El Gamal Mixnets and Implementation of a Verifier
SA104x Degree Project in Engineering Physics
KTH Royal Institute of Technology
School of Computer Science and Communication
Supervisor: Douglas Wikström

Carl Svensson carlsven@kth.se & Erik Larsson erikl3@kth.se

April 11, 2013

Contents

1 Introduction

Why cryptography?

Classical vs Modern cryptography

Public key cryptosystem

Uses (Communication, Signatures?, Verification)

Mix networks

Verificatum Mixnet

Verification and its importance

Implementation

2 El Gamal Cryptography

2.1 Definition

The El-Gamal cryptosystem is defined over a group $G_q = \langle g \rangle$ of prime order q, generated by $g \in G_q$. A private key $x \in \mathbb{Z}_q$ is chosen randomly and is used to compute the public key $(g, y) \in G_q \times G_q$ where $y = g^x$.

Encryption of a plaintext $m \in G_q$ is done by choosing a random $s \in Z_q$ and computing $\text{Enc}_{pk}(m, s) = (u, v) \in G_q \times G_q$ where $u = g^s$ and $v = y^s m$. Decryption of a ciphertext $(u, v) \in G_q \times G_q$ is achieved by using the private key x to compute $m = \text{Dec}_{pk}(u, v) = u^{-x} v$.

2.2 Security

Let $b = g^a \in G_q$ where $a \in \mathbb{Z}_q$. Then a is said to be the discrete logarithm of b in the group G_q. There is currently no known efficient classical algorithm that given (G_q, g, b) is able to calculate a in a reasonable amount of time (polynomial time). The discrete logarithm problem is thus considered to be a hard problem. (Källa)

The security of the El Gamal cryptosystem relies on the difficulty of discrete logarithm in finite cyclic groups G_q. This means that the El Gamal cryptosystem is secure as long as no one is able to compute the discrete logarithm in G_q efficiently. (Källa)

2.3 Properties

The El Gamal cryptosystem is a homomorphic cryptosystem. This means that for any two messages $m_1, m_2 \in G_q$ and randomnesses $s_1, s_2 \in \mathbb{Z}_q$

$$\text{Enc}_{pk}(m_1, s_1)\text{Enc}_{pk}(m_2, s_2) = (g^{s_1}, y^{s_1} m_1)(g^{s_2}, y^{s_2} m_2) =$$

$$= (g^{s_1+s_2}, y^{s_1+s_2} m_1 m_2) = \text{Enc}_{pk}(m_1 m_2, s_1 + s_2)$$

By choosing $m_1 = m$ and $m_2 = 1$ one obtains

$$\text{Enc}_{pk}(m, s_1)\text{Enc}_{pk}(1, s_2) = \text{Enc}_{pk}(m, s_1 + s_2)$$

This homomorphic property of the El Gamal Cryptosystem may be used to reencrypt an already encrypted message. If $s_1 \in \mathbb{Z}_q$ and $s_2 \in \mathbb{Z}_q$ are both chosen with the uniform randomness, then $s_1 + s_2 \in \mathbb{Z}_q$ will be uniformly random as well (utveckla + källa).

2.4 Generalization

A generalization of the El Gamal Cryptosystem over a group G_q can be achieved by considering the plaintext group M_w to be $G_q \times .. \times G_q = G_q^w$ and the ciphertext group to be $C_w = M_w \times M_w$ (källa: douglas doc). Encryption and decryption is done componentwise and the group operation of M_w will also be performed componentwise.

Förklara mer?

3 Cryptographic Primitives

PRGs ROs

4 Mix Networks

4.1 Overview

Intuitiv beskrivning (gör bättre)

http://www.rsa.com/rsalabs/staff/bios/ajuels/publications/universal/Universal.pdf

One purpose of mix networks, or mixnets, is to provide untraceability to its users. A mixnet may, for example, take as input a list of encrypted messages of different origins. These messages pass through the mixnet and is output

decrypted and in a randomized order. This property may be used to enable anonymous voting systems.

A reencryption mixnet consists of a number of servers which sequentially process the messages and reencrypts the list of messages and outputs them in a randomized order. After passing through all servers, the list of ciphertexts is decrypted and the result is a list of the messages in random order. It is impossible to deduce from where each element came.

4.2 El Gamal Mixnets

4.3 Operation

4.4 Verification

5 Specification

The documentation describing the Verificatum Mixnet verifier—.

The document describes a number of subtasks, some of which may be implemented independently. The subtasks include include implementation of general representation of data, an arithmetic library needed to perform group operations, the cryptographic primitives and the structure of files created during an execution of the mixnet. Furthermore, all algorithms executed during the verification are described in detail in order to allow an independent verifier.

Here follows some suggestions to improve the readability and —.

Inkonsekvent namngivning i vissa algoritmer h-vektorn som inte står med i argumentlistan Beskrivning av Pedersen commitments

Vilken dokumentation har vi använt oss av?

6 Implementation of the Verifier

6.1 General Design Choices

Programming language?

Objects, UML

6.2 Third Party Libraries

6.2.1 Arithmetic Library

GMP why?

6.2.2 XML Parser

RapidXML why?

6.2.3 Cryptographic Primitives

OpenSSL why?

6.2.4 Testing

Google Test why?

6.3 Math Library

Hur och varför har vi gjort som vi gjort?

6.4 Pseudorandom Generators and Random Oracles

6.5 Verifier

6.6 Tests

6.7 Performance

Viktigt?

7 Conclusion

Kunde dokumentationen ha gjorts bättre?

8 References