Towards Automated Design of Multi-party Rational Exchange Security Protocols

Almudena Alcaide, Juan M. Estevez-Tapiador, Julio C. Hernandez-Castro, Arturo Ribagorda
Computer Science Department, Carlos III University of Madrid
Avda. Universidad 30, 28911, Leganes, Madrid (Spain)
{aalcaide, jestevez, jcesar, arturo}@inf.uc3m.es

Abstract

It is only recently that rational exchange schemes have been considered as an alternative solution to the exchange problem. A rational exchange protocol cannot provide fairness but it ensures that rational (i.e. self-interested) parties would have no reason to deviate from the protocol, as misbehaving is not beneficial. The common understanding of rationality as a security property has encouraged researchers to look for methods to formally analyze and categorize rational protocols. By contrast, in this paper, we adopt a completely new approach as we present an automated tool to design multi-party rational exchange security protocols. Given a specific set of initial parameters the tool performs an heuristic search in the space of exchanging protocols, producing a rational exchange scheme as a solution. As this is work in progress, we will present the first results obtained executing the application in a three entity environment and a total set of six exchanging items.

1 Introduction

Rational exchange protocols do have the enormous advantage over *fair* exchange schemes in not requiring a trusted third party. Informally, a *rational* exchange protocol cannot provide fairness, but it ensures that rational (i.e. self-interested) parties have no reason to deviate from the protocol as misbehaving does not result beneficial. As it is only recently that rational exchange schemes have been considered as an alternative solution to the exchange problem, there are very few rational exchange protocols proposed in the literature ([3], [6]). Furthermore, the design and analysis of rational exchange security protocols pose a greater challenge when considering a multi-party environment. Some rational multi-party solutions do exist although, rationality is applied to solve other types of problems such as secret sharing and multi-party computation [4].

1.1 Automated analysis and design of exchanging protocols

Regarding the formal analysis of security protocols, several automated tools have been presented over the years, each one implementing a different methodology. These tools have served to formally study classic security requirements such as confidentiality, authentication or integrity. More recently, other methodologies have been considered to evaluate new security properties such as rationality ([1], [2], [3]). However, in every case, the approach has always been on the analysis of existing schemes. By adopting a different approach we intend to automatically and formally design protocols ensuring such desired properties. Our work is novel and has focused on the automated design of multi-party *rational* exchange security protocols (m-RES protocols).

1.2 Overview of our work

The automated design of m-RES protocols consists of an application designed to produce rational exchange protocols for a particular environment specified by a series of formatted parameters. Although the program flow and the data structures being used are highly flexible in terms of number of entities and type of exchanging items, this particular work is based on exploring the space of 3-party exchanging protocols in search of rational solutions (3-RES protocols). The resulting solution/s will ensure protocol participants a minimum set of requirements expressed by the appropriate utility values, as well as ensuring rationality.

The program uses heuristic technique Simulating Annealing to search for the best possible solution/s.

The rest of the paper is organized as follows. Section 2 gives a brief description of the initial assumptions on the exchanging scenario. In Section 3, we describe the main aspects of the implementation regarding data structures and program flow. In Section 4, we show the results obtained when executing the application with the given parameters. Finally, in Section 5, we summarize the paper by presenting

the main conclusions and some future research lines.

2 m-RES protocols

A multi-party cryptographic rational exchange protocol is a cryptographic protocol allowing several parties to exchange commodities in such a way that, if one or more parties deviate from the protocol description, then they may bring other participants to a disadvantageous situation, but they cannot gain any advantage by doing so.

In this paper we will be focusing on a particular multi-party exchanging scenario where an entity E_0 aims to collect a series of items from other participant entities $E_1, .., E_n$, delivering the appropriate tokens in return.

2.1 3-RES protocol initial assumptions

For the purpose of this first study we have focused on a 3-entity scenario (entities E_0, E_1 and E_2) for which we are giving a series of initial assumptions. These will be passed to the application appropriately formatted. The application will then resolve the problem by designing a *rational* scheme in which entities will see all their requirements fulfilled. The initial assumptions are:

- All entities are considered to be rational (self-interested) and aimed at maximizing their own utility functions.
- An entity E_0 aims to collect a series of electronic items from entities E_1 and E_2, delivering the appropriate tokens in return.
- No item in isolation is of any value to entity E_0. In other words, E_0 is interested in collecting all or none of these items.
- Additionally, the nature of these items is such that, their utilities, only become available when the corresponding tokens are delivered in return. Although this restriction seems hard and unrealistic, there are few real life examples where items are of this nature. For example, E_0 could be a user trying to book a holiday package consisting of accommodation, flights and tickets for a local tourist attraction. User E_0 needs either all or none of the required items, additionally, no item becomes available unless the providers of the required services have received payment.
- Participant entities E_1 and E_2, are part of a visible and recognizable PKI (Public Key Infrastructure). By contrast, this is not a restriction on entity E_0 who can maintain anonymous his/her real identity. No other trusted or semi-trusted parties need to be involved in the scheme.
- All messages sent by E_1 and E_2 must be signed with their corresponding private keys and all messages received by these entities must be encrypted with their corresponding public keys.

3 Automated design of 3-RES solutions

As mentioned before, each of the initial assumptions is implemented as a parameter within the application. Any other kind of environment can be appropriately formatted and will produce a completely different kind of result.

Next is the description of those aspects of the application which are essential for the understanding of the tool implemented.

3.1 Data representation

The following are the main components of the program:

- A protocol is represented by a matrix where each row is interpreted as a message. The first two components of every message identify the sender and the receiver respectively. The rest represent the tokens being sent as part of the message.
- A total of six tokens are involved in the exchange and they are summarized as follows:

  $desc\_item_1$**:** Request for $item_1$. It contains a description of $item_1$, required from entity E_1.

  $desc\_item_2$**:** Request for $item_2$. It contains a description of $item_2$, required from entity E_2.

  $item_1$**:** Customized item issued by entity E_1 as specified by E_0 in $desc\_item_1$.

  $item_2$**:** Customized item issued by entity E_2 as specified by E_0 in $desc\_item_2$.

  $return\_item_1$**:** Token issued by E_0 for E_1 in return for $item_1$.

  $return\_item_2$**:** Token issued by E_0 for E_2 in return for $item_2$.
- A fitness function is individually defined for each protocol participant. All participants assign every token in the protocol a positive, null or negative value, depending on whether that entity is highly interested in the item, indifferent or uninterested, respectively. The search is aimed at finding exchanging schemes which maximize each individual fitness function.
- A global protocol fitness function is also defined. All protocols evaluated are shortened to the step where the fitness was the highest. Additionally, participants who have reached their maximum utility values are forced to quit the protocol at that step.

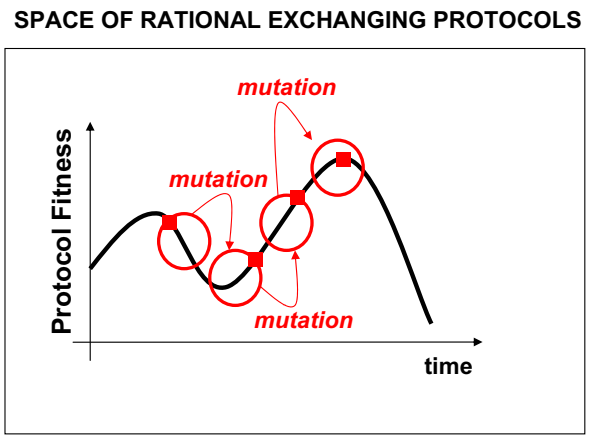

Figure 1. Graphical representation. Phase I of the process (Expansion and Cooperation) is performed in each red circle being able to find the local maximum marked with a square. Phase II consists of consecutively mutating the current local maximum and generating the next individuals.

3.2 Searching technique

As mentioned before the searching algorithm is based on Simulated Annealing. Roughly, this technique requires a first individual (exchange protocol) which is randomly generated and presented for evaluation. Such evaluation consists of computing the fitness values individually obtained by each protocol participant when executing the scheme, as well as the global protocol fitness value. The first randomly generated protocol is then *evolved by mutation* to a different scheme which is also evaluated. If the new protocol reaches a higher level of overall fitness than the original one, then it is *accepted* as a new valid individual from which to generate the next one. If the new protocol represents a solution worse than the previous one, the new scheme could only be *accepted* if a certain dynamically decreasing parameter (temperature) is above a specific value. In other words, better solutions are always accepted and worse solutions are accepted when the temperature is still above a certain threshold. The process is repeated a fixed number of times depending on the *initial temperature* and the *cooling-rate*. Details of the changes applied to protocols to evolve them towards better solutions, are given in the following paragraphs. Besides, the process is graphically illustrated in Figure Fig.1.

Each individual (protocol exchange) is subject to changes in two different phases:

1. **Phase (I): Search of the local maximum.** For any possible solution the program runs a utility in search of the local maximum. In other words, for any given protocol, the program tries to find the best possible amongst similar protocols to that given. This routine is based on two different procedures called *Expansion* and *Cooperation*.

   - Expansion consists of adding new random messages to every shortened solution in search of higher fitness values.
   - Cooperation consists of forcing participants to relay items which have no value to them but which might be valuable to others.

   In the search for local maximum we always choose the best found solution. No "bad movements" are allowed in this phase. The resulting protocol is defined as a local best solution or local maximum.

2. **Phase (II): Mutation.** The local maximum from phase I suffers a transformation consisting of a *Mutation* routine. A mutation is a permutation of the message order plus an expansion of the resulting individual. If the new protocol does not represent a better solution, it will only be accepted if the temperature is high enough to accept "bad movements".

   Phase II is repeated from an initial temperature down to zero (for a given cooling rate) and, for each new individual generated Phase I is executed. At the end of the process the application returns the best found solution.

3.3 Rationality

Two main aspects of the application ensure rational solutions: 1. Entities that have achieved their maximum utility values are forced to quit the protocol. In other words, the application will not consider rational those moves taken by a participant after achieving all their goals. 2. Because all solutions are shortened to where the utility values were maximum, participant entities are rationally forced to follow the protocol description to obtain maximum possible payoff values.

4 Results

The following rational protocol is automatically produced by the tool:

- Entity E_0 sends entity E_1, a message including $desc\_item_1$ and $desc\_item_2$, descriptions of the required items.

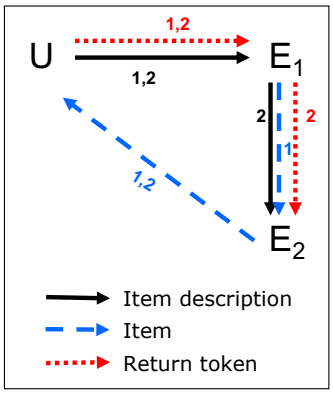

Figure 2. Sketch of a 3-RES protocol. Arrows represent different message content: bold lines represent the requests for items, dashed lines represent tailored $item_i$ destined to E_0 and dotted lines represent return tokens.

- Entity E_1 produces, according to the appropriate description, a customized $item_1$ destined to E_0.
- Entity E_1 sends E_2 a message containing $item_1$ and the description-token $desc\_item_2$.
- Entity E_2 sends E_0 items $item_1$ and $item_2$.
- Participant E_0 sends entity E_1 a message including $return\_item_1$ and $return\_item_2$, the return-tokens for the items received.
- E_1 receives a message with two return-tokens. It takes $return\_item_1$ and sends entity E_2, the token $return\_item_2$.

Figure 2 represents the complete set of messages exchanged to successfully terminate the 3-RES protocol execution.

Note that, from entity E_0's point of view, rationality is derived from the fact that $item_1$ and $item_2$ are of no use until the corresponding return-items $return\_item_1$ and $return\_item_2$ have reached entities E_1 and E_2 respectively. Furthermore, since entity E_0 requires either all or none of the items, entity E_1 is then rationally forced to relay items $desc\_item_2$ and $return\_item_2$ to entity E_2, otherwise, $item_1$ is of no use to E_0 and it will not report benefits. In a similar way, entity E_2 is willing to deliver $item_1$ to entity E_0 as only by doing so, will $item_2$ be of any value to entity E_0.

5 Conclusions

Traditionally, automated tools have always been applied to the analysis of security protocols. In this paper we have adopted a completely new approach, ensuring rationality as part of the automated design of an exchange scheme. Furthermore, complex multi-party scenarios can easily be parameterized to pass them as arguments to the tool.

Summarizing, our automated application takes four main arguments: number of participants, number and type of exchanging items, initial assumptions about the exchanging dynamics and the values each entity assigns to every token involved in the exchange. It then produces a rational scheme. This, we have proven for a particular 3-party scenario. Our future work will be focused on executing and analyzing the results obtained in other exchanging contexts as well as developing a more extensive and detailed experimental project.

References

[1] A. Alcaide, J.M. Estévez-Tapiador, J.C. Hernández Castro, A. Ribagorda. "An Extended Model of Rational Exchange Based on Dynamic Games of Imperfect Information". In *ETRICS'06*, LNCS Vol. 3995/2006, pp. 396-408. June 2066. Springer-Verlag.

[2] A. Alcaide, J.M. Estévez-Tapiador, J.C. Hernández Castro, A. Ribagorda. "Bayesian rational exchange". To appear in *International Journal of Information Security*. Springer-Verlag.

[3] L. Buttyán. "Building Blocks for Secure Services: Authenticated Key Transport and Rational Exchange Protocols". *Ph.D. Thesis*. Laboratory of Computer Communications and Applications, Swiss Federal Institute of Technology. Lausanne.

[4] J. Halpern, V. Teague. "Rational Secret Sharing and Multiparty Computation: Extended Abstract". In *STOC'04*, ACM 1-58113-852-0/04/0006.

[5] H. Pagnia and F.C. Gärtner. "On the impossibility of fair exchange without a trusted third party". Darmstadt University of Technology, Department of Computer Science. Technical Report TUD-BS-1999-02. March 1999.

[6] P. Syverson. "Weakly secret bit commitment: Applications to lotteries and fair exchange". In *Proc. 11th IEEE Computer Security Foundations Workshop*, pp. 2–13, 1998.