PhD-FSTC-2016-47
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defence held on 24/10/2016 in Luxembourg

To obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Masoud TABATABAEI
Born on 22 September 1978 in Kashmar (Iran)

GAMES AND STRATEGIES IN ANALYSIS OF SECURITY PROPERTIES

Dissertation defense committee:

Dr Peter Y.A. Ryan, dissertation supervisor
Professor, Université du Luxembourg

Dr Wojciech Jamroga, vice-chairman
Professor, Institute of Computer Science, Polish Academy of Sciences

Dr Leon Van der Torre, chairman
Professor, Université du Luxembourg

Dr Vanessa Teague
Senior Lecturer, University of Melbourne

Dr Steve Schneider
Professor, University of Surrey

Games and Strategies in Analysis of Security Properties

Masoud Tabatabaei

Supervisors:

Prof. Peter Y.A. Ryan *(University of Luxembourg)*

Prof. Wojciech Jamroga *(Institute of Computer Science, Polish Academy of Sciences)*

The author was employed at the University of Luxembourg and received support from the National Research Fund Luxembourg in the project "Games and Information Algebras in Analysis of Voting Systems” (AFR Code:5884506).

Abstract

Information security problems typically involve decision makers who choose and adjust their behaviors in the interaction with each other in order to achieve their goals. Consequently, game theoretic models can potentially be a suitable tool for better understanding the challenges that the interaction of participants in information security scenarios bring about. In this dissertation, we employ models and concepts of game theory to study a number of subjects in the field of information security.

In the first part, we take a game-theoretic approach to the matter of preventing coercion in elections. Our game models for the election involve an honest election authority that chooses between various protection methods with different levels of resistance and different implementation costs. By analysing these games, it turns out that the society is better off if the security policy is publicly announced, and the authorities commit to it.

Our focus in the second part is on the property of noninterference in information flow security. Noninterference is a property that captures confidentiality of actions executed by a given process. However, the property is hard to guarantee in realistic scenarios. We show that the security of a system can be seen as an interplay between functionality requirements and the strategies adopted by users, and based on this we propose a weaker notion of noninterference, which we call *strategic noninterference*. We also give a characterisation of strategic noninterference through unwinding relations for specific subclasses of goals and for the simplified setting where a strategy is given as

a parameter.

In the third part, we study the security of information flow based on the consequences of information leakage to the adversary. Models of information flow security commonly prevent *any* information leakage, regardless of how grave or harmless the consequences the leakage can be. Even in models where each piece of information is classified as either sensitive or insensitive, the classification is "hardwired" and given as a parameter of the analysis, rather than derived from more fundamental features of the system. We suggest that information security is not a goal in itself, but rather a means of preventing potential attackers from compromising the correct behavior of the system. To formalize this, we first show how two information flows can be compared by looking at the adversary's ability to harm the system. Then, we propose that the information flow in a system is *effectively secure* if it is as good as its idealized variant based on the classical notion of noninterference.

Finally, we shift our focus to the strategic aspect of information security in voting procedures. We argue that the notions of receipt-freeness and coercion resistance are underpinned by existence (or nonexistence) of a suitable strategy for some participants of the voting process. In order to back the argument formally, we provide logical "transcriptions" of the informal intuitions behind coercion-related properties that can be found in the existing literature. The transcriptions are formulated in the modal game logic **ATL***, well known in the area of multi-agent systems.

I dedicate this thesis

to my father, for his endless encouragement to continue my progress in life,
and to my mother who always wishes me nothing but health and happiness.

Acknowledgement

I would first like to express my sincere gratitude to my supervisors Prof. Peter Ryan and Prof. Wojtek Jamroga, for their continuous support for my PhD study. I was very lucky to have two highly knowledgeable, but also very friendly and kind supervisors by my side during my research.

I would also like to thank the rest of my defence committee: Prof. Leon Van der Torre, Dr. Vanessa Teague and Prof. Steve Schneider for their insightful comments and raising very interesting discussions during my defence.

I thank Prof. Marc Pouly who as a member of my thesis supervisory committee, significantly helped me to have regular outputs and an organized plan for my research during my studies.

I thank my friend Arash Atashpendar for helping me in proofreading the thesis and giving comments on how to improve the structure of my (not so great) writing.

I thank my friend Marjan Skrobot for helping me in re-structuring my slides for the defence, his ideas made my presentation much easier to follow.

Finally and foremost, I thank my companion Chista for being with me through all the hassles and joys we experienced together during these memorable four years of our lives.

Contents

List of Figures

Chapter 1

Introduction

Typically, most security problems by their nature involve the participation of independent decision makers, often referred to as agents or players. These agents can be individuals, pieces of software or devices that are capable of making decisions and taking actions. In the context of security problems, these agents may act selfishly, maliciously, or cooperatively. In these problems, malicious players, often referred to as the adversary or the attacker, correspond to the agents whose goals are opposed to, or not in agreement with the proper functionality of the system. These often choose their attack methods strategically and intelligently to achieve their goals. Accordingly, there is an underlying relationship between a security problem, and the motivations and strategic capabilities of the agents involved. In this regard, game theoretic models can be used in information security problems to both better understand the challenges of the problems, and to achieve more efficient defence strategies for them. Game theory provides mathematical tools for modelling multi-agent systems, where one agent's outcome depends not only on his decisions but also on those of the other players in the interaction, which is clearly the setting for most security scenarios.

One can apply game theory to the analysis of security problems on different levels. Here we mention some of the approaches in which the study of information security can benefit from game theoretic models. We should emphasise that these approaches are neither exhaustive nor disjoint.

One of the use cases of game theoretic models in the field of information security has to do with expressing and analysing security properties that by definition include notions of strategy and goal for the players involved. For instance, one such property is coercion-resistance in an e-voting system. As we will see in more detail in Chapter 2, the informal definition of coercion-resistance in the literature is often similar to the following statement: whatever strategy the coercer adopts, the voter always has a strategy to vote as she intended while appearing to comply with all the coercer's requirements. Therefore, the intuition behind the property evidently includes strategies and goals of the players. However, few works in this area have used game theoretic models to express and analyse the coercion-resistance property.

Another approach for using game theoretical models and concepts in the field of information security is to extend the security properties that do not explicitly include notions of strategy or goal, to variants which consider these notions. This approach can help us to better understand, study, and implement the information security properties in realistic situations in which agents follow their goals strategically. For instance, noninterference (and its variants) denotes an information flow security property that is meant to prevent the leakage of information about the activity of a group of high clearance agents (or *High players*) to a group of low clearance agents (or *Low players*). In general, noninterference is a very restrictive property and implementing it in a real system is very difficult, sometimes even impossible. For this reason, several weaker notions of noninterference have been proposed in the literature. For achieving a weaker yet practical notion of noninterference, one may also take into account the strategies and goals of the players. On the one hand, we can look at High players and what they consider to be the *correct* behaviour of the system (which we can call the goal of the High players or the goal of the system). When all the High players are trusted and well informed about their goal, we can expect them to follow only those possible behaviours (strategies) that are in line with their goal. We can then relax the noninterference property to prohibit information flow only in those runs of the system that follow these High players' strategies. On the other hand, we can also focus on the goal of the adversaries in an information security scenario, which can in general be stated as harming the goal of the system, i.e., preventing the system from functioning correctly. The ability of the adversaries to opt for a strategy that damages the goal of the system depends partly (but not only) on the amount of information leaked to them from the High players. We can argue that we need to prevent the leakage of only those pieces of information that increase the strategic ability of the adversary to harm the goal of the system.

Models of game theory can also be used in the field of information security, not for defining or expressing security properties, but also at the level of implementation, as a decision-making tool for choosing the right security policy among the possible options. In this sense, game theory can be seen as a means to analyse the economics of security, which is now an emerging area of study. By taking into account the incentives and the information available to the players, game theory models can help to allocate the limited resources for implementing different security measures in order to balance the perceived security risks. One example of such situations is the security of elections. For an election to function properly, several security properties have been defined. To obtain these security properties, the decision makers of the election usually have a vast number of possible measures and tools, each with a different level of security and a different cost of implementation. Moreover, they also need to make decisions about higher level strategies, like whether or not to publicly announce the details of their security measures, whether or not to allow polling before the election, and who can have access to the results of the polling, etc. Game theoretic models can shed light on these situations and help the decision makers to choose the right strategies, and also to better understand the risks and the consequences of their chosen strategies.

Game models have also been used to provide a proof system for deciding security properties. In this sense, the players and their strategies in the defined games usually do not represent the actual agents taking a role in the security scenario and their respective possible strategies. Instead, they are serving as a mathematical tool to prove the security property in question.

1.1 Aims and Objectives

In this thesis, we take a game theoretical approach to the study of coercion-resistance in e-voting, and information flow security. We can divide our aims and objectives into three main strands. Each strand can be seen as representing one of the possible approaches to applying game theoretic models in information security. For keeping this chapter more concise, we will give a more detailed introduction about each strand in their respective chapters in the thesis.

1.1.1 Strand 1: Preventing Coercion in Elections

In this strand, we use game models to study coercion in elections. We analyse the strategies of the coercer, and of the decision makers of an election in the process of deciding the security policies. We try to find a strategy for the decision makers that minimises the damages that the coercion would inflict upon what we consider to be the common good of the society.

Objective 1

To model and analyse an election system in the presence of potential coercers, in order to find optimal strategies for the decision makers of the election, and to minimise the risk and expected damage of coercion.

1.1.2 Strand 2: Information Flow Security and Strategic Abilities of Players

In this strand, we aim to explore new perspectives on information flow security by taking into account the strategic abilities and goals of the players involved. We consider two approaches: The first one is to focus on the goals of the High players and their strategic abilities to achieve these goals. The other one is to emphasise the strategic abilities of the Low players to harm the goal of the system.

Objective 2.a

To define a weaker notion of noninterference by considering the strategic ability of the High players and their incentive to ensure the correct behaviour of the system.

Objective 2.b

To study the significance of information flow in a system based on its influence on increasing the strategic ability of the adversary to harm the goal of the system.

1.1.3 Strand 3: Expressing Receipt-Freeness and Coercion-Resistance in Logics of Strategic Ability

In this part, we study the intuitions behind various definitions of receipt-freeness and coercion-resistance properties in an election. Specifically, we would argue that these properties are usually defined based on the existence (or nonexistence) of some suitable strategy for the participants in the election. Therefore, game theoretic models seem appropriate for expressing and analysing these properties.

Objective 3

To use logics of strategic ability to provide "transcriptions" of informal intuitions behind the definitions of receipt-freeness and coercion-resistance properties in the literature.

1.2 Contributions

The first contribution of this dissertation addresses *Objective 1* of the thesis. We present game models for an election in the presence of a coercer, both in complete and incomplete information settings. We study the models based on different solution concepts: Nash equilibrium, Stackelberg equilibrium and minmax. The results of this study show that in general, announcing the security methods publicly before the election and committing to following them are beneficial for minimising the risk and expected damage of coercion.

The second contribution of this dissertation follows *Objective 2.a*. In this part, we show that the security of a system can be seen as an interplay between the functionality requirements of the system and the strategies adopted by the players. We then define a security property called strategic noninterference, which weakens the standard notions of noninterference by taking into account only those runs of the system that are in line with the strategies of the High players, assuming that they intend to ensure the correctness of the system. We also give a characterization of strategic noninterference through unwinding relations for some subclasses of goals and for the simplified setting where the strategy of the High players is provided as a parameter.

The third contribution addresses *Objective 2.b*. We suggest that information security is not a goal in itself, but rather a means of preventing potential attackers from compromising the correct behaviour of the system. To formalise this, we first show how two information flows can be compared by looking at the adversary's ability to harm the system. Then, we propose that the information flow in a system is effectively secure if it does not allow for more damage than its idealised variant based on the classical notion of noninterference. This contribution includes a definition of *the noninterferent idealised variant* of the system and its uniqueness theorem.

The fourth contribution of the thesis aims at exploring *Objective 3*. We argue that the notions of receipt-freeness and coercion-resistance are underpinned by existence (or nonexistence) of a suitable strategy for some participants of the voting process. To back the argument formally, we provide logical "transcriptions" of the informal intuitions behind the definitions of coercion-resistance properties in the literature. The transcriptions are formulated in the modal game logic **ATL***, well known in the area of multi-agent systems.

1.3 Outline

We have structured this dissertation in nine chapters. In the following, we outline the contents of each chapter.

Chapter 2: On Formal Definitions of Receipt-Freeness and Coercion-Resistance. This chapter serves as a partial literature review for strands 1 and 3 of the thesis. We first give an overview of some of the standard methodologies for obtaining proofs for security properties. Next, we present a brief literature review on the formal definitions of receipt-freeness and coercion-resistance properties. Then we take a closer look at some of the significant works that have given formal definitions of these properties.

Chapter 3: Towards Game-Theoretic Analysis of Coercion Resistance. This chapter serves as a preliminary for Objective 1 of the thesis. We explain normal form representation of games and some of the important solution concepts for them. We then take a look at incomplete information games and one of their important solution concepts, the Bayesian Nash equilibrium.

Chapter 4: Preventing Coercion in Elections, a Game Theoretic Approach. This is the main chapter related to Objective 1 of the thesis. First, we take a look at related works. Then, we present a game model in normal form for an election in the presence of coercion. The players in the game consist of an "honest election authority" who acts on behalf of society, and a coercer who may try to change the result of the election by coercing the voters. We analyse the games using different solution concepts for normal form games. Furthermore, we extend our model to incorporate an incomplete information setting where players are not certain about the number of the voters that the coercer needs to coerce to change the result of the election.

Chapter 5: Strategies and Information Flow: Preliminaries. This chapter is a preliminary for Objective 2 of the thesis. First, we give a brief literature review on the noninterference property and its variants. Then, we present the transition network model, which is similar to the one used in the original definition of noninterference in the seminal work of Goguen and Meseguer [GM82]. We also explain some related terminologies which will be used further in the thesis. In the next part, we present the definition of the standard noninterference property based on the transition network model. Then, we present the unwinding relation which is a characterization of the noninterference property. Lastly, formal definitions of the strategies and the temporal goals of the players in a transition network model are explained.

Chapter 6: Strategic Noninterference. This is the main chapter contributing to Objective 2.a of the thesis. We start with a motivating example, and then present an overview of related works. In the next section, we propose our new concept of strategic noninterference. We give a proof showing that a tractable characterization of strategic noninterference in its general form does not exist. Then, we present characterizations of this property for certain classes of goals, and for a setting where the strategy of the High players is given as a parameter.

Chapter 7: Effective Security. This chapter includes the main results related to Objective 2.b of the thesis. We begin by presenting a motivating example, and then continue by taking a look at related works. We define the concept of effective security as a means to compare the security of two models based on the strategic abilities of the adversary to harm the goal of the system. Then we look more specifically at information flow security and show how it can be defined based on the relation between the attacker's observational capabilities and his ability to compromise the goals of the system. We define the noninterferent idealised variant of a system, as a variant which does not allow any relevant information flow by construction. Finally, we extend our results to models that are not total on input (unlike the transition network models of Goguen and Meseguer[GM82]).

Chapter 8: Expressing Receipt-Freeness and Coercion-Resistance in Logics of Strategic Ability. This chapter contains the contribution of the thesis towards Objective 3. We begin by explaining the motivation and by giving an overview of related works. The next part includes a preliminary section on logics of the strategic ability **ATL** and **ATL***. In the rest of the chapter, we go through some of the significant works introducing formal definitions of the receipt-freeness and coercion-resistance properties and give a transcription of the intuition behind their definitions in the logic of strategic ability **ATL***.

Chapter 9: Conclusion and Future Works. In this chapter we discuss the results presented throughout the thesis and outline some of the future works regarding each strand.

1.4 Publications

The research carried out during the course of this PhD has resulted in the following publications:

- Tabatabaei, Masoud, Wojciech Jamroga, and P. Y. Ryan. “Preventing Coercion in E-Voting: Be Open and Commit.” Proceedings of the 1st Workshop on Hot issues in Security Principles and Trust (HotSpot), 2013.

- Jamroga, Wojciech, and Masoud Tabatabaei. “Strategic Noninterference.” IFIP International Information Security Conference. Springer International Publishing, 2015.

- Tabatabaei, Masoud, Wojtek Jamroga and Peter Y. A. Ryan. “Expressing Receipt-Freeness and Coercion-Resistance in Logics of Strategic Ability: Preliminary Attempt.” The International Workshop on AI for Privacy and Security (PrAISe), 2016.

- Jamroga, Wojtek and Masoud Tabatabaei. “Information Security as Strategic (In)effectivity.” 12th International Workshop on Security and Trust Management (STM), 2016.

- Jamroga, Wojciech, and Masoud Tabatabaei. “Information Security as Strategic (In)effectivity (Extended Version).” arXiv preprint arXiv:1608.02247 (2016).

- Jamroga, Wojtek and Masoud Tabatabaei. “Preventing Coercion in E-Voting: Be Open and Commit." 12th International Joint Conference on Electronic Voting (E-Vote-ID), 2016.

Chapter 2

On Formal Definitions of Receipt-Freeness and Coercion-Resistance

Voting is a mechanism of utmost importance to social processes, as many important decisions for the society are made through elections and referenda. Digital technology holds out the promise of greater citizen engagement in decision-making. Throughout the history of democracy, elections have been the target of attempts to manipulate the outcome. To counter the threats of coercion or vote buying, *ballot confidentiality* was recognised an important property of voting systems. More recently, cryptographers and security experts have been looking at using cryptographic mechanisms to provide *voter-verifiability*, i.e., the ability of voters to confirm that their votes are correctly registered and counted. Voter-verifiability strengthens the integrity of the voting procedure, but, if it is not done carefully, it can introduce new threats to confidentiality. This leads to the introduction of more sophisticated notions: *receipt-freeness* and *coercion-resistance*. Receipt-freeness focuses on the resources needed to construct a coercion attack and requires that the voter can obtain no certified information (a receipt) which could be used to prove to a coercer that she voted in a certain way. Coercion-resistance, on the other hand, is intended to capture broader security concerns, guided by the following intuition: whatever strategy the coercer adopts, the voter always has a strategy to vote as they intend while appearing to comply with all the coercer's requirements. In this chapter, we take a look at formal definitions of receipt-freeness and coercion-resistance properties in the literature. The content of this chapter serves as a part of literature review for Objectives 1 and 3 of the thesis (Chapters 4 and 8).

Outline of the chapter. In Section 2.1 we have an overview on more conventional methodologies and models used for obtaining proofs of security properties. Section 2.2 contains a brief literature review on the development of the formal definitions of the properties of receipt-freeness and coercion-resistance. Then in Section 2.3 we take a closer look at some of the significant works that have presented formal definitions of these properties. Finally in Section 2.4 we concludes the chapter.

2.1 Common Methods for Modeling the Security Protocols

In order to obtain proofs that security protocols are correct, one first needs to model them mathematically. Two types of models that are commonly used for providing proofs of security for the protocols are symbolic models and computational models. In the following, a brief overview of these models is provided.

Symbolic models: In the symbolic model, the cryptographic primitives are represented by functions considered as black-boxes. The messages are terms on these functions. In this model, the adversary is restricted to compute only using the given functions. Moreover the only equalities that hold are those explicitly given by equations in the model. So, this model assumes perfect cryptography. For instance, if shared-key encryption is modeled by two function *enc* and *dec* and the equality $dec(enc(x,k),k) = x$, then the adversary can decrypt $enc(x,k)$ only when she has the key k. The symbolic model is in particular suitable for automation.

Computational (Cryptographic) models: In computational models, the messages are bitstrings and the cryptographic primitives are functions from bitstrings to bitstrings. The adversary is any probabilistic Turing machine. A security property is considered to hold when the probability that it does not hold is negligible in a security parameter (which is normally the length of a key used in the protocol). For instance, shared-key encryption can be modeled by two functions *enc* and *dec* with the $dec(enc(x,k),k) = x$. Then the security of encryption is expressed by saying that the adversary has a negligible probability of distinguishing encryptions of two messages of the same length. The computational model is much more realistic, but the proofs are usually more complicated, and these proofs are often only manual.

One can also see two main approaches to defining security properties in computational models. One is referred to as "simulation-based models" and the other "game-based models". In a simulation-based security definition, besides the real world, we also consider an *ideal world*. The ideal world often doesn't involve any cryptography and usually is defined regarding the physical security assumptions. We would say a cryptographic scheme is now secure if any 'attack' possible in the real world can also be applied to the ideal functionality in the ideal world. In a game-based security definition, the security is defined by a game between a challenger and an adversary. Usually, the challenger has access to all secret parameters, whereas the adversary only has access to some oracles (e.g., public hash functions) plus whatever is given by the challenger during the game (e.g. public parameters). The adversary then provides some output depending on the defined challenge. The advantage of an adversary roughly corresponds to how much 'better' the adversary can do at the game than a trivial adversary that just guesses its output. A cryptographic scheme is said to satisfy the security definition if and only if no 'efficient' adversary has a substantial advantage in the game.

2.2 A Brief Literature Review

In 1994, Benolah and Tuinstra [BT94] introduced receipt-freeness as a required property for avoiding coercion in e-voting systems. Later Michels et al. [MH96] extended the concept by considering different levels of voter-control for the coercers, and varying levels of collusion between the coercer and other parties in the election. Okamoto and Tatsuaki [Oka98] developed the formal definition of [BT94] to make it more appropriate for large scale elections. In 2005, Juels et al. [JCJ05] introduced coercion-resistance as the property of being receipt-freeness, plus resisting against randomization, forced abstention and simulation attacks.

Delaune et al. have a series of works [DKR05, DKR06, DKR10] in which they have formalised receipt-freeness and coercion-resistance properties in applied pi calculus. Moran and Naor [MN06] introduced a simulation based definition for coercion-resistance. Backes et al. [BHM08] introduced a definition of coercion-resistance in the symbolic model which was more suitable for automation than previous works. Meng [Men09] provided a state of the art survey on the definitions of receipt-freeness and coercion-resistance and on what technologies are used at the time to implement these properties in voting schemes.

Kuesters et al. [KTV10] introduced a formalisation of coercion-resistance which could provide a probabilistic measure of the amount of coercion-resistance in a voting scheme. Dreier et al. [DLL12] provided formal definitions of various privacy notions in applied pi calculus, and showed how they are related to each other. Also, some works such as [JP06, BRS07, KT09] have used formal logic to express coercion-resistance property in elections.

Several works such as [MBC01, LK03, ALBD04, LBD$^+$04, KH04, WAB07, ARR$^+$10b, SHKS12], without introducing new definitions of receipt freeness and coercion-resistance properties, have developed weaker, more practical, or more efficient ways to realize the needed assumptions for achieving these properties (assumptions such as existence of untappable channels, anonymous channels, etc.).

2.3 Some of the Significant Works

1994: ***Receipt-free secret ballot elections*** **[BT94]**

This paper introduced the notion of receipt-freeness. They used some examples to show why giving a “receipt” to the voter can be harmful, as it prevents the voter from being able to deceive the coercer. Consequently, it defines receipt-freeness exactly as the property of "not giving the voter any receipt during the election that makes her able to prove that her vote was cast in a particular way". Through the paper “uncoercibility” is regarded equivalent to receipt freeness.

The formal definition of “uncoercability” in this paper is given in computational model. It assumes two possible protocols (choices) for the voter. The property requires that firstly the coercer needs to have same public views over the runs of the election for the two choices. Secondly, a voter does not get to have a “receipt” from its private info that can be used as a proof to show what protocol she has used.

2005: ***Coercion-resistant electronic elections*** **[JCJ05]**

This paper introduced the coercion-resistance property. They argue that in real world scenarios, receipt-freeness is too weak. Because it fails to protect an election system against several forms of serious, real-world attack, namely randomization attack, forced abstention attack, and simulation attack. Briefly, in the randomization attack, the coercer asks the voter to use some randomization method for choosing her vote. In the forced abstention attack, the attacker wants the voter to avoid voting, and in the simulation attack, the attacker himself simulate the role of the voter (for example by causing her to divulge her private keying material after the registration, but before the election process).

So, the paper suggests that the coercion-resistance property is one that besides not enabling the voter to produce a proof of his vote, can also protect against randomization, forced abstention and simulation attacks. The formal definition of coercion-resistance in the paper is given in computational model and as a game based definition. The game is between the adversary and a voter targeted by the adversary for a coercive attack. The structure of the game is as follows: a coin is flipped, and the outcome is represented by a bit b. If $b = 0$, then the voter casts a ballot of its choice and provides the adversary with a false voting key. In other words, the voter attempts to evade adversarial coercion. If $b = 1$ on the other hand, then the voter submits to the coercion of the adversary. She merely provides the adversary with her valid voting key and does not cast a ballot herself. The challenge of the adversary is to guess the value of the coin b, that is, to determine whether or not the targeted voter, in fact, cast a ballot. A coercion resistant voting system is thus one that, no efficient adversary has a non-negligible advantage in this game.

2005: ***Receipt-freeness: Formal definition and fault attacks*** **[DKR05]**

This paper proposed a formalisation of receipt-freeness in the symbolic model, in applied pi calculus. It also comments on that if one wants to satisfy both the receipt-freeness property and the individual verifiability property, then some restriction on the communication between the voter and the coercer is needed (Individual verifiability guarantees that a voter can obtain proof that her vote was counted in the final tally of the election.).

The formal definition of receipt-freeness in this paper is based on the concept of observational indistinguishability of processes. Intuitively, a voting protocol is receipt-free if for all voters A, the process in which A votes according to the intruder's wishes is indistinguishable from the one in which she votes something else, even if all secrets are (apparently) shared with the intruder. The definition also demands the existence of some counterbalancing voter B with whom voter A swaps her vote, to avoid the case in which the intruder can distinguish the situations merely by counting the votes at the end.

2006: ***Coercion-resistance and receipt-freeness in electronic voting*** **[DKR06]**

In this work the authors have continued on their previous works and provided formal definitions of coercion-resistance and receipt-freeness in applied pi calculus. They stated that the difference between the definition of coercion-resistance and receipt-freeness lies in the powers of the coercer to interact with the voter during the voting stage. In receipt freeness, the assumption is that the coercer just examines evidence gained from observing the election process, including those provided by the cooperating voter (e.g., the voter's private key and random coins used for probabilistic encryption). In coercion-resistance on the other hand, the coercer has additional capabilities, as he can interact with the cooperating voter (e.g., by adaptively preparing messages which the voter will send during the process).

2006: ***Receipt-free universally-verifiable voting with everlasting privacy*** **[MN06]**

This work gives a formal definition for receipt-freeness in computation model, as a simulation-based definition. They state that a receipt-free protocol must specify, in addition to the strategy for honest voters, a coercion strategy. When a voter is coerced by the adversary, she begins following the coercion strategy. The coercion strategy tells the voter how to fake its responses to the adversary.

The formal definition uses the real world vs. the ideal world model. In the ideal world model, the only interaction between the parties is the public outputs of the election. Loosely speaking, a protocol is receipt free if any coercion attack that the adversary can do in the real world, it can also do in the ideal world. This may require that in the real world the coerced voter switches to a "coercion strategy", which specifies how to respond to coercer's queries and commands. The authors also have stated that even when a protocol is receipt-free by this definition, it may still be possible to coerce the voters. What the definition does promise is that if it is possible to coerce a voter in the real world, it is also possible to coerce her in the ideal world.

2008: ***Automated verification of remote electronic voting protocols in the applied pi-calculus***

[BHM08]

In this paper, a formalisation of coercion-resistance and receipt-freeness in applied pi calculus is presented. It improves on the definitions from [DKR06] by introducing a formal definition which is more suitable for automation, and also taking forced abstention attacks into consideration. They define coercion resistance as the immunity to the simulation attacks and later prove that this definition implies both the immunity to the forced abstention attacks and receipt-freeness (They didn't address the randomization attacks in this paper).

The idea of the formal definition can be expressed as follows: One considers two processes for the voter, one which complies with the request of the coercer and one which "cheats" and in which the voter provides the coercer with some fake information while voting as she intends. Then the property states that the two processes (running in parallel with other voters and the the voting system itself) must be observational equivalent for the coercer.

2010: ***A game-based definition of coercion-resistance and its applications*** **[KTV10]**

This work gives a formal definition of coercion-resistance in the form of a game-based definition. A game is defined in which the coerced voter chooses either to follow the strategy given by the coercer or to follow a counter-strategy which results as she intended. The definition states that for any coercer, the probability of the coercer guessing correctly the choice of the coerced voter must be less than some given value δ. The minimum value of δ for which this property holds can also be used for measuring the coercion-resistance property in a system.

2012: ***A formal framework for modelling coercion resistance and receipt freeness*** **[HS12]**

This work introduces a formal framework in the symbolic model, in process algebra CSP, for modelling different definitions of coercion-resistance in the literature. The framework is intended to help analysing a voting system to see which of the definitions it satisfies.

The framework is based on a process equivalence between the coercer's view of the voting system when the voter follows his instruction, and when the voter follows some behaviour that results in her deliberate choice of candidate. The general definition states that for any coercer and any instructed strategy, there must exist a voter process such that the process equivalence is satisfied. Different flavours of coercion-resistance then can be modelled by modifying the process instructed by the coercer.

2.4 Summary

In this chapter, we had a brief overview on some of the significant works that have formalised the receipt-freeness and the coercion-resistance properties. In Chapter 8 we return to some of these works, and will use logics of strategic abilities to express the informal definition behind each of them. Also in Chapter 4 we model an election system in the presence of possible coercion and a set of security measures available for implementation. However in that chapter we take these measures as some methods and won't mention any technical details about them.

Chapter 3

Towards Game-Theoretic Analysis of Coercion Resistance

Game theory is the mathematical study of interaction (conflict and cooperation) between agents who are pursuing their own interests (which may or may not be in conflict with other agents' interests). Game theory has vastly been used in disciplines such as economics, biology, sociology, computer science and political science, among others.

Because of the nature of the security problem scenarios, which usually consists of interactions between defenders and attackers, game theory is well-suited as a tool for studying these situations. In this chapter, we review the models of normal form games with both complete and incomplete information of players, together with some solution concepts that we will use in the following chapters.

Outline of the chapter. Section 3.1 explains modelling the games in normal form, and the concept of mixed strategies in normal form games. In Section 3.2 we present some of the important solution concepts for the games in normal form. In Section 3.3, we take a look at incomplete information games, and explain the Bayesian Nash equilibrium. Finally Section 3.4 concludes the chapter.

3.1 Games in Normal Form

Representing a game in normal form - also known as the strategic form - is the most common way of modelling the interaction between the agents in game theory. When a game is represented in normal form, it is usually assumed that the players choose their actions simultaneously, or at least without knowing the action selected by other players beforehand. In a game represented in normal form, the utility of each player is uniquely determined by the actual state of the world, which is in its turn determined uniquely by the players' combined actions. While explaining the games with incomplete information in section 3.3, we will see that even when we consider that the state of the world depends on some randomness in the environment, we still can model the game in normal form.

Definition 3.1.1 (Normal form game)**.** *A finite normal form game with n player is a tuple $\langle N, \Sigma, u \rangle$, where:*

- *N is a finite set of n players,*
- *$\Sigma = \Sigma_1 \times \cdots \times \Sigma_n$, where Σ_i is a the set of actions available to player i,*
- *$u = (u_1, \cdots, u_n)$, where $u_i : \Sigma \to \mathbb{R}$ is a utility (or payoff) function for player i.*

$a = (a_1, \cdots, a_n) \in \Sigma$ is called an action profile. A game in normal form for two players is usually represented by a matrix, where the rows of the matrix represents the actions of the first player and the columns represents the actions of the second player. The utility of each player for each action profile is written in the corresponding cell of the matrix.

Example 3.1.1 (Prisoner's Dilemma)**.** *Figure 3.1 shows a normal form representation for a famous example in game theory which is called the "prisoner's dilemma". The story is about two prisoners who are captured by the police for committing a crime. The two prisoners have two choices, either to* confess *to the crime, or to* deny *it. If they both deny the crime they are both sentenced to stay*

	Deny	Confess
Deny	$-1,-1$	$-5,0$
Confess	$0,-5$	$-3,-3$

Figure 3.1 The matrix representation of normal form model of the prisoner's dilemma scenario

one year in prison. If they both confess then they are both sentenced to stay three years in prison, and if one confesses and the other one denies then the prisoner who confessed is released, and the one who denied stays five years in prison.

We can represent the prisoner's dilemma in normal form, as the tuple $\langle N, \Sigma, u \rangle$ *where* $N = \{p_1, p_2\}$ *is the set of two players (prisoners);* $\Sigma = \Sigma_1 \times \Sigma_2$ *is the set of action profiles where* Σ_1=$\Sigma_2 = \{D, C\}$ *are the sets of actions for the two players (D representing to deny and C representing to confess); and* $u = (u_1, u_2)$ *is the utility function of the players defined by* $u_1(D,D) = u_2(D,D) = -1$*,* $u_1(C,C) = u_2(C,C) = -5$*,* $u_1(D,C) = u_2(C,D) = -3$*, and* $u_1(C,D) = u_2(D,C) = 0$.

In Figure 3.1 each row of the matrix shows one possible action for the first player and each column represents one possible action for the second player. The four possible outcomes based on these actions are described by the four cells of the matrix. In each cell, the first element is the utility of the first player and the second element the utility of the second player.

3.1.1 Strategies in Normal Form Games

Strategies represent the choices of the players in a game. A *pure strategy* of a player expresses the selection of a single possible action by that player. We call a tuple of pure strategies for each of the players in the game a *pure strategy profile*. We will denote a pure strategy of a player, and a pure strategy profile, respectively by the same notation we use for a player's action and an action profile.

A player may also decide to randomise between its possible actions based on some probability distribution. We call such a randomised choice a *mixed strategy* of a player. If D is a set, we show the set of all probability distribution over members of D by $\Delta(D)$.

Definition 3.1.2 (Mixed strategies)**.** *If $\langle N, \Sigma, u \rangle$ is a normal form game, then the set of mixed strategies of player i is $S_i = \Delta(\Sigma_i)$.*

Also $S = S_1 \times \cdots \times S_n$ is the set of all strategy profiles. By $s_i(a_j)$ we denote the probability of an action $a_j \in \Sigma_i$ being played under strategy s_i of player i.

Definition 3.1.3 (Expected utility of a mixed strategy)**.** *Given a normal form game $\langle N, \Sigma, u \rangle$, the expected utility u_i of player i for the mixed strategy profile $s \in S$ is defined as*

$$u_i(s) = \sum_{a \in \Sigma} u_i(a) \prod_{j=1}^{n} s_j(a_j)$$

If $s = \langle s_1, \cdots, s_i, \cdots, s_n \rangle \in \Sigma$ is a strategy profile and i is a player in the game, we may represent s by (s_i, s_{-i}) where $s_{-i} = \langle s_1, \cdots, s_{i-1}, s_{i+1}, \cdots, s_n \rangle$. Similarly, we may represent an action profile $a \in \Sigma$, by $a = (a_i, a_{-i})$.

Definition 3.1.4 (Best response)**.** *The best response of player i to the strategy profile s_{-i} is a mixed strategy $s_i^* \in S_i$ such that $u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i})$ for all strategies $s_i \in S_i$.*

A best response action (pure strategy) of a player is defined in a similar way. We call a strategy s_i a *dominant strategy* of player i, if it is a best response to all strategy profiles $s_{-i} \in S_1 \times \cdots S_{i-1} \times S_{i+1} \times \cdots S_n$. A dominant action (pure strategy) of a player is also defined similarly.

Example 3.1.2. *In Figure 3.2 , the best response of the first player for each action of the second player is depicted in blue. Also, the best response of the second player for each action of the first player is represented in red.*

	Deny	Confess
Deny	$-1,-1$	$-5,0$
Confess	$0,-5$	$-3,-3$

Figure 3.2 The best responses of players for each action of the other player is depicted in color. The Nash equilibrium is at (Confess, Confess).

3.2 Solution Concepts for Normal Form Games

In game theory, *solution concepts* are used to define which collective behaviours are "rational" and should (or may) be selected by the players in a game. Different solution concepts correspond to different notions of rationality. More precisely, they correspond to different models of the deliberation process that leads to selecting one or the other strategy. In this section, we provide a brief overview of three such concepts that we will also use in the following chapters: maxmin, Nash equilibrium and Stackelberg equilibrium.

3.2.1 Maxmin

Maxmin for player i selects the strategy of player i which guarantees the best minimal outcome of the strategy. In a way, it boils down to assuming that the other players may try to inflict as much damage as possible to i. Therefore, one may argue that maxmin captures decision making of "paranoid" agents.

Definition 3.2.1 (Maxmin)**.** *The maximin strategy for player i is defined as*

$$arg\ max_{s_i} min_{s_{-i}} u_i(s_i, s_{-i})$$

.

3.2.2 Nash Equilibrium

Nash equlibrium represents a play which can emerge when players adapt their choices to what they expect from the other players. Formally, a strategy profile $s \in S$ is a Nash equilibrium iff no player can unilaterally change her strategy in s so that she increases her utility (the strategies of the other players are assumed to stay the same). Nash equilibrium often captures the collective behaviour that emerges "organically", through a sequence of strategy adjustments from different players that leads to a point when nobody is tempted to change their strategy anymore.

Example 3.2.1. *In Figure 3.2, the Nash equilibrium of the game is at (Confess, Confess), where each player's action is the best response to the other players action.*

3.2.3 Stackelberg Equilibrium

Stackelberg equilibrium represents a rational play in a game with a designated *leader* player. The leader has the power to choose her action first. She can also publicly commit to her choice so that the other players are fully aware of it. Formally, Stackelberg equilibrium is defined as the best response to the best response. That is, for every strategy s_i of the leader (player i) we find the response s_{-i} that maximizes the utilities of the opponents; then, we select the s_i which maximizes $u_i(s_i, s_{-i})$.

It is important to notice that when the Stackelberg equilibrium and maxmin of a player coincide, the leader does not have an incentive to commit openly to her strategy: if the opponents don't choose the best respond, the leader can only gain by that but never lose. Conversely, when Stackelberg equilibrium is different from maxmin, *the leader is better off openly committing to her policy* because this way she forces the other players to respond in a desirable way.

3.3 Games with Incomplete Information

So far in the games we discussed, all the players know what game is being played. In other words, the number of players, the actions available to each player, and the payoff of all players for each action profile are all common knowledge among all the players. In this section, we look at games with incomplete information, also known as Bayesian games, where players may be uncertain about some aspects of the game being played.

Harsanyi [HS72] suggested that one can simulate the uncertainties about the games being played, and also about the beliefs of the other players about the game being played, by introducing nature as an additional player that chooses the actual game based on some probability distribution. One assumption we make is that all the possible games have the same number of agents and the same strategy space for each agent. There are various ways of formalising Bayesian games. We use the following definition in this work:

Definition 3.3.1 (Bayesian game). *A Bayesian game with n player is a tuple $\langle N, \Omega, \Sigma, T, \tau, p, u \rangle$, where:*

- *N is a finite set of n players,*
- *Ω is a finite set of states of the nature. Eech state of the nature uniquely determines one of the possible games being played.*
- *$\Sigma = \Sigma_1 \times \cdots \times \Sigma_n$, where Σ_i is a the set of actions available to player i,*
- *$T = T_1 \times \cdots \times T_n$, where T_i is the set of possible signals that may be observed by player i and gives her some information about the game being played. We call each $t_i \in T_i$ a* type *of player i.*
- *$\tau = (\tau_1, \cdots, \tau_n)$, where $\tau_i : \Omega \to T_i$ decides the signal observed by player i for a given state of the nature. In other words τ_i decides the type of player i for each state of the nature.*

- *$p = (p_1, \cdots, p_n)$, where $p_i \in \Delta(\Omega)$ is the probability distribution over the states of the nature for player i. It holds that $p_i(\tau_i^{-1}(t_i)) > 0$ for all $t_i \in T_i$.*
- *$u = (u_1, \cdots, u_n)$, where $u_i : \Omega \times \Sigma \to \mathbb{R}$ is a utility (or payoff) function for player i. The payoff of a player depends both on the action profile chosen by the players, and on the actual state of the nature.*

A strategy of a player in a Bayesian game assigns choices for the player for each of the player's type. Therefore a mixed strategy of a player is defined as $s_i : T_i \to \Delta(\Sigma_i)$. We show the set of all mixed strategy profiles of a Bayesian game by $\hat{S}$. By $s_i(t_i, a_i)$ we denote the probability of choosing action a_i by player i when its type is t_i. The expected utility of a mixed strategy given a state of the nature is defined similarly to the Definition 3.1.2 :

Definition 3.3.2 (Expected utility for a given state of nature)**.** *Given a Bayesian game $\langle N, \Omega, \Sigma, T, \tau, p, u \rangle$, the expected utility u_i of player i for the mixed strategy profile $s \in \hat{S}$ in the state of the nature ω is defined as*

$$u_i(\omega, s) = \sum_{a \in \Sigma} u_i(\omega, a) \prod_{j=1}^{n} s_j(\tau_j(\omega), a_j)$$

Next, we calculate the *ex-ante* expected utility of a player for a given strategy profile. This is the expected utility of a player, assuming that the player does not know its type.

Definition 3.3.3 (Ex-ante expected utility)**.** *Given a Bayesian game $\langle N, \Omega, \Sigma, T, \tau, p, u \rangle$, the ex-ante expected utility $\hat{u}_i : S \to \mathbb{R}$ of player i for the mixed strategy profile $s \in \hat{S}$ is defined as*

$$\hat{u}_i(s) = \sum_{\omega \in \Omega} pr_i(\omega) \cdot u_i(\omega, s).$$

Now we can transform construct a normal form game where the possible choices (actions) of players are the strategies they can take in the Bayesian game, and their utility function is the ex-ante expected utility of them in the Bayesian game. The Bayesian Nash equilibrium of the Bayesian game is then defined as the Nash equilibrium of this new game in normal form.

Definition 3.3.4 (Bayesian Nash equilibirum)**.** *The Bayesian Nash equilibrium of the Bayesian game* $\langle N,\Omega,\Sigma,T,\tau,p,u\rangle$*, is defined to be the Nash equilibrium of normal form game* $\langle N,\hat{S},\hat{u}\rangle$*.*

3.4 Summary

In this section, we had an overview of some of the important concepts in game theory. We use most of the material explained in this chapter in Chapter 4. We will meet the concept of the *strategy* again in Chapters 6, 7 and 8. However in those chapters we use the term with a different nuance (as will be explained in Chapter 5) which is closer to the definition of the strategy in the games in *extensive form* (not covered in this chapter).

Chapter 4

Preventing Coercion in Elections, a Game Theoretic Approach

As we have seen in Chapter 2, receipt-freeness and coercion-resistance are considered essential properties of voting systems that are intended to counter threats of coercion or vote buying. Achieving coercion-resistance is extremely challenging, especially in the context of the internet and remote voting (e.g. postal). A number of schemes have been proposed that provide it, but typically this comes at a cost, in particular in terms of usability. In this chapter, we take a game theoretic approach to analyse the trade-offs between the costs of implementing coercion-resistance mechanisms on one hand, and the costs to the society due to the coercion attacks on the other hand.

Unlike most existing works, we do not propose a new coercion-resistant voting scheme, nor prove that a scheme is secure in that respect. Instead, we focus on the context of coercion attempts in e-voting, namely costs and benefits of involved parties. The main question we try to answer is: *Should the society invest in coercion-resistant procedures, and if so, in what way?*. We do not aim at devising a secure voting procedure, but rather at exposing conditions under which security of a procedure is relevant at all.

Here we do not represent the coercion resistance property of a voting system explicitly. Instead, we model the coercion resistance level as a simple scalar, usually indicating how much effort/cost it would take to break it. Also, our game-theoretic models are very simple: the society's interests are represented by a single agent that we call the "election authority" and there is only one coercer in the game. We also study two cases; First we consider a case where the structure of the game (i.e., strategies and their outcomes) are common knowledge among the participants. Hence we model it as a normal form game with complete information. Then we consider the uncertainty of the players about some parameters of the game, modelling it as a strategic game with incomplete information of players.

Outline of the chapter. Section 4.1 contains a literature review of related works. In Section 4.2 we present a game model for an election in the presence of possible coercion, where the players involved have complete information about the game being played. We start with a model where a perfectly secure method is available for being implemented by the authority player, and then extend it to a model where none of the available methods is perfectly secure. In Section 4.3 we study the setting where players have incomplete information about the number of voters that is needed to be coerced by the coercer to change the result of the election. We consider both a uniform and a normal probability distribution for this number, and in each case we study the solution concepts for the resulting incomplete information game model. We conclude the chapter in Section 4.4.

4.1 Related Works

Related work can be roughly divided into three strands: definitions of the concept of the coercion-resistance (and its relation to privacy), proposals of coercion-resistant voting procedures, and studies of the context of coercion-resistance.

The works in the first strand is discussed in detailed in Chapter 2. The second strand overlaps with the first: [JCJ05, GGR09] all propose voting protocols that satisfy their definitions of coercion-resistance while [KTV10] proves coercion resistance of two previously existing protocols. Another coercion-resistant voting scheme was introduced – and proved – in [ARR$^+$10a]. Several other papers proposed voting schemes which provably satisfy privacy as an intuitive argument for coercion-resistance, cf. e.g. [Rya10].

Putting coercion resistance in a broader economic or social context has been, to our best knowledge, mostly left untouched. The only paper in this strand that we are aware of is [BM07]. The authors compare two voting systems using game models, more precisely zero-sum two-player games based on attack trees. There are two actions available for the attacker (performing the attack or not), and the authority is presumably choosing one of the two voting systems. The utility of the attacker is the expected probability of successful coercion minus the expected probability of being caught. The value is computed for the two systems using empirical data. In contrast, we consider a more general game where coercion – and resistance measures – come at a *cost* (instead of simply assuming probability distributions for the possible events), and we look for the rational choices of the players using game-theoretic solution concepts. We also argue that the coercion game is not zero-sum, with important consequences for the best policy to be chosen.

4.2 Modeling Coercions in Elections: Complete Information Setting

We model the election as a game in normal form. In the general case, we may have several coercers who try to change the result of election in different ways. But in the models we use in this chapter, we consider that there is only one coercer acting in the election. We consider that the election includes a set of candidates $\Omega = \{\omega_1, ..., \omega_g\}$ and a set of n voters. However we don't consider candidates or voters as the players in the game model, but rather as parameters of the model.

We model the election as a strategic game $\langle\{A,C\},\Sigma,(u_A,u_C)\rangle$, where $\Sigma = \Sigma_A \times \Sigma_C$ with the following ingredients:

Players. A and C are the players in the game. The player A is an honest election authority who acts on behalf of the society. We assume that the goal of this player is in line with what we may call "the common good of the society". Player A has no preference for any of the candidates and tries to make the result of the election as similar as possible to the result of the election without any coercion, i.e when the voters vote only on basis of their own preferences over the candidates.

Player C represent a potential coercer. This player may try to change the result of the election by threatening or bribing voters in order to make them vote based on the coercer's plan, rather than the voters' own preferences over the candidates.

Note that we do not represent the voters explicitly as players in the game. Their interests are globally represented by the preferences of A.

Actions and strategies. $\Sigma_A = \{\alpha_0, ..., \alpha_{Max}\}$ is the set of privacy methods available to be implemented by the election authority A. These represent the security measures that are possible to be implemented in order to prevent, or make it harder, for a coercer to find out about the actual value of the votes of the voters. It is assumed that α_0 represents the case of no privacy.

$\Sigma_C = \{0, ..., n^*, ..., n\}$ is the set of actions for the coercer C. The actions of the coercer are the number of voters the coercer attempts to bribe. So action t shows the action of attempting to bribe t voters by the coercer. The minimum number of voters that the coercer needs to bribe in order to change the result of the election in its favour is n^*, where $0 \leq n^* \leq n$ (where n is the number of voters). It is assumed that coercing more than n^* voters always only add the cost of coercion for the coercer and is always dominated for the coercer by n^*. Therefore, when analysing the model, if we assume that the coercer knows the value of n^*, we normally don't consider the actions $n^* + 1$ to n.

Preferences. Preferences are represented by utility functions over possible combinations of strategies. For each player, their utility combines several factors. The utility if the election authority A is defined as $u_A(\sigma) = v_A(out(\sigma)) - imp(\alpha_j)$, where:

- $\sigma = (\alpha_j, k) \in \Sigma$ is an action profile, chosen by the players in the game.
- $out : \Sigma \to \Omega$ gives the result of the election. We assume that the result of the election is depends only on the actions chosen by the players in the game.
- $imp : \Sigma_A \to \mathbb{R}$ is the implementation cost function. $imp(\alpha_j)$ shows the cost of implementing the privacy method α_j for the player A. It is assumed that $imp(\alpha_0) = 0$, and $imp(\alpha_t) \leq imp(\alpha_{t'})$ if $t < t'$.
- $v_A : \Omega \to \mathbb{R}$ is a function that gives the value of the election for the player A. Our assumption is that for all α_j, $v_A(out(\alpha_j, k_1)) \leq v_A(out(\alpha_j, k_2))$ if $k_1 < k_2$. It means that adding a new bribed voter can only decrease the value of the election for player A. The function v_A is defined such that it can take only two values: for a given σ either $v_A(out(\sigma)) = v_A^*$, when the result of the election is the same as its result without coercion, or $v_A(out(\sigma)) = v_A^* - \varepsilon_A$, when the result of election has changed because of coercion, where ε_A is a positive number. We assume the value ε_A is larger than implementation cost of any privacy method: for all $\alpha_i \in \Sigma_A$, $imp(\alpha_i) < \varepsilon_A$.

The utility function for the coercer C is defined as $u_C(\sigma) = v_C(out(\sigma)) - k \times (d_C(\alpha_j) + \beta_C)$, where:

- $\sigma = (\alpha_j, k) \in \Sigma$ is an action profile, chosen by the players in the game, where α_j is the security measure chosen by player A and k is the number of voters that coercer attepmts to coerce.
- $out : \Sigma \to \Omega$ gives the result of the election, given an action profile.

- $v_C : \Omega \to \mathbb{R}$ is a function that gives the value of the election for the player C. The function v_C is defined such that it can take only two values: for a given σ either $v_C(out(\sigma)) = v_C^*$, when the result of the election is in favor of the coercer, or $v_C(out(\sigma)) = v_C^* - \varepsilon_C$, when the result of the election is not in favor of the coercer, where ε_C is a positive number.

- β_C is the cost of bribing one voter for the player C. It is assumed that this number is constant for any number of bribed voters and for all the privacy methods used.

- $d_C : \Sigma_A \to \mathbb{R}$ is a difficulty function of the security method, such that $d_C(\alpha_j)$ shows the cost of verifying the vote of one voter by the coercer, when the privacy method α_j is implemented. It is assumed that $d_C(\alpha_0) = 0$, and $d_C(\alpha_t) \leq d(\alpha_{t'})$ if $t < t'$.

  We sometimes write $cost_C(\alpha_j)$ instead of the value $d_C(\alpha_j) + \beta_C(\alpha_j)$. It represents the total cost that the coercer must pay for successfuly coercing one voter, which consists of both the bribing cost, and the cost of overcoming the security measure.

- k is the number of voters that player C attempts to coerce.

In the following, first we consider a strategic game model with complete information, and we distinguish two possible settings: in one setting there exists a perfect security measure available for A, and in the other one player A can increase the cost of breaking the privacy but it cannot rule out the possibility of breaking it completely. Then we consider a strategic game model with incomplete information, where both the authority A and the coercer C have uncertainty about the value of n^* (the number of voters needed to be coerced by the coercer so that the result of election changes).

4.2.1 Perfect Security Measure

The first case that we study is the situation where the election authority has a choice between no security measure and a perfect security measure. In this case $\Sigma_A = \{\alpha_0, \alpha_1\}$, where α_0 represents

	0	n^*
α_0	$(v_A^*, v_C^* - \varepsilon_C)$	$(v_A^* - \varepsilon_A, v_C^* - \beta_C \cdot n^*)$
α_1	$(\mathbf{v_A^*} - \mathbf{imp}(\boldsymbol{\alpha_1}), \mathbf{v_C^*} - \boldsymbol{\varepsilon_C})$	$(v_A^* - imp(\alpha_1), v_C^* - \varepsilon_C - \beta_C \cdot n^*)$

Figure 4.1 Game model for perfect privacy. The Stackelberg equilibrium for the authority is shown in bold. There is no Nash equilibrium in pure strategies.

no privacy and α_1 represents perfect security measure.

α_1, being a perfect security measure, implies that the coercer cannot change the result of the election no matter how many voters he attempts to bribe, as there is no way for him to verify the values of the votes. Therefore in this case the utility of the coercer would be $v_C^* - \varepsilon_C - k \cdot \beta_C$, where k is the number of voters he attempts to bribe. Notice that we assume a coercion attempt is successful only if the coercer can verify the votes. Also we assume that when the security measure is perfect, there will be no difficulty cost for the coercer, but the result of election won't change no matter how many voters are attempted to be coerced.

We consider that n^*, the exact number of voters that the coercer needs to coerce in order to change the result of the election in his favour, is known to both the coercer and the authority. Because for player C the actions 1 to $n^* - 1$ are all dominated by action 0, we remove them from the game table. Therefore the only actions considered for the coercer are 0 and n^*. The game table of this game is depicted in Figure 4.1. For each action of the player A we show the best response of player C in red, and respectively for each action of C show the best response of player A in blue.

This game has no pure Nash equilibrium. In its mixed Nash equilibrium the authority chooses "no privacy" with the probability $p = \frac{\beta_C \cdot n^*}{\varepsilon_C}$, and the coercer attempts to coerce the voters with the probability $q = \frac{imp(\alpha_1)}{\varepsilon_A}$. In this case the expected utility of the authority is $v_A^* - imp(\alpha_1)$, which is also its utility at its maximin strategy which is α_1. Therefore the authority can as well always choose α_1. In this scenario, the Stackelberg equilibrium for the authority coincides with its maxmin strategy, so the authority does not gain by making its strategy public and committing to it. On the other hand if we consider that the authority would prefer no coercion attempt over an unsuccessful coercion attempt (this can be modeled by adding $-k \cdot \beta_A$ to the utility of the player A, where k is the number of bribed voters and β_A is some small positive number), then the Stackelberg equilibrium does not coincide with maxmin (because $v_A^* - imp(\alpha_1) > v_A^* - imp(\alpha_1) - n^* \cdot \beta_A$). Thus, A can, in fact, improve its utility by bringing the game to a Stackelberg equilibrium. It can do that by making its commitment to choose α_1 public. As a result, the coercer not anymore preferring to attempt to coerce changes its choice from mixing between b_0 and b_{n^*}, to not to coerce at all.

4.2.2 Imperfect Privacy

The next case we study is where the election authority has several choices for the privacy method, such that none of them is perfect. In other words, under all the security measures the coercer is able to coerce successfully. Again we assume that the value of n^* (the amount of voters needed to be coerced to change the result of the election in favour of the coercer) is known to both the coercer and the authority. The game model is $\langle \{A,C\}, \Sigma, (u_A, u_C) \rangle$, in which $\Sigma = \Sigma_A \times \Sigma_C$ and $\Sigma_A = \{\alpha_0, ..., \alpha_{Max}\}$. As the authority changes the privacy method from α_0 to α_{Max}, the difficulty of coercing for the coercer increases. For a given privacy method α, if $v_C^* - cost_C(\alpha) \cdot n^*$ is larger than $v_C^* - \varepsilon_C$ then the coercer prefers coercing over not coercing. But it might be the case that from some privacy method α_m on, the cost of coercing for the coercer is more than ε_C. In this case, the coercer, although being able to coerce successfully, prefers not to attempt to coerce the election. We assume the coercer prefers to coerce when the privacy method is among α_0 to α_{m-1}, and prefers not to coerce when the privacy method is among α_m to α_{Max}. In other words for $\alpha_i, m \leq i \leq Max$, we have $(v_C^* - \varepsilon_C) \geq (v_C^* - cost_C(\alpha_i) \cdot n^*)$. For the similar reasons as the previous case, the only actions considered for the coercer are 0 and n^*. The game table of this game for four chosen security measures α_0, α_{m^*-1}, α_{m^*} and α_{Max} is depicted in Figure 4.2. For each action of the player A we show the best response of player C in red, and respectively for each action of C show the best response of player A in blue.

	0	n^*
α_0	$(v_A^* \, , v_C^* - \varepsilon_C)$	$(v_A^* - \varepsilon_A \, , v_C^* - cost_C(\alpha_0) \cdot n^*)$
α_{m^*-1}	$(v_A^* - imp(\alpha_{m^*-1}) \, , v_C^* - \varepsilon_C)$	$(v_A^* - imp(\alpha_{m^*-1}) - \varepsilon_A \, , v_C^* - cost_C(\alpha_{m^*-1}) \cdot n^*)$
α_{m^*}	$(\mathbf{v_A^* - imp(\alpha_{m^*})}, \mathbf{v_C^* - \varepsilon_C})$	$(v_A^* - imp(\alpha_{m^*}) - \varepsilon_A \, , v_C^* - cost_C(\alpha_{m^*}) \cdot n^*)$
α_{Max}	$(v_A^* - imp(\alpha_{Max}) \, , v_C^* - \varepsilon_C)$	$(v_A^* - imp(\alpha_{Max}) - \varepsilon_A \, , v_C^* - cost_C(\alpha_{Max}) \cdot n^*)$

Figure 4.2 Game model for breakable privacy. The Nash equilibrium is at (α_0, b_{n^*}), and the Stackelberg equilibrium for the authority is shown in bold.

This game has a pure Nash equilibrium at (α_0, n^*). In this equilibrium, the coercer attempts coercing enough voters, and the authority chooses the least safe privacy method among the possible choices. This equilibrium obviously is not a preferred one for the authority. One approach for the authority to change this equilibrium to a more preferred one is to use a Stackelberg equilibrium. If the authority commits itself to choose the privacy method α_m and makes this commitment public (such that the coercer believes the commitment), then the coercer chooses 0 (not to attempt coercing any voter) because its utility would be lower if he attempts to coerce rather than not coercing. By using the Stackelberg strategy, the authority improves its utility by $\varepsilon_A - imp(\alpha_m)$. So if the implementation cost of α_m is lower than the cost of a successfully coerced election for the authority, it can definitely benefit from changing the Nash equilibrium to the Stackelberg equilibrium by making its commitment public. This result suggests that even if the authority chooses a privacy method which is difficult enough to break, such that the coercers prefer not to attempt to coerce, the authority cannot benefit from this choice unless it makes this decision public and makes the coercers believe its commitment to choose the privacy method.

Like in the previous case, it is easy to see that Stackelberg is different from maxmin (the maximin strategy of player A is α_0 and its maximin value is $v_A^* - \varepsilon_A$). Thus, again, the election authority should publicly commit to its coercion-resistance strategy α_{m^*}.

4.3 Modeling Coercions in Elections: Incomplete Information Setting

In the previous scenarios, we assumed that players have complete information about the game. This is not in general true, as players may not be certain about some aspects of the game they are playing. For example, they may be uncertain about another player's preference between some of the outcomes, or even their own utility. In this section, we consider some scenarios involving incomplete information of the players. Specifically, we consider that the players don't know the exact number of voters that the coercer need to coerce to change the result of the election in his favour (This number is shown by n^* in the models).

Although the incomplete information about the value of n^* in the model represent the uncertainty of players about the exact number voters to coerce, in the real world it can also represent the uncertainty of players about some other parts of the system. For example, a coercer may have to choose his targets blindly, without knowing their initial preferences. Therefore it is possible that he attempts to coerce someone who initially was going to vote for his side regardless of the coercion. This makes the coercer uncertain about how many voters exactly he needs to coerce in order to have the needed amount of change of votes. So the effect of this "colliding" situations can also be modelled as the incomplete information of the coercer on the number of voters he has to coerce.

Although the players don't know the exact value of n^*, we assume that they have some beliefs about this value. In particular, we assume that the players know the probability distribution the value of n^*. Moreover, in the scenarios we discuss, we assume that both the coercer and the authority have the same beliefs about the probability distribution of n^*. This assumption may not be true in all the games of this kind, but in the election scenarios where these beliefs are based on the polling results which are usually made public, it seems to be realistic.

4.3.1 Bayesian Game Model

We model the incomplete information scenarion as a Bayesian game model $\langle\{A,C\},\Omega,\Sigma,T,\tau,p,(\hat{u}_A,\hat{u}_C)\rangle$, with the following ingredients:

Players. A and C are the players in the game, similar to the previous scenarios.

States of the world. Ω is the set of states of the world. In our scenario, each state of the world corresponds to one possible value for the n^*, the number of voters needed to be coerced so the result of election changes in coercer's favour.

Actions. $\Sigma = \Sigma_A \times \Sigma_C$ is the set of action profiles for the players, which is similar to the previous scenarios. We assume that the set of possible actions for the players is the same in all states of the world.

Player types. $T = T_A \times T_C$ is the set of type profiles of the players. We consider that $T_A = \{t_A\}$ and $T_C = \{t_C\}$. In other words, we consider that there is only one possible type for each player, because in our scenarios players don't have uncertainty about each other's beliefs. Both players consider the probability distribution of n^* as common knowledge.

Signal function. $\tau = (\tau_A, \tau_C)$, where $\tau_A : \Omega \to T_A$, and $\tau_C : \Omega \to T_C$. The signal function assigns a type to each player based on the actual state of nature. In our scenario the assigned type is always the same for each player.

Probability distribution. $p \in \Delta(\Omega)$ is the probability distribution over the states of nature as believed by both players. Notice that in general, each player may assign a different probability distribution over the states of nature, but in our scenarios, we assume this distribution is common knowledge. Although the values of n^* are discrete, when the number of voters is large we can use continuous probability distributions to estimate the probability of it being in different ranges. In this chapter, we will consider two probability distributions: uniform distribution and normal distribution.

Utility functions. $\hat{u}_A, \hat{u}_C : \Omega \times \Sigma \to \mathbb{R}$ are utility functions of the players. They are defined similarly to the ones in the complete information setting, with the difference that they depend not only on the action profile chosen by the players, but also on the actual state of the nature.

For analysing the solution concepts of this Bayesian game, as we already saw in Chapter 3, we can transform it into a normal form game. This transformation was introduced by Harsany [HS72]. In each of the following scenarios we transform the Bayesian game $\langle \{A,C\}, \Omega, \Sigma, T, \tau, p, (\hat{u}_A, \hat{u}_C) \rangle$ into a normal form game $\langle \{A,C\}, \Sigma, (u_A, u_C) \rangle$, such that for $s \in \Sigma$

$$u_A(s) = \mathbf{E}_{\omega \in \Omega}[\hat{u}_A(\omega, s)]$$

and

$$u_C(s) = \mathbf{E}_{\omega \in \Omega}[\hat{u}_C(\omega, s)]$$

4.3.2 Uniform Probability Distribution

In our first scenario with incomplete information, we consider that the players know that the value of n^* has uniform probability distribution in range $[n_a, n_b]$, where $n \geq b \geq a \geq 0$. This may be a very simplistic way to model the uncertainty of the players about n^*, but it can also reflect some realistic situations. For example consider that the coercer doesn't have any information about the number of voters who support him in a majority based election, and the only information he has is the number of voters. This situation can be modelled as the coercer believing that n^* has a uniform distribution between zero and half of the number of voters plus one.

In order to have normal form game transformation $\langle \{A,C\}, \Sigma, (u_A, u_C) \rangle$, we need to compute $u_A(\alpha, n)$ and $u_C(\alpha, n)$ for a security measure α chosen by player A and the number of voters n chosen by the player C to coerce.

Utility of the coercer player. We consider three ranges for n and compute $u_C(\alpha, n)$ in each range separately:

- If $n < a$ then in all states of the nature $n < n^*$, therefore:

  $$u_C(\alpha, n) = v_C^* - \varepsilon_C - n \cdot cost_C(\alpha).$$

  In this range the action 0 is the best response of player C. By choosing this action the utility of the coercer is $v_C^* - \varepsilon_C$.

- If $n \geq b$ then in all states of the nature $n \geq n^*$, therefore:

  $$u_C(\alpha, n) = v_C^* - n \cdot cost_C(\alpha).$$

  In this range the action b is the best respond of the player C, which corresponds to the utility $v_C^* - b \cdot cost_C(\alpha)$ for the coercer.

- If $a \leq n < b$ then

  $$\begin{aligned} u_C(\alpha, n) = \mathbf{E}_{\omega \in \Omega}[\hat{u}_C(\omega, (\alpha, n))] &= v_C^* - n \cdot cost_C(\alpha) - \frac{b-n}{b-a} \cdot \varepsilon_C \\ &= v_C^* - \frac{b}{b-a} \cdot \varepsilon_C + n \cdot (\frac{\varepsilon_C}{b-a} - cost_C(\alpha)). \end{aligned}$$

  If $\frac{\varepsilon_C}{b-a} - cost_C(\alpha)$ is positive then $u_C(\alpha, n)$ is increasing in n and otherwise it is decreasing in n.

Utility of the authority player. For the utility of the authority $u_A(\alpha, n)$, given a security measure α chosen by player A and the number of voters n chosen by the player C to coerce, we again consider three ranges:

- If $n < a$ then in all the states of the nature $n < n^*$, therefore:

$$u_A(\alpha, n) = v_A^* - imp(\alpha).$$

- If $n \geq b$ then in all states of the nature $n \geq n^*$, therefore:

$$u_A(\alpha, n) = v_A^* - imp(\alpha) - \varepsilon_A.$$

- If $a \leq n < b$ then

$$u_A(\alpha, n) = \mathbf{E}_{\omega \in \Omega}[\hat{u}_A(\omega, (\alpha, n))] = v_A^* - imp(\alpha) - \frac{n-a}{b-a} \cdot \varepsilon_A.$$

Normal form model. By observing the values of u_C, we can see that in the range $[0, a]$, and also when $n > b$, $u_C(\alpha, n)$ is decreasing in n. In the range $[a, b]$, based on the sign of $(\frac{\varepsilon_C}{b-a} - cost_C(\alpha))$ it can be increasing or decreasing in n. So the best response of the coercer is always one of the actions 0 or b (action a is always dominated by 0). Therefore we need only to consider these two actions for player C. We have that $u_C(\alpha, 0) = v_C^* - \varepsilon_C$ and $u_C(\alpha, b) = v_C^* - b.cost_C(\alpha)$. The coercer profits more by coercing b voters when $cost_C(\alpha) < \frac{\varepsilon_C}{b}$, and otherwise would prefer to not to coerce. We assume that from α_0 to α_{m^*-1}, it holds that $cost_C(\alpha) < \frac{\varepsilon_C}{b}$ and from α_{m^*} on, it holds that $cost_C(\alpha) > \frac{\varepsilon_C}{b}$.

Figure 4.3 shows the game table for the uniform distribution of n^*. The best responses of the coercer for each choice of the authority is depicted in red, and the best response of the player A for each choice of the coercer is depicted in blue. Similar to the scenario with complete information, the game has a pure Nash equilibrium which is (α_0, b). Again, as this is not a preferred equilibrium, the authority can use the Stackelberg equilibrium of the game, which is at $(\alpha_{m^*}, 0)$. As the Stackelberg equilibrium is different from maxmin of player A (which is at (α_{m^*}, b)), player A will

	0	b
α_0	$(v_A^*\ ,\ v_C^* - \varepsilon_C)$	$(v_A^* - \varepsilon_A\ ,\ v_C^* - \beta_C \cdot b)$
α_{m^*-1}	$(v_A^* - imp(\alpha_{m^*-1})\ ,\ v_C^* - \varepsilon_C)$	$(v_A^* - imp(\alpha_{m^*-1}) - \varepsilon_A\ ,\ v_C^* - b \cdot cost_C(\alpha_{m^*-1}))$
α_{m^*}	$(\mathbf{v_A^* - imp(\alpha_{m^*})}, \mathbf{v_C^* - \varepsilon_C})$	$(v_A^* - imp(\alpha_{m^*}) - \varepsilon_A\ ,\ v_C^* - b \cdot cost_C(\alpha_{m^*}))$
α_{Max}	$(v_A^* - imp(\alpha_{Max})\ ,\ v_C^* - \varepsilon_C)$	$(v_A^* - imp(\alpha_{Max}) - \varepsilon_A\ ,\ v_C^* - b \cdot cost_C(\alpha_{Max}))$

Figure 4.3 Incomplete information game model, where the number of voters needed to coerce is estimated by a uniform probability distribution. The Nash equilibrium is is at $(\alpha_0, b_{n_\mu+\sigma})$, and the Stackelberg equilibrium for the authority is shown in bold.

need to commit to choosing the method α_m and to make this commitment public.

4.3.3 Normal Probability Distribution

In our second scenario with incomplete information, we consider that the players know that the value of n^* has a normal probability distribution with mean value μ and standard deviation σ. Considering a normal probability distribution for n^* is usually more realistic than the uniform distribution.

Again, for constructing the stategic game transformation $\langle\{A,C\},\Sigma,(u_A,u_C)\rangle$, we need to compute $u_A(\alpha,n)$ and $u_C(\alpha,n)$ for a security measure α chosen by player A and the number of voters n chosen by the player C to coerce.

Utility of the coercer player. When n^* has a normal distribution with mean μ and standard deviation σ, the probability of a chosen n being more than n^* is:

$$Pr[n^* \leq n] = \frac{1}{2}[1 + erf(\frac{n-\mu}{\sigma\sqrt{2}})].$$

where:

$$erf(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2} \cdot dt.$$

Therefore $u_C(\alpha,n)$ can be calculated as:

$$u_C(\alpha,n) = \mathbf{E}_{\omega\in\Omega}[\hat{u}_C(\omega,(\alpha,n))] = v_C^* - n\cdot cost_C(\alpha) - \frac{1}{2}[1-erf(\frac{n-\mu}{\sigma\sqrt{2}})]\cdot\varepsilon_C$$
$$= v_C^* - \mu\cdot cost_C(\alpha) - \frac{\varepsilon_C}{2} + \gamma(n).$$

where:

$$\gamma(n) = \frac{\varepsilon_C}{2}\cdot erf(\frac{n-\mu}{\sigma\sqrt{2}}) - (n-\mu)\cdot cost_C(\alpha).$$

Analysing the changes of function $\gamma(n)$ shows that if $cost_C(\alpha) > \frac{\varepsilon_C}{\sigma\sqrt{2}}$ then $\gamma(n)$ is decreasing in n. In this case $u_c(\alpha,n)$ has its maximum at $n=0$. If $cost_C(\alpha) < \frac{\varepsilon_C}{\sigma\sqrt{2}}$ then $\gamma(n)$, and hence $u_C(\alpha,n)$, has a maximum at

$$n_\alpha^{max} = \mu + \sqrt{2\sigma^2 ln(\frac{\varepsilon_C}{\sqrt{2\pi}\cdot cost_C(\alpha)\cdot\sigma})}.$$

Notice that this number is decreasing in $cost_C(\alpha)$. We denote the value of $u_C(\alpha,n)$ at this point by $u_C^{max,\alpha}$, where:

$$u_C^{max,\alpha} = v_C^* - n_\alpha^{max}\cdot cost_C(\alpha) - \frac{1}{2}[1-erf(\frac{n_\alpha^{max}-\mu}{\sigma\sqrt{2}})]\cdot\varepsilon_C$$

$u_C^{max,\alpha}$ is positive and is increasing in σ and decreasing in $cost_C(\alpha)$.

Utility of the authority player. If the coercer chooses to coerce n voters and the security measure α is implemented, the utility of the authority in normal form transformation is:

$$u_A(\alpha,n) = v_A^* - imp(\alpha) - \frac{1}{2}[1+erf(\frac{n-\mu}{\sigma\sqrt{2}})]\cdot\varepsilon_A.$$

Notice that if we fix n, this function is decreasing in $imp(\alpha)$.

Normal form model. We can consider two cases: If $cost_C(\alpha) > \frac{\varepsilon_C}{\sigma\sqrt{2}}$ then the best response for the coercer is 0, and otherwise his best response is n_α^{max}. We assume that from α_0 to α_{m^*-1}, it holds that $cost_C(\alpha) < \frac{\varepsilon_C}{\sigma\sqrt{2}}$ and from α_{m^*} on, it holds that $cost_C(\alpha) > \frac{\varepsilon_C}{\sigma\sqrt{2}}$.

Figure 4.4 shows the strategic game model for the normal distribution of n^*. For the choices of the authority, we have only shown four security measures: α_0, α_{m^*-1}, α_{m^*} and α_{Max}. For the choices of the coercer, we only included the ones that are the best responses to one of the depicted choices of the authority. The choice 0 is the best response for the coercer when authority chooses any security measure from α_{m^*} on. The choice n_0^{max} is the best response when authority chooses α_0, and the choice $n_{m^*-1}^{max}$ is the best response when authority's choice is α_{m^*-1}. For each choice of the authority, the best response of the coercer is depicted in red, and for each choice of the coercer the best response of the authority is shown in blue.

The game has a pure Nash equilibrium at (α_0, n_0^{max}), which clearly is not a good result for the authority player. However, if the implementation cost of the security measure α_{m^*} is less than the expected damage that player A gets from the coercion at the Nash equilibrium, i.e if $imp(\alpha_{m^*}) < \frac{1}{2}[1 + erf(\frac{n_0^{max}-\mu}{\sigma\sqrt{2}})] \cdot \varepsilon_A$, then the authority can use a Stackelberg equilibrium at $(\alpha_{m^*}, 0)$ by committing itself to choose the method α_{m^*} and to make this commitment public.

Now consider that the authority cannot, or does not prefer to implement α_{m^*} or more secure privacy methods (for example because of the high cost of it) and the strongest security measure that can be implemented is a suboptimal security measure α_{m^*-1}. By announcing his choice and committing to it, the authority can achieve an equilibrium at $(\alpha_{m^*-1}, n_{m^*-1}^{max})$. In this equilibrium the estimated cost of a successfully coerced election for the authority ($\frac{1}{2}[1 + erf(\frac{n-n_\mu}{\sigma\sqrt{2}})] \cdot \varepsilon_A$) is lower than ones in the pure Nash equilibrium of the game. If this reduction of cost is worthwhile for the authority (in comparison to the extra implementation cost of α_{m^*-1} comparing to α_0), the authority can benefit from announcing and committing to his strategy even in a suboptimal security measure.

Notice that by increasing the uncertainty about the number of needed votes to buy, i.e., by increasing σ, the value of m^* decreases. It means that the Stackelberg equilibrium can be moved to a one with lower implementation cost for the authority. This may suggest that the authority can

	0	$n^{max}_{m^*-1}$	n^{max}_0
α_0	$(v_A^* \, , v_C^* - \varepsilon_C)$	$(u_A(\alpha_0, n^{max}_{m^*-1}) \, , u_C(\alpha_0, n^{max}_{m^*-1}))$	$(u_A(\alpha_0, n^{max}_0) \, , u_C^{max,\alpha_0})$
α_{m^*-1}	$(v_A^* - imp(\alpha_{m^*-1}) \, , v_C^* - \varepsilon_C)$	$(u_A(\alpha_{m^*-1}, n^{max}_{m^*-1}) \, , u_C^{max,\alpha_{m^*-1}})$	$(u_A(\alpha_{m^*-1}, n^{max}_0) \, , u_C(\alpha_{m^*-1}, n^{max}_0))$
α_{m^*}	$(\mathbf{v_A^* - imp(\alpha_{m^*}), v_C^* - \varepsilon_C})$	$(u_A(\alpha_{m^*}, n^{max}_{m^*-1}) \, , u_C(\alpha_{m^*}, n^{max}_{m^*-1}))$	$(u_A(\alpha_{m^*}, n^{max}_0) \, , u_C(\alpha_{m^*}, n^{max}_0))$
α_{Max}	$(\mathbf{v_A^* - imp(\alpha_{Max}), v_C^* - \varepsilon_C})$	$(u_A(\alpha_{Max}, n^{max}_{m^*-1}) \, , u_C(\alpha_{Max}, n^{max}_{m^*-1}))$	$(u_A(\alpha_{Max}, n^{max}_0) \, , u_C(\alpha_{Max}, n^{max}_0))$

Figure 4.4 Incomplete information game model, where the number of voters needed to coerce is estimated by a normal probability distribution. The Nash equilibrium is is at $(\alpha_0, b_{n_0^{max}})$, and the Stackelberg equilibrium for the authority is shown in bold.

in fact benefit from restrictions on making *very accurate* pollings be available to the public before the election.

4.4 Summary

In this chapter, we looked at simple social models of coercion-resistance in voting procedures. The models are two-person non-zero-sum noncooperative games, where one player represents the society and the other a potential coercer in the election. The models are arguably extremely simple. Still, even at this stage, some interesting patterns can be observed. Most importantly, we show that in all games that we consider, Stackelberg equilibrium is different from Nash equilibrium. In other words, it is in the interest of the society *not* to adapt to the expected strategy of the coercer. Rather, the society should decide on its coercion-resistance policy in advance. Moreover, the Stackelberg equilibrium does not coincide with maxmin, which suggests that the society will benefit from announcing its policy openly. This way, the coercer is best off when refraining from coercion altogether.

Chapter 5

Strategies and Information Flow: Preliminaries

Information flow security is one of the methods used for preventing insecure information propagation in systems. In the literature of information flow security, the term *noninterference* applies to a class of formal security properties, with the intuition that an agent at a high-security level is unable to interfere with an agent at a lower security level. The term was first introduced in the seminal work by Goguen and Meseguer [GM82] as a formalisation of information flow security. The concept can be informally described as follows: one group of users, using a certain set of actions, is noninterfering with another group of users if what the first group does has no effect on what the second group of users can see. The idea is to prevent any information about the behaviour of the first group (which we call *H* players) to flow to the second group (which we call *L* players).

This chapter serves as a preliminary for Chapters 6 and 7, where we build our proposals around the standard notion of noninterference by Goguen and Meseguer [GM82].

Outline of the chapter. In Section 5.1 first we have a brief literature review on noninterference and its variants. Then we present the transition network model, similar to the one used in the original definition of noninterference in the seminal work of Goguen and Meseguer [GM82]. Subsequently, in this section, we give a definition of standard noninterference using the transition network model. We also present the unwinding relation as a characterisation for the noninterference property. Section 5.2 contains the definition of the strategy of players in the transition network model. In Section 5.3 we discuss how to define temporal goals of players and sure-winning strategies in this model. Finally, Section 5.4 concludes this chapter.

5.1 Information Flow Security and Noninterference

From its appearance in [GM82], noninterference has been vastly used to define confidentiality properties in programs and concurrent processes. Since then, several variations have been suggested for the concept, such as *nondeducibility* [Sut86], *noninference* [O'H90], and *restrictiveness* [McC88]. Although noninterference was originally introduced for systems modelled as finite state machines for formalising information security in operating systems verification, it was later redefined, generalised, and extended in the framework of process algebras [All91, RWW94, Ros95, RG99, RS01] to describe the security properties of concurrent processes. Noninterference and its variants have been studied from different perspectives. Some works dealt with composability of noninterference, e.g., in [McC88, ZL95, SS09]. Another group of papers studied the properties of intransitive noninterference, e.g., in [RG99, BP03, vdM07, EvdMZ12]. Probabilistic noninterference and quantitative noninterference have been investigated, e.g., in [GI90, WJ90, MM03, PHW04, LZ05, Smi09].

We build our proposals in Chapers 6 and 7 around the standard notion of noninterference by Goguen and Meseguer [18]. We use the *transition network model*, similar to the one in the work of Goguen and Meseguer, to represent the interaction between actions of different agents and to define the security properties. We have chosen this model as the basis for our proposals because it is relatively easy to express game theoretic concepts such as the goals and the strategies of the players in it. Accordingly, defining security notions that are related to these game theoretical concepts is more straightforward in the transition network models. However, we should also mention the restrictions that these models bring about. Firstly, Goguen and Meseguer's models define agents' observations based on the states only, whereas it is sometimes more convenient to model the information flow due to directly observing each others' actions. Secondly, The models are fully asynchronous in the sense that if each user "submits" a sequence of actions to be executed then every interleaving of the submitted sequences can occur as the resulting behaviour of the system. No synchronisation is possible. Thirdly, the models are "total on input" (each action label is available to every user in every state), and hence no synchronisation mechanism can be encoded via the availability of actions.

Especially the last two features imply that models of Goguen and Meseguer allow for representation of a very limited class of systems. More expressive classes of models include various kinds of transition systems [WN95], concurrent programs [KVW00], interpreted systems [FHMV95], reactive modules [AH99], multi-agent transition networks (a.k.a. concurrent game structures) [AHK02], and many more.

5.1.1 Transition Network Model

Definition 5.1.1 (Transition network model)**.** *A multi-agent asynchronous transition network M (which from now on we simply call a transition network) is a tuple $\langle St, s_0, \mathfrak{U}, \mathfrak{A}, Obs, obs, do\rangle$ where:*

- *St is the set of* states*,*
- *s_0 is the initial state,*
- *$\mathfrak{U}$ is the set of* agents *(or* users*),*
- *$\mathfrak{A}$ is the set of* actions *(or* commands*),*

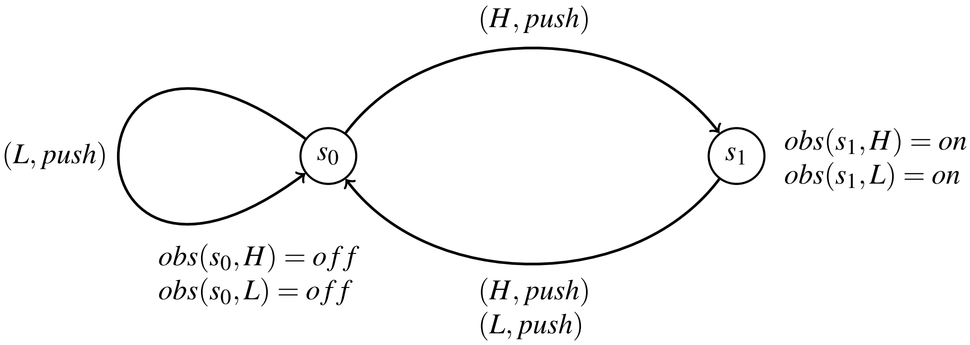

Figure 5.1 A simple transition network model for the scenario of Example 5.1.1

- *Obs is the set of possible* observations *(or* outputs*);*

- $obs : St \times \mathfrak{U} \to Obs$ *is the observation function.*

- $do : St \times \mathfrak{U} \times \mathfrak{A} \rightharpoonup St$ *is the transition function that specifies the (deterministic) outcome* $do(s,u,a)$ *of action a if it is executed by user u in state s.*

We will sometimes write $[s]_u$ instead of $obs(s,u)$. We will call a pair *(user, action)* a *personalized action*.

We construct the multi-step transition function $exec : St \times (\mathfrak{U} \times \mathfrak{A})^* \rightharpoonup St$ so that, for a finite string $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$ of personalized actions, $exec(s,\alpha)$ denotes the state resulting from execution of α from s on. We may sometimes write $s \xrightarrow{\alpha} t$ instead of $exec(s,\alpha) = t$, and $exec(\alpha)$ instead of $exec(s_0,\alpha)$.

In this chapter and throughout Chapters 6 and 7, wherever we refer to H players and L players, unless explicitly stated otherwise, we assume that H and L are disjoint and partition $\mathfrak{U}$.

Example 5.1.1. *Consider an engine with a very simple push down button for turning it on or off. The button has a sensor to distinguish between the H and the L players. The H player can turn the engine on or off by pushing the button. The L player though, can only turn the engine off. If L player* *pushes the button while the engine is off, the state of the engine does not change. We can model this scenario as a transition network* $M = \langle St, s_0, \mathfrak{U}, \mathfrak{A}, Obs, obs, do\rangle$ *where* $St = \{s_0, s_1\}$, $\mathfrak{U} = \{H, L\}$, $\mathfrak{A} = \{push\}$, $do(s_0, H, push) = s_1$, $do(s_1, H, push) = s_0$, $do(s_0, L, push) = s_0$, $do(s_1, L, push) = s_0$, $Obs = \{on, off\}$, $obs(s_0, H) = off$, $obs(s_1, H) = on$, $obs(s_0, L) = off$, *and* $obs(s_1, L) = on$. *Figure 5.1 shows this transition network.*

Definition 5.1.2 (Purge function). *If* $U \subseteq \mathfrak{U}$, $A \subseteq \mathfrak{A}$, *and* $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$, *then by* $Purge_U(\alpha)$ *we mean the subsequence of* α *obtained by eliminating all the pairs* (u, a) *with* $u \in U$. *Also* $Purge_{U,A}(\alpha)$ *denotes the subsequence of* α *obtained by eliminating all the pairs* (u, a) *with* $u \in U$ *and* $a \in A$.

In general, in a transition network model a state can be reached after different sequences of personalised actions. So the current state of the model cannot tell exactly what has happened so far. That's why sometimes it is easier to study the behaviour of the agents using *the tree unfolding* of a transition network model because in a tree unfolding of a transition network two different histories of personalised actions always result in different nodes of the tree.

Definition 5.1.3 (Tree unfolding). *If* $M = \langle St, s_0, \mathfrak{U}, \mathfrak{A}, Obs, obs, do\rangle$ *is a transition network, the tree unfolding of* M *is a transition network* $T(M) = \langle S', s'_0, \mathfrak{U}, \mathfrak{A}, Obs', obs', do'\rangle$, *and is defined as follows:*

- $S' \subseteq St^+$,
- $s'_0 = \langle s_0\rangle$,
- *if* $s \xrightarrow{(u,a)} \bar{s}$ *and* $\langle s_0 s_1 \ldots s_m = s\rangle \in S'$, *then* $\langle s_0 s_1 \ldots s_m \bar{s}\rangle \in S'$ *and* $\langle s_0 s_1 \ldots s_m\rangle \xrightarrow{(u,a)} \langle s_0 s_1 \ldots s_m \bar{s}\rangle$, *moreover* do' *contains no other transitions than the ones defined this way.*
- $Obs' = Obs^+$,
- $obs(u, \langle s'_0\rangle) = \langle obs(u, s_0)\rangle$,

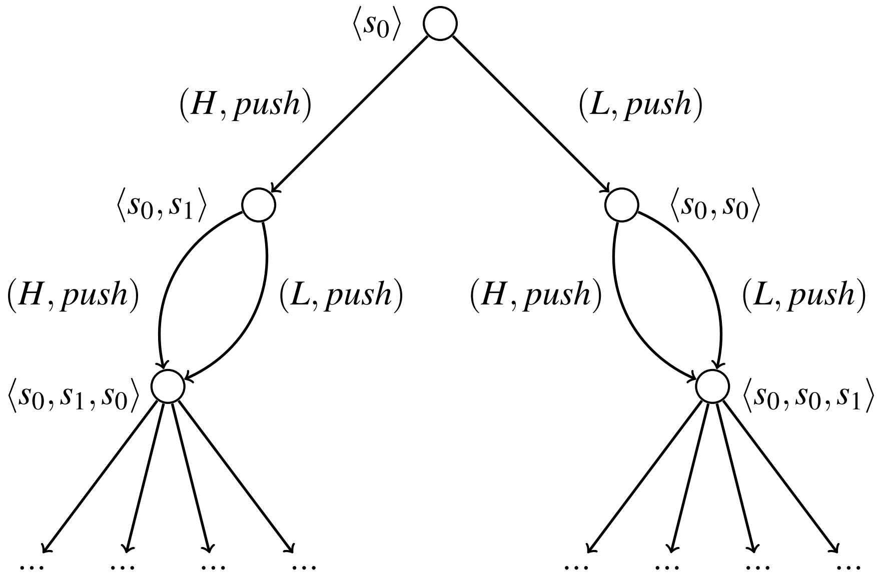
$\langle s_0 \rangle$
$(H, push)$
$(L, push)$
$\langle s_0, s_1 \rangle$
$\langle s_0, s_0 \rangle$
$(H, push)$
$(L, push)$
$(H, push)$
$(L, push)$
$\langle s_0, s_1, s_0 \rangle$
$\langle s_0, s_0, s_1 \rangle$
...

Figure 5.2 Tree unfolding of the transition network of Figure 5.1. The observations are not depicted in the picture.

- *if* $h = \langle s_0 s_1 \ldots s_m \rangle$ *, then* $obs'(u, h \circ q) = obs'(u, h)$ *if* $obs(u, s_m) = obs(u, q)$ *, and* $obs(u, h \circ q) = obs'(h) \circ obs(u, q)$ *otherwise.*

Example 5.1.2. *The tree unfolding of the transition network of the scenario of the Example 5.1.1 is shown in Figure 5.2 (the observations of the players are not depicted in the picture).*

5.1.2 Modelling Noninterference Using the Transition Network Model

We first recall the standard notion of noninterference by Goguen and Meseguer [GM82] and then discuss some remarks about the concept.

Definition 5.1.4 (Noninterference [GM82]). *Given a transition network M and sets of agents H and L, we say that H* is non-interfering with *L iff for all* $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$ *and all* $u_l \in L$ *we have* $[exec(\alpha)]_{u_l} = [exec(Purge_H(\alpha))]_{u_l}$*. We denote the property by* $NI_M(H, L)$*.*

In other words, for every sequence of actions α_H that H can execute, there is no "response" sequence from L which, interleaved with α_H, might reveal that H have done anything. Assuming that H need to hide only occurrences of some "sensitive" actions $A \subseteq \mathfrak{A}$, the concept of noninterference is refined as follows.

Definition 5.1.5 (Noninterference on sensitive actions [GM82]). *Given a transition network M, sets of agents H, L, and a set of actions $A \subseteq \mathfrak{A}$, we say that H* is non-interfering with *L* on *A iff for all $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$ and all $u_l \in L$ we have $[exec(\alpha)]_{u_l} = [exec(Purge_{H,A}(\alpha))]_{u_l}$. We denote the property by $NI_M(H,L,A)$.*

It is easy to see that $NI_M(H,L)$ iff $NI_M(H,L,\mathfrak{A})$.

Two remarks are in order. First, noninterference focuses solely on the information flow in the system. If L can detect any activity of H then noninterference is lost, regardless of the nature of the activity and the possible uses of the information. In real systems, the impact of information flow goes well beyond the information itself. Information is sought and preserved for a reason, not for its sake. Typically, L want to obtain information about H because they want to use it to achieve their goals more effectively (i.e., conclude a business contract, submit a better bid in an auction, get unauthorised access to a bank account, etc.). On the other hand, H want to protect their private information from L because their goals may be in conflict with the goals of L. This is especially the case when the Low players are labelled as "attackers" or "intruders".

Secondly, detecting H's actions may require L to engage in "diagnostic" activity, i.e., executing a sequence of actions whose only purpose is to determine if H was active or not. This becomes an issue when we see information as a resource used to obtain one's goals, rather than *the* goal of the user's activity. Then, gathering more information about H can be in conflict with what L must do in order to achieve their real goals. Thus, on one hand, L need more information to construct a better strategy for their goals, but on the other hand to acquire the information they may have to depart from the successful strategy.

5.1.3 Unwinding Relation for the Noninterference Property

Noninterference is typically characterised by the unwinding relations [GM84, Rus92, vdMZ10]. Intuitively, an unwinding relation connects states that are indistinguishable to the L agents, in the sense that L have no "diagnostic procedure" that would distinguish one from the other. Thus, if H proceed from one such state to another, no information leaks to the adversaries. Unwinding relations are important because they characterise noninterference in purely structural terms, similar to well-known bisimulation relations. Moreover, the existence of an unwinding relation is usually easier to verify than proving noninterference directly.

Definition 5.1.6 (Unwinding for Noninterference [GM84, Rus92]). *Let M be a transition network, H a set of High agents, and L a set of Low agents. Then, $\sim_{NI_L} \subseteq St \times St$ is an* unwinding relation *iff it is an equivalence relation satisfying the conditions of* output consistency (OC), step consistency (SC), *and* local respect (LR). *That is, for all states $s,t \in St$:*

(OC) *If $s \sim_{NI_L} t$ then $[s]_L = [t]_L$;*

(SC) *If $s \sim_{NI_L} t$, $u \in L$, and $a \in \mathfrak{A}$ then $do(s,u,a) \sim_{NI_L} do(t,u,a)$;*

(LR) *If $u \in H$ and $a \in \mathfrak{A}$ then $s \sim_{NI_L} do(s,u,a)$.*

Proposition 5.1.1 ([GM84, Rus92]). *$NI_M(H,L)$ iff there exist an unwinding relation $\sim_{NI_L}$ on the states of M that satisfies (OC), (SC) and (LR).*

5.2 Defining Strategies of Players

Strategy is a game-theoretic concept which captures behavioral policies that an agent can consciously follow in order to realize some objective [vNM44, LBS08]. We begin with an abstract formulation and then mention the most representative examples of strategy types. Let $T(M)$ be

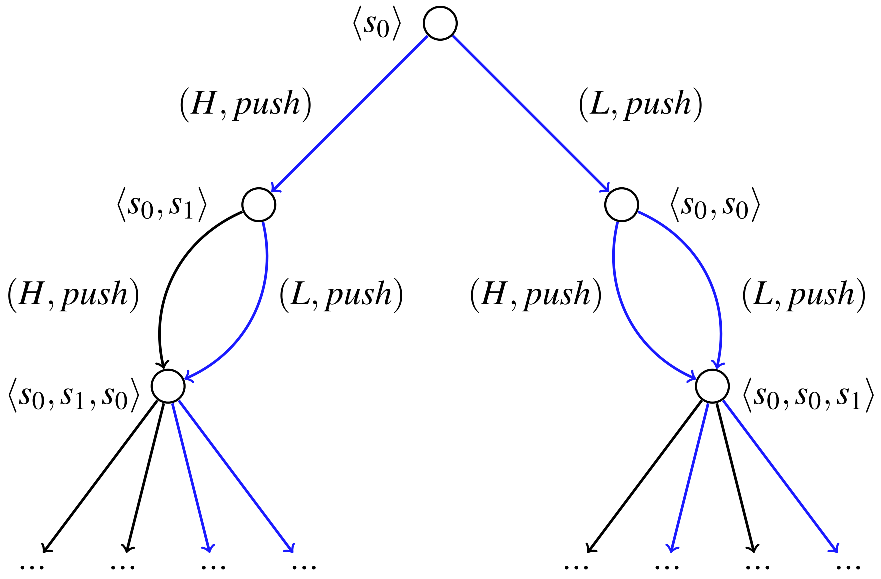
$\langle s_0 \rangle$
$(H, push)$
$(L, push)$
$\langle s_0, s_1 \rangle$
$\langle s_0, s_0 \rangle$
$(H, push)$
$(L, push)$
$(H, push)$
$(L, push)$
$\langle s_0, s_1, s_0 \rangle$
$\langle s_0, s_0, s_1 \rangle$
…
…
…
…
…
…
…
…

Figure 5.3 The subtree shown in blue corresponds to the strategy of H player in Example 5.2.1. It indicates the strategy of H never pushing the button when the engine is on.

the *tree unfolding* of M. Also if $U \subseteq \mathfrak{U}$ is a subset of agents, let T' be a *U-trimming* of tree T iff T' is a subtree of T starting from the same root and obtained by removing an arbitrary subset of transitions labeled by actions of agents from U. For the moment, we assume that each subset of agents $U \subseteq \mathfrak{U}$ is assigned a set of available coalitional strategies Σ_U. The most important feature of a strategy $\sigma_U \in \Sigma_U$ is that *it constrains the possible behaviours of the system.* We represent it formally by the *outcome function* $out_M(\sigma_U)$ that removes the executions of the system that strategy σ_U would never choose. Therefore, for every $\sigma_U \in \Sigma_U$, its outcome $out_M(\sigma_U)$ is a U-trimming of $T(M)$. We may use $nodes(out_M(\sigma_U))$ both to represent the states of $out_M(\sigma_U)$, or to represent the states of M that are the last element in the states of $out_M(\sigma_U)$.

Let h be a node in tree T corresponding to a particular finite history of interaction. We denote the sequence of personalised actions leading to h by $act^*(h)$. Furthermore, $act^*(T) = \{act^*(h) \mid h \in nodes(T)\}$ is the set of finite sequences of personalised actions that can occur in T.

Example 5.2.1. *In the scenario of Example 5.1.1, consider H player chooses the strategy of never pushing the push button when the engine is on. The subtree corresponding to this strategy is depicted in blue in Figure 5.3.*

Positional and perfect recall strategies. Strategies are usually constructed as mappings from possible situations that the player can recognise in the game, to actions of the player (or subsets of actions if we allow for nondeterministic strategies). Two types of such strategies are commonly used in the literature on game-like interaction: positional strategies and perfect recall strategies.

Positional strategies represent conditional plans where the decision is solely based on what the agents see in the current state of the system. Formally, for $u \in \mathfrak{U}$, the set of individual positional strategies of u is

$$\Sigma_u^{\mathfrak{Pos}} = \{\sigma_u : St \to \mathscr{P}(\mathfrak{A}) \mid \forall q, q' \in St \cdot [q]_u = [q']_u \Rightarrow \sigma_u(q) = \sigma_u(q')\},$$

where $\mathscr{P}(X)$ denotes the powerset of X. Notice the “uniformity” constraint which enforces that the agent must specify the same action(s) in states with the same observations. Now, coalitional positional strategies for a group of agents $U \subseteq \mathfrak{U}$ are simply tuples of individual strategies, i.e., $\Sigma_U^{\mathfrak{Pos}} = \times_{u \in U}(\Sigma_u^{\mathfrak{Pos}})$. The outcome of $\sigma_U \in \Sigma_U^{\mathfrak{Pos}}$ in model M is the tree obtained from $T(M)$ by removing all the branches that begin from a node containing state q with a personalised action $(u,a) \in U \times \mathfrak{A}$ such that $a \notin \sigma_U(q)$.

Perfect recall strategies capture conditional plans where the agents can base their decisions on the whole history of the game until that moment. Formally, the set of perfect recall strategies of agent u is

$$\Sigma_u^{\mathfrak{Rec}} = \{\sigma_u : nodes(T(M)) \to \mathscr{P}(\mathfrak{A}) \mid obs_u(h) = obs_u(h') \Rightarrow \sigma_u(h) = \sigma_u(h')\},$$

where $obs_u(.)$ denotes the observation function of $T(M)$ as defined in Definition 5.1.3. That is, what u has learned along h is equivalent to the sequence of observations she has seen, modulo removal of "stuttering" observations. Again, coalitional strategies of perfect recall for a group of agents $U \subseteq \mathfrak{U}$ are combinations of individual strategies, i.e., $\Sigma_U^{\mathfrak{Rec}} = \times_{u \in U}(\Sigma_u^{\mathfrak{Rec}})$. The outcome of $\sigma_U \in \Sigma_U^{\mathfrak{Rec}}$ in model M is the tree obtained from $T(M)$ by removing all the branches that begin from a node h with a personalised action $(u,a) \in U \times \mathfrak{A}$ such that $a \notin \sigma_U(h)$.

5.3 Temporal Goals and Winning Strategies

A goal is a property that some agents may attempt to enforce by selecting their behaviour accordingly. In game-theoretic models, goals are typically phrased as properties of the final state in the game. In our case, there is no final state – the interaction can in principle go forever. Because of that, we understand goals as properties of the full temporal trace that executes the sequence of actions selected by users. We base our approach on the concepts of *paths* and *path properties*, used in temporal specification and verification of systems [B6̈2, McN66]. Let $paths(M)$ denote the set of infinite sequences of states that can be obtained by subsequent transitions in M. Note that every sequence of states in $paths(M)$ corresponds to a sequence of states in $paths(T(M))$ and vice versa. Therefore we may sometimes abuse the notation $paths(T(M))$ to represent $paths(M)$. Consequently, we will use $paths_M(\sigma)$ as a shorthand for $paths(out_M(\sigma))$ to represent the set of infinite sequences of states in M that corresponds to the sequences of states in $out_M(\sigma)$.

Definition 5.3.1 (Temporal goal [McN66]). *A* goal *in M is any $\Gamma \subseteq paths(M)$. A goal can be equivalently seen as a subset of paths in the tree unfolding of M.*

Most common examples of such goals are safety and reachability goals.

Definition 5.3.2 (Safety and reachability goals [McN66]). *Given a set of safe states $\mathbb{S} \subseteq St$, the* safety goal $\Gamma_{\mathbb{S}}$ *is defined as $\Gamma_{\mathbb{S}} = \{\lambda \in paths(M) \mid \forall i.\lambda[i] \in \mathbb{S}\}$. Moreover, given a set of target* *states* $\mathbb{T} \subseteq St$*, the* reachability goal $\Gamma_{\mathbb{T}}$ *can be defined as* $\Gamma_{\mathbb{T}} = \{\lambda \in paths(M) \mid \exists i.\lambda[i] \in \mathbb{T}\}$.

Often, agents select their behavior in order to achieve their goal. However, it is not always possible for an agent to ensure its goal regardless of the actions selected by the other agents. If an agent (or a coalition of agents) have the ability to ensure a goal, we say that they have a surely winning strategy to achieve that goal.

Definition 5.3.3 (Winning strategies)**.** *Given a transition network M, a set of agents* $U \subseteq \mathfrak{U}$ *with goal* Γ_U*, and a set of strategies* $\Sigma_U^{\mathfrak{Rec}}$*, we say that U have a (surely winning) strategy to achieve* Γ_U *iff there exists a strategy* $\sigma_U \in \Sigma_U^{\mathfrak{Rec}}$ *such that* $paths_M(\sigma_U) \subseteq \Gamma_U$.

Notice that here we defined winning strategies, considering perfect recall for the agents. We could alternatively consider positional strategies to use in the definition. In the following chapters we use this definition unless it is explicitly stated otherwise.

5.4 Summary

In this chapter we presented the noninterference property and its formal definition using the transition network model. We then gave definitions of positional and perfect recall strategies of players. Temporal goals such as safety goals and teachability goals were also defined in this model. We then showed the definition of the winning strategies of the players regarding some temporal goals. These definitions are mainly used in Chapters 6 and 7 which contains our contributions towards Objective 2.a and 2.b of the thesis.

Chapter 6

Strategic Noninterference

As much as the notion of noninterference as an information flow security property is appealing in theory, several challenges make it less useful in practice. Noninterference is a very restrictive concept, and implementing a practical system that satisfies it entirely is hard or even impossible. It becomes even more difficult when integrating an already deployed infrastructure with an information flow policy defined on top of it (cf. [Zda04]). Last but not least, in many applications, a downward flow of information is either permitted or is inevitable in some possible runs of the system. In this chapter, we propose to restrict the property of noninterference to only a subset of possible behaviours of the system. This leads to defining the *strategic noninterference* which is a weaker notion of noninterference.

Outline of the chapter. In Section 6.1 we explain the motivating idea for our proposed strategic noninterference property. Section 6.2 gives an overview of related works. Section 6.3 presents the formal definition of strategic noninterference. Then in Section 6.4 we distinguish between the two case where the strategy of high players is public versus where it is private. In Section 6.5 we discuss the characterization of strategic noninterference in the form of unwinding relation. Finally, in Section 6.6 we give a summary of the chapter.

6.1 Motivation

The proposal follows an observation that, in most systems, not all possible behaviours actually happen. If the High players pursue a particular goal, they may do so by executing a *strategy*. Then, only those runs of the system can occur which are consistent with the strategy. But in that case, it should suffice to preserve confidentiality only in the runs that can happen when the strategy is executed. In other words, High do not need to worry about the leakage of information that their strategy prevents.

For sophisticated systems, different security strategies are available that constrain the behaviour of the system. Such a strategy can consist of implementation guidelines for programs, fixing some parameters of the software (e.g., the schedule of automated updates, time windows for entering new data, etc.), institutional policies in organisations, as well as imposing constraints on the behaviour of human components (e.g., who is allowed to enter new data). We propose that security of the system can be seen as an interplay between the goal of the system, phrased in terms of a functionality requirement, and the security strategy being used. The scenario of Example 6.1.1 shows how one may ensure noninterference in an intrinsically insecure system, by committing to a strategy which both satisfies the desired goal and prevents information flow.

Example 6.1.1 (Motivating Example)**.** *Consider the following scenario: A health care system is responsible for gathering medical data from the hospitals in the area and storing them on the servers of the centre. The centre also provides limited internet access for public users who can run allowed queries on the database. The querying interface is accessible all of the time. Moreover, the data centre runs an updating procedure whenever new data is available at one of the hospitals. To ensure the integrity of the answers, the querying interface returns "out of service" while the update is running.*

Unfortunately, it has turned out that a user may be able to relate the time of the update (= the time of observing the "out of service" message) to the hospital from which the data comes. He *may then gain unauthorised information about the hospital by checking the results of the queries before and after the update.*

The data centre provides multiple functionalities (storing, updating, and providing access to the data). Moreover, requirements on the functionalities can be specified differently. An important observation is that depending on the actual functionality requirement, there might exist a strategy that fulfils the requirement and *satisfies a given security property (in our case, noninterference). Consider, for instance, the following condition:* "the system should be updated as soon as new data is available, and the querying interface should be running all day". *It is easy to see that, for this functionality requirement, the system is bound to be vulnerable. More formally, no strategy satisfies the requirement and at the same time guarantees noninterference. However, if the functionality requirement is changed to a weaker one:* "the system should be updated at most 24 hours after new data is available, and the querying interface should be running at least 22 hours a day", *then such a strategy exists. The strategy can be closing the interface for one hour every day, and postponing the updates to the nearest closing time of the interface.*

6.2 Related Works

A short literature review on the noninterference properties can be found in Section 5.1. Out of all the works, only [RS01] comes closer to our proposal, as the authors suggest that, for systems that do not satisfy noninterference in general, the property can be possibly restored for a suitably constrained version of the system. However, the behavioural constraint has to be given explicitly, and it can be of an entirely abstract nature. In particular, it does not have to specify an executable strategy for any participants. Moreover, the functionality-related side (i.e., goals) is not treated explicitly in [RS01].

When reasoning about information leakage, it is important to distinguish between two methodological views on confidentiality. According to the first view, the Low users may attempt to read directly or deduce indirectly information that they are not authorised to obtain, and they are trying to do this on their own. The second view assumes possible cooperating agents among the High players, for example, malicious spy processes, that help the Low players to get the unauthorised information. This is usually done through *covert channels* [Lam73, WJ90]. In our approach, we assume that either the High players are not malicious, or the commitment mechanism is powerful enough so that even malicious players follow the selected strategy. We should also mention that our proposal is inherently different from *nondeducibility on strategies* [WJ90]. While in [WJ90] strategies are considered as a means to transfer information from the High player to the Low player, in our approach it is used by the High player to prevent the leakage of information.

6.3 Strategic Noninterference: Definition

Our main idea can be summarised as follows. If the High agents H are going to behave in a certain way, they do not need to worry about information leakage in *all* executions of the system but only in those executions that can happen. In particular, if H execute strategy σ_H then they should not care about the traces that are outside the outcome traces of σ_H. Moreover, the agents can choose σ_H in such a way that they avoid leaks. This leads to the following attempt at refining noninterference for agents who play strategically.

Definition 6.3.1 (Strategic Noninterference, first attempt)**.** *Given a transition network M, a set of High agents H with coalitional strategies Σ_H, a set of Low agents L, and a set of "sensitive" actions A, we say that H* is strategically non-interfering with L on A *iff there exists a strategy $\sigma_H \in \Sigma_H$ such that for all $\alpha \in act^*(out_M(\sigma_H))$ and all $u_l \in L$ we have $[exec(\alpha)]_{u_l} = [exec(Purge_{H,A}(\alpha))]_{u_l}$.*

Unfortunately, the above definition is not very practical. True, in many cases the High agents could avoid leakage of information – for instance, by refraining from doing anything but the most

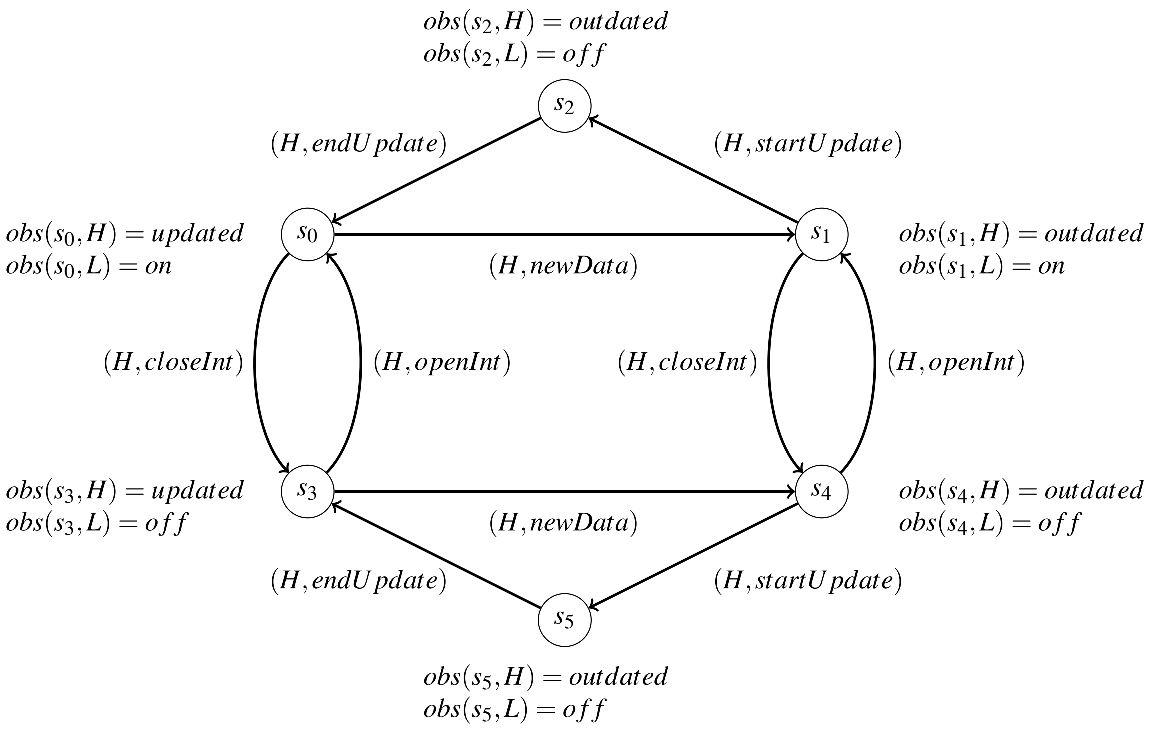

Figure 6.1 Transition network for the healthcare example. Reflexive arrows for transitions that do not change the state of the system are omitted from the picture

conservative actions. In that case, however, they would never obtain what they want. Thus, we need to take into account the *goals* of H in the definition of noninterference.

We can now propose a weaker concept of noninterference, parameterised with the goal that the High agents pursue.

Definition 6.3.2 (Strategic Noninterference)**.** *Given a transition network M, a set of High agents H with goal Γ_H and coalitional strategies Σ_H, a set of Low agents L, and a set of "sensitive" actions A, we say that H* is strategically non-interfering with L on actions A for goal Γ_H *iff there exists a strategy $\sigma_H \in \Sigma_H$ such that: (i) $paths_M(\sigma_H) \subseteq \Gamma_H$, and (ii) for every $\alpha \in act^*(out_M(\sigma_H))$ and $u_l \in L$ we have $[exec(\alpha)]_{u_l} = [exec(Purge_{H,A}(\alpha))]_{u_l}$.*

We will denote the property by $SNI_M(H,L,A,\Gamma_H)$.

Example 6.3.1 (Strategic noninterference)**.** *Consider the model in Figure 6.1 for the health care scenario from Example 6.1.1. There are two agents H and L, and the initial state is s_0. The possible observations for agent H are updated and outdated, showing if the data centre is up-to-date or not. The possible observations for agent L are on and off, showing if L sees the working interface or the "out of service" message. The available actions are: newData used by H to signal that new data is available from a hospital, startUpdate used by H to start the updating process, endUpdate used by H to finish the process, openInt and closeInt used by H to open and close the interface, and query used by L to run a query.*

Let $A = \{newData, startUpdate, endUpdate\}$. Clearly, it is not the case that H noninterferes with L on A, because $Purge_{H,A}(\langle (H, newData), (H, startUpdate)\rangle = \langle\rangle$, but $[s_2]_L \neq [s_0]_L$. However, if the goal Γ_H is defined as the system being updated after any opening of the interface, then player H can obtain Γ_H by avoiding action startUpdate in state s_1 and avoiding openInt in s_4. For this strategy, H's behaviour is noninterfering with L on A. □

Note that the variant of strategic noninterference from Definition 6.3.1 is captured by $SNI_M(H, L, A, paths(M))$. Moreover, the following is straightforward:

Proposition 6.3.1. *$SNI_M(H, L, A, \Gamma_H)$ if and only if there exists $\sigma_H \in \Sigma_H$ such that $paths_M(\sigma_H) \subseteq \Gamma_H$ and $NI_{out_M(\sigma_H)}(H, L, A)$.*

6.4 Private vs. Public Strategies

According to Definition 6.3.2, L can only use what they observe to determine if H have done a sensitive move. We implicitly assume that L do not know the strategy being executed by H; in this sense, the strategy of H is private. Another possibility is to assume that L are aware of the strategy of H. Then, L can detect in two ways that an action of H has occurred: (i) by getting to an observation that could not be obtained without an interleaved action from H, or (ii) by passing through a state where H's strategy forces H to execute something. In the model we use, where the system is asynchronous and total on actions for all players, the condition (ii) can only happen if the strategy of H includes some restriction on some of its non-sensitive actions. In this situation, those actions become information-revealing if the L player knows the strategy of H.

It is often appropriate to assume that H's strategy is known to the adversaries. This can be adopted as a worst case assumption, e.g., when a long-term pattern of H's behaviour is used by L to predict their future strategy. A similar situation arises when H's goals (or incentives) are easy to guess. It is also known that announcing a strategy publicly and committing to it can sometimes increase security, especially in case of a government agency (cf. e.g. [KYK^{+}11], or Chapter 4 of this thesis).

Definition 6.4.1 (Strategic Noninterference in Public Strategies)**.** *Given a transition network M, a set of High agents H with goal Γ_H and coalitional strategies Σ_H, a set of Low agents L, and a set of "sensitive" actions $A \subseteq \mathfrak{A}$, we say that H is strategically non-interfering with L on A for goal Γ_H* in public strategies *iff there exists a strategy $\sigma_H \in \Sigma_H$ such that: (i) $paths_M(\sigma_H) \subseteq \Gamma_H$, and (ii) for every $\alpha \in act^*(out_M(\sigma_H))$ and $u_l \in L$ we have that $[exec(\alpha)]_{u_l} = [exec(Purge_{H,A}(\alpha))]_{u_l}$ and $Purge_{H,A}(\alpha) \in act^*(out_M(\sigma_H))$.*

We will denote the property by SNI-Pub$_M(H,L,A,\Gamma_H)$.

Example 6.4.1 (Public vs. private strategies)**.** *Consider the transition system in Figure 6.2, with two agents H and L and the initial state s_0. The set of possible actions is $\mathfrak{A} = \{a,b,c,d\}$ and the set of sensitive actions is $A = \{c,d\}$. The observations for both agents are shown in the picture. Let goal Γ_H be that whenever system goes to s_3, it must have been at some previous point in s_2. Agent H can obtain this goal by using strategy σ_1 of avoiding action b in s_0 and avoiding action a in s_1. Moreover, when using σ_1, H noninterferes with L on A in private strategies but not in public strategies. To see why, note that if $\alpha = \langle (H,c)(H,b) \rangle$ then $\alpha \in act^*(out(\sigma_1))$ but $Purge_{H,A}(\alpha) = \langle (H,b) \rangle$ is not in $act^*(out(\sigma_1))$. Therefore, although H can obtain Γ_H by using*

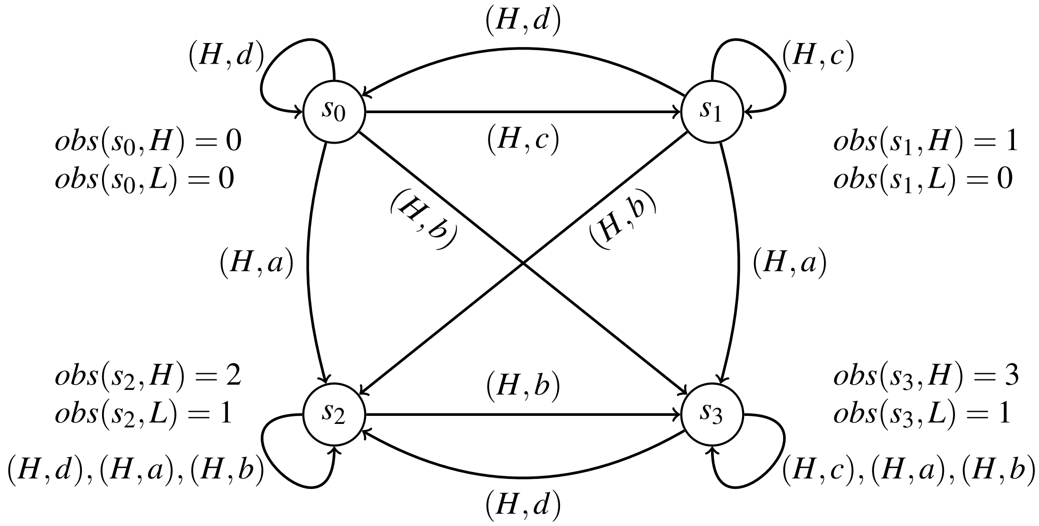

Figure 6.2 Noninterference in public and private strategies

strategy σ_1 while preserving noninterference, the security can be only achieved if L does not know the strategy of H.

Now consider goal Γ'_H being that whenever the system is at s_0 or s_1, the state changes in the next step if an action from H gets executed. H can obtain this goal by using strategy σ_2 of avoiding action d in s_0 and avoiding action c in s_1. Using σ_2, H noninterferes with L on A. Moreover, the system satisfies noninterference in public strategies, so the agent H may let the agent L know his strategy σ_2 without compromising noninterference. □

Strategic noninterference is a weaker notion than ordinary noninterference. Out of the two notions of SNI, noninterference in public strategies is stronger.

Proposition 6.4.1. *$NI_M(H,L,A) \Rightarrow$ SNI-Pub$_M(H,L,A,\Gamma_H) \Rightarrow$ SNI(H,L,A,Γ_H). The converse implications do not universally hold.*

Proof. The implications are straightforward from the definitions. Non-validity of the converse implications follows from Examples 6.3.1 and 6.4.1. □

Models of Goguen and Meseguer allow only to represent systems that are fully asynchronous and where all actions are available to each user in each state. As it turns out, revealing H's strategy makes a difference only when H have both sensitive and insensitive actions. Thus, if H are to conceal all their actions then it actually doesn't matter whether their strategy is publicly known. Before showing this formally, we make the following observations.

Observation 6.4.2. *In a transition network M, if $u \in \mathfrak{U}$, $\sigma_H \in \Sigma_H$, and $u \notin H$ then for all $\alpha \in act^*(out_M(\sigma_H))$ and $a \in \mathfrak{A}$ we have that $\alpha \circ (u,a) \in act^*(out_M(\sigma_H))$, where $\alpha \circ (u,a)$ denotes concatenation of α and (u,a). This is because M is asynchronous and in each state any agents may get its action executed before the others. On the other hand, σ_H only restricts the behaviour of agents in H. Therefore any outgoing transition from a node in $T(M)$ by an agent outside H must remain in the trimmed tree given by $out_M(\sigma_H)$.*

Observation 6.4.3. *In the tree given by $out_M(\sigma_H)$, sequences of actions are prefix-closed. In other words, for every sequence α, we have $\alpha \circ (u,a) \in act^*(out_M(\sigma_H)) \Rightarrow \alpha \in act^*(out_M(\sigma_H))$.*

Proposition 6.4.4. *$SNI_M(H,L,\mathfrak{A},\Gamma_H)$ iff $SNI\text{-}Pub_M(H,L,\mathfrak{A},\Gamma_H)$.*

Proof. By Proposition 6.4.1 we have that $SNI\text{-}Pub_M(H,L,A,\Gamma_H)$ implies $SNI_M(H,L,A,\Gamma_H)$. For the other direction it suffices to show that if $SNI_M(H,L,\mathfrak{A},\Gamma_H)$ then for every $\alpha \in act^*(out_M(\sigma_H))$ and $\sigma_H \in \Sigma_H$ it holds that $Purge_{H,\mathfrak{A}}(\alpha) \in act^*(out_M(\sigma_H))$. We prove this by induction on the size of α.

Induction base: if $\alpha = \langle\rangle$, then $Purge_{H,A}(\alpha) = \langle\rangle$ and also $\langle\rangle \in act^*(out_M(\sigma_H))$, therefore $Purge_{H,A}(\alpha) \in act^*(out_M(\sigma_H))$.

Induction step: We want to show that if

$$\text{(I) } \alpha \in act^*(out_M(\sigma_H)) \Rightarrow Purge_{H,A}(\alpha) \in act^*(out_M(\sigma_H))$$

then for all $u \in \mathfrak{U}$ and $a \in \mathfrak{A}$:

$$\text{(II) } (\alpha \circ (u,a)) \in act^*(out_M(\sigma_H)) \Rightarrow Purge_{H,A}(\alpha \circ (u,a)) \in act^*(out_M(\sigma_H))$$

We prove it as follows. If (I) then either $\alpha \notin act^*(out_M(\sigma_H))$, in which case by Observation 6.4.3 we have $\alpha \circ (u,a) \notin act^*(out_M(\sigma_H))$ and therefore (II) is true; or $Purge_{H,A}(\alpha) \in act^*(out_M(\sigma_H))$, in which case we have two possibilities: (a) If $u \in H$ then $Purge_{H,A}(\alpha \circ (u,a)) = Purge_{H,A}(\alpha)$. We assumed that $Purge_{H,A} \in act^*(out_M(\sigma_H))$ so $Purge_{H,A}(\alpha \circ (u,a)) \in act^*(out_M(\sigma_H))$ and hence (II) is true. (b) If $u \notin H$ then $Purge_{H,A}(\alpha \circ (u,a)) = Purge_{H,A}(\alpha) \circ (u,a)$. This together with Observation 6.4.2, $u \notin H$ and $Purge_{H,A}(\alpha) \in act^*(out_M(\sigma_H))$ implies that $Purge_{H,A}(\alpha.(u,a)) \in act^*(out_M(\sigma_H))$, therefore (II) is true. □

6.5 Formal Characterization of Strategic Noninterference

As mentioned in Chapter 5, noninterference is typically characterised through so called unwinding relations [GM84, Rus92, vdMZ10]. In this part, we try to characterize strategic noninterference in a similar way. That is, we look for unwinding relations corresponding to strategies that obtain a given goal and at the same time prevent information leakage. There are two possible perspectives to this. First, we can look for unwinding relations whose existence corresponds to *existence* of a suitable strategy. Then, we may look for unwindings whose existence guarantees strategic noninterference *for a given strategy.*

6.5.1 Unwinding for Strategic Noninterference

We begin with the following negative result.

Proposition 6.5.1. *There is no succinct characterization of strategic noninterference with respect to goals definable in Linear Time Logic.*

Proof. We base our proof on the widely conjectured assumption that **P≠NP**. We show that if there exist a succinct characterization of strategic noninterference, then it contradicts this assumption.

Suppose, to the contrary, that there exists a deterministic[1] condition Φ which: (i) is of polynomial size with respect to the size of the model and the length of the goal formula, and (ii) guarantees that $SNI_M(H,L,\mathfrak{A},\Gamma)$ iff there is an unwinding relation satisfying Φ for $M,H,L,\mathfrak{A},\Gamma$. Note that the model checking problem for Linear Time Logic can be embedded in checking strategic noninterference by assuming that $H=\emptyset$ and that L have the same observation in every state. Then, $SNI_M(H,L,\mathfrak{A},\Gamma)$ iff Γ is satisfied on every possible path in M. If we have a polynomial time algorithm for the construction of the unwinding relation, and knowing that Φ is also polynomial in the size of the model, this would give us a nondeterministic polynomial-time procedure for model checking Linear Time Logic (because the composition of two polynomials is another polynomial). But as we already know that model checking Linear Time Logic is **PSPACE**-complete [Sch03], it implies that **PSPACE =P**, which in turn implies **P=NP**, which contradicts our assumption. □

It is evident from the proof that the impossibility stems from the hardness of finding a strategy that obtains a given goal, and not necessarily from the noninterference part. We will now show that strategic noninterference can indeed be succinctly characterised for a particular class of goals, namely safety goals.

Definition 6.5.1 (Unwinding Relation for Safety Goal)**.** *Let M,H,L be as usual, and $\Gamma_{\mathbb{S}}$ be a safety goal with safe states $\mathbb{S}\subseteq St$. Moreover, let $reach(U)=\{s\mid \exists\alpha\in(U,\mathfrak{A})^*\ ,\ s=exec(\alpha)\}$ denote the set of reachable states for agents U. We say that $\sim_{\Gamma_{\mathbb{S}}}\subseteq St\times St$ is an unwinding relation for $\Gamma_{\mathbb{S}}$ iff $\sim_{\Gamma_{\mathbb{S}}}$ satisfies the following properties:*

($OC_{\mathbb{S}}$) *For all $s,t\in reach(L)$, if $s\sim_{\Gamma_{\mathbb{S}}} t$ then $[s]_L=[t]_L$;*

($SC_{\mathbb{S}}$) *For all $s,t\in reach(L)$, $u\in L$, and $a\in\mathfrak{A}$, if $s\sim_{\Gamma_{\mathbb{S}}} t$ then $do(s,u,a)\sim_{\Gamma_{\mathbb{S}}} do(t,u,a)$.*

[1] By “deterministic”, we essentially mean “quantifier-free”. Note that quantification over elements of the model (e.g., states, agents, and actions) is not a problem, since it can always be unfolded to a quantifier-free form by explicitly enumerating all the possible values. Such an unfolding incurs only polynomial increase of the size of Φ.

Proposition 6.5.2. *SNI*$(H,L,\mathfrak{A},\Gamma_{\mathbb{S}})$ *iff* $reach(\mathfrak{U}\setminus H) \subseteq \mathbb{S}$ *and there exists an unwinding relation* $\sim_{\Gamma_{\mathbb{S}}}$ *for the safety goal* $\Gamma_{\mathbb{S}}$.

Proof. "$\Leftarrow$" Suppose that $reach(\mathfrak{U}\setminus H) \subseteq \mathbb{S}$ and there exists an unwinding relation $\sim_{\Gamma_{\mathbb{S}}}$. We show that there exists a strategy σ_H for agents H such that (i) $path_M(\sigma_H) \subseteq \Gamma_{\mathbb{S}}$, and (ii) for every $\alpha \in act^*(out_M(\sigma_H))$ and $u_l \in L$ we have $[exec(\alpha)]_{u_l} = [exec(Purge_{H,\mathfrak{A}}(\alpha))]_{u_l}$. We choose σ_H to be a positional strategy defined as $\sigma_H(s) = \emptyset$ for all $s \in St$.

i) By the definition of σ_H, we know that $act^*(out_M(\sigma_H)) \subseteq (\mathfrak{U}\setminus H,\mathfrak{A})^*$. This together with $reach(\mathfrak{U}\setminus H) \subseteq \mathbb{S}$ and the definition of safety goal, implies that $path_M(\sigma_H) \subseteq \Gamma_{\mathbb{S}}$.

ii) For every $\alpha \in act^*(out_M(\sigma_H))$ and $u_l \in L$, we have $\alpha \in (\mathfrak{U}\setminus H,\mathfrak{A})^*$ by (i), and hence $Purge_{H,\mathfrak{A}}(\alpha) = \alpha$. Therefore $[exec(Purge_{H,\mathfrak{A}}(\alpha)]_{u_l} = [exec(\alpha)]_{u_l}$.

By i) and ii) we have that *SNI*$(H,L,\mathfrak{A},\Gamma_{\mathbb{S}})$ holds.

"$\Rightarrow$" Suppose that *SNI*$(H,L,\mathfrak{A},\Gamma_{\mathbb{S}})$, and σ_H is a strategy that satisfies the conditions of strategic noninterference. We show that there exists an unwinding relation $\sim_{\Gamma_{\mathbb{S}}}$ for the safety goal $\Gamma_{\mathbb{S}}$. Let $\sim_{\Gamma_{\mathbb{S}}}$ be the relation such that $s \sim_{\Gamma_{\mathbb{S}}} t$ if $s,t \in nodes(out_M(\sigma_H))$ and for all $\alpha \in (L,\mathfrak{A})^*$ and $u_l \in L$, $[exec(s,\alpha)]_{u_l} = [exec(t,\alpha)]_{u_l}$. We show that $\sim_{\Gamma_{\mathbb{S}}}$ is an unwinding relation for the safety goal $\Gamma_{\mathbb{S}}$.

i) If $\alpha \in (\mathfrak{U}\setminus H,\mathfrak{A})^*$ then by Observation 6.4.2 we have that $\alpha \in act^*(out_M(\sigma_H))$, and therefore $exec(\alpha) \in \mathbb{S}$ (by strategic noninterference). So $reach(\mathfrak{U}\setminus H) \subseteq \mathbb{S}$.

ii) If we take $\alpha = \langle\rangle$, then by definition of $\sim_{\Gamma_{\mathbb{S}}}$ we have that for all $u_l \in L$ and all $s,t \in reach(L)$, $[exec(s,\alpha)]_{u_l} = [exec(t,\alpha)]_{u_l}$. So $[exec(s,\langle\rangle)]_{u_l} = [exec(t,\langle\rangle)]_{u_l}$, or $[s]_{u_l} = [t]_{u_l}$ which proves that $\sim_{\Gamma_{\mathbb{S}}}$ satisfies (OC$_{\mathbb{S}}$).

iii) Lastly, we need to prove that $\sim_{\Gamma_{\mathbb{S}}}$ satisfies (SC$_{\mathbb{S}}$). Suppose there exists $s,t \in reach(L)$, $u \in L$ and $a \in \mathfrak{A}$ such that $s \sim_{\Gamma_{\mathbb{S}}} t$ and $do(s,u,a) \not\sim_{\Gamma_{\mathbb{S}}} do(t,u,a)$. Then there exists $\alpha \in (L,\mathfrak{A})^*$ such that $[exec(do(s,u,a),\alpha)]_{u_l} \neq [exec(do(t,u,a),\alpha)]_{u_l}$ for some $u_l \in L$. It implies that $[exec(s,((u,a) \circ \alpha))]_{u_l} \neq [exec(t,((u,a) \circ \alpha))]_{u_l}$, which contradicts $s \sim_{\Gamma_{\mathbb{S}}} t$. Therefore $\sim_{\Gamma_{\mathbb{S}}}$ satisfies (SC$_{\mathbb{S}}$). □

It would be interesting to characterize strategic noninterference for other subclasses of goals in a similar way. For example reachability goals are another promising class that we leave for future work.

6.5.2 Strategy-Specific Unwinding Relations

We now turn to characterising strategic noninterference when a strategy is given as a parameter of the problem. First we define the maximum coverage of a strategy of a player in a state as the set of all actions that the strategy may allow for the player in some traces. Let σ_H be a strategy for H in M. We define the maximum coverage of σ_H in state s as $maxcover(\sigma_H, s) = \{a \in \mathfrak{A} \mid \exists \alpha \in act^*(out_M(\sigma_H)), u_h \in H.\ exec(\alpha) = s$ and $\alpha \circ (u_h, a) \in act^*(out_M(\sigma_H))\}$. Then we define a strategy to be *state-consistent* if that strategy always allows the same set of actions in a state regardless of the history of actions that resulted in that state. We call a strategy σ_H state-consistent iff for all $s \in St$, if $\alpha \in act^*(out_M(\sigma_H))$, $exec(\alpha) = s$ and $a \in maxcover(\sigma_H, s)$ then $\alpha \circ (u_h, a) \in act^*(out_M(\sigma_H))$.

Now we can define the strategy-specific unwinding relation:

Definition 6.5.2 (Strategy-Specific Unwinding Relation)**.** *Let M, H, L be as usual, Γ be a goal, and σ_H a strategy for H. We call $\sim_{\sigma_H} \subseteq St \times St$ a* strategy-specific unwinding relation for σ_H *iff it satisfies the following properties:*

(OC$_\sigma$) *For all $s, t \in nodes(out_M(\sigma_H))$ and $u \in L$, if $s \sim_{\sigma_H} t$ then $[s]_u = [t]_u$;*

(SC$_\sigma$) *For all $s, t \in nodes(out_M(\sigma_H))$, $u \in L$, and $a \in \mathfrak{A}$, if $s \sim_{\sigma_H} t$ then $do(s, u, a) \sim_{\sigma_H} do(t, u, a)$;*

(LR$_\sigma$) *For all $s \in nodes(out_M(\sigma_H))$, $u \in H$, and $a \in maxcover(\sigma_H, s)$, we have that $s \sim_{\sigma_H} do(s, u, a)$.*

Proposition 6.5.3. *Let M, H, L, Γ be as before, and σ_H be a positional strategy for H that obtains Γ (formally: $paths_M(\sigma_H) \subseteq \Gamma_H$). If there exists a strategy-specific unwinding relation for σ_H then M satisfies strategic noninterference with respect to σ_H (formally: for every $\alpha \in act^*(out_M(\sigma_H))$ and $u_l \in L$ we have that $[exec(\alpha)]_{u_l} = [exec(Purge_{H,\mathfrak{A}}(\alpha))]_{u_l}$).*

Proof. By (OC$_\sigma$) it is enough to show that for all $\alpha \in act^*(out_M(\sigma_H))$, $exec(\alpha) \sim_{\sigma_H} exec(Purge_{H,\mathfrak{A}}(\alpha))$. We prove this by induction on the size of α.

Induction base: For $\alpha = \langle\rangle$, we have $\langle\rangle \in act^*(out_M(\sigma_H))$ and $Purge_{H,\mathfrak{A}}(\langle\rangle) = \langle\rangle$. Therefore $exec(\langle\rangle) \sim_{\sigma_H} exec(Purge_{H,\mathfrak{A}}(\langle\rangle))$, because $\sim_{\sigma_H}$ is reflexive.

Induction step: Suppose that for some $\alpha \in act^*(out_M(\sigma_H))$, $exec(\alpha) \sim_{\sigma_H} exec(Purge_{H,\mathfrak{A}}(\alpha))$. We show that for any (u,a) such that $u \in L$ and $a \in \mathfrak{A}$, either $exec(\alpha \circ (u,a)) \sim_{\sigma_H} exec(Purge_{H,\mathfrak{A}}(\alpha \circ (u,a))$, or $\alpha \circ (u,a) \notin act^*(out_M(\sigma_H))$. We consider three cases:

(i) If $u \in H$ and $a \notin \sigma_H(exec(\alpha))$, then $\alpha \circ (u,a) \notin act^*(out_M(\sigma_H))$.

(ii) If $u \in H$ and $a \in \sigma_H(exec(\alpha))$, then $Purge_{H,\mathfrak{A}}(\alpha \circ (u,a)) = Purge_{H,\mathfrak{A}}(\alpha)$. By (LR$_\sigma$) we have that $exec(\alpha) \sim_{\sigma_H} exec(\alpha \circ (u,a))$. This together with induction step assumption and transitivity of $\sim_{\sigma_H}$ implies that $exec(Purge_{H,\mathfrak{A}}(\alpha)) \sim_{\sigma_H} exec(\alpha \circ (u,a))$. By substituting $Purge_{H,\mathfrak{A}}(\alpha)$ with $Purge_{H,\mathfrak{A}}(\alpha \circ (u,a))$ we have $exec(\alpha \circ (u,a)) \sim_{\sigma_H} exec(Purge_{H,\mathfrak{A}}(\alpha \circ (u,a))$.

(iii) If $u \in L$ then $exec(Purge_{H,\mathfrak{A}}(\alpha \circ (u,a))) = do(exec(Purge_{H,\mathfrak{A}}(\alpha)), u, a)$. This, together with the induction step assumption and (SC$_\sigma$), implies that $do(exec(\alpha)), u, a) \sim_{\sigma_H} do(exec(Purge_{H,\mathfrak{A}}(\alpha)), u, a)$. Therefore $exec(\alpha \circ (u,a)) \sim_{\sigma_H} exec(Purge_{H,\mathfrak{A}}(\alpha \circ (u,a))$. □

Proposition 6.5.4. *Let $M, H, L, \Gamma, \sigma_H$ be as in Proposition 6.5.3. If M satisfies strategic noninterference with respect to σ_H then there exists a strategy-specific unwinding relation for σ_H.*

Proof. Let $\sim_{\sigma_H}$ be the relation such that $s \sim_{\sigma_H} t$ if $s,t \in nodes(out_M(\sigma_H))$ and for all $\alpha \in (L,\mathfrak{A})^*$ and $u_l \in L$, $[exec(s,\alpha)]_{u_l} = [exec(t,\alpha)]_{u_l}$. We show that $\sim_{\sigma_H}$ has the conditions of strategy-specific unwinding relation for strategy σ_H.

(i) If we take $\alpha = \langle\rangle$, then by definition of $\sim_{\sigma_H}$, for all $u_l \in L$ and all $s,t \in nodes(out_M(\sigma_H)$, $[exec(s,\langle\rangle)]_{u_l} = [exec(t,\langle\rangle)]_{u_l}$, or $[s]_{u_l} = [t]_{u_l}$ which proves that $\sim_{\sigma_H}$ satisfies OC_σ.

(ii) Suppose there exists $s,t \in nodes(out_M(\sigma_H))$, $a \in \mathfrak{A}$ and $u_l \in L$ such that $s \sim_{\sigma_H} t$ but $do(s,u_l,a) \not\sim_{\sigma_H} do(t,u_l,a)$. Then there should exist $\alpha \in (L,\mathfrak{A})^*$ and $u \in L$ such that $[exec(do(s,u_l,a),\alpha)]_u \neq$

$[exec(do(t,u_l,a),\alpha]_u$. Therefore $[exec(s,(u_l,a)\circ\alpha)]_u \neq [exec(t,(u_l,a)\circ\alpha)]_u$, contradicting $s \sim_{\sigma_H} t$. So it proves that $\sim_{\sigma_H}$ satisfies (SC$_\sigma$).

(iii) Suppose that $s \in nodes(out_M(\sigma_H))$, $a \in maxcover(\sigma_H, s)$, $\alpha \in (L,\mathfrak{A})^*$, $u_l \in L$ and $u_h \in H$. Then there exist $\lambda \in act^*(out_M(\sigma_H))$ such that $exec(\lambda) = s$. By strategic noninterference property, $[exec(\lambda\circ\alpha)]_{u_l} = [exec(Purge_{H,\mathfrak{A}}(\lambda\circ\alpha_)]_{u_l}$ and $[exec(\lambda\circ(u_h,a)\circ\alpha)]_{u_l} = [exec(Purge_{H,\mathfrak{A}}(\lambda\circ(u_h,a)\circ\alpha))]_{u_l}$. We also know that $Purge_{H,\mathfrak{A}}(\lambda\circ(u_h,a)\circ\alpha) = Purge_{H,\mathfrak{A}}(\lambda\circ\alpha)$. Using these equalities we have that $[exec(\lambda\circ\alpha)]_{u_l} = [exec(\lambda\circ(u_h,a)\circ\alpha)]_{u_l}$, i.e $[exec(s,\alpha)]_{u_l} = [exec(do(s,u_h,a),\alpha)]_{u_h}$, therefore $s \sim_{\sigma_H} do(s,u_h,a)$ (by the definition of $\sim_{\sigma_H}$) and so (LR$_{\sigma_H}$) holds. □

6.6 Summary

In this chapter, we proposed how to relax the classical requirement of noninterference by taking into account a strategy that the High players may follow in order to achieve their goals. We also studied characterization of strategic noninterference through unwinding relations. While showing that the characterization result is impossible for arbitrary goals, we presented some characterisations in the form of unwinding relation for specific subclasses of goals and for the settings where a strategy is given as parameter.

Chapter 7

Effective Security

Many information flow security properties, including noninterference property, are understood as preventing any information leakage, regardless of how grave or harmless consequences the leakage can have. Even in models where each piece of information is classified as either sensitive or insensitive, the classification is "hardwired" and given as a parameter of the analysis, rather than derived from more fundamental features of the system. In this chapter, we suggest that information security is not a goal in itself, but rather a means of preventing potential attackers from compromising the correct behaviour of the system. To formalise this, we first show how two information flows can be compared by looking at the adversary's ability to harm the system. Then, we propose that the information flow in a system is effectively information-secure if it does not allow for more damage than its idealised variant based on the classical notion of noninterference.

Outline of the chapter. Section 7.1 presents a detailed explanation of the motivation and the main idea behind the contribution of this chapter. Then in Section 7.2 we give an overview on some of related works. Section 7.3 reminds some of the preliminary concepts. In Section 7.4, we define the generic concept of effective security. In Section 7.5, we look specifically at the security of information flow and show how it can be defined based on the relation between the attacker's observational capabilities and his ability to compromise the goals of the system. In Section 7.6 we extend our results to models that are not total on input. Finally, we summarise the work in Section 7.7.

7.1 Motivation and the Main Idea

Information plays multiple roles in interactions between agents (be it humans or artificial entities, e.g., software agents). First, it can be the commodity that the agents compete for; in that case, it often defines the outcome of the “interaction game”. Key exchange protocols are a good example here, as the involved honest parties strive to learn the key of the other agent while at the same time preventing any information leak to the intruder. Secondly, information can define the semantic content of an action: Typically, most actions specified in a security protocol consist in transmitting or processing some information. Thirdly, information can be a resource that enables actions and influences the outcome of the game. This is because agents need information to construct and execute plans that can be used to achieve their goals.

Most approaches to information flow security adopt the first perspective. That is, information defines the ultimate goal of the interaction. Classical information security properties specify *what* information must not leak, and *how* it could leak (i.e., what channels of information leakage are considered), but they do not give an account of *why* the information should not leak to the intruder. For example, the property of *noninterference* [GM82] assumes that the “low clearance” users cannot learn anything about the activities of the “high clearance” users. In order to violate this, the “low” users can try to analyse their observations, or execute a sequence of explorative actions of their own. *Nondeducibility on strategies* [WJ90] makes the same assumption about *what* should not leak, but also takes into account covert channels that some “high” users can use to send signals to the “low” agents according to a previously agreed code. *Anonymity* in voting [Cha81, FOO92] captures that an observer cannot learn what candidate a particular has voted for by looking at the voter's behaviour, scanning the web bulletin board, coercing the voter to hand in the vote receipt, etc.

As a consequence, the classical properties of information security can only distinguish between relevant and irrelevant information leaks if the distinction is given explicitly as a parameter, e.g., by classifying available actions into sensitive and insensitive [GM82]. However, it is usually hard (if not impossible) to obtain such a distinction based on the internal characteristics of the actions. We illustrate the point below through a real-life example.

7.1.1 Motivating Example: Phone Banking

In some phone banking services, the maiden name of the user's mother is used as a part of authentication, e.g., to change the settings of the account. That is, the user is typically asked to spell her name, birth date, current address, and her mother's maiden name in order to change the credit limit in the account, block/unblock ATM use in specified geographical areas, and so on. Note that information about one's birth date and address is fairly easy to obtain in public directories or repositories kept and marketed by various web services that require the data for registration. So, the mother's maiden name plays the role of a "strong test of identity" in this scenario.[1] Consider now a user posting an essay about some ancestor of hers on her blog, mentioning the name of the ancestor. If the essay is about the user's mother, it reveals potentially dangerous information. This is, among other things, because an intruder can use the information to: (1) access the phone banking service, (2) authenticate impersonating the user, (3) change (in the user's profile) the telephone number used for web banking password recovery and sms authentication of web banking transactions, (4) change the web banking password of the user, and finally (5) log in and transfer money from the user's account. On the other hand, if the post is about some other member of the user's

[1] This is a real-life example from with BNP Paribas in one of EU countries. For similar security questions, used by various phone or web services, cf. e.g. [Lev08].

family (father, grandmother, paternal grandfather, etc.) revealing the name of the person is probably harmless. Note that it is impossible to distinguish between the two pieces of information (say, the mother's maiden name vs. the grandmother's maiden name) based on their internal features. Both have the same syntactic structure of a single word (i.e., a string of characters with no blank spaces) and the same semantic content (a family name of a person; more precisely, the family name of the person at birth). The only difference lies in the context: the first kind of information is used in some important social procedures, while the second one is not.

7.1.2 Information as Strategic Resource

We claim that a broader perspective is needed to model and analyse appropriately such scenarios. Agents compete for information not for its sake, but for reasons that go beyond purely epistemic advantages. An intruder may want to know the password of a PayPal user to impersonate the user and steal some *real* money by making a payment to his own benefit (possibly via an account of a suitable "mule"). An industry player may need encryption keys used in internal communication between employees of its main competitor to find out about the competitor's current business strategy. A political activist needs the ability to learn the value of someone's vote in order to effectively coerce that person into voting for the candidate that the activist is rallying for. Thus, in most security scenarios, information is a resource rather than a commodity. More precisely, information is a commodity that the players compete for in an "information security game" but the game is played in the context of a "real" game where information is only a resource, enabling (some) players to achieve their non-epistemic goals. As players obtain new information, their uncertainty is reduced, and they increase their ability to choose a good strategy in the real game.

What would a *significant information leak* be in this view? To answer the question, we draw inspiration from the concept of the *value of information* from decision theory [How66]: a piece of information is worth as much as it increases the expected payoff of the player. Similarly, an information leak is significant if it increases the ability of the attacker to construct a damaging attack strategy in the real game.

7.1.3 Main Idea and Contribution of the Chapter

The main idea behind our contribution in this chapter of the thesis can be summarised as follows. We consider three questions:

- How can we evaluate the ability of an adversary to harm the goal of the system?
- How can we compare two systems with regard to the ability of attackers to damage the goal of the system in them?
- How can we know whether the ability (or inability) of the attacker to harm the goal of the system is because of some leakage of information to the attacker or not?

This chapter is structured to discuss these questions in order. First, we use the concept of *surely winning strategies* from game theory to analyse the adversary's strategic ability to disrupt the correct behaviour of the system. This can be a functionality property, a security property, or a combination of the two kinds. Also, it can arise from a goal of the "high clearance" agents or an objective assigned to the system by its designers or its owners. Preventing the attacker from having a winning attack strategy is what the designer of the system may want to achieve. We will see the *effective security* of the system as the attacker's *in*ability to come up with such a strategy. Secondly, we use the notion of *effective security* for comparing two systems by looking at the strategic ability of an adversary to harm the goal of the system. Thirdly, a successful attack strategy can exist due to flawed design of either the control flow or the information flow in the system. Here, we are interested in the latter. That is, we want to distinguish between vulnerabilities coming from the control vs. the information flow, and single out systems where redesigning the flow of information alone can make the system more secure. To this end, we define the noninterferent idealised variant of the system, which has the same control flow as the original system, but with the information reduced so that the system satisfies noninterference. Then, we define the system to be *effectively information-secure* if it is as good as its noninterfering idealised variant. As the main technical result, we show that the concept is well defined, i.e., the maximal noninterferent variant exists for every state-transition model.

7.2 Related Works

An overview of the works on information flow security and noninterference properties is given in Section 5.1. Most of these works assume that the information flow in the system is secure only when no information ever flows from High to Low players. Here, we want to discard irrelevant information leaks, and only look at the significant ones (in the sense that the leaking information can be used to construct an attack on a higher-order correctness property).

The problem of how to weaken noninterference to capture security guarantees for real systems has also been extensively studied. Most notably, postulates and policies for *declassification* (called also *information release*) were studied, cf. [SS05] for an introduction. This submission can be viewed as an attempt to determine *what information is acceptable to declassify*. In this sense, our results can useful in proposing new declassification policies and evaluating existing ones. We note, however, that the existing works on declassification are mainly concerned by the question *what* information can be released, *when*, *where*, and by *whom*. In contrast, we propose an argument for *why* it can be released. Moreover, declassification is typically about intentional release of information, whereas we do not distinguish between intentional and accidental information flow. Finally, the research on declassification assumes that security is defined by some given "secrets" to be protected. In our approach, no information is intrinsically secret, but the information flow is harmful if it enables the attacker to gain more strategic ability against the goals of the system.

Parameterised noninterference [GM04] can be seen as a theoretical counterpart of declassification, where security of information flow is parameterised by the analytic capabilities of the attacker. Again, that research does not answer why some information must be kept secret while some other needs not, and in particular it does not take the strategic power of the attacker into account.

Economic and strategic analysis of security properties is a growing field in general, cf. e.g. [AMNO07, MA11, DR07, BM07, YKK$^+$10, KYK$^+$11]. A number of papers have applied game-theoretic concepts to define the security of information flow [MH99, HNS02, HJR$^+$13, Dim14, FPM$^+$14, JT15]. However, most of those papers [MH99, HNS02, HJR$^+$13, Dim14] use games only in a narrow mathematical sense to provide a proof system (called the *game semantics*) for deciding security properties. We are aware of only a handful of papers that investigate the impact of participants' incentives and available strategies on the security of information flow. In [AG04, GCC08], economic interpretations of privacy-preserving behavior are proposed. [FPM$^+$14] uses game-theoretic solution concepts (in particular, Nash equilibrium) to prescribe the optimal defence strategy against attacks on information security. In contrast, our approach is analytic rather than prescriptive, as we do not propose how to manage information security. Moreover, in our view, privacy is not the goal but rather the means to achieve some higher-level objectives. We also in Chapter 6 proposed a weaker variant of noninterference by allowing the High players to select an appropriate strategy, while here we look at the potential damage inflicted by adverse strategies of the Low users.

Our idea of looking at the unique most precise non-interfering variant of the system is related on the technical level to [GM04]. There, attackers displaying different analytical capabilities are defined by abstract interpretation, which leads to a lattice of noninterference variants with various strength. Attackers with weakened observational powers were also studied in [ZM03].

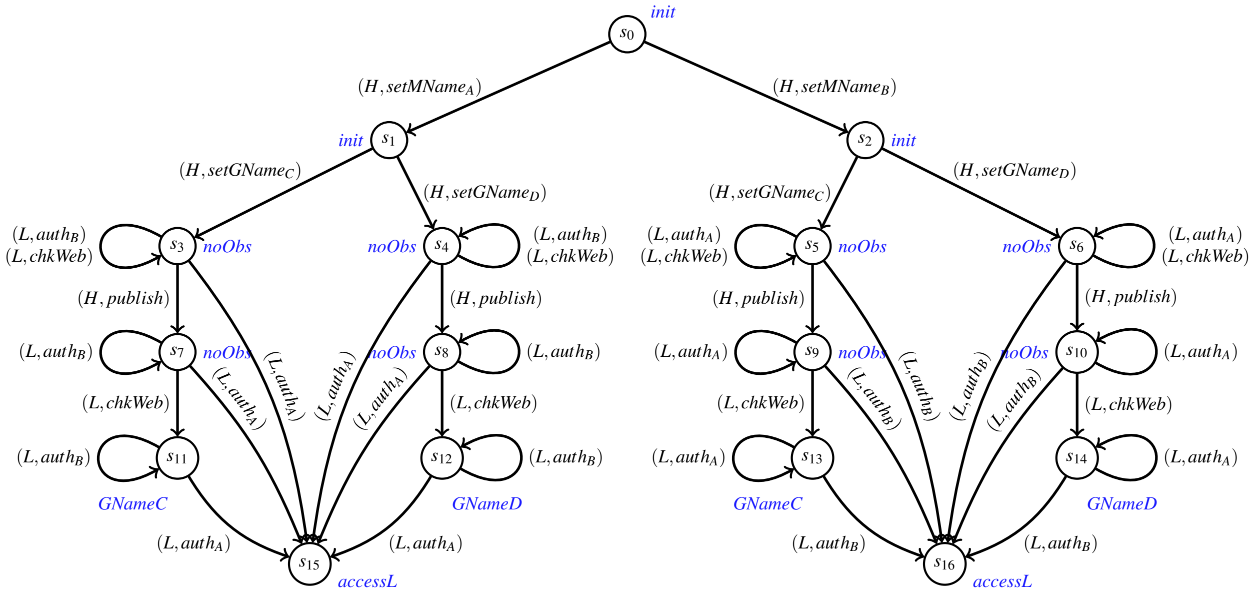

Figure 7.1 Transition network M_a in which the High player publishes her grandmother's maiden name on her blog. Only the observations of L are shown

7.3 Revisiting the Preliminaries

In this chapter we revisit the concepts of noninterference, transition networks, strategies and goals of the players, which are explained in Chapter 5. We remind these concepts using the scenario introduced in Section 7.1

Example 7.3.1. *Consider a simplified version of the phone banking scenario of Subsection 7.1.1. Figure 7.1 and Figure 7.2 presents simple transition networks for the scenario. The users are H who has an account in the bank, and L who may try to impersonate H. We consider two alternative variants: one where H publishes her grandmother's maiden name on the blog (Figure 7.1), and one where she publishes her mother's maiden name (Figure 7.2). To keep the graph simple, we assume that the possible names are A and B in the former case, and C and D in the latter. Labels on transitions show the personalised actions resulting in the transition, and the observations of users in each state are shown beside the state. The observations of L are shown beside each state.*

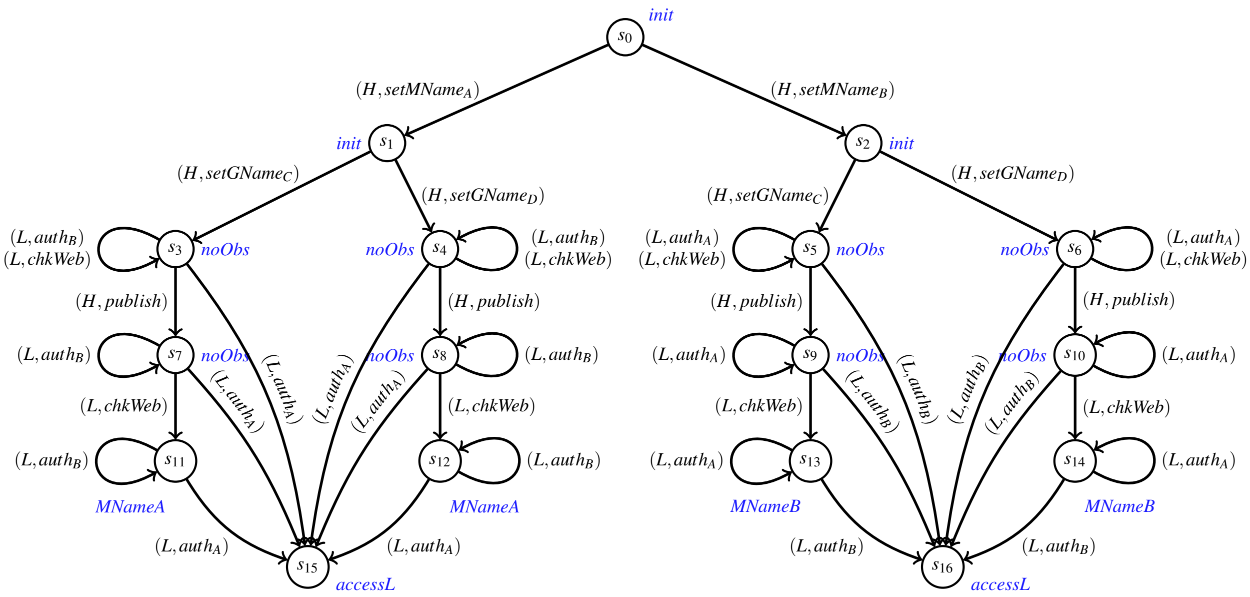

Figure 7.2 Transition network M_b in which H publishes her mother's maiden name

The observations for H are omitted, as they will be irrelevant for our analysis.

Each model begins by initialization of the relevant names, represented by virtual actions of agent H. Then, H publishes an essay on her blog. In the first variant, the essay mentions the maiden name of H's grandmother. In the second variant, it mentions her mother's maiden name. After H has published the essay, L can check the blog (action chkWeb). The resulting observation of L depends on what is published. Then, authentication proceeds: to log in, a user must give the correct value of H's mother's maiden name.

Note that, for mathematical completeness, we must define the outcome of every user-action pair in every state. We assume that there are two "error states" s_{HErr}, s_{LErr} *in models* M_a *and* M_b *(not shown in the graphs). Any action of H not depicted in the figure leads to* s_{HErr}*, and any action of L not depicted in the figure leads to* s_{LErr}*. We will later use the error states in the definition of the players' goals, in such a way that L will always want to avoid* s_{LErr} *and H will want to avoid* s_{HErr}*. This way we can (however imperfectly) simulate some synchronisation in the restricted framework* *of Goguen and Meseguer.*

It is easy to see that neither M_a nor M_b satisfies noninterference from H to L. For instance, in the model from Figure 7.1, the observation of L after sequence $\alpha = \langle (H, setMName_A), (H, setGName_D), (H, publish), (L, chkWeb) \rangle$ is GNameD, but the observation of L after $Purge_H(\alpha) = \langle (L, chkWeb) \rangle$ is noObs, which is clearly different.

Example 7.3.2. *Consider the models in Figure 7.1 and Figure 7.2, and suppose that L wants to access H's bank account. This can be expressed by the reachability goal $\Gamma_{\mathbb{T}}$ with $\mathbb{T} = \{s_{15}, s_{16}\}$ as the target states. In fact, L also wins if H executes an out-of-place action (cf. Example 7.3.1 for detailed explanation). In consequence, the winning states for L are $\mathbb{T} = \{s_{15}, s_{16}, s_{HErr}\}$. Note that L has no strategy that guarantees $\Gamma_{\mathbb{T}}$ in model M_a (although information is theoretically leaking to L as the model does not satisfy noninterference). Even performing the action chkWeb does not help, because L cannot distinguish between states s_{11} and s_{13}, and there is no single action that succeeds for both s_{11}, s_{13}. Thus, L does not know whether to use $auth_A$ or $auth_B$ to get access to H's bank account.*

On the other hand, L has a winning strategy for $\Gamma_{\mathbb{T}}$ in model M_b. The strategy is to execute chkWeb after H publishes her mother's maiden name, and afterwards do $auth_A$ in states s_{11}, s_{12} (after observing MNameA) or $auth_B$ if the system gets to s_{13}, s_{14} (i.e., after observing MNameB).

□

7.4 Security as Strategic Property

The property of noninterference looks for *any* leakage of *any* information. If one can possibly happen in the system, then the system is deemed insecure. In many cases, this view is too strong. There are lots of information pieces that can leak out without bothering any interested party. Revealing the password to your web banking account can clearly have much more disastrous effects than revealing the price that you paid for metro tickets on your latest trip to Paris. Moreover, the relevance of an information leak cannot, in general, be determined by the type of the information. Think, again, of revealing the maiden name of your mother vs. the maiden name of your grandmother. The former case is potentially dangerous since the maiden name of one's mother is often used to grant access to manage banking services by telephone. Revealing the latter is quite harmless to most ends and purposes.[2]

We suggest that the relevance of information leakage should be judged by the extent of damage that the leak allows the attackers to inflict on the goal of the system. Thus, as the first step, we define the security of the system in terms of damaging abilities of the L players.

In order to assess the relevance of information flow from H to L, we will look at the resulting strategic abilities of L players. For this, two design choices have to be made. First, what type of strategies are L players supposed to use? Secondly, what is the goal that they are assumed to pursue? The second question is of particular importance, because typically we do not know (and often do not care about) the real goals of potential attackers. What we are aware of, and what we want to protect, is the objective that the system is built for.

We follow the game-theoretic tradition of looking at the worst case and assuming the opponents to be powerful and adversary. Thus, we assume L players to use perfect recall strategies. Moreover, we assume that the goal of L players is to violate a given goal of the system. The goal can be a functionality or a security requirement, or a combination of both. Moreover, it can originate from a private goal of the H players, an objective ascribed to the system by its designer (e.g., the designer of a contract signing protocol), or a combination of requirements specified by the owner/primary stakeholder in the system (for instance, a bank in case of a web banking infrastructure).

Definition 7.4.1 (Effective security). *Let M be a transition network with some Low players $L \subseteq \mathfrak{U}$, and let Γ be the goal of the system. We say that M is* effectively secure *for (L,Γ) iff L does not have*

[2] Note, however, that revealing the maiden name of your maternal grandmother is potentially dangerous to your *mother* if she enables banking by telephone.

a strategy to enforce $\overline{\Gamma}$, where $\overline{X}$ denotes the complement of set X. That is, the system is effectively secure iff the attackers do not have a strategy that ensures an execution violating the goal of the system. We will use $ES(M,L,\Gamma)$ to refer to this property.

Besides judging the effective security of a system, we can also use the concept to compare the security level of two models.

Definition 7.4.2 (Comparative effective security)**.** *Let M,M' be two models, and Γ be a goal in M,M' (i.e., $\Gamma \subseteq paths(M) \cup paths(M')$). We say that:*

- *M has* strictly less effective security *than M' for (L,Γ), denoted $M \prec_{L,\Gamma} M'$, iff $ES(M',L,\Gamma)$ but not $ES(M,L,\Gamma)$. That is, L can enforce a behavior of the system that violates its goal in model M but not in M'. We denote the relationship by $M \prec_{L,\Gamma} M'$;*
- *M' is* at least as effectively secure as *M for (L,Γ), denoted $M \preceq_{L,\Gamma} M'$, iff $ES(M,L,\Gamma)$ implies $ES(M',L,\Gamma)$;*
- *M is* effectively equivalent to *M' for (L,Γ), denoted $M \simeq_{L,\Gamma} M'$, iff either both $ES(M,L,\Gamma)$ and $ES(M',L,\Gamma)$ hold, or both do not hold.*

Thus, if in one of the models L can construct a more harmful strategy, then the model displays lower effective security than the other model. Conversely, if both models allow only for the same extent of damage, then they have the same level of effective security. This way, we can order different alternative designs of the system according to the strategic power they give away to the attacker.

Example 7.4.1. *Consider models M_a,M_b from Figure 7.1 and Figure 7.2, and let the goal Γ be to prevent L from accessing H's bank account. Thus, Γ is the safety goal $\Gamma_{\mathbb{S}}$ with $\mathbb{S} = St \setminus \{s_{15}, s_{16}, s_{HErr}\}$, and therefore $\overline{\Gamma} = \Gamma_{\mathbb{T}}$ with $\mathbb{T} = \{s_{15}, s_{16}, s_{HErr}\}$. As we saw in Example 7.3.2, L has no strategy to guarantee $\overline{\Gamma}$ in M_a, but she has a surely winning strategy for $\overline{\Gamma}$ in M_b. Thus, M_b is strictly less effectively secure than M_a, i.e., $M_b \prec_{L,\Gamma} M_a$.*

We will further use the concept to compare security of alternative information flows based on the same (or similar) action-transition structures.

7.5 Effective Information Security

We will now propose a scheme that allows determining whether a given model of interaction leaks relevant information or not. We use the idea of refinement checking from process algebras, where a process is assumed to be correct, if and only if it refines the ideal process [RHB97]. A similar reasoning scheme is also used in the analysis of multi-party computation protocols (a protocol is correct iff it is equivalent to the ideal model of the computation [GMW87]).

To this end, we need a suitable notion of refinement or equivalence and a suitable definition of the ideal model. The former is straightforward: we will use the $\simeq_{L,\Gamma}$ relation. The latter is more involved. If the reference model ascribes too much observational capabilities to the L players then the concept will be ill-defined (it will classify insecure systems as secure). If the reference model assigns L players with too little information, then the concept will be useless (no realistic system will be ever classified as secure).

In what follows, we first explain and define the concept of an idealised variant of a model. Then in Section 7.5.2 we do our first take on the idealised variant by defining the *blind variant* of a model. In Section 7.5.3 we first define the *non-interfering idealised* variant of a model.

7.5.1 Ability-Based Security of Information Flows

Definition 7.4.2 allows for comparing the effective security of two alternative information flows. We will say two models differ only in their information flow if they are *transition equivalent*:

Definition 7.5.1 (Transition-equivalent models)**.** *The* action-transition frame *of a model M, which we denote by F_M, is the network M minus the observation functions $obs(\cdot)$. We will denote the set* *of models based on frame F by $\mathscr{M}(F)$. Two models are* transition-equivalent *iff they are based on the same frame.*

Then , $(F,obs) \prec (F,obs')$ says that the observation function *obs* leaks more relevant information than *obs'* in the transition-action frame F. However, we usually do not want to compare several alternative information flows. Rather, we want to determine if a single given model M reveals relevant information or not. How can we achieve that? A natural idea is to compare the effective security of M to an *ideal model*, i.e., a model that is transition equivalent to the original model and moreover leaks no relevant information by construction. Then, a model is effectively information-secure if it has the same level of effective security as its idealised variant:

Definition 7.5.2 (Effective information security)**.** *Let M be a transition network with some Low players $L \subseteq \mathbb{U}$, and let Γ be the goal of the system. Moreover, let $Ideal(M)$ be the idealised variant of M. We say that M is* effectively information-secure *for (L,Γ) iff $M \simeq_{L,\Gamma} Ideal(M)$.*

How do we construct the idealised variant of M? The idea is to "blur" observations of Low so that we obtain a variant of the system where the observational capabilities of the attackers are minimal. What observational capabilities are "minimal"? We start with the following, rather naive definition of idealisation.

7.5.2 Blinding the Low Players: First Attempt

By using the idealised model, we intend to distinguish to what extent the damaging abilities of L players are due to the "hard" actions available in the system, and to what extent they are due to the available information flow. In other words, we want to see how far one can minimize the strategic ability of the L players by reducing their observational abilities in the model.

The first take to define an idealised model is to assume that L *never sees anything*. To this end, we simply assume that $obs(s,L)$ is the same in all states $s \in St$.

Definition 7.5.3 (Idealised model, first take)**.** *Having a transition network M based on frame F, and a set of players L, we define the* blind variant of M *as* $M' = (F, obs')$ *such that* $obs'(q,l) = obs(q',l)$ *for every* $q,q' \in St$ *and* $l \in L$.

In many scenarios this is too much. In particular, a L agent may have access to perfectly legitimate observations that are inherent to maintaining their private affairs, such as checking the balance of their bank account, listing the files stored on in their private file space, etc.

7.5.3 Idealised Models Based on Noninterference

Below we propose a weaker form of "blinding" that will be used to single out the damaging abilities that are due to *L observing H's actions*, rather than due to *any* observations that L players can happen to make. We begin by recalling the notion of *term unification* which is a fundamental concept in automated theorem proving and logic programming [Rob65]. Given two terms t_1, t_2, their unification ($t_1 \equiv t_2$) can be understood as a declaration that, from now on, both terms refer to exactly the same underlying object. In our case, the terms are observation labels from the set *Obs*. Unification can be seen as an equivalence relation on observation labels, or equivalently as a partitioning of the labels into equivalence classes. The application of the unification to a model yields a similar model where the equivalent observations are "blurred".

Definition 7.5.4 (Unification of observations)**.** *Given a set of observation labels Obs, a* unification *on Obs is any equivalence relation* $\mathcal{U} \subseteq Obs \times Obs$.

Given a model $M = \langle St, s_0, \mathfrak{U}, \mathfrak{A}, Obs, obs, do \rangle$ *and a unification* $\mathcal{U} \subseteq Obs \times Obs$, *the* application of $\mathcal{U}$ to M *is the model* $\mathcal{U}(M) = \langle St, s_0, \mathfrak{U}, \mathfrak{A}, Obs', obs', do \rangle$, *where:* $Obs' = \{[o]_{\mathcal{U}} \mid o \in Obs\}$ *replaces Obs by the set of equivalence classes defined by* $\mathcal{U}$, *and* $obs'(q,u) = [obs(q,u)]_{\mathcal{U}}$ *replaces the original observation in q with its equivalence class for any* $u \in \mathfrak{U}$.

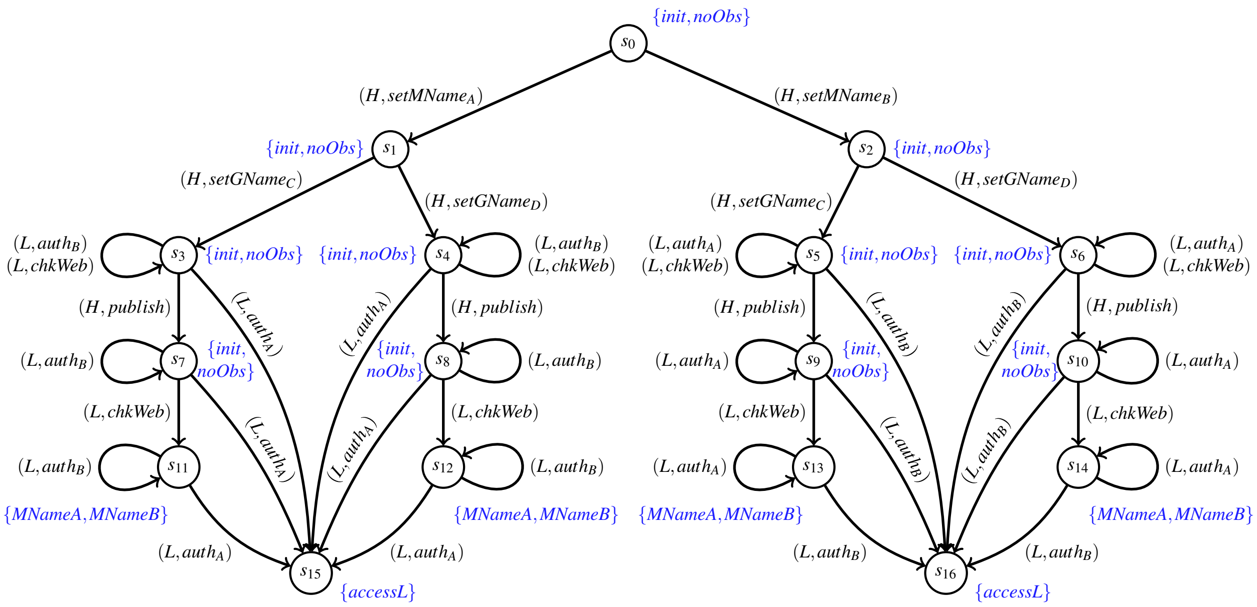

Figure 7.3 An example of unification of observations.

Example 7.5.1. *Figure 7.3 depicts the model obtained from M_b by unifying observations MNameA and MNameB into $\{MNameA, MNameB\}$, observations init and noObs into $\{init, noObs\}$, and observation accessL into $\{accessL\}$.*

Our reference model for M will be the variant of M where noninterference is obtained by the minimal necessary "blurring" of L's observations.

Definition 7.5.5 (Noninterferent idealised model)**.** *Having a transition network M and a set of "Low" players L, we define the* noninterfering idealised variant of M *as $\mathscr{U}(M)$ such that:*

(i) $NI_{\mathscr{U}(M)}(H, L)$, and

(ii) for every $\mathscr{U}' \subsetneq \mathscr{U}$ it is not the case that $NI_{\mathscr{U}'(M)}(H, L)$.

We need to show that the concept of noninterferent idealised model is well defined. The proof is constructive, i.e., given a model M, we first show how one can build its idealised variant, and then show that it is unique.

Theorem 7.5.1. *For every transition network M, there is always a unique unification $\mathscr{U}$ satisfying properties (i) and (ii) from Definition 7.5.5.*

The proof of Theorem 7.5.1 needs some preliminary steps. First use the concept of unwinding relations [GM84, Rus92, vdMZ10], which was discussed in Chapter 5, to define the relation R_M^* on the states of a transition network M. Then we use this relation for constructing and proving the uniqueness of the idealised variant of M.

For defining relation R_M^*, first we relate any two states of M if one of them can be reached from the other one by a sequence of H personalised actions. Then in each step, we relate the pair of states that are reached by a similar Low personalized action from any two states that are already related. Also, we enforce transitivity on the set. We continue adding related states until the relation becomes stable.

The mathematical definition of R_M^* is as follows:

Definition 7.5.6 (Relation R_M^* for a transition network M)**.** *Given a model $M = \langle St, s_0, \mathfrak{U}, \mathfrak{A}, Obs, obs, do\rangle$ and sets of High players H and Low players L, we define the relation $R_M^* \subseteq St \times St$ as the least fixpoint of the following function F, transforming relations on St:*

$$
\begin{aligned}
F(R) \;\; = \;\; & R_0 \cup \\
& \{(t_1,t_2) \mid \exists (s_1,s_2) \in R, l \in L, a \in \mathfrak{A}.\ do(s_1,l,a) = t_1, do(s_2,l,a) = t_2\} \cup \\
& \{(t_1,t_2) \mid \exists s \in St.\ (t_1,s) \in R \& (s,t_2) \in R\},
\end{aligned}
$$

where $(s_1,s_2) \in R_0$ iff for some sequence of personalized actions of High players α, either $s_1, \xrightarrow{\alpha} s_2$, or $s_2 \xrightarrow{\alpha} s_1$.

It is straightforward to see that R_M^* is an equivalence relation (for reflexivity, notice that for any $s \in St$, $s \xrightarrow{\alpha} s$ for $\alpha = \langle\rangle$, and therefore $(s,s) \in R_0$). We will now show that *if M satisfies* *noninterference then R_M^* is the smallest unwinding relation.* Conversely, if M does not satisfy noninterference then R_M^* indicates pairs of states that *must bear the same observations for Low if we want to make the model M non-interferent.* We will later show that it is sufficient to unify Low's observations in states connected by R_M^* in order to obtain a non-interferent variant of M. In consequence, R_M^* generates the minimal unification that achieves the task.

The following proposition shows that if M satisfies the noninterference property, then R_M^* is a subset of any unwinding relations on the states of M.

Proposition 7.5.2. *Given a model M and sets of players H and L, if $\sim_{NI_L}$ is an unwinding relation on the states of M and relation R_M^* is defined as in Definition 7.5.6, then $R_M^* \subseteq \sim_{NI_L}$.*

Proof. As the relation R_M^* is constructed by adding related states in several steps, we do the proof by induction on these steps. We show that firstly $R_0 \subseteq \sim_{NI_L}$, and secondly all related pair of states added in each step also is in $\sim_{NI_L}$.

Induction base: If $(s_1, s_2) \in R_0$, then for some sequence of personalized actions of High players α, either $s_1, \xrightarrow{\alpha} s_2$, or $s_2 \xrightarrow{\alpha} s_1$, hence by property (LR) , and transitivity of the unwinding relation it holds that $(s_1, s_2) \in \sim_{NI_L}$. Therefore $R_0 \subseteq \sim_{NI_L}$.

Induction step: We show that if $R_i \subseteq \sim_{NI_L}$, then $F(R_i) \subseteq \sim_{NI_L}$ holds. $F(R_i)$ is constructed by union of three sets. We show that all these three sets are subsets of $\sim_{NI_L}$:

i- $R_i \in \sim_{NI_L}$ by the induction step assumption.

ii- If $(s_1, s_2) \in R_i$ then by induction step assumption $(s_1, s_2) \in \sim_{NI_L}$. So for any $t_1, t_2 \in St$, $a \in \mathfrak{A}$ and $l \in L$ such that $do'(s_1, l, a) = t_1$ and $do'(s_2, l, a) = t_2$, by property (SC) of unwinding relation it holds that $(t_1,t_2) \in \sim_{NI_L}$. Therefore:

$$\begin{aligned}&\{(t_1,t_2) \mid \exists (s_1,s_2) \in R, l \in L, a \in \mathfrak{A} \cdot \\ &\qquad do'(s_1,l,a) = t_1, do'(s_2,l,a) = t_2\} \\ &\subseteq \sim_{NI_L}.\end{aligned}$$

iii- If $(t_1,s) \in R_i$ and $(s,t_2) \in R_i$, then by induction step assumption it holds that $(t_1,s) \in \sim_{NI_L}$ and $(t_2,s) \in \sim_{NI_L}$. Therefore by transitivity of $\sim_{NI_L}$ it entails that $(t_1,t_2) \in \sim_{NI_L}$, hence:

$$\begin{aligned}&\{(t_1,t_2) \mid \exists s \in St \cdot (t_1,s) \in R \text{ and } (s,t_2) \in R\} \\ &\subseteq \sim_{NI_L}.\end{aligned}$$

By i, ii, and iii we infer that $F(R_i) \subseteq \sim_{NI_L}$, and therefore by induction base and induction step we have that $R^*_M \subseteq \sim_{NI_L}$. □

Now we show that if the model M, L players have the same observations at any two states related by R^*_M, then M satisfies noninterference.

Lemma 7.5.3. *In a model M with sets of players H and L, if for any $l \in L$, $s_1,s_2 \in St$ it is the case that $(s_1,s_2) \in R^*_M$ implies $obs(s_1,l) = obs(s_2,l)$, then R^* is an unwinding relation on the states of M and therefore it holds that $NI_M(H,L)$.*

Proof. We prove this by showing that R^*_M satisfies the conditions of Definition 5.1.6: The relation R^*_M is an equivalence relation, condition (OC) follows from the assumption of this lemma, and conditions (SC) and (LR) follow from the definition of the relation R^*_M. Therefore it holds that $NI_M(H,L)$. □

And as the last step before introducing the unification of function $\mathscr{U}^*_M$, we show that if M

satisfies noninterference, then R^*_M is an unwinding relation on its states (and by Proposition 7.5.2 it is in fact the smallest unwinding relation).

Proposition 7.5.4. *In a model M with sets of players H and L, if $NI_M(H,L)$ then R^*_M is an unwinding relation on states of M.*

Proof. If $NI_M(H,L)$ then by Proposition 5.1.1 there is an unwinding relation $\sim_{NI_L}$ on the states of M. By Proposition 7.5.2 $R^*_M \subseteq \sim_{NI_L}$ and therefore for any $l \in L$, $s_1, s_2 \in St$ such that $(s_1, s_2) \in R^*_M$ it is the case that $(s_1, s_2) \in \sim_{NI_L}$ and therefore $obs(s_1, l) = obs(s_2, l)$. Hence by Lemma 7.5.3 R^*_M is an unwinding relation on the states of M. □

Now, by using relation R^*_M, we define the unification of observations $\mathcal{U}^*_M$ that will provide the noninterferent idealised variant of M.

Definition 7.5.7 (Unification for noninterference $\mathcal{U}^*_M$)**.** *We define the unification of observations $\mathcal{U}^*_M \subseteq Obs \times Obs$ as follows. For any $o_1, o_2 \in Obs$, we have $(o_1, o_2) \in \mathcal{U}^*_M$ iff there exist $s_1, s_2, t_1, t_2 \in St$ and $l \in L$ such that:*

(a) $obs(s_1, l) = o_1$,

(b) $obs(s_2, l) = o_2$,

(c) $(s_1, t_1) \in R^*_M$,

(d) $(s_2, t_2) \in R^*_M$, and

(e) $obs(t_1, l) = obs(t_2, l)$.

It then holds that $\mathcal{U}^*_M(M)$ satisfies the noninterference property (Proposition 7.5.6) and no refinement of $\mathcal{U}^*_M$ achieves that (Proposition 7.5.7). The following lemma states that if two states are related by R^*_M, then their observations are unified by $\mathcal{U}^*_M$.

Lemma 7.5.5. *In a model M, for any $s_1, s_2 \in St$ and $l \in L$, if $(s_1,s_2) \in R^*_M$ then $(obs(s_1,l), obs(s_2,l)) \in \mathscr{U}^*_M$.*

Proof. Assume $(s_1,s_2) \in R^*_M$ and $l \in L$. For proving that $(obs(s_1,l), obs(s_2,l)) \in \mathscr{U}^*_M$ we verify the conditions in Definition 7.5.7. By taking $obs(s_1,l) = obs_1$ and $obs(s_2,l) = obs_2$, conditions (a) and (b) are satisfied trivially. If we take $t_1 := s_2$ and $t_2 := s_2$, then $(s_1,t_1) \in R^*_M$ by the proposition assumption and $(s_2,t_2) \in R^*_M$ by reflexivity of R^*_M. These prove conditions (c) and (d). Condition (e) is also satisfied because $t_1 = t_2$. Therefore $(obs(s_1,l), obs(s_2,l)) \in \mathscr{U}^*_M$. □

As the next step, we show that $\mathscr{U}^*_M(M)$ satisfies the noninterference property.

Proposition 7.5.6. *Given a model M, and $\mathscr{U}^*_M(M) = \langle St, s_0, \mathfrak{U}, \mathfrak{A}, Obs^*, obs^*, do\rangle$ defined as in Definition 7.5.7 on M, it holds that $NI_{\mathscr{U}^*_M(M)}(H,L)$.*

Proof. For the proof, we are going to use Lemma 7.5.3 and show that for any two states s_1, s_2 and $l \in L$, if $(s_1,s_2) \in R^*_{\mathscr{U}^*_M(M)}$ then $obs^*(s_1,l) = obs^*(s_2,l)$. First notice that $R^*_M = R^*_{\mathscr{U}^*_M(M)}$, because M and $\mathscr{U}^*_M(M)$ differ only in their observation functions and the definition of R^* relation does not depend on the observation function of the model. So for any $(s_1,s_2) \in R^*_{\mathscr{U}^*_M(M)}$ and $l \in L$ we have that $(s_1,s_2) \in R^*_M$, and by Lemma 7.5.5 it follows that $(obs(s_1,l), obs(s_2,l)) \in \mathscr{U}^*_M$, and therefore $obs^*(s_1,l) = obs^*(s_2,l)$. Hence by Lemma 7.5.3 it holds that $R^*_{\mathscr{U}^*_M(M)}$ is an unwinding relation for $\mathscr{U}^*_M(M)$'and therefore $NI_{\mathscr{U}^*_M(M)}(H,L)$. □

As the last step before proving Theorem 7.5.1, we show that $\mathscr{U}^*_M$ is the minimal unification that makes the model M noninterfering.

Proposition 7.5.7. *Given a model M, and sets of players H and L, for any unification of observations U where $U(M) = \langle St, s_0, \mathfrak{U}, \mathfrak{A}, Obs', obs', do\rangle$, if $NI_{U(M)}(H,L)$ then $\mathscr{U}^*_M \subseteq U$.*

Proof. Assume U is a unification of observations for model M such that $NI_{U(M)}(H,L)$ and assume $(obs_1, obs_2) \in \mathscr{U}^*_M$. We show that $(obs_1, obs_2) \in U$ and hence $\mathscr{U}^*_M \subseteq U$. By the definition of

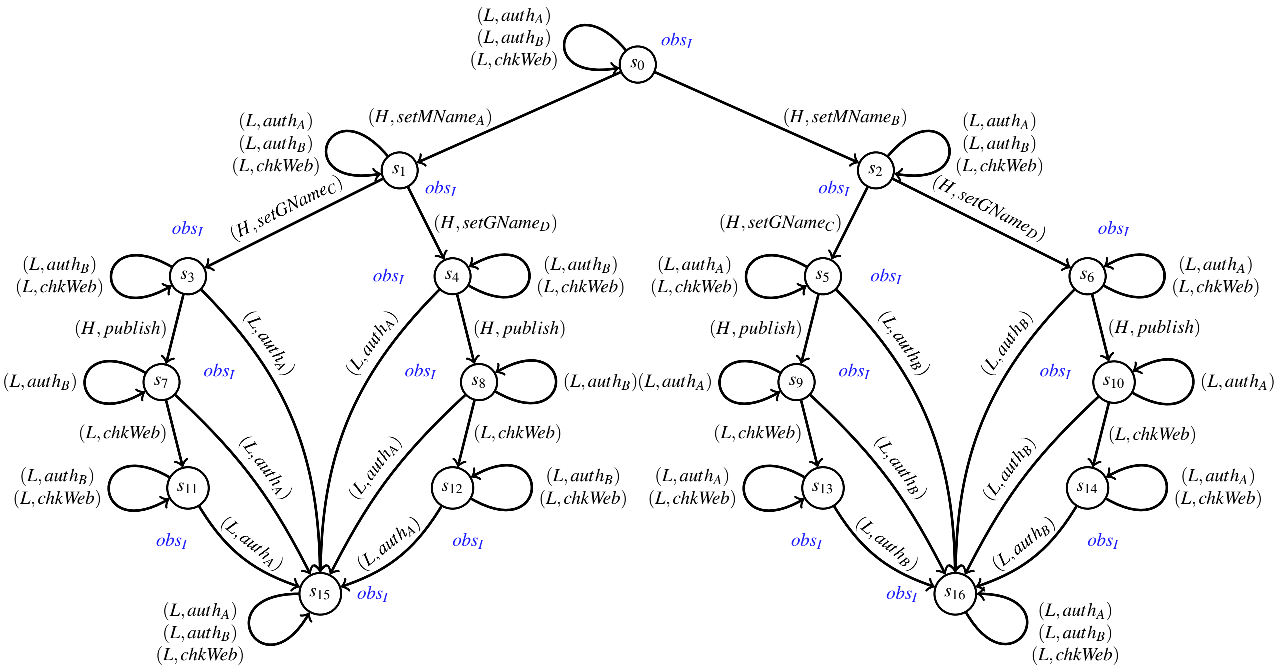

Figure 7.4 The noninterfering idealised variant of the banking model M_b, where obs_I is the unification of *init*, *noObs*, *MNameA* and *MNameB*

.

$\mathscr{U}^*_M$, there exists $s_1, s_2, t_1, t_2 \in St$, $l \in L$ such that $obs(s_1, l) = obs_1$, $obs(s_2, l) = obs_2$, $(s_1, t_1) \in R^*_M$, $(s_2, t_2) \in R^*_M$ and $obs(t_1, l) = obs(t_2, l)$. By $NI_{U(M)}(H, L)$ and Proposition 7.5.4 we have that $R^*_{U(M)}$ is an unwinding relation for $U(M)$. So as $R^*_M = R^*_{U(M)}$, R^*_M is also an unwinding relation for $U(M)$. Therefore by property (OC) of unwinding relation, from $(s_1, t_1) \in R^*_M$ and $(s_2, t_2) \in R^*_M$ we entail that $obs'(s_1, l) = obs'(t_1, l)$ and $obs'(s_2, l) = obs'(t_2, l)$. Using the definition of $obs'(.)$ we have that $(obs(s_1, l), obs(t_1, l)) \in U$ and $(obs(s_2, l), obs(t_2, l)) \in U$. So, as $obs(t_1, l) = obs(t_2, l)$ and by transitivity property of U, we infer that $(obs(s_1), obs(s_2)) \in U$, and it follows that $(obs_1, obs_2) \in U$. Therefore $\mathscr{U}^*_M \subseteq U$. □

We can now complete the proof of Theorem 7.5.1.

Proof of Theorem 7.5.1. We want to prove that, given a model M, set of players H and L, and any unification of observations $\mathscr{U}$, if $\mathscr{U}(M)$ is a noninterfering idealised variant of M, then $\mathscr{U} = \mathscr{U}^*_M$. Assume that $\mathscr{U}(M)$ is a noninterfering idealised variant of M. By property (i) of Definition 7.5.5 and Proposition 7.5.7 we infer that $\mathscr{U}^*_M \subseteq \mathscr{U}$. Also, by Proposition 7.5.6, we have that $NI_{\mathscr{U}^*_M(M)}(H,L)$. Therefore by property (ii) of Definition 7.5.5 it holds that $\mathscr{U} = \mathscr{U}^*_M$. □

From now on, we assume that $Ideal(M)$ refers to the noninterfering idealised variant of M.

Example 7.5.2. *Consider models M_a, M_b in Figure 7.1 and Figure 7.1. We recall that both models are not noninterferent. Figure 7.4 shows the idealised variant $Ideal(M_b)$ of M_b, obtained by unification $U^*_{M_b}$ (see Definition 7.5.7). In the noninterferent idealised variant of M_b, observations init, noObs, MnameA, and MNameB of L are unified and replaced by the equivalence class $\{init, noObs, MNameA, MNameB\}$, represented by obs_I in the figure. The idealised variant of M_a is constructed analogously by unification of init, noObs, MnameC, and MNameD. Clearly, L has no surely winning strategy to guarantee $\overline{\Gamma} = \Gamma_{\mathbb{T}}$ for $\mathbb{T} = \{s_{15}, s_{16}, s_{HErr}\}$ in both $Ideal(M_a)$ and $Ideal(M_b)$.*

Recall from Example 7.4.1 that L has no winning strategy for $\overline{\Gamma}$ in M_a, but she has one in M_b. So, $M_a \simeq_{L,\Gamma} Ideal(M_a)$, but $M_b \not\simeq_{L,\Gamma} Ideal(M_b)$. Thus, M_a is effectively information-secure for (L,Γ), but M_b is not. □

It is important to notice that noninterferent variants are indeed idealisations:

Proposition 7.5.8. *For every M, L, and Γ, we have that $M \preceq_{L,\Gamma} Ideal(M)$.*

Proof. Note that M and $Ideal(M)$ differ only in their observation functions. Also we have that for any pair of states $s_1, s_2 \in St$, if $[s_1]^M_L = [s_2]^M_L$ then $[s_1]^{Ideal(M)}_L = [s_2]^{Ideal(M)}_L$. Therefore all the strategies of L in $Ideal(M)$ are also L's strategies in M. Thus for any for any goal $\Gamma \subseteq paths(M)$, if L have a surely winning strategy to enforce $\overline{\Gamma}$ in $Ideal(M)$ then they also have a surely winning strategy for $\overline{\Gamma}$ in M, qed. □

Finally, note that the concept of noninterference in our construction of effective security can be in principle replaced by an arbitrary constraint of information leakage. The same reasoning scheme could be applied to noninference, nondeducibility, strategic noninterference, and so on. The pattern does not change: given a "classical" property $\mathscr{P}$ of information security, we define the idealised variant of M through the minimal unification U such that that $U(M)$ satisfies $\mathscr{P}$. Then, M is effectively secure in the context of property $\mathscr{P}$ iff it is strategically equivalent to $U(M)$.

We leave the investigation of which information security properties have unique minimal unifications for future work.

7.6 Extending the Results to a Broader Class of Models

As mentioned before, the models of Goguen and Meseguer are "total on input," i.e., each action label is available to every user at every state. This makes modeling actual systems very cumbersome. We have seen that in the previous examples where spurious states had to be added to the analysis to allow for some synchronization between actions of different agents. In this section, we consider a broader class of models, and show how our results carry over to the more expressive setting. That is, we consider *partial transition networks (PTS)* $M = \langle St, s_0, \mathfrak{U}, \mathfrak{A}, Obs, obs, do \rangle$ which are similar to the transition networks defined in Chapter 5, except that the transition function $do : St \times \mathfrak{U} \times \mathfrak{A} \rightharpoonup St$ can be a partial function. By $do(s,u,a) = undef$ we denote that action a is unavailable to user u in state s; additionally, we define $act(s,u) = \{a \in \mathfrak{A} \mid do(s,u,a) \neq undef\}$ as the set of actions available to u in s. Moreover, we assume that players are aware of their available actions, and hence can distinguish states with different repertoires of choices – formally, for any $u \in \mathfrak{U}, s_1, s_2 \in St$, if $obs(s_1,u) = obs(s_2,u)$ then $act(s_1,u) = act(s_2,u)$.

We begin by a suitable update of the definition of noninterference:

Definition 7.6.1 (Noninterference for partial transition networks)**.** *Given a PTS M and sets of* *agents H,L, such that $H \cup L = \mathfrak{U}, H \cap L = \emptyset$, we say that H* is non-interfering with *L iff for all $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$ and all $u_l \in L$, if $exec(\alpha) \neq undef$ then $[exec(\alpha)]_{u_l} = [exec(Purge_H(\alpha))]_{u_l}$. We denote the property also by $NI_M(H,L)$, thus slightly overloading the notation.*

Note that Definition 5.1.4 is a special case of Definition 7.6.1. We now define the noninterferent idealised variant based on the *total extension* of a PTS.

Definition 7.6.2 (U-total extension)**.** *Given a PTS $M = \langle St, s_0, \mathfrak{U}, \mathfrak{A}, Obs, obs, do \rangle$ and a subset of users $U \subseteq \mathfrak{U}$, we define the U-total variant of M as $total_U(M) = \langle St, s_0, \mathfrak{U}, \mathfrak{A}, Obs, obs, do' \rangle$ where the transition function $do'(.)$ is defined as follows: for every $s \in St$, $v \in \mathfrak{U}$ and $a \in \mathfrak{A}$, $do'(s,v,a) = s$ if for some $u \in U$ we have $v = u$ and $do(s,u,a) = undef$, otherwise $do'(s,v,a) = do(s,v,a)$.*

Definition 7.6.3 (Noninterferent idealised model for PTS)**.** *Given a partial transition network M and a set of "low" players L, we define the* noninterferent idealised variant of *M as $\mathcal{U}(total_L(M))$ such that:*

(i) $NI_{\mathcal{U}(total_L(M))}(H,L)$, and

(ii) for every $\mathcal{U}' \subsetneq \mathcal{U}$ it is not the case that $NI_{\mathcal{U}'(total_L(M))}(H,L)$.

The uniqueness theorem is then stated similar to Theorem 7.5.1:

Theorem 7.6.1. *For every partial transition network M, there is always a unique unification $\mathcal{U}$ satisfying properties (i) and (ii) from Definition 7.6.3.*

The proof is similar to the proof of Theorem 7.5.1, with the difference that we use $R^*_{total_L(M)}$ instead of R^*_M for constructing the idealised variant. However, as we use the concept of unwinding relation as the basis for using the R^* relation for constructing the idealised variant, we first need to modify the definition of the unwinding relation in Definition 5.1.6 and its corresponding proposition, Proposition 5.1.1 to adapt them to the new model:

Definition 7.6.4 (Unwinding for Noninterference in PTN). *Let M be a transition network, H a set of High agents, and L a set of Low agents. Then,* $\sim_{NI_L} \subseteq St \times St$ *is an* unwinding relation *iff it is an equivalence relation satisfying the conditions of* output consistency (OC)*,* step consistency (SC)*, and* local respect (LR)*. That is, for all states* $s,t \in St$*:*

(OC) *If* $s \sim_{NI_L} t$ *then* $[s]_L = [t]_L$*;*

(SC) *If* $s \sim_{NI_L} t$*,* $u \in L$*, and* $a \in \mathfrak{A}$ *then* $a \in act(s,u)$ *implies* $do(s,u,a) \sim_{NI_L} do(t,u,a)$*;*

(LR) *If* $u \in H$ *and* $a \in \mathfrak{A}$ *then* $a \in act(s,u)$ *implies* $s \sim_{NI_L} do(s,u,a)$*.*

Proposition 7.6.2. $NI_M(H,L)$ *iff there exist an unwinding relation* $\sim_{NI_L}$ *on the states of M that satisfies (OC), (SC) and (LR).*

Proof. **"⇐"** Suppose that there exists an unwinding relation $\sim_{NI_L}$ satisfying (OC), (SC) and (LR). We show for all $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$ and $u_l \in L$, if $exec(\alpha) \neq undef$ then $[exec(\alpha)]_{u_l} = [exec(Purge_H(\alpha))]_{u_l}$. We prove by induction on the size of α.

Induction base: $\alpha = \langle\rangle$. In this case $Purge_H(\alpha) = \langle\rangle$, and therefore $exec(\alpha) = exec(Purge_H(\alpha)) = s_0$. By the reflexivity of $\sim_{NI_L}$ we have that $exec(\alpha) \sim_{NI_L} exec(Purge_H(\alpha))$.

Induction step: Suppose for some $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$, $exec(\alpha) \neq undef$ implies $exec(\alpha) \sim_{NI_L} exec(Purge_H(\alpha))$. We show that for all $a \in \mathfrak{A}$ and $u \in \mathfrak{U}$ it holds that $exec(\alpha \circ (u,a)) \neq undef$ implies $exec(\alpha \circ (u,a)) \sim_{NI_L} exec(Purge_H(\alpha \circ (u,a)))$ (where $\circ$ denotes the concatenation operator). We consider three cases:

i) If $exec(\alpha \circ (u,a)) = undef$ then it holds that $exec(\alpha \circ (u,a)) \neq undef$ implies $exec(\alpha \circ (u,a)) \sim_{NI_L} exec(Purge_H(\alpha \circ (u,a)))$.

ii) If $exec(\alpha \circ (u,a)) \neq undef$ and $u \in L$ then firstly notice that $exec(Purge(\alpha \circ (u,a)) \neq undef$. Because by induction step assumption and (OC) it holds that $obs(exec(\alpha),u) = obs(exec(Purge_H(\alpha)),u)$ and so because $a \in act(exec(\alpha),u)$, by our model restrictions it holds that $a \in act(exec(Purge_H(\alpha),u))$.

Therefore by $exec(\alpha) \sim_{NI_L} exec(Purge_H(\alpha))$ (induction step assumption), $u \in L$, and (SC) we have $exec(\alpha \circ (u,a)) \sim_{NI_L} exec(Purge_H(\alpha \circ (u,a)))$.
iii) If $exec(\alpha \circ (u,a)) \neq undef$ and $u \in H$ then by (LR) property of $\sim_{NI_L}$, $exec(\alpha) \sim_{NI_L} exec(\alpha \circ (u,a))$. By this, induction step assumption and $Purge_H(\alpha) = Purge_H(\alpha \circ (u,a))$ we infer that $exec(\alpha \circ (u,a)) \sim_{NI_L} exec(Purge_H(\alpha \circ (u,a)))$.

"$\Rightarrow$" Suppose that $NI_M(H,L)$, we show there exists an unwinding relation on the states of M. Consider the relation $\sim$ defined as follows: for any $s,t \in St$, $s \sim t$ if for all $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$ and $u_L \in L$, it holds that if $exec(s,\alpha) \neq undef$ and $exec(t,\alpha) \neq undef$, then $[exec(s,\alpha)]_{u_L} = [exec(t,\alpha))]_{u_L}$. It can easily be seen that $\sim$ is an equivalence relation, we prove that it satisfies (OC), (SC) and (LR) properties.

(OC): If $s \sim t$ and we take $\alpha = \langle\rangle$, by $[exec(s,\alpha)]_{u_L} = [exec(s,\alpha)]_{u_L}$ it holds that $[s]_{u_L} = [t]_{u_L}$and therefore $\sim$ satisfies (OC).

(SC): Suppose that for some $s,t \in St$, $u \in L$ and $a \in \mathfrak{A}$ such that $s \sim t$, it holds that $do(s,u,a) \neq undef$ and $do(s,u,a) \not\sim do(t,u,a)$. Then there exists $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$, $u_L \in L$ such that $exec(do(s,u,a),\alpha) \neq undef$, $exec(do(t,u,a),\alpha) \neq undef$, and $[exec(do(s,u,a),\alpha)]_{u_L} \neq [exec(do(t,u,a),\alpha))]_{u_L}$. Therefore $[exec(s,((u,a) \circ \alpha)]_{u_L} \neq [exec(t,((u,a) \circ \alpha)]_{u_L}$, which contradicts $s \sim t$.

(LR): Suppose that for some $s \in St$, $u \in H$ and $a \in \mathfrak{A}$, it holds that $do(s,u,a) \neq undef$ and $s \not\sim do(s,u,a)$. Then there exists $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$, $u_L \in L$ such that $exec(do(s,u,a),\alpha) \neq undef$, $exec(s,\alpha) \neq undef$, and $[exec(s,\alpha)]_{u_L} \neq [exec(do(s,u,a),\alpha))]_{u_L}$. Because s is reachable, we have that $s = exec(\beta)$ for some $\beta \in (\mathfrak{U} \times \mathfrak{A})^*$. Therefore $[exec(\beta \circ \alpha)]_{u_L} \neq [exec(\beta \circ (u,a) \circ \alpha))]_{u_L}$. But this is a contradiction because by $NI_M(H,L)$ it holds that
$[exec(\beta \circ (u,a) \circ \alpha))]_{u_L} = [exec(Purge_H(\beta \circ (u,a) \circ \alpha)))]_{u_L}$ and
$[exec(\beta \circ \alpha))]_{u_L} = [exec(Purge_H(\beta \circ \alpha)))]_{u_L}$ and we have that
$Purge_H(\beta \circ (u,a) \circ \alpha) = Purge_H(\beta \circ \alpha)$. □

The rest of the proof of Theorem 7.6.1 follows analogously.

Example 7.6.1. *With PTS, the scenario from Example 7.3.1 can be modeled directly, without spurious states that ruled out illegal transitions. Thus, our models M_a, M_b for the two variants of the scenario are now exactly depicted in Figures 7.1 and 7.2.*

The noninterferent idealised variants of M_a (resp. M_b) is again obtained by the unification of observations init, noObs, MnameC, and MNameD (resp. init, noObs, MnameA, and MNameB). Clearly, L has no surely winning strategy to guarantee $\overline{\Gamma} = \Gamma_{\mathbb{T}}$ for $\mathbb{T} = \{s_{15}, s_{16}\}$ in M_a, Ideal(M_a), and Ideal(M_b). Moreover, he has a surely winning strategy in M_b. In consequence, M_a is effectively information-secure for (L,Γ), but M_b is not. □

The noninterferent variant was indeed an idealisation in simple transition networks of Goguen and Mesguer. Is it still the case in partial transition networks? That is, is it always the case that L has no more abilities in $Ideal(M)$ than in M? In general, no. On one hand, L's observational capabilities are more limited in $Ideal(M)$, and in consequence some strategies in M are no longer uniform in $Ideal(M)$. On the other hand, unification U^* possibly adds new transitions to M, that can be used by L in $Ideal(M)$ to construct new strategies. However, under some assumptions, $Ideal(M)$ does provide idealisation. First, notice that the potential additional strategic ability that L players may gain in the $Ideal(M)$ can only come from the added reflexive transitions. The only way that L can harm the goal of the system using these reflexive transitions is by keeping the system remaining in those states by continually selecting the reflexive actions. Consequently, the following propositions are straightforward:

Proposition 7.6.3. *Let M be a PTN such that for every state s in M there is $a \in \mathfrak{A}$ and $u \in L$ such that $do(s,u,a) = s$. Then, for any Γ, we have that $M \preceq_{L,\Gamma} Ideal(M)$.*

Proposition 7.6.4. *For any PTN M and safety goal Γ, we have $M \preceq_{L,\Gamma} Ideal(M)$.*

7.7 Summary

In this chapter, we introduced the concept of *effective information security*. The idea is aimed at assessing the relevance of information leakage in a system, based on how much the leakage enables an adversary to harm the correct behaviour of the system. We say that two information flows are *effectively equivalent* if the strategic ability of the adversary is similar in both of them. Moreover, one of them is *less effectively secure* than the other one if the amount of information leaked to the adversary in it increases the damaging ability of the adversary. In order to determine how critical the information leakage in a given system is, we compare the damaging ability of the adversary to his ability in the idealised variant of the model. We defined the idealised model based on noninterference and showed that the construction is well defined. We proved this first for the total-on-input models of Goguen and Meseguer, and then extended the results to non-total-on-input structures that allow for a more flexible modelling of interaction.

Chapter 8

Expressing Receipt-Freeness and Coercion-Resistance in Logics of Strategic Ability

In this chapter, we focus on the strategic aspect of information security in voting procedures. We argue that coercion-related properties are underpinned by existence (or nonexistence) of a suitable strategy for some participants: typically for the voter, the coercer, or both. Such strategic behaviour has been studied in game theory, social choice theory, and theory of multi-agent systems. In particular, a number of *game logics* have been proposed that can be used to specify properties related to strategic ability. Here, we use formulae of the game logic $\mathbf{ATL}^*$ to encode and disambiguate different flavours of receipt-freeness and coercion-resistance.

Outline of the chapter. Section 8.1 explains the motivating idea of the contribution of this chapter. Section 8.2 presents the syntax and semantics of the logic of strategic abilities $\mathbf{ATL}^*$. In Section 8.3 we discuss some of the significant works that have given formal definitions of receipt-freeness and coercion-resistance. We provide logical expressions for the informal definitions included in those works. Finally Section 8.4 concludes the chapter.

8.1 Motivation

In the existing literature, coercion-related properties are typically formulated on two levels of abstraction (For a more thorough literature review on formal definitions of coercion-resistance properties refer to Chapter 2.). On one hand, the *informal intuition* usually builds upon abilities of participants in the interaction between the voter and the potential coercer. That is, it refers to the existence or nonexistence of suitable strategies for players in the *real* game between the voter and the coercer.

On the other hand, the*formal definition* specifies a mathematical structure to which the property is related, and defines how to evaluate the property based on the structure. Some of the formal definitions are game-based, but the games used there are primarily mathematical devices to define the concept, much like in the case of the game semantics for first-order predicate logic, or the game semantics of programming languages. It is *not* the real game between participants of the voting process, but rather an abstract game between the "verifier" trying to prove the property true, and the "falsifier" that attempts the opposite. Thus, strategy-based definitions of coercion-resistance and receipt-freeness are either informal or use strategies that have no obvious relation to the real behaviour of actual participants in the voting process. The closest work we know of, that has formalized the coercion-resistance property as strategic abilities of the participants, is the work of Kusters et al. [KTV10]. They have formalised the property as a quantitative measure showing how well a coercer can distinguish between the strategy he has prescribed to the coerced voter and a counter-strategy used by the voter. However for formalising the property they have also used a cryptographic model, rather than a model oriented for specifying strategic abilities of players.

Our aim in this chapter is to provide logical "transcriptions" of the informal intuitions that can be found in the literature. It is important to mention that this is essentially a position chapter. We put forward some specifications of the coercion related properties that can trigger discussions. These discussions hopefully can lead to the improvement of the formalizations in the future. Also, we do not claim that the formalizations we formulate in this chapter get the concepts of coercion-resistance and receipt-freeness *completely right*. In fact, the issue whether they "get it right" is not entirely well-formed, because the model and the semantics of the specifications are not formulated precisely. Nevertheless, the transcriptions formally expose the strategic nature of coercion-related properties, and allow to demonstrate some interesting differences between the existing approaches.

8.2 Logics of Strategic Ability

The idea behind here is to capture the intuitive meaning of coercion-related properties by formal specifications that explicitly refer to the strategic interaction between the voter(s) and the coercer(s). We will show a number of logical formulae that refer to the existence (or nonexistence) of strategies to coerce (resp. to defend from coercion). To this end, we will use *modal logics of strategic ability*, or *modal game logics* [BP88, AHK02, Sch04, CHP07, MMV10], that have gained much popularity within Artificial Intelligence in the last 20 years.

There are many syntactic and semantic variants of game logics. In this work, we use *alternating-time temporal logic* **ATL** whose formulae allow for expressing statements about the existence of a surely winning strategy to achieve a given temporal goal.

8.2.1 What Agents Can Achieve: ATL and ATL*

Alternating-time temporal logic [AHK97, AHK02] generalizes branching-time temporal logic **CTL*** [Eme90] by replacing path quantifiers E, A with *strategic modalities* $\langle\langle A \rangle\rangle$. Informally, $\langle\langle A \rangle\rangle \gamma$ says that a group of agents A has a collective strategy to enforce temporal property γ. **ATL***

formulas can include temporal operators: "$\bigcirc$" ("in the next state"), "$\Box$" ("always from now on"), "$\Diamond$" ("now or sometime in the future"), and $\mathcal{U}$ (strong "until"). Similarly to **CTL*** and **CTL**, we consider two syntactic variants of the alternating-time logic, namely **ATL*** and **ATL**.

Syntax. Formally, let $\mathbb{A}$ be a finite set of agents, and Π a countable set of atomic propositions. The language of **ATL*** is defined as follows:

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\langle A \rangle\!\rangle \gamma,$$
$$\gamma ::= \varphi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \bigcirc\gamma \mid \gamma\,\mathcal{U}\gamma.$$

where $A \subseteq \mathbb{A}$ and $\mathsf{p} \in \Pi$. Derived boolean connectives and constants $(\vee, \top, \bot)$ are defined as usual. "Sometime", "weak until", and "always from now on" are defined as $\Diamond\gamma \equiv \top\,\mathcal{U}\gamma$, $\gamma_1\,\mathcal{W}\gamma_2 \equiv \neg((\neg\gamma_2)\,\mathcal{U}(\neg\gamma_1 \wedge \neg\gamma_2))$, and $\Box\gamma \equiv \gamma\,\mathcal{W}\bot$.

ATL (without "star") is the syntactic variant in which strategic and temporal operators are combined into compound modalities:

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\langle A \rangle\!\rangle \bigcirc \varphi \mid \langle\!\langle A \rangle\!\rangle \varphi\,\mathcal{U}\varphi \mid \langle\!\langle A \rangle\!\rangle \varphi\,\mathcal{W}\varphi.$$

Models. The semantics of **ATL*** is defined over a variant of synchronous multi-agent transition systems.

Definition 8.2.1 (CGS). *A* concurrent game structure (CGS) *is a tuple $M = \langle \mathbb{A}, St, Act, d, o, \Pi, V \rangle$ which includes nonempty finite sets of: agents $\mathbb{A} = \{1, \dots, k\}$, states St, actions Act, atomic propositions Π, and a propositional valuation $V : St \to 2^\Pi$. The function $d : \mathbb{A} \times St \to \mathcal{P}(Act)$ defines availability of actions. The (deterministic) transition function o assigns a successor state $q' = o(q, \alpha_1, \dots, \alpha_k)$ to each state $q \in St$ and any tuple of actions $\alpha_i \in d(i, q)$ that can be executed by $\mathbb{A}$ in q.*

A pointed CGS *is a pair (M, q_0) consisting of a concurrent game structure M and an initial state q_0 in M.*

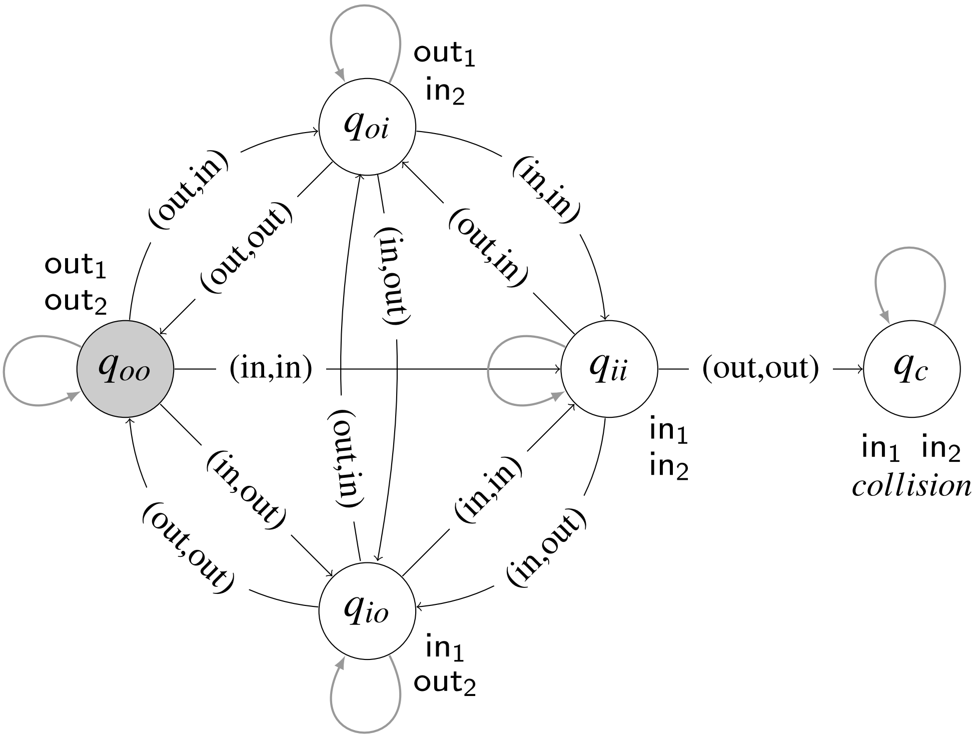

Figure 8.1 Autonomous vehicles at the intersection: model M_1

Example 8.2.1 (Driving agents)**.** *Consider an intersection with k autonomous vehicles around it. Each vehicle is modelled as a separate agent, whose local state is characterised by either the proposition* out$_i$ *(when the vehicle is outside the intersection) or* in$_i$ *(when the vehicle is inside it). The available actions are: in ("drive in" or "stay in", depending on the current state) and out ("drive out" or "stay out"). Transitions update the state accordingly, except for one case: when both agents are in and decide to leave at the same time, a collision occurs (*collision*).*

Figure 8.1 presents a pointed CGS modeling the scenario for $k = 2$. The combinations of actions that are not displayed in the graph do not change the state of the system.

Strategies and their outcomes. Given a CGS, we define the strategies and their outcomes as follows. A *strategy* for a is a function $s_a : St \to Act$ such that $s_a(q) \in d(a,q)$.[1] The set of such

[1] This corresponds to the notion of *memoryless* or *positional* strategies. In other words, we assume that the memory

strategies is sometimes denoted by Σ_a^{Ir}, with the capital "I" referring to perfect **I**nformation, and the lowercase "r" for possibly imperfect **r**ecall. A *collective strategy* for a group of agents $A = \{a_1, \ldots, a_r\}$ is a tuple of individual strategies $s_A = \langle s_{a_1}, \ldots, s_{a_r} \rangle$. The set of such strategies is denoted by Σ_A^{Ir}.

A *path* $\lambda = q_0 q_1 q_2 \ldots$ in a CGS is an infinite sequence of states such that there is a transition between each q_i, q_{i+1}. $\lambda[i]$ denotes the ith position on λ (starting from $i = 0$) and $\lambda[i, \infty]$ the suffix of λ starting with i. The "outcome" function $out(q, s_A)$ returns the set of all paths that can occur when agents A execute strategy s_A from state q onward. Function $out(q, s_A)$ returns the set of all paths $\lambda \in St^{\omega}$ that may occur when agents A execute strategy s_A from state q onward, defined as follows:

$$out(q, s_A) = \{\lambda = q_0, q_1, q_2 \ldots \mid q_0 = q \text{ and for each } i = 0, 1, \ldots \text{ there exists } \langle \alpha^i_{a_1}, \ldots, \alpha^i_{a_k} \rangle \text{ such that } \alpha^i_a \in d_a(q_i) \text{ for every } a \in \mathfrak{A}, \text{ and } \alpha^i_a = s_A[a](q_i) \text{ for every } a \in A, \text{ and } q_{i+1} = o(q_i, \alpha^i_{a_1}, \ldots, \alpha^i_{a_k})\}.$$

Semantics. The semantics of **ATL*** is defined by the following clauses:

$M, q \models \mathsf{p}$ iff $q \in V(\mathsf{p})$, for $\mathsf{p} \in \Pi$;

$M, q \models \neg\varphi$ iff $M, q \not\models \varphi$;

$M, q \models \varphi_1 \wedge \varphi_2$ iff $M, q \models \varphi_1$ and $M, q \models \varphi_2$;

$M, q \models \langle\langle A \rangle\rangle \gamma$ iff there is a strategy $s_A \in \Sigma_A^{\mathrm{Ir}}$ such that, for each path $\lambda \in out(q, s_A)$, we have $M, \lambda \models \gamma$.

$M, \lambda \models \varphi$ iff $M, \lambda[0] \models \varphi$;

$M, \lambda \models \neg\gamma$ iff $M, \lambda \not\models \gamma$;

$M, \lambda \models \gamma_1 \wedge \gamma_2$ iff $M, \lambda \models \gamma_1$ and $M, \lambda \models \gamma_2$;

$M, \lambda \models \bigcirc\gamma$ iff $M, \lambda[1, \infty] \models \gamma$; and

of agents is explicitly defined by the states of the model.

$M,\lambda \models \gamma_1 \mathcal{U} \gamma_2$ iff there is an $i \in \mathbb{N}_0$ such that $M,\lambda[i,\infty] \models \gamma_2$ and $M,\lambda[j,\infty] \models \gamma_1$ for all $0 \leq j < i$.

Example 8.2.2 (Driving agents, ctd.). *For model M_1, we have $M_1, q_{oo} \models \langle\langle 1\rangle\rangle \Box \neg \mathsf{collision}$: agent 1 can avoid the collision forever (the obvious strategy is to never enter the crossroads). On the other hand, the agent cannot ensure the collision even if it wants to: $M_1, q_{oo} \models \neg\langle\langle 1\rangle\rangle \Diamond \mathsf{collision}$. This can only be guaranteed if the agents cooperate: $M_1, q_{oo} \models \langle\langle 1,2\rangle\rangle \Diamond \mathsf{collision}$. Moreover, $M_1, q_{oo} \models \langle\langle 1\rangle\rangle \Diamond \mathsf{in}_1 \wedge \langle\langle 2\rangle\rangle \Diamond \mathsf{in}_2$: each agent is able to enter the intersection. Still, it cannot successfully drive through the crossroads on its own (e.g., $M_1, q_{oo} \not\models \langle\langle 1\rangle\rangle \Diamond(\mathsf{in}_1 \wedge \Diamond \mathsf{out}_1)$). Finally, if the agents cooperate, they can make sure that they successfully drive through the crossroads: $M_1, q_{oo} \models \langle\langle 1,2\rangle\rangle \Diamond(\mathsf{in}_1 \wedge \Diamond \mathsf{out}_1)$.*

8.2.2 Abilities under Imperfect Information

ATL* was originally proposed for reasoning about agents in perfect information scenarios. However, realistic multi-agent interaction always includes some degree of limited observability [Sch04, JvdH04, Ågo06, HT06, JÅ07, Sch10]. Here, we use the classical variant of "**ATL*** with imperfect information" from [Sch04], defined as follows.

First, we extend concurrent game structures with indistinguishability relations $\sim_1, \ldots, \sim_k$, one per agent in $\mathbb{A}$. Now, strategies must specify identical choices in indistinguishable situations. That is, strategies with imperfect information (ir strategies, for short) are functions $s_a : St \to Act$ such that (1) $s_a(q) \in d(a,q)$, and (2) if $q \sim_a q'$ then $s_a(q) = s_a(q')$.[2] As before, collective strategies for $A \subseteq \mathbb{A}$ are tuples of individual strategies for $a \in A$. We denote the set of A's imperfect information strategies by Σ_A^{ir}.

The semantics of "**ATL*** with imperfect information" differs from the one presented in Section 8.2.1 only in the clause for strategic modality:

[2] Again, we consider only positional strategies here.

$M, q \models \langle\langle A \rangle\rangle \gamma$ iff there is a strategy $s_A \in \Sigma_A^{\text{ir}}$ such that, for every agent $a \in A$, state q' such that $q \sim_a q'$, and path $\lambda \in out(q', s_A)$, we have that $M, \lambda \models \gamma$.

In other words, the agents in A should have an executable strategy which enforces γ from all the states that at least one member of the coalition considers possible.

Example 8.2.3 (Intersection with limited visibility)**.** *Take model M_1 from Example 8.2.1, and assume that no agent sees the location of the other vehicle. This can be modeled by the following indistinguishability relations: $q_{oo} \sim_1 q_{oi}$ and $q_{ii} \sim_1 q_{io}$; $q_{oo} \sim_2 q_{io}$ and $q_{oi} \sim_2 q_{ii}$. Now, we still have e.g. that $M_2, q_{oo} \models \langle\langle 1 \rangle\rangle \Box \neg \mathsf{collision}$ (it suffices that agent 1 executes action "out" regardless of anything) On the other hand, $M_2, q_{oo} \models \neg \langle\langle 1,2 \rangle\rangle \Diamond \mathsf{collision}$ (the agents cannot make sure that a collision will happen, even if they want to). We leave it up to the interested reader to check the latter.*

8.2.3 Knowledge and Belief Modalities

Coercion-resistance and receipt-freeness are privacy-type properties. In this sense, they are related to the *knowledge* and/or *beliefs* of the adversary about a given secret. In the case of elections, the secret is usually the value of the voter's vote. Thus, we will need modalities for knowledge (resp. beliefs). The former is formalised by epistemic formulae of type $K_a\varphi$, expressing that agent *a knows that φ holds*, with the following semantics:

$M, q \models K_a\varphi$ iff, for every state q' such that $q \sim_a q'$, we have that $M, q' \models \varphi$.

It is interesting to observe that this modality is in fact superfluous in "**ATL*** with imperfect information," since we can equivalently express $K_a\varphi$ by $\langle\langle a \rangle\rangle \varphi \mathcal{U} \varphi$.

The modality for beliefs is very similar. $B_a\varphi$ expresses that *a believes* that ϕ holds, and exactly the same semantic clause can be used to interpret $K_a\varphi$ and $B_a\varphi$. The difference lies in the axiomatic properties. For knowledge, the indistinguishability relation is assumed to be an equivalence (i.e., reflexive, symmetric, and transitive), whereas for beliefs it is sufficient to have it serial, symmetric and Euclidean. Thus, whenever the indistinguishability relation $\sim_a$ is an equivalence, one can use K_a to address the subjective view of player a; otherwise B_a should be used.

8.3 Expressing Informal Definitions of Coercion-Related Properties

In order to express security properties of a voting system, we assume that the voting process is modelled as a concurrent game structure where the set of players $\mathbb{A}$ includes the set of voters V, the coercer c, and possibly some other players. Let Bal be the set of possible "ballot values," i.e., ways in which a ballot can be cast by a voter. In a simple majority voting procedure where each voter votes for one of the candidates, Bal is the set of candidates. We assume that the states where voter $v \in V$ has already voted are labelled by the atomic proposition $\mathsf{voted}_{\mathsf{v,i}}$, where $i \in Bal$ indicates how v voted.

For this work, we have chosen several important papers on preventing coercion in elections. In each of the papers, an informal intuition is first given and later followed by a formal definition that typically uses some heavy mathematical machinery. Here, we only look at the informal intuitions to provide their transcriptions in the game logic **ATL***. To make the list easier to read, we label the properties to be transcribed as either **(RF***x***)** for variants of receipt-freeness properties, and **(CR***x***)** for variants of coercion-resistance.

8.3.1 Benaloh and Tuinstra (1994)

(RF1) For a voting system to be uncoercable, no voter should be able to convince any other participant of the value of its vote. [BT94]

This paper introduced the notion of receipt-freeness. They used some examples to show why giving a “receipt” to the voter can be harmful, as it prevents the voter from being able to deceive the coercer. Therefore through the paper “uncoercibility” is regarded equivalent to receipt-freeness.

For expressing this definition, we need to interpret two terms *to convince’* and *other participants*. We can consider “other participants” to be the set of all voters, or to be the set of all players. The more subtle term to interpret is “to convince”. It can both mean to prove to someone about one’s vote value, and to make someone believe that the voter has voted in a particular way. In the first case, the knowledge modality is the right one to use and in the second case the belief modality. However, because in this definition “being unable to convince” is used only for the actual vote of the voter, we decide to use knowledge modality for expressing the property. Therefore definition **(RF1)**, if we consider “other participants” to be the set of voters, can be expressed as:

$$\bigwedge_{\substack{v,v' \in V \setminus \{c\} \\ v \neq v'}} \bigwedge_{i \in Bal} \neg \langle\langle v \rangle\rangle \Diamond (\mathsf{voted}_{\mathsf{v,i}} \wedge K_{v'} \mathsf{voted}_{\mathsf{v,i}}),$$

or if we consider “other participants” to be any other player in the model:

$$\bigwedge_{v \in V} \bigwedge_{\substack{a \in \mathfrak{A} \\ v \neq a}} \bigwedge_{i \in Bal} \neg \langle\langle v \rangle\rangle \Diamond (\mathsf{voted}_{\mathsf{v,i}} \wedge K_{a} \mathsf{voted}_{\mathsf{v,i}}).$$

Note that $\bigwedge$ is not a first-order quantifier but a conjunction of finitely many subformulae. Thus, the above specifications are propositional modal formulae of finite length.

8.3.2 Juels, Catalano, and Jakobsson (2005)

This paper introduced the notion of coercion-resistance property as an improvement over receipt-freeness. We start by the definition of receipt-freeness as given in this work:

(RF2) Receipt-freeness is the inability of a voter to prove to an attacker that she voted in a particular manner, even if the voter wishes to do so. [JCJ05]

This definition is very similar to **(RF1)** The difference is that instead of any other player, we use a coercer player as the adversary. Although one might think the two interpretations imply each other, they can, in fact, have different nuances. One may define specific abilities for the adversaries in the model that are different from those accessible to the voters, or other players in the system. Also, one might consider some (maybe powerful) players in the model as trustworthy and decide not to include them in the set of coercers. The definition **(RF2)** then can be expressed as follows:

$$\bigwedge_{v\in V\setminus\{c\}} \bigwedge_{i\in Bal} \neg\langle\langle v\rangle\rangle\Diamond(\mathsf{voted}_{\mathsf{v,i}} \wedge K_c\mathsf{voted}_{\mathsf{v,i}}).$$

The definition of the coercion-resistance property given in this paper is meant to give extra protection, where receipt-freeness fails to protect an election system against several forms of serious, real-world attack, specifically against randomization attacks, forced abstention attacks, and simulation attacks. In randomization attack, the coercer asks the voter to use some randomization method for choosing her vote. In forced abstention attack the attacker wants the voter to avoid voting in the election, and in simulation attack, the attacker himself simulate the role of the voter (for example by causing her to divulge her private keying material after the registration, but before the election process).

(CR1) A coercion-resistant voting system is one in which the user can deceive the adversary into thinking that she has behaved as instructed, when the voter has in fact cast a ballot according to her own intentions. [JCJ05]

This definition includes *instructions* of the coercer to the voter. The paper explains that these instructions can be voting for a specific candidate, but also abstaining from voting, randomising the vote, and in general any specific behaviour during the election process. Here, we focus only on instructing to vote for a specific candidate and abstaining from voting, and we discuss the other cases later. Notice that in this definition, the voter intends to deceive the adversary to accept the voter has voted in a way which is not the actual vote of the voter. This means that the knowledge modality cannot be used here because in classical epistemic logic, knowledge about a proposition implies the truth of it in all possible worlds. Hence we use belief modalities in this case. The - narrowly interpreted - expression of definition **(CR1)** then can be as follows:

$$\bigwedge_{v \in V \setminus \{c\}} \bigwedge_{i,j \in Bal} \langle\langle v \rangle\rangle \Diamond (\mathsf{voted}_{\mathsf{v,i}} \wedge B_c \mathsf{voted}_{\mathsf{v,j}}).$$

8.3.3 Delaune, Kremer, and Ryan (2005)

(RF3) An election protocol is receipt-free if a voter *A* cannot prove to a potential coercer *C* that she voted in a particular way. We assume that *A* wishes to cooperate with *C*; receipt-freeness guarantees that such cooperation will not be worthwhile, because it will be impossible for *C* to obtain proof about how *A* voted. [DKR05]

The paper proposed a formalization of receipt-freeness in applied pi calculus. Here, unlike the previous definitions, a cooperation between the coercer and the voter has been mentioned explicitly. Expressing this definition can be as follows:

$$\bigwedge_{v \in V \setminus \{c\}} \bigwedge_{i \in Bal} \neg \langle\langle c, v \rangle\rangle \Diamond (\mathsf{voted}_{\mathsf{v,i}} \wedge K_c \mathsf{voted}_{\mathsf{v,i}}).$$

It is important to note that in the semantics of **ATL**, the existence of a collective strategy for a group of players doesn't imply that the players are committed to follow that strategy. It also doesn't necessary mean that each player has a way of distinguishing whether the other players in the coalition are following the strategy or not. To show how these affect the definition, suppose in some election the voter has options of voting for candidates *A* or *B*. Then she can *choose a receipt* for either candidates *A* or *B*. This is entirely the voter's choice and both choices are offered no matter which vote was cast. The voter then can give the receipt to the coercer. It is clear that there is always a strategy for the voter of voting for a candidate and always taking the receipt for that candidate. Under that strategy, in all possible executions of the system the coercer knows which way the voter has voted. Therefore the voter and the coercer have a joint cooperative strategy in which the coercer knows how the voter voted. However it is obvious that such a system cannot be considered to be receipt-free. This problem arises because in **ATL**, the transcription of the definition given above, does not enforce the voter to follow the (agreed) strategy, nor does it imply that the coercer can distinguish whether the voter has diverged from the strategy or not. For having these properties we need a more expressive language than **ATL**. We will mention this issue again while talking about the future works in Chapter 9.

8.3.4 Moran and Naor (2006)

(RF4) A voting system is receipt free, if a voter is unable to convince a third party of her vote even if she wants to do so. [MN06]

This work gives a formal definition for receipt-freeness in computation model. The differences between this definition and **(RF3)** is that firstly here the adversary can be any other player, and secondly the term "to convince" is used instead of "to prove". We can include the latter by replacing the knowledge modality with belief modality. So **(RF4)** can be expressed as follows:

$$\bigwedge_{v \in V} \bigwedge_{\substack{a \in \mathfrak{U} \\ v \neq a}} \bigwedge_{i \in Bal} \neg \langle\langle v \rangle\rangle \Diamond (\mathsf{voted}_{\mathsf{v,i}} \wedge B_a \mathsf{voted}_{\mathsf{v,i}}).$$

8.3.5 Backes, Hritcu, and Maffei (2008)

(RF5) A voting system satisfies receipt-freeness, if a coercer cannot force a voter to cast a certain vote and to provide a receipt that would certify her vote. [BHM08]

This paper provided a formalisation of coercion-resistance and receipt-freeness in applied pi calculus. A key term in the informal definition here is “to force”. Because there is not a way to exactly express *forcing someone to do something* in ATL, we interpret it as though the coercer has a way to make voter to commit to a mutual strategy. In this way, *forcing the voter* can be interpreted as *having a mutual strategy with the voter*. The other key term here is “the receipt”. Again, for being able to express the informal definition in ATL, we replace the concept of *existence of a receipt*’ with a more general concept of *existence a strategy to prove the value of the vote*. With these interpretations we can express the definition as follows, which is similar to **(RF3)**:

$$\bigwedge_{v\in V\setminus\{c\}} \bigwedge_{i\in Bal} \neg\langle\langle c,v\rangle\rangle\Diamond(\mathsf{voted}_{\mathsf{v,i}} \wedge K_c\mathsf{voted}_{\mathsf{v,i}}).$$

8.3.6 Delaune, Kremer, and Ryan (2010)

(RF6) A voting system is receipt free, if the voter does not obtain any artefact (a “receipt”) which can be used later to prove to another party how she voted. [DKR10]

Here again similar to definition **(RF5)**, we translate the *having a receipt* to *having a strategy to prove the value of the vote*. Therefore the definition can be expressed as:

$$\bigwedge_{v\in V} \bigwedge_{\substack{a\in\mathfrak{A}\\ v\neq a}} \bigwedge_{i\in Bal} \neg\langle\langle v\rangle\rangle\Diamond(\mathsf{voted}_{\mathsf{v,i}} \wedge K_a\mathsf{voted}_{\mathsf{v,i}}).$$

(CR2) A voting system is coercion-resistant, if the link between a voter and her vote cannot be established by an attacker, even if the voter cooperates with the attacker during the election process. We assume that the voter and the attacker can communicate and exchange data at any time during the election process. [DKR10]

If we translate *establishing a link between a voter and her vote* by *knowing the value of the vote of the voter*, then this definition can be expressed similar to definitions **(RF3)** and **(RF5)**:

$$\bigwedge_{v\in V\setminus\{c\}} \bigwedge_{i\in Bal} \neg\langle\langle c,v\rangle\rangle\Diamond(\mathsf{voted}_{\mathsf{v,i}} \wedge K_c\mathsf{voted}_{\mathsf{v,i}}).$$

On the other hand, if we take it as a more general concept of *finding any correlation between the voter and his vote*, then we can express it as:

$$\bigwedge_{v\in V\setminus\{c\}} \bigwedge_{i\in Bal} \neg\langle\langle c,v\rangle\rangle\Diamond(\mathsf{voted}_{\mathsf{v,i}} \wedge \bigvee_{j\in Bal\setminus\{i\}} K_c\neg\mathsf{voted}_{\mathsf{v,j}}).$$

8.3.7 Kusters, Truderung and Vogt (2010)

(CR3) A voting system is coercion-resistant, if there exists a counter-strategy for the voter such that the coercer cannot tell whether the coerced voter is in fact following the coercer's instructions or whether she is just running the counter-strategy, and hence, achieves her own goal. [KTV10]

The counter-strategy of the voter in this definition has to satisfy two conditions. Firstly it has to be indistinguishable from the instructed strategy of the coercer and secondly, makes the voter achieves her goal. Here again, we focus only on the simple case where the coercer's instruction is basically voting for a certain candidate, and the goal of the voter is voting for her preferred candidate. The definition then can be expressed as follows:

$$\bigwedge_{v\in V\setminus\{c\}} \bigwedge_{\substack{i,j\in Bal\\ i\neq j}} \langle\langle v\rangle\rangle\Diamond(\mathsf{voted}_{\mathsf{v,i}} \wedge \Box\neg K_c\neg\mathsf{voted}_{\mathsf{v,j}}).$$

That is, the voter always has a strategy to eventually vote for her preferred candidate, and be sure that the coercer never finds out that she has disobeyed his instruction.

8.3.8 Randomization and Forced Abstention Attacks

In several definitions of coercion-resistance, like **(CR1)**, the security property is meant to satisfy receipt-freeness but also protect against forced abstention attacks and randomization attack. We have omitted those kinds attacks in the previous subsections. Here, we tentatively suggest how resistance to randomization and forced abstention attacks can be specified.

Randomization attack happens when the coercer wants the voter to cast her vote in a way that the result follows some probability distribution, for example, uniform distribution over the set of candidates. However, if the coercer instructs the voter to "vote at random with a uniform distribution over the candidates", he will have no way of checking whether the voter followed his instruction. Therefore, in the case of election, a randomization attack is possible only when there exist a more tangible property (from the point of view of the coercer) for verifying the randomness of a single cast vote. It must be, at least in principle, possible for the coercer to verify the property based on the actual sequence of events (and not the voter's behaviour as a whole, which is inaccessible to the coercer). For example in Prêt à Voter [CRS05, Rya10], the list of candidates are printed in a entirely random way on each ballot. Therefore if the coercer asks the voter to "cross the first slot in the ballot", this is potentially verifiable, and indirectly it implies random voting of the voter. In fact, the voter always has the possibility of auditing a ballot and obtaining a new one, so the voter has a counter strategy to get a ballot with her candidate in the coercer specified position, but of course this would be tedious to execute for complex ballots.

To make this idea more general, we can represent *feasible* randomization attacks by a state property p, such that the occurrence of p implies random behaviour of the voter, in the way intended by the coercer. For example, the above scenario can be represented by $\mathsf{p} \equiv \mathsf{crossed}_{\mathsf{v},1}$, where $\mathsf{crossed}_{\mathsf{v,n}}$ expresses that voter v has crossed the nth slot on the ballot. Then, resistance to randomization attacks can be approximated by the following formula:

$$\bigwedge_{v \in V \setminus \{c\}} \neg\langle\langle c, v\rangle\rangle \Diamond K_c \mathsf{p}.$$

That is, there is no collective strategy for the coercer and the voter (even assuming that they fully cooperate) such that at some point the coercer will know that p has occurred, and hence conclude that the voter has followed his instruction.

Forced abstention attack happens when the coercer wants the voter to behave such that her vote does not affect the final ballot counting result. It can be only asking the voter not to cast a vote, but also can be wanting the voter to cast a vote in a way that is considered a spoilt vote. For the case of asking for casting a spoilt vote, the formalisation can be similar to the case of randomization attack, where the occurrence of a potentially verifiable state property p means that the voter casts a spoilt vote. For the case where the coercer asks the voter not to participate in the election(or not to cast a vote) the protection against forced abstention attack can be expressed as follows:

$$\bigwedge_{v \in V \setminus \{c\}} \neg\langle\langle c, v\rangle\rangle \Box \left(\bigwedge_{i \in Bal} \neg\mathsf{voted}_{\mathsf{v,i}} \wedge K_c \bigwedge_{i \in Bal} \neg\mathsf{voted}_{\mathsf{v,i}} \right).$$

That is, there is no collective strategy for the coercer and the voter such that at any time after the end of the election time, the coercer knows that the voter has voted for a candidate (any candidate, even if the coercer cannot figure out which one).

8.4 Summary

In this chapter, we have provided logical transcriptions of the informal form of the definitions of the coercion-resistance and receipt-freeness properties that can be found in the literature. Our focus in these transcriptions was to show the role of strategic abilities of the participants in an election system in defining coercion related properties. To this end, we used formulae of the game logic **ATL*** for expressing these properties, and demonstrated some differences between the existing approaches.

Chapter 9

Conclusions and Future Work

In this dissertation, we used game theoretic approaches for analysing security properties within three separate strands of work. In the following, for each strand, we summarise the main results and contribution of the thesis, and present a few suggestions for future work.

Strand 1: Preventing Coercion in Elections. The primary objective of this strand was to model and analyse an election system in the presence of potential coercers, in order to find optimised strategies for the decision makers of the election, and to minimise the risk and expected damage of coercion.

We presented simple game models for an election with the possibility of coercion. The models are two-person non-zero-sum noncooperative games, where one player represents the society and the other a potential coercer in the election. We considered both complete information and incomplete information settings where the players are not sure about the number of voters that the coercer needs to coerce to change the result of the election in his favour. In the incomplete information setting, we have studied both the uniform distribution and normal distribution for the number of voters needed to be coerced. We showed that in all the games we considered, Stackelberg equilibrium is different from Nash equilibrium, and also the Stackelberg equilibrium does not coincide with maxmin. It means that it is in the interest of the society *not* to adapt to the expected strategy of the coercer. Rather, the society should decide on its coercion-resistance policy in advance, announce it openly and commit to using it. This way, the coercer is best off when refraining from coercion altogether.

For future work, it would be interesting to study models where the game consists of multiple coercers whose interests conflict. Extending models to include incomplete information of players about other aspects of the game, like the value of coercion for the coercer, or the cost of bribing voters, is another possible line of research for future work.

Strand 2: Information Flow Security and Strategic Abilities of Players. The first objective of this strand was to define a weaker notion of noninterference as an information flow security property, by considering the strategic ability of High players and their incentive to ensure the correct behaviour of the system.

We proposed how to relax the classical requirement of noninterference by taking into account a strategy that the High players may follow in order to achieve their goals. The idea is of particular importance for the design and analysis of confidentiality in realistic systems where full noninterference and nondeducibility can seldom be guaranteed. Moreover, strategic noninterference in a system can be obtained not only by strengthening security measures, but also by "fine-tuning" functionality requirements: even if it does not hold for the current goals, there may exist weaker yet still acceptable goals that allow for confidentiality-preserving behaviour. Thus, the new concept helps to realise which objectives can be achieved while avoiding information leakage.

Regarding the technical results, we studied the characterisation of strategic noninterference through unwinding relations. On the one hand, we proved that a general characterisation result is impossible for arbitrary goals. On the other hand, we presented some characterizations for specific subclasses of goals and for the simplified setting where a strategy is given as a parameter. The proofs are constructive and can be used to obtain practical algorithms that check for strategic noninterference. We also showed that, in the classical models of Goguen and Meseguer, knowing the strategy of High players usually does not increase the ability of Low players to break noninterference. It is worth mentioning that, in a realistic system, the usefulness of strategic noninterference relies heavily on the ability of High players to select specific behaviours. In a system where High players have no such ability, the notions of noninterference and strategic noninterference coincide.

The models we used are deterministic asynchronous transition networks of the original definition of noninterference [GM82]. We plan to extend our study to richer models in future work. In particular, the generalised form of non-interference by Ryan and Schneider [RS01] seems very promising for a formulation of strategic noninterference in process-algebraic models.

The second objective of this strand was to study the significance of information flow in a system, based on its influence on increasing the strategic ability of the adversary to harm the goal of the system.

As our contribution in this line of work, we introduced the concept of *effective information security*. The idea is aimed at assessing the relevance of information leakage in a system, based on the extent to which the leakage enables an adversary to harm the correct behaviour of the system. This is in contrast with the common approach to information flow security where revealing any information is seen as being intrinsically harmful. We say that two information flows are *effectively equivalent* if the strategic ability of the adversary is similar in both of them. Moreover, one of them is *less effectively secure* than the other one if the amount of information leaked to the adversary in it increases the damaging ability of the adversary.

To determine how critical the information leakage is in a given system, we compare the damaging ability of the adversary to his ability in the idealised variant of the model. We defined idealised models based on noninterference, and show that the construction is well defined. We proved this first for the deterministic, fully asynchronous transition networks of Goguen and Meseguer, and then extended the results to structures that allow for a more flexible modelling of interaction. The construction includes an algorithm that computes the idealised variant of each model in polynomial time with respect to the size of the model.

Note that the concept of noninterference in our construction of effective security can be in principle replaced by an arbitrary property of information flow. The same reasoning scheme could be applied to noninference, nondeducibility, strategic noninterference, and so on. The pattern does not change: given a property $\mathscr{P}$, we define the idealised variant of M through the minimal unification U such that $U(M)$ satisfies $\mathscr{P}$. Then, M is effectively information-secure in the context of property $\mathscr{P}$ iff it is strategically equivalent to $U(M)$. Therefore one suggestion for future work is to investigate which information security properties have unique minimal unifications.

Moreover, we plan to work on a more refined version of effective information security based on coalitional effectivity functions [AK91], in which the strategic ability of the adversary is not only compared at the initial state of the system, but across the whole state space. This will allow us to achieve a more refined comparison between the security of different information flows.

Strand 3: Expressing Receipt-Freeness and Coercion-Resistance in Logics of Strategic Ability. The main objective of this strand was to use logics of strategic ability to provide "transcriptions" of informal intuitions behind the definitions of receipt-freeness and coercion resistance properties in the literature.

In this line of work, we chose some of the significant works in the literature that have provided formal definitions of coercion-resistance and receipt-freeness, and used formulae of the game logic **ATL*** for expressing the informal form of the definitions in these works. Our objective in these transcriptions was to show the role of strategic abilities of the participants in defining coercion related security properties, and to demonstrate some differences between the existing approaches to these properties.

There are many possible paths for future work. Among other things, we plan to refine our specifications using the more flexible language of Strategy Logic [MMV10, FMV12, MMPV14] that allows for explicit quantification over strategies in *k*-player concurrent games. In particular, this should allow for a more general specification of resistance to randomization and abstention attacks, by directly encoding the fact that the coercer is unable to distinguish between the actual behaviour of the voter and the behaviour prescribed by the coercer.

Perhaps more importantly, we will try to map the *formal* definitions of receipt-freeness and coercion resistance from [BT94, JCJ05, DKR05, MN06, BHM08, DKR10, KTV10] to models and formulae of game logics, in order to study the precise relationship between the informal intuitions and their formalizations. Adapting the **ATL** model checking algorithms so that they can be used to verify coercion-related properties is the third line of research that we envisage for future research.

Finally, we plan to study how *opacity*, as defined by Bryans et al. [BKMR05, BKR05] can be expressed in **ATL**, and more specifically how to define coercion-resistance as a flavour of opacity, similar to the style suggested in [PR06].

Bibliography

[AG04] A. Acquisti and J. Grossklags. Privacy attitudes and privacy behavior - losses, gains, and hyperbolic discounting. In *Economics of Information Security*, volume 12 of *Advances in Information Security*, pages 165–178. Springer, 2004.

[Ågo06] T. Ågotnes. Action and knowledge in alternating-time temporal logic. *Synthese*, 149(2):377–409, 2006. Section on Knowledge, Rationality and Action.

[AH99] R. Alur and T. A. Henzinger. Reactive modules. *Formal Methods in System Design*, 15(1):7–48, 1999.

[AHK97] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 100–109. IEEE Computer Society Press, 1997.

[AHK02] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002.

[AK91] J. Abdou and H. Keiding. *Effectivity Functions in Social Choice*. Springer, 1991.

[ALBD04] Riza Aditya, Byoungcheon Lee, Colin Boyd, and Ed Dawson. An efficient mixnet-based voting scheme providing receipt-freeness. In *Trust and Privacy in Digital Business*, pages 152–161. Springer, 2004.

[All91] P.G. Allen. A comparison of non-interference and non-deducibility using CSP. In *Proceedings of CSFW*, pages 43–54, 1991.

[AMNO07] R. Anderson, T. Moore, S. Nagaraja, and A. Ozment. Incentives and information security. In *Algorithmic Game Theory*. 2007.

[ARR$^+$10a] Roberto AraÃžjo, Narjes Rajeb, Riadh Robbana, Jacques TraorÃľ, and Souheib Youssfi. Towards practical and secure coercion-resistant electronic elections. In Swee-Huay Heng, RebeccaN. Wright, and Bok-Min Goi, editors, *Cryptology and Network Security*, volume 6467 of *Lecture Notes in Computer Science*, pages 278–297. Springer Berlin Heidelberg, 2010.

[ARR$^+$10b] Roberto Araújo, Narjes Ben Rajeb, Riadh Robbana, Jacques Traoré, and Souheib Youssfi. Towards practical and secure coercion-resistant electronic elections. In *Cryptology and Network Security*, pages 278–297. Springer, 2010.

[B6̈2] J.R. Büchi. On a decision method in restricted second order arithmetic. In *Logic, Methodology and Philosophy of Science. Proc. 1960 Intern. Congr.*, pages 1–11. Stanford University Press, 1962.

[BHM08] Michael Backes, Catalin Hritcu, and Matteo Maffei. Automated verification of remote electronic voting protocols in the applied pi-calculus. In *Computer Security Foundations Symposium, 2008. CSF'08. IEEE 21st*, pages 195–209. IEEE, 2008.

[BKMR05] Jeremy W Bryans, Maciej Koutny, Laurent Mazaré, and Peter YA Ryan. Opacity generalised to transition systems. In *Formal Aspects in Security and Trust*, pages 81–95. Springer, 2005.

[BKR05] Jeremy W Bryans, Maciej Koutny, and Peter YA Ryan. Modelling opacity using petri nets. *Electronic Notes in Theoretical Computer Science*, 121:101–115, 2005.

[BM07] Ahto Buldas and Triinu Mägi. Practical security analysis of e-voting systems. In *Proceedings of IWSEC*, volume 4752 of *Lecture Notes in Computer Science*, pages 320–335. Springer, 2007.

[BP88] N. Belnap and M. Perloff. Seeing to it that: A canonical form for agentives. *Theoria*, 54(3):175–199, 1988.

[BP03] Michael Backes and Birgit Pfitzmann. Intransitive non-interference for cryptographic purposes. In *Proceedings of S&P*, pages 140–152. IEEE, 2003.

[BRS07] Anguraj Baskar, Ramaswamy Ramanujam, and SP Suresh. Knowledge-based modelling of voting protocols. In *Proceedings of the 11th conference on Theoretical aspects of rationality and knowledge*, pages 62–71. ACM, 2007.

[BT94] Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 544–553. ACM, 1994.

[Cha81] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24:84âĂŞ–90, 1981.

[CHP07] K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. In *Proceedings of CONCUR*, pages 59–73, 2007.

[CRS05] David Chaum, Peter Y. A. Ryan, and Steve A. Schneider. A practical voter-verifiable election scheme. In *Proceedings of ESORICS*, pages 118–139, 2005.

[Dim14] A.S. Dimovski. Ensuring secure non-interference of programs by game semantics. In *Security and Trust Management*, pages 81–96. Springer, 2014.

[DKR05] Stéphanie Delaune, Steve Kremer, and Mark D Ryan. Receipt-freeness: Formal definition and fault attacks. In *Proceedings of the Workshop Frontiers in Electronic Elections (FEE 2005), Milan, Italy*. Citeseer, 2005.

[DKR06] S. Delaune, S. Kremer, and M. Ryan. Coercion-resistance and receipt-freeness in electronic voting. In *Computer Security Foundations Workshop, 2006. 19th IEEE*, pages 12–pp. IEEE, 2006.

[DKR10] Stéphanie Delaune, Steve Kremer, and Mark Ryan. Verifying privacy-type properties of electronic voting protocols: A taster. In *Towards Trustworthy Elections*, pages 289–309. Springer, 2010.

[DLL12] Jannik Dreier, Pascal Lafourcade, and Yassine Lakhnech. A formal taxonomy of privacy in voting protocols. In *Communications (ICC), 2012 IEEE International Conference on*, pages 6710–6715. IEEE, 2012.

[DR07] Y. Dodis and T. Rabin. Cryptography and game theory. In *Algorithmic Game Theory*, chapter 8, pages 181–208. 2007.

[Eme90] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier Science Publishers, 1990.

[EvdMZ12] Kai Engelhardt, Ron van der Meyden, and Chenyi Zhang. Intransitive noninterference in nondeterministic systems. In *Proceedings of CCS*, pages 869–880. ACM, 2012.

[FHMV95] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[FMV12] G. Perelli F. Mogavero, A. Murano and M.Y. Vardi. What makes ATL* decidable? a decidable fragment of strategy logic. In *Proceedings of CONCUR*, pages 193–208, 2012.

[FOO92] A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In *Proceedings of AUSCRYPT*, pages 244–âĂŞ251, 1992.

[FPM$^+$14] A. Fielder, E. Panaousis, P. Malacaria, C. Hankin, and F. Smeraldi. Game theory meets information security management. *IFIP Advances in Information and Communication Technology*, 428:15–29, 2014.

[GCC08] J. Grossklags, N. Christin, and J. Chuang. Secure or insure? A game-theoretic analysis of information security games. In *Proceedings of WWW*, pages 209–218. ACM, 2008.

[GGR09] Ryan W. Gardner, Sujata Garera, and Aviel D. Rubin. Coercion resistant end-to-end voting. In Roger Dingledine and Philippe Golle, editors, *Financial Cryptography and Data Security*, volume 5628 of *Lecture Notes in Computer Science*, pages 344–361. Springer Berlin Heidelberg, 2009.

[GI90] James W Gray III. Probabilistic interference. In *Proceedings of S&P*, pages 170–179. IEEE, 1990.

[GM82] Joseph A Goguen and José Meseguer. Security policies and security models. In *Proceedings of S&P*, pages 11–20. IEEE Computer Society, 1982.

[GM84] Joseph A Goguen and José Meseguer. Unwinding and inference control. In *IEEE Symposium on Security and Privacy*, pages 75–75. IEEE Computer Society, 1984.

[GM04] R. Giacobazzi and I. Mastroeni. Abstract non-interference: parameterizing non-interference by abstract interpretation. In *Proceedings of POPL*, pages 186–197. ACM, 2004.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play ANY mental game. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing STOC '87*, pages 218–229. ACM, 1987.

[HJR$^+$13] W.R. Harris, S. Jha, T.W. Reps, J. Anderson, and R.N.M. Watson. Declarative, temporal, and practical programming with capabilities. In *Proceedings of SP*, pages 18–32. IEEE Computer Society, 2013.

[HNS02] C. Hankin, R. Nagarajan, and P. Sampath. Flow analysis: Games and nets. In *The Essence of Computation, Complexity, Analysis, Transformation. Essays Dedicated to Neil D. Jones [on occasion of his 60th birthday]*, volume 2566 of *Lecture Notes in Computer Science*, pages 135–156. Springer, 2002.

[How66] R. A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, pages 22–26, 1966.

[HS72] John C Harsanyi and Reinhard Selten. A generalized nash solution for two-person bargaining games with incomplete information. *Management Science*, 18(5-part-2):80–106, 1972.

[HS12] J. Heather and S. Schneider. A formal framework for modelling coercion resistance and receipt freeness. *FM 2012: Formal Methods*, pages 217–231, 2012.

[HT06] A. Herzig and N. Troquard. Knowing how to play: Uniform choices in logics of agency. In *Proceedings of AAMAS'06*, pages 209–216, 2006.

[JÅ07] W. Jamroga and T. Ågotnes. Constructive knowledge: What agents can achieve under incomplete information. *Journal of Applied Non-Classical Logics*, 17(4):423–475, 2007.

[JCJ05] A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 61–70. ACM, 2005.

[JP06] Hugo L Jonker and Wolter Pieters. Receipt-freeness as a special case of anonymity in epistemic logic. 2006.

[JT15] W. Jamroga and M. Tabatabaei. Strategic noninterference. In *Proceedings of the 30th International Conference on ICT Systems Security and Privacy Protection IFIP SEC 2015*, volume 455 of *IFIP Advances in Information and Communication Technology*, pages 67–81. Springer, 2015.

[JvdH04] W. Jamroga and W. van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63(2–3):185–219, 2004.

[KH04] Wei-Chi Ku and Chun-Ming Ho. An e-voting scheme against bribe and coercion. In *e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004 IEEE International Conference on*, pages 113–116. IEEE, 2004.

[KT09] Ralf Kusters and Tomasz Truderung. An epistemic approach to coercion-resistance for electronic voting protocols. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 251–266. IEEE, 2009.

[KTV10] R. Küsters, T. Truderung, and A. Vogt. A game-based definition of coercion-resistance and its applications. In *Proceedings of the 2010 23rd IEEE Computer Security Foundations Symposium*, pages 122–136. IEEE Computer Society, 2010.

[KVW00] O. Kupferman, M.Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, 2000.

[KYK$^+$11] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artificial Intelligence Research*, 41:297–327, 2011.

[Lam73] B.W. Lampson. A note on the confinement problem. *Communications of the ACM*, 16(10):613–615, 1973.

[LBD$^+$04] Byoungcheon Lee, Colin Boyd, Ed Dawson, Kwangjo Kim, Jeongmo Yang, and Seungjae Yoo. Providing receipt-freeness in mixnet-based voting protocols. In *Information Security and Cryptology-ICISC 2003*, pages 245–258. Springer, 2004.

[LBS08] K. Leyton-Brown and Y. Shoham. *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*. Morgan & Claypool, 2008.

[Lev08] J. Levin. In what city did you honeymoon? and other monstrously stupid bank security questions. *Slate*, 2008.

[LK03] Byoungcheon Lee and Kwangjo Kim. Receipt-free electronic voting scheme with a tamper-resistant randomizer. In *Information Security and CryptologyâĂŤICISC 2002*, pages 389–406. Springer, 2003.

[LZ05] Peng Li and Steve Zdancewic. Downgrading policies and relaxed noninterference. In *ACM SIGPLAN Notices*, volume 40, pages 158–170. ACM, 2005.

[MA11] T. Moore and R. Anderson. Economics and internet security: a survey of recent analytical, empirical and behavioral research. Technical Report TR-03-11, Computer Science Group, Harvard University, 2011.

[MBC01] Emmanouil Magkos, Mike Burmester, and Vassilis Chrissikopoulos. Receipt-freeness in large-scale elections without untappable channels. In *Towards The E-Society*, pages 683–693. Springer, 2001.

[McC88] Daryl McCullough. Noninterference and the composability of security properties. In *Proceedings of S&P*, pages 177–186. IEEE, 1988.

[McN66] R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9:521âĂŞ–530, 1966.

[Men09] Bo Meng. A critical review of receipt-freeness and coercion-resistance. *Information Technology Journal*, 8(7):934–964, 2009.

[MH96] Markus Michels and Patrick Horster. Some remarks on a receipt-free and universally verifiable mix-type voting scheme. In *Advances in CryptologyâĂŤASIACRYPT'96*, pages 125–132. Springer, 1996.

[MH99] P. Malacaria and C. Hankin. Non-deterministic games and program analysis: An application to security. In *Proceedings of LICS*, pages 443–452. IEEE Computer Society, 1999.

[MM03] Annabelle McIver and Carroll Morgan. A probabilistic approach to information hiding. *Programming Methodology*, pages 441–460, 2003.

[MMPV14] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y Vardi. Reasoning about strategies: On the model-checking problem. *ACM Transactions on Computational Logic (TOCL)*, 15(4):34, 2014.

[MMV10] F. Mogavero, A. Murano, and M.Y. Vardi. Reasoning about strategies. In *Proceedings of FSTTCS*, pages 133–144, 2010.

[MN06] Tal Moran and Moni Naor. Receipt-free universally-verifiable voting with everlasting privacy. In *Advances in Cryptology-CRYPTO 2006*, pages 373–392. Springer, 2006.

[O'H90] Colin O'Halloran. A calculus of information flow. In *Proceedings of ESORICS*, pages 147–159, 1990.

[Oka98] Tatsuaki Okamoto. Receipt-free electronic voting schemes for large scale elections. In *Security Protocols*, pages 25–35. Springer, 1998.

[PHW04] Alessandra Di Pierro, Chris Hankin, and Herbert Wiklicky. Approximate non-interference. *Journal of Computer Security*, 12(1):37–81, 2004.

[PR06] Thea Peacock and PYA Ryan. Coercion-resistance as opacity in voting systems. *TECHNICAL REPORT SERIES-UNIVERSITY OF NEWCASTLE UPON TYNE COMPUTING SCIENCE*, 959, 2006.

[RG99] A.W. Roscoe and M.H. Goldsmith. What is intransitive noninterference? In *Proceedings of CSF*, pages 228–228. IEEE, 1999.

[RHB97] A. W. Roscoe, C. A. R. Hoare, and R. Bird. *The Theory and Practice of Concurrency*. Prentice Hall PTR, 1997.

[Rob65] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.

[Ros95] A.W. Roscoe. CSP and determinism in security modelling. In *Proceedings of S&P*, pages 114–127. IEEE, 1995.

[RS01] Peter YA Ryan and Steve A Schneider. Process algebra and non-interference. *Journal of Computer Security*, 9(1):75–103, 2001.

[Rus92] John Rushby. *Noninterference, transitivity, and channel-control security policies*. SRI International, Computer Science Laboratory, 1992.

[RWW94] A.W. Roscoe, J.C.P. Woodcock, and L. Wulf. Non-interference through determinism. In *Proceedings of ESORICS*, pages 31–53. Springer, 1994.

[Rya10] Peter Y. A. Ryan. The computer ate my vote. In *Formal Methods: State of the Art and New Directions*, pages 147–184. Springer, 2010.

[Sch03] Ph. Schnoebelen. The complexity of temporal model checking. In *Advances in Modal Logics, Proceedings of AiML 2002*. World Scientific, 2003.

[Sch04] P. Y. Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2):82–93, 2004.

[Sch10] Henning Schnoor. Strategic planning for probabilistic games with incomplete information. In *Proceedings of AAMAS'10*, pages 1057–1064, 2010.

[SHKS12] Michael Schlapfer, Rolf Haenni, Reto Koenig, and Oliver Spycher. Efficient vote authorization in coercion-resistant internet voting. In *E-Voting and Identity: Third International Conference, VoteID 2011, Tallinn, Estonia, September 28-20, 2011, Revised Selected Papers*, volume 7187, page 71. Springer, 2012.

[Smi09] Geoffrey Smith. On the foundations of quantitative information flow. In *Foundations of Software Science and Computational Structures*, pages 288–302. Springer, 2009.

[SS05] A. Sabelfeld and D. Sands. Dimensions and principles of declassification. In *Proceedings of CSFW-18*, pages 255–269. IEEE Computer Society, 2005.

[SS09] Fredrik Seehusen and Ketil Stølen. Information flow security, abstraction and composition. *IET Information Security*, 3(1):9–33, 2009.

[Sut86] David Sutherland. A model of information. In *Proc. 9th National Computer Security Conference*, pages 175–183, 1986.

[vdM07] Ron van der Meyden. What, indeed, is intransitive noninterference? In *Proceedings of ESORICS*, pages 235–250. Springer, 2007.

[vdMZ10] R. van der Meyden and C. Zhang. A comparison of semantic models for noninterference. *Theoretical Computer Science*, 411(47):4123–4147, 2010.

[vNM44] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behaviour*. Princeton University Press: Princeton, NJ, 1944.

[WAB07] Stefan G Weber, Roberto Araujo, and Johannes Buchmann. On coercion-resistant electronic elections with linear work. In *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*, pages 908–916. IEEE, 2007.

[WJ90] J.T. Wittbold and D.M. Johnson. Information flow in nondeterministic systems. In *IEEE Symposium on Security and Privacy*, pages 144–144, 1990.

[WN95] G. Winskel and M. Nielsen. Handbook of logic in computer science (vol. 4). chapter Models for Concurrency, pages 1–148. Oxford University Press, 1995.

[YKK$^+$10] Z. Yin, D. Korzhyk, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. Nash in security games: interchangeability, equivalence, and uniqueness. In *Proceedings of AAMAS*, pages 1139–1146. IFAAMAS, 2010.

[Zda04] Steve Zdancewic. Challenges for information-flow security. In *Proceedings of the 1st International Workshop on the Programming Language Interference and Dependence (PLIDâĂŹ04)*, 2004.

[ZL95] Aris Zakinthinos and E Stewart Lee. The composability of non-interference. *Journal of Computer Security*, 3(4):269–281, 1995.

[ZM03] S. Zdancewic and A.C. Myers. Observational determinism for concurrent program security. In *Proceedings of CSFW-16*, pages 29–43. IEEE Computer Society, 2003.