massachussets, USA. IEEE Computer Society Press.

[Lysyanskaya and Triandopoulos, 2006] Lysyanskaya, A. and Triandopoulos, N. (2006). Rationality and adversarial behavior in multi-party computation. In *Advances in CryptologyCrypto 2006*, volume 4117, pages 180–197.

[Maynard-Smith and Price, 1973] Maynard-Smith, J. and Price, G. (1973). The logic of animal conflict. *Nature*, 246:15–18.

[Meadows, 1991] Meadows, C. (1991). A system for the specification and verfication of key management protocols. In *IEEE Computer Society Symposium on Research in Security and Privacy*, pages 182–197. Washington - Brussels- Tokyo. IEEE.

[Meadows, 1994] Meadows, C. (1994). Formal verification of cryptographic protocols. In *ASIACCRYPT 1994*.

[Meadows, 2003a] Meadows, C. (2003a). Formal methods for cryptographic protocol analysis: emerging issues and trends. *IEEE journal on selected areas in communications*, 21(1).

[Meadows, 2003b] Meadows, C. (2003b). What makes a cryptographic protocol secure? the evolution of requirements specification in formal cryptographic protocol analysis.

[Moran and Naor, 2006] Moran, T. and Naor, M. (2006). Polling with physical envelopes: A rigorous analysis of a human-centric protocol. In *Advances in Cryptology - EUROCRYPT 2006*.

[Nash, 1950] Nash, J. (1950). Equilibrium points in n-person games. *Proceedings of the National Academy of the USA*, 36(1):48–49.

[Needham and Schroeder, 1978] Needham, R. M. and Schroeder, M. D. (1978). Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999.

[Nenadic et al., 2004] Nenadic, A., Zhang, N., and Barton, S. (2004). A protocol for certified e-goods delivery. In *Proceedings. ITCC 2004. International Conference on Information Technology: Coding and Computing*, pages 22 – 28.

[Nielsen et al., 2007] Nielsen, J. B., Alwen, J., Cachin, C., Pereira, O., Sadeghi, A., Schoenmakers, B., Shelat, A., and Visconti, I. (2007). Summary report on

rational cryptographic protocols. Technical report, ECRYPT - European Network of Excellence in Cryptology.

[Nurmi, 2006] Nurmi, P. (2006). A bayesian framework for online reputation systems. In *Proceedings of the Advanced Int. Conf. on Telecomm.* French Caribbean, IEEE Computer Society (2006).

[Onieva et al., 2003] Onieva, J., Zhou, J., Carbonell, M., and López, J. (2003). Intermediary non-repudiation protocols. In *Proceedings of 2003 IEEE Conference on Electronic Commerce*, pages 207–214.

[Pagnia and Gärtner, 1999] Pagnia, H. and Gärtner, F. (1999). On the impossibility of fair exchange without a trusted third party. Technical report, Darmstadt University of Technology, Department of Computer Science.

[Palomar et al., 2006a] Palomar, E., Estévez-Tapiador, J., Hernández-Castro, J., and Ribagorda, A. (2006a). Certificate-based access control in pure p2p networks. In *Proceedings of the 6th International Conference on Peer-to-Peer Computing*, pages 177–184. Cambridge, UK, IEEE (2006).

[Palomar et al., 2006b] Palomar, E., Estévez-Tapiador, J., Hernández-Castro, J., and Ribagorda, A. (2006b). A protocol for secure content distribution in pure p2p networks. In *Proceedings of the 3th Int Workshop on P2P Data Management, Security and Trust*, pages 712–716. Krakow, Poland, IEEE (2006).

[Park and Hong, 2005] Park, K. and Hong, C. (2005). Cryptographic protocol design concept with genetic algorithms. In *KES (2)*, pages 483–489.

[Paulson, 1998] Paulson, L. C. (1998). The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1-2):85–128.

[Poundstone, 1992] Poundstone, W. (1992). *Prisoner's Dilemma*. Doubleday, New York.

[Roscoe, 1995] Roscoe, A. W. (1995). Modelling and verifying key-exchange protocols using csp and fdr. In *Proceedings of the 8th IEEE Computer Security Foundations Workshop*, pages 98–107. IEEE Computer Society Press.

[Roughgarden, 2005] Roughgarden, T. (2005). *Selfish routing and the price of anarchy*. MIT Press.

[Roughgarden and Tardos, 2000] Roughgarden, T. and Tardos, E. (2000). How bad is selfish routing? In *41st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 93–102.

[Rusinowitch and Turuani, 2001] Rusinowitch, M. and Turuani, M. (2001). Protocol insecurity with finite number of sessions is np-complete. In *Proceedings of the 14th IEEE Computer Security Foundations Workshop*, pages 174–190.

[Syverson, 1994] Syverson, P. (1994). A taxonomy of replay attacks. In *Proceedings of the 7th IEEE Computer Security Foundations Workshop*, pages 187–191. IEEE Computer Society Press.

[Syverson, 1998] Syverson, P. (1998). Weakly secret bit commitment: Applications to lotteries and fair exchange. In *Proceedings of the 11th IEEE Computer Security Foundations Workshop*, pages 2–13.

[Syverson and van Oorschot, 1994a] Syverson, P. and van Oorschot, P. (1994a). On unifying some cryptographic protocol logics. In *IEEE Computer Society Symposium on Research in Security and Privacy*, pages 14–28.

[Syverson and van Oorschot, 1994b] Syverson, P. and van Oorschot, P. C. (1994b). On unifying some cryptographic protocol logics. In *IEEE Symposium on Research in Security and Privacy*, pages 14–28. IEEE Computer Society Press. Oakland, CA. IEEE COmputer Society, Technical Committee on Security and Privacy.

[von Neumann and Morgenstern, 1944] von Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press. ISBN 13: 978-0-691-13061-3.

[Wang et al., 2005] Wang, H., Guo, H., Yin, J., He, Q., Lin, M., and Zhang, J. (2005). Abuse-free item exchange. In *International Conference on Computational Science and its Applications (ICCSA05)*, pages 1028–1035.

[Wenberger, 1990] Wenberger, E. D. (1990). Correlated and uncorrelated fitness landscape and how to tell the difference. *Biological Cybernetics*, 63:325–336.

[Zhang et al., 2004] Zhang, N., Shi, Q., and M., M. (2004). A unified approach to a fair document exchange system. *Journal of System and Software*, 72(1):83–96.