بسم الله الرحمن الرحيم

University of Isfahan

Faculty of Engineering

Department of Computer Engineering

M.Sc. Thesis

Automatic Verification of Authentication Protocols using Genetic Programming

Supervisor
Dr. Behrouz Tork Ladani

By

Hasan Taha

September 2010

اصفهان دانشگاه

دانشکده فنی مهندسی

گروه کامپیوتر

عنوان تحت حسن طه آقای هوش مصنوعی کامپیوترگرایش ی رشته ارشد کارشناسی ی نامه پایان

وارسی پروتکلهای احراز اصالت با استفاده از برنامه نویسی ژنتیک

در تاریخ ۲۰۱۱/۰۱/۲۹ توسط هیأت داوران زیر بررسی و با درجه عالی به تصویب نهایی رسید.

۱ -استاد راهنمای پایان نامه دکتر بهروز ترک لادانی با مرتبه ی علمی استادیار امضا

۲ -استاد مشاور پایان نامه با مرتبه ی علمی امضا

۳ -استاد داور داخل گروه دکتر .... با مرتبه ی علمی استادیار امضا

۴ -استاد داور خارج از گروه دکتر .... با مرتبه ی علمی استادیار امضا

امضای مدیر گروه

Dedication

To my father, the origin of me, who helped me whenever I needed, the reason that I have not felt any shortage; spiritual or material.

To Mohammad Ali (Mahdi), who endured me, and accepted, being far from his grandfather and family only for my eyes.

To my dear Hanan, who changed my life in both the presence and absence.

Acknowledgements

I am extremely grateful to Doctor Behrouz Tork Ladani my supervisor for his support, help, advice, perfect ideas, experience, and knowledge; without his support, I would not have been able to complete my research. On the human side, he has taught me how to face unsuccessful experiments and learn from them, and how to evaluate the successful ones.

In Addition, I would like to give special thanks to Doctor Ahmad Reza Naghsh Nilchi, and Doctor Amir Hasan Monadjjemi who helped me in my foreignness; without their assistance, I may not be able to be able to live abroad.

Finally I would like to thank my family who endured and supported me; in special my father and wife.

Abstract

Implicit and unobserved errors and vulnerabilities issues usually arise in cryptographic protocols and especially in authentication protocols. This may enable an attacker to make serious damages to the desired system, such as having the access to or changing secret documents, interfering in bank transactions, having access to users' accounts, or may be having the control all over the system. Many methods have been used to verify the cryptographic protocols such as logical, algebraic, inductive, and mathematical complexity methods. Each of these methods has its special shortages, in addition to the general disadvantages of these methods. Nevertheless, no one has tried to use the computational methods as cryptographic protocol's verifiers. In this thesis, we represent genetic programming – a type of computational methods- as a new verification method for the cryptographic protocols. GAProver; new genetic programming based system has been built and successfully used to verify some known authentication protocols. We wish a new horizon would be opened in the field of cryptographic protocol verification because of this effort. Moreover, we believe that further development of this method may solve many problems that encountered by researchers in the field of protocol verification because of the use of formal methods.

Keywords: Authentication protocols, Authentication protocol verification, Computational methods, Genetic programming.

Table of Contents

Title **page**

Chapter 1 :Introduction

1.1. Authentication Protocols 1
1.2. Verification of Authentication Protocols 2
1.3. Motivation 2
1.4. The structure of the thesis 3

Chapter 2 Verification Methods of Authentication Protocols

2.1. Introduction 5
2.2. Authentication protocols 5
2.2.1. Authentication Building blocks and types 6
2.2.2. Protocol notation 7
2.3. Facts (hypotheses & goals) about authentication protocols 9
2.4. Vulnerability of authentication protocols 10
2.5. Verification methods of Authentication Protocols 11
2.5.1. Why security protocol analysis is hard 11
2.5.2. The decidability problem 12
2.5.3. Formal specification 12
2.5.4. Dolev-Yao model 13
2.5.5. Formal verification 14
2.6. Logical Methods of Protocol Verification 15
2.7. Algebraic Methods 17
2.8. Complexity-Theoretic Proof of Security 19
2.9. Inductive Methods 20
2.10. Key Dates 20
2.11. Some authentication protocol verification tools 21
2.11.1. Murphi 21
2.11.2. NRL Analyzer 22
2.11.3. Protocol Verifier 22

Title **page**

2.11.4. AVISPA ... 22

2.12. How to continue the research ... 23

2.13. Conclusion ... 23

Chapter 3 :Genetic Programming as a type of Evolutionary Computation

2.1. Introduction ... 24

2.2. The Biological Aspect ... 25

2.3. Evolutionary computation methods ... 25

2.3.1. The history ... 25

2.3.2. Search Space ... 26

2.3.3. NP-Complete Problems ... 26

2.3.4. The typical evolutionary algorithm ... 27

2.4. Genetic algorithms ... 27

2.4.1. The general algorithm ... 28

2.4.2. Encoding of a Chromosome ... 28

2.4.3. Crossover ... 29

2.4.4. Mutation ... 29

2.5. Genetic Programming ... 29

2.5.1. Preparatory steps of genetic programming ... 30

2.5.2. Executional steps of genetic programming ... 31

2.5.3. Example of a run of genetic programming ... 36

2.6. Conclusion ... 40

Chapter 4: A Model for Verification of Authentication Protocols using Genetic Programming

4.1. Introduction ... 41

4.2. Modeling the protocol in a hostile environment ... 41

4.3. Analyzing the role of the intruder ... 44

4.3.1. Basic capabilities of an intruder ... 44

4.3.2. The intruder strategy ... 46

4.4. Protocol verification as a planning problem ... 48

Title **page**

4.4.1. What is planning 48
4.4.2. Why Planning 49
4.4.3. The GAProver as a planning problem 50
4.4.4. Using search in our planning problem 52
4.5. Architecture of the GAProver 52
4.6. A Genetic programming view of Gap-Gift Model 53
4.6.1. Modeling the set of terminals 53
4.6.2. Modeling the set of primitive functions 57
4.6.3. Representation of the Genetic Protocol 64
4.6.4. The Fitness Function 67
4.6.5. Certain Parameters for controlling the run 70
4.6.6. Extraction the intrusion scenario 71
4.7. The complete Algorithm of the GAProver 72
4.8. Conclusion 76
Chapter 5: Implementation and Experimental Results
5.1. Introduction 78
5.2. JGAP tool 78
5.2.1. JGAP's overview 79
5.2.2. How to code in JGAP 80
5.3. The GAProver 84
5.3.1. How to use GAProver 84
5.3.2. The configuration file 84
5.4. Experimental results 85
5.4.1. The simple protocol 85
5.4.2. Encrypted Key Exchange (EKE) protocol 89
5.4.3. BAN-Andrew protocol 96
5.5. Conclusion 100
Chapter 6: Conclusion and future work
6.1. Introduction 101

Title **page**

6.2. Discussed topics and ideas 101

6.3. Future work 103

Bibliography 105

Table of Figures

Title **page**

Figure 2-1: Simple two-way authentication protocol. 8
Figure 2-2: Another way to represent a protocol. 8
Figure 2-3: An intrusion for the simple authentication protocol 11
Figure 3-1: Example of a search space. 26
Figure 3-2: Typical flowchart for an evolutionary algorithm, (Fogel, 1966). 27
Figure 3-3: A prototypical genetic algorithm. 28
Figure 3-4: Main loop of genetic programming..30
Figure 3-5: Flowchart of genetic programming. 32
Figure 3-6: Creation of a seven-point tree using the "Full" initialization method (t=time)..33
Figure 3-7: Creation of a five-point tree using the "Grow" initialization method (t=time)..34
Figure 3-8: Pseudo code for recursive program generation with the "Full" and "Grow" methods...34
Figure 3-9: Initial population of four randomly created individuals of generation 0. 38
Figure 3-10: The fitness of each of the four randomly created individuals of generation 0 is equal to the area between two curves. 38
Figure 3-11: Population of generation 1 (after one reproduction, one mutation, and one two-offspring crossover operation...38
Figure 4-1 Standard Search system..50
Figure 4-2: The main units of the GAProver (The First Level ERD).................52
Figure 4-3: Simple information with an encryption function root.......................55
Figure 4-4: An example of the refined style of information. 55
Figure 4-5: Refined Information Extraction algorithm. 56
Figure 4-6: Standard Encrypt function...58
Figure 4-7: An example of Encrypt function, and the new generated information 59
Figure 4-8: Recursive algorithm checks if a principal has knowledge. 59
Figure 4-9: Information validator algorithm. 60
Figure 4-10: The prototype of the protocol cromosome validator.......................62
Figure 4-11: The prototype of message analyzing algorithm. 63
Figure 4-12: Algorithm of decryption ability of a principal. 64
Figure 4-13: The prototype of the representProtocol algorithm. 65

Title **page**

Figure 4-14: An example of a step structure. 66
Figure 4-15: Selkow's tree similarity algorithm. 68
Figure 4-16: The prototype of the GAProver fitness function 69
Figure 4-17: Find Intrusion Path Algorithm. 71
Figure 4-18: The prototype of the protocol intrusion path extractor. 72
Figure 4-19: Prototypical genetic programming based GAProver. 74
Figure 4-20: Multi Genetic engine GAProver. 75
Figure 5-1: Chromosome representation as a tree in JGAP. 79
Figure 5-2: Sample of code for building GP configuration 80
Figure 5-3: Sample code of initial genotype creation. 82
Figure 5-4: Sample code of custom function creation. 83
Figure 5-5: The GAProver configuration file. 84
Figure 5-7: The configuration file for the verification of the simple protocol 86
Figure 5-6: Simple two-way authentication protocol 86
Figure 5-8: An intrusion for the simple authentication protocol found by GAProver. 87
Figure 5-9: Graphical Intrusion scenario for simple protocol generated using jgraphx package. 88
Figure 5-10: The Conversion of the GAProver to reach an attack for the simple protocol 89
Figure 5-11: Encrypted Key Exchange protocol 90
Figure 5-12: The configuration file for the verification of the EKE protocol. 91
Figure 5-13: An intrusion for EKE protocol found by GAProver. 92
Figure 5-15: Graphical Intrusion scenario for EKE protocol generated using jgraphx package (first subgoal). 93
Figure 5-14: The standard format of the intrusion of EKE protocol that found by the system. 93
Figure 5-16: Graphical Intrusion scenario for EKE protocol generated using jgraphx package (second subgoal). 94
Figure 5-17: The Conversion of the GAProver to reach an attack for the first subgoal of EKE Protocol. 95
Figure 5-18: The Conversion of the GAProver to reach an attack for the second subgoal of EKE Protocol. 96
Figure 5-20: The configuration file for the verification of the BAN-Andrew protocol 97
Figure 5-19: BAN-Andrew Protocol. 97
Figure 5-21: An intrusion for BAN-Andrew protocol found by GAProver. 98

Title **page**

Figure 5-22: The standard format of the intrusion of EKE protocol that found by the system. 99
Figure 5-23: The Conversion of the GAProver to reach an attack for the goal of BAN-Andrew Protocol...100
Figure 6-1: One suggested way to represent a full protocol run. 103

List of Tables

Title **page**

Table 2-1: Notation of the authentication protocols. 7
Table 2-2: The syntax of BAN logic.. 15
Table 2-3: The key dates of cryptographic protocol verification. 20

Abbreviations

Abbreviation	Meaning
IT	**I**nformation **T**echnology
EA	**E**volutionary **A**lgorithms
GA	**G**enetic **A**lgorithms
GP	**G**enetic **P**rogramming
DES	**D**ata **E**ncryption **S**tandard
AES	**A**dvanced **E**ncryption **S**tandard
RSA	**R**ivest, **S**hamir, **A**dleman
ACP	**A**lgebra of **C**ommunicating **P**rocesses
GAProver	**G**enetic Programming **A**uthentication **Pro**tocol **Ver**ification System
NP	**N**ondeterministic **P**olynomial
EKE	**E**ncrypted **K**ey **E**xchange

Chapter 1
Introduction

1.1. Authentication Protocols

The revolution in Information Technology (IT) field in the last three decades has allowed people to engage many new activities in their daily life such as the Internet, wireless communication, and e-commerce. This revolution led to access very critical information, and this access created many advantages and many serious risks. Many of these risks appear because of security breaches, and the cost is often very expensive. One of the main problems is how to authenticate identity of the people that a service is dealing with in a secure way. This problem appears because many of protocols that were thought to be safe for long periods, has been revealed to be unsafe and vulnerable for serious dangers.

Network based authentication mechanism requires a principal to authenticate to a single system either local or remote (BISHOP, 2003). Authentication is used to verify the identity of users in order to control access to resources, to prevent unauthorized users from gaining access to the system and to record the activities of the users in order to hold them accountable for their activities (Matthew, 2002). This is the reason why external entity must provide information to enable the system to confirm its identity. Authentication process consists of obtaining the authentication information from an entity, analyzing the data and determines if it is associated with that entity.

The information comes from what the entity knows (Password, secret information), what the entity has (Badge or card), where the entity is (Terminal), and what the entity is (Fingerprint, odour, retina, hand geometry).

1.2. Verification of Authentication Protocols

In cryptographic protocols and especially authentication protocols there is sometimes an opportunity to find some unobserved errors or vulnerabilities. This may enables an attacker to make serious damages to the desired system, such as having the access to or changing secret documents, interfering in bank transactions, having access to users' accounts, or may be having the control all over the system (Whalen, et al., 2005).

A window for formal verification of the protocols has been opened by Needham and Schroeder by reporting that some errors in the security protocols are hard to discover manually (Needham, et al., 1978). After some years, Denning and Sacco have succeeded in building the first formal protocol verifier that discovered an intrusion to the Needham-Schroeder authentication protocol (Denning, et al., 1981). After that, many methods have been applied to verify the authentication protocols such as logical methods, algebraic methods, and inductive methods.

Logical methods of protocol verification have been started by Burrows, Abadi, and Needham (BAN) (Burrows, et al., 1989, 1990). In the same time, Milner opened the way to use the algebra as a formal method of protocol verification by the extension of the capability of process algebra and developing Pi calculus(Milner, 1989, 1990). In 1998, Abadi and Gordon continued Milner's way and added Spi calculus, which had the structures of cryptography(Abadi, et al., 1998). In 1998, Paulson started the use of induction as a technique for the verification of cryptographic protocols was pioneered by using the proof tool called Isabelle, which uses Higher Order Logic (HOL) (Paulson, 1998). Since that time, there are no efforts to solve the problem of verification of protocols in new methods. However, all that has been achieved is to extend these methods and increase their capacity to include new protocols.

1.3. Motivation

In the previous section, a short description for the methods of authentication protocol verification has been mentioned. The common feature of these methods is that nearly all of them are based on symbolic intelligence. As will be discussed later all of these methods has their limitations such as validating the vulnerable protocols to be secure, and proving that formal methods has always have shortcomings in the verification of authentication protocols. These limitations have encouraged us to try the computational methods for the purpose of authentication protocol verification.

In this research, we present a new method for analysis and automated detection of attacks on protocols. Our objective is to identify the possible executions making an attack to a protocol (if there is any). For this purpose, we applied genetic programming, a systematic method to search the space of all possible protocol executions, to find the possible attack procedures. We apply the proposed method on some two-party shared-key authentication protocols such as a simple challenge-response authentication protocol, Encrypted Key Exchange (EKE) protocol, and BAN-Andrew protocol and show that genetic programming could find the attack scenario to this protocol by running in an acceptable time.

For this purpose, first we define a suitable model for our problem and then specify the genetic programming parameters. Then we use this technique to find the possible sequence of steps that an intruder may do to impersonate another agent and so could attack to the protocol.

Briefly, all that we have done is to use a new paradigm to verify the authentication protocols; this method may open a new horizon in the field of protocol verification, and we believe that further development of this method may solve many problems that encountered by researchers in the field of protocol verification because of the use of formal methods.

1.4. The structure of the thesis

In this chapter and in the rest of this thesis we will refer to the system that we have developed with GAProver (pronounced as Gap Prover), which is derived from **G**enetic-Programming **A**uthentication-**Pro**tocol **Ver**ification.

In the rest of this thesis, in chapter 2, we will mention the concepts, general hypothesis and goals of authentication protocols, speak about their formal verification methods, and introduce some verification tools based on some of these methods. Firstly, we will light a spot on the general properties of the formal methods. Then we will discuss the formal verification methods including logical methods, algebraic methods, complexity theoretic methods, and the inductive methods. After that, we will give a table of brief history for all of the formal methods. At last, some authentication protocol verification tools based on some of the mentioned methods will be introduced. In the discussion of this chapter, we will give the limitations of the current methods that motivate us to try a different paradigm for the protocol verification.

In chapter 3, we will discuss the evolutionary algorithms as a kind of computational intelligence; the overall procedure of evolutionary algorithms will be discussed. Then we will discuss genetic algorithms as a kind of evolutionary algorithms, in this section we will discuss the genetic algorithms' prototype in detail because it will help us to understand how genetic programming works. Then we will discuss the genetic programming as an extension of genetic algorithms. In this chapter we also will give an example to understand how genetic programming works.

Chapter 4 is the heart of this thesis. In this chapter, first, we will explain the work of (Tork Ladani, 2005) which defines a framework for modeling and verification of authentication protocols using the intruder Gap-Gift strategy, this model will be the base to develop our authentication protocol verification system (GAProver) based on genetic programming. Then, we will explain how to represent the GAProver as planning problem, then; we will explain how to utilize researching in our planning problem to reach a hybrid system that has an informed search feature. After that, main units of GAProver will be presented with the basic functions of each unit. Then, a genetic programming view of the Gap-Gift model will be discussed in detail. Lastly, we will introduce the complete GAProver system based on genetic programming. We will introduce two possible structures for the GAProver; the first one will be a mono-engine genetic programming system, while the second is a multi-engine genetic programming system.

Chapter 5 is the chapter of tests and experimental results. First, we will introduce the *JGAP*, a java based toolbox for genetic programming, which is the used to implement the GAProver. Then, we will give an overview of how to use GAProver. After that, we will discuss the results of the system tests on three protocols including, a simple challenge-response authentication protocol, Encrypted Key Exchange (EKE) protocol, and BAN-Andrew protocol. Lastly, a comparison between the GAProver and two other systems will be done.

In chapter 6, the thesis will be concluded. After that, some ideas will be given about how to continue this work and to improve the GAProver and genetic programming in whole.