IN THE NAME OF GOD

SIMULATION OF RSA & ELGAMAL PUBLIC KEY CRYPTOSYSTEMS

BY
MAHNAZ MOHAMMADI

THESIS

SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE (M.Sc.)

IN
ELECTRICAL ENGINEERING (COMMUNICATION)
SHIRAZ UNIVERSITY
SHIRAZ, IRAN

EVALUATED AND APPROVED BY THE THESIS COMMITTEE AS: VERY GOOD

..........A. Zolghadr Asli, Ph.D., Assistant Prof. of Electrical Engineering (Chairman)

..........A.Sheikhi Ph.D., Assistant Prof. of Electrical Engineering

..........Sh. Golbahar Haghighi Ph.D., Assistant Prof. of Electrical Engineering

OCTOBER 2001

To my parents

Acknowledgement

I would like to express my deepest appreciation for the help, guidance and concern of my adviser Dr. A. Zolghadr Asli throughout this work.

I am truly grateful to the member of my advisory committee Dr. A. Sheikhi and Dr. Sh. Golbahar Haghighi for their valuable suggestions, criticism and advice in developing this work.

I am partially grateful to Dr. E. Farjah the Chairman of Electrical Engineering Department for his cooperation and helps during my graduate work.

Abstract

Simulation of RSA & ELGAMAL Public Key Cryptosystems

By

Mahnaz Mohammadi

Two types of public-key cryptosystems in key generation , encryption and decryption schemes are considered and implemented in this project: *RSA public-key cryptosystem* (based on factoring a large integer) and *ElGamal public-key cryptosystem* (based on discrete logarithm modulo a large prime).

Since both systems require computations in algebraic structure *Zn* (the integers modulo *n*) where *n* is a large positive integer (may or may not be a prime) and none of the available hardware supports calculation in this range, so to carry out the computation efficiently, arithmetic operations are simulated in the project using some mathematical algorithms. At the end these two systems are compared to each other from different parameters points of view such as performance, security and applications. To have a good comparison and also to have a good level of security correspond to users need the systems are designed flexibly in terms of the key size.

Table of Contents

Content page

List of Tables

List of Figures

Chapter 1

Introduction

Cryptography deals with the transformation of ordinary text (plaintext) into coded form (ciphertext) by encryption, and transformation of ciphertext into plaintext by decryption. Normally these transformations are parameterized by one or more keys. The motive for encrypting text is security for transmissions over insecure channels.

Three of the most important services provided by cryptosystems are secrecy, authenticity, and integrity. Secrecy refers to denial of access to information by unauthorized individuals. Authenticity refers to validating the source of a message; i.e., that it was transmitted by a properly identified sender and is not a replay of a previously transmitted message. Integrity refers to assurance that a message was not modified accidentally or deliberately in transit, by replacement, insertion or deletion. A fourth service which may be provided is nonrepudiation of origin, i.e., protection against a sender of a message later denying transmission. Variants of these services, and other services, are discussed in [23].

Classical cryptography deals mainly with the secrecy aspect. It also treats keys as secret. In the past 15 years two new trends have emerged:

a. Authenticity as a consideration which rivals and sometimes exceeds secrecy in importance.

b. The notion that some key material need not be secret.

The first trend has arisen in connection with applications such as electronic mail systems and electronic funds transfers. In such settings an electronic equivalent of the handwritten signature may be desirable. Also, intruders into a system often gain entry by masquerading as legitimate users; cryptography presents an alternative to password systems for access control.

The second trend addresses the difficulties which have traditionally accompanied the management of secret keys. This may entail the use of couriers or other costly, inefficient and not really secure methods. In contrast, if keys are public the task of key management may be substantially simplified.

An ideal system might solve all of these problems concurrently, i.e., using public keys; providing secrecy; and providing authenticity. Unfortunately no single technique proposed to date has met all three criteria. Conventional systems such as DES (see sec. 2.5) require management of secret keys; systems using public key components may provide authenticity but are inefficient for bulk encryption of data due to low bandwidths.

Fortunately, conventional and public-key systems are not mutually exclusive; in fact they can complement each other. Public- key systems can be used for signatures and also for the distribution of keys used in systems such as DES . Thus it is possible to construct hybrids of conventional and public-key systems which can meet all of the above goals: secrecy, authenticity and ease of key management.

The concept of public-key cryptography was invented by *Whitfield Diffie* and *Martin Hellman* in 1976[11], since that time numerous public-key algorithms have been proposed but many of them are insecure and of those still considered secure, many are impractical, either they have too large keys or the ciphertext they produce is much longer than the plaintext (original information) which is a big disadvantage. Of the secure and practical algorithms some are only suitable for key distribution or digital signatures and only few of them work well for both encryption and digital signing.

A typical class of these techniques is *RSA-Rabin*, which is the combination of the polynomial time algorithm of finding a root of a polynomial over a finite field and the intractability of factoring problem. This class includes *RSA* [50], *Rabin* [46], *Williams*[75], *Kurosawa-Itoh-Takeuchi*[29], *Cubic RSA* [45].

Another typical class of techniques is *Diffie-Hellman-ElGamal*, which is the combination of the commutative property of the logarithm in a finite Abelian group and the intractability of the discrete logarithm problem. This class includes *Diffie-Hellman* [11], *ElGamal* [15] and *elliptic curve versions of the Diffie-Hellman* and *ElGamal* [27].

The idea behind all public-key algorithms is the same, these algorithms are generally based on one of the *NP-hard* problems (see sec. 4.1.2) . This project focuses on efficient implementation and analysis of two most popular of these algorithms, *RSA* and *ElGamal* just for key generation and the encryption scheme (encryption / decryption operation). *RSA* relies on difficulty of prime factorization of a very large number, and the hardness of *ElGamal* algorithm is essentially equivalent to the hardness of finding discrete logarithm modulo a large prime.

The reminder of this report is organized as follows. Chapter 2 provides background material and basic concepts in cryptography required for this project. Simulation of arithmetic and modular operations in a suitable way for efficient implementation of public-key cryptosystems and some of these algorithms are represented in chapter 3. In Chapter 4 some of the elementary theorem in number theory and number-theoretic computational problems, related to public-key algorithms which form the security bases for encryption schemes of these algorithms are considered. Prime generation and some of the primality test algorithms are also discussed in this chapter. Examples of some public key cryptosystems are treated in chapter 5 . RSA public-key cryptosystem is the topic of chapter 6, in this chapter key generation and encryption scheme (encryption / decryption operations) algorithms are presented as well as security discussions. Chapter 7 includes the same material as chapter 6 but for ElGamal public-key cryptosystem. Implementation details is the title of chapter 8, this chapter contains the implementation issues in four main stages of the project, which are arithmetic operations, prime generation and primality tests, implementation of RSA, ElGamal systems and data conversion needed for both systems. The advantage and disadvantage of each implemented public-key cryptosystems are mentioned and compared in chapter 9, this chapter also contains a conclusion and security recommendations for public-key systems.

Chapter 2

Background and Basic Concepts

A major goal of information security techniques is "*confidentiality*" ensuring that adversaries gain no intelligence from a transmitted message in a network. There are two major methods for achieving confidentiality :

- ***Steganography***: the art of hiding a secret message within a larger one in such a way that the adversary can not discern the presence or contents of the hidden message. For example, a message might be hidden within a picture by changing the low-order pixel bits to be the message bits (refer to [71] for more information on steganography).
- ***Cryptography***: transforming the message into a ciphertext such that an adversary who overhears the ciphertext can not determine the message sent. The legitimate receiver posses a secret key that allows him/her to reverse the encryption transformation and retrieve the message. The sender may have used the same key to encrypt the message (with symmetric scheme) or used a different but related key (with public-key scheme).

2.1 Overview of cryptography

The practice of encryption messages has been in existence for a long time and cryptosystems have been used by the military and by the diplomatic services through out the centuries[25]. Conventionally a cryptographic algorithm, also called a cipher, was a mathematical function which by nature was used for both encryption and decryption of messages. However the security of the algorithm was dependent on keeping its operation a secret, which was popularly turned as the restrictions of that algorithm.

Modern cryptography uses a system of keys to solve the problems of conventional algorithms. This key might be one among several possible in a large key-space. Both the encryption and decryption operations use this key. Mathematically :

$$E_k(M) = C$$

$$D_k(C) = M$$

where k is the key, E is the encryption operation, D is the decryption operation and C is the ciphertext.

Some algorithms use different keys for encryption and decryption but the idea is the same and all the security is in the key rather than the algorithm. This means that the algorithm can be safely published. A cryptosystem is an algorithm plus all possible plaintexts(messages), ciphertexts and keys.

In a conventional cryptosystem, E and D are parameterized by a single key K, so that we have $D_k(E_k(M))$ = M. It is often the case that the algorithms for obtaining D_k and E_k from K are public, although both E_k and D_k are secret.

2.2 Purposes of cryptography

Besides providing confidentiality, cryptographic systems have been extensively used for jobs such as:

- ***Secrecy*** : refers to denial of access to information by unauthorized individuals.
- ***Authentication***: ensures that the origin of a message is correctly identified, which an assurance that the identity is not false.
- ***Integrity***: it should be possible for the receiver of a message to verify that the message has not been modified while it is transmitted.
- ***Non-repudiation***: neither the sender nor the receiver of a message should be able to deny the transmission.

These play a vital role in today's social interactions via computers. Hence the cryptographic systems are most compared on the basis of their capability to provide these facilities.

2.2.1 Requirements for secrecy

Secrecy requires that a cryptanalyst (i.e., a would-be intruder into a cryptosystem) should not be able to determine the plaintext corresponding to given ciphertext, and should not be able to reconstruct D by examining ciphertext for known plaintext. This translates into two requirements for a cryptosystem to provide secrecy:

a. A cryptanalyst should not be able to determine M from E(M); i.e., the cryptosystem should be immune to ciphertext-only attacks.

b. A cryptanalyst should not be able to determine D given {E(M_i)} for any sequence of plaintexts {M_1, M_2,...} ; i.e. the cryptosystem should be immune to known-plaintext attacks. This should remain true even when the cryptanalyst can choose {M_i} (chosen-plaintext attack), including the case in which the cryptanalyst can inspect {E(M_1),...,E(M_j)} before specifying M_{j+1} (adaptive chosen-plaintext attack).