

El Gamal Mix-Nets and Implementation of a Verifier

Erik Larsson (erikl3@kth.se) and Carl Svensson (carlsven@kth.se)

INTRODUCTION

A mix-net is a cryptographic protocol based on public key cryptography which enables untraceable communication through a collection of nodes. One important application is electronic voting where it enables the construction of systems which satisfies many voting security requirements, including verifiability of correct execution. Verificatum is an implementation of a mix-net by Douglas Wikström.

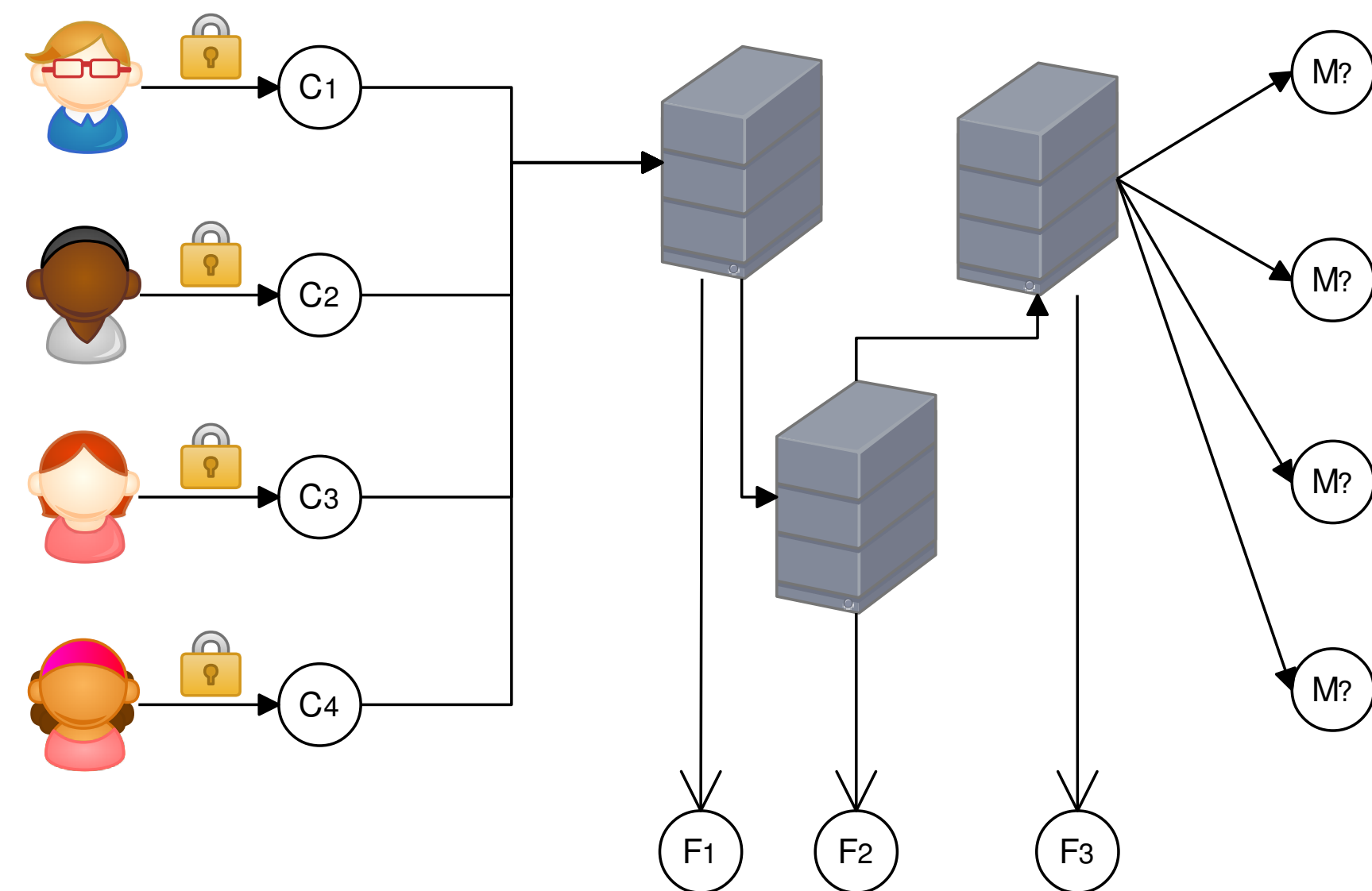
Our work concerns the implementation of a verifier for the Verificatum mix-net and evaluation of a document describing the implementation. The purpose of the document is to enable third parties to convince themselves that the mix-net has behaved correctly without revealing any secret information. By implementing a verifier, the document is examined and feedback is provided to its author. This is one step in making the document correct and more accessible.

MIX NETWORKS

A mix-net takes as input a list of encrypted messages. Verificatum is a reencryption mix-net. Such a mix-net consists of a number of servers, mix 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 the messages output in random order.

In the context of electronic voting a reencryption mix-nets may work as follows.

1. The mix servers prepare the mix-net by generating public and secret keys.
2. Each voter encrypts his vote and appends it to a public list of encrypted votes.
3. In sequential order each mix server takes as input the list of encrypted votes, reencrypts and outputs them in a randomized order, replacing the previous list of encrypted votes.
4. After all mix servers have processed the list, each vote is jointly decrypted and posted on a bulletin board making the outcome of the election universally available without revealing how anyone voted.

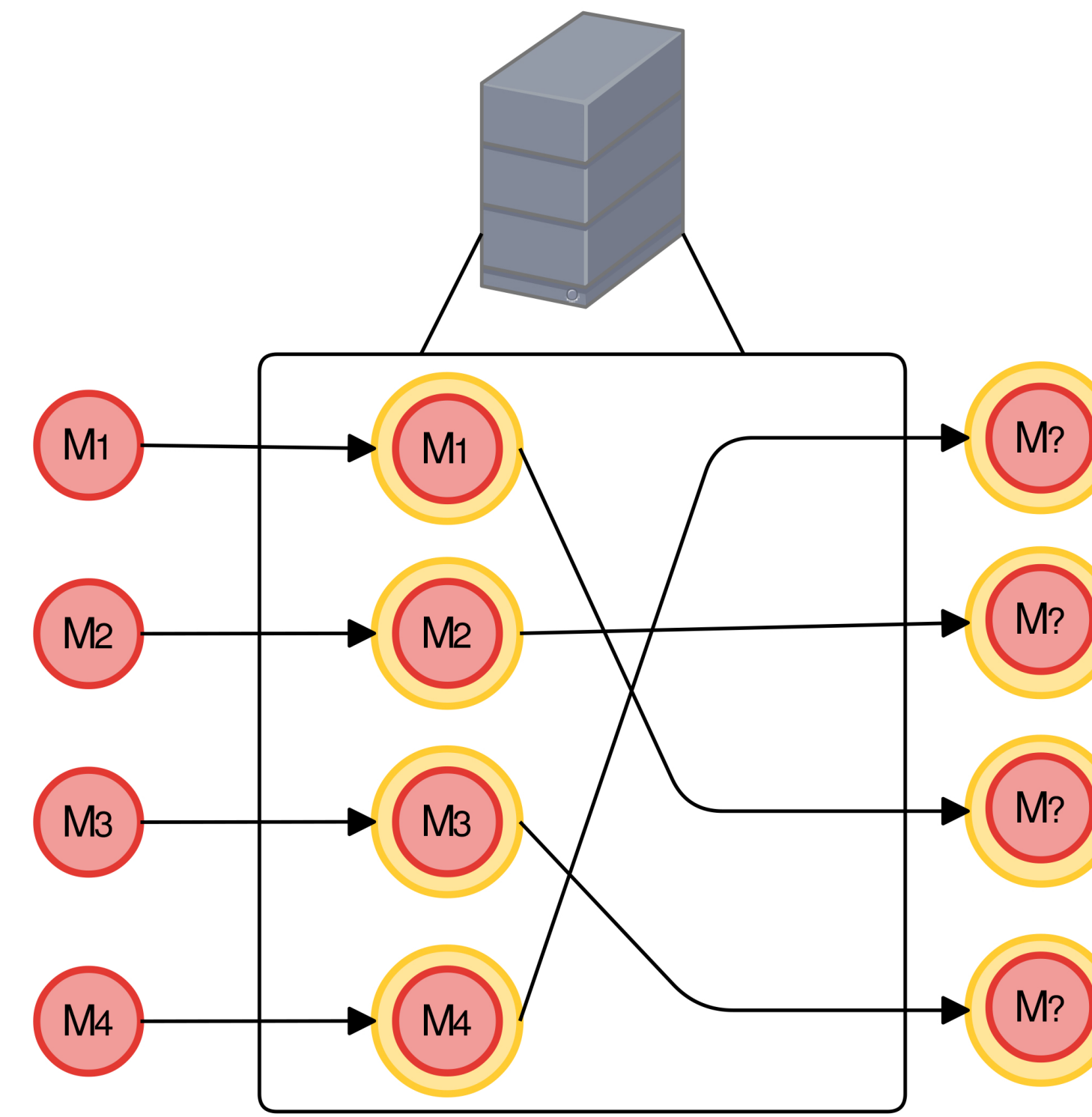


MIX SERVER

Each server in the mix-net performs the same actions during execution.

1. It is given the list of ciphertexts from the previous server.
2. It reencrypts these ciphertexts with its private key.
3. It randomly shuffles the list of ciphertexts.
4. It passes the reencrypted and shuffled list of ciphertexts to the next server.

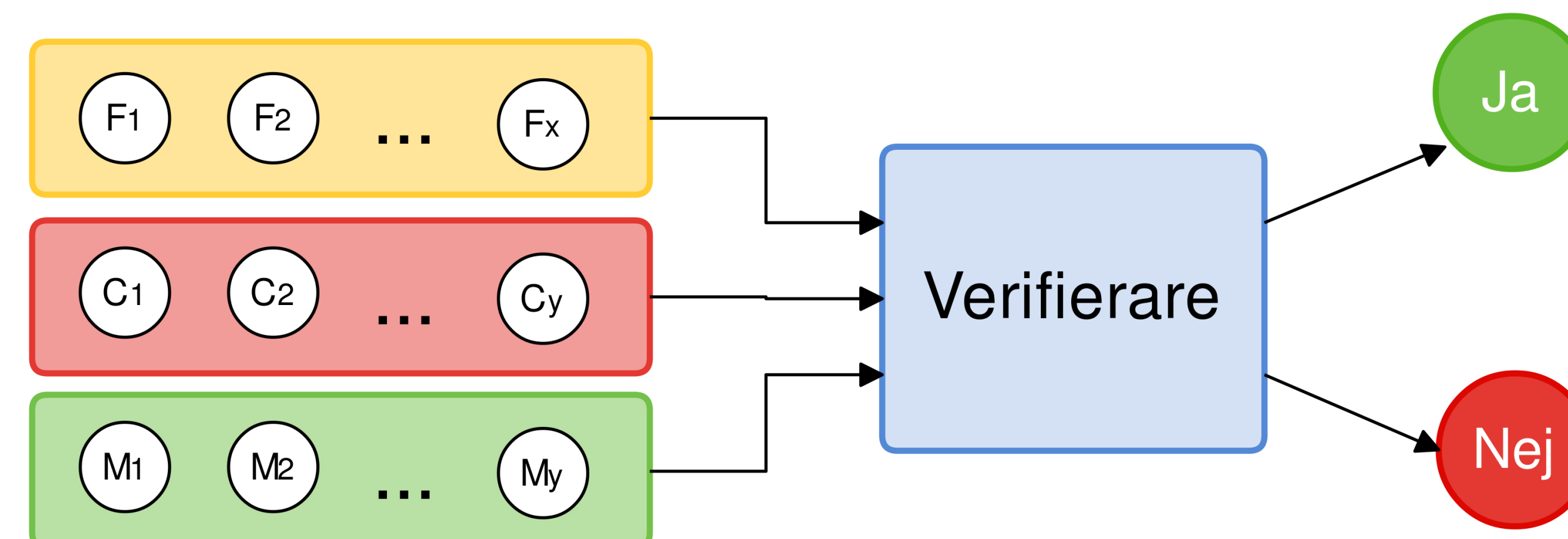
During these steps a zero-knowledge proof is produced which can prove that the server indeed did output a list of shuffled and reencrypted ciphertexts without tampering with the ciphertexts.



VERIFIER

The verification of execution relies on a concept called zero-knowledge proof. A zero-knowledge proof is a cryptographic protocol that can be used by one party, the prover, to prove to another party that some statement is true without revealing additional information.

The verifier takes as input the ciphertexts, the output messages and the extra data, the zero-knowledge proofs, created by each mix-net server. It then performs a series of calculations which prove whether the mix-net was executed correctly or not and outputs this.



EL GAMAL CRYPTOGRAPHY

The El Gamal cryptosystem is a public key cryptosystem based on the difficulty of computing discrete logarithms in cyclic groups.

The cryptosystem is defined over a group G_q of order q with generator g . The secret key x is chosen randomly in \mathbb{Z}_q and the public key y is created as follows

$$y = g^x$$

Encryption of a plaintext $m \in G_q$ is done by choosing a random $s \in \mathbb{Z}_q$ and computing

$$\text{Enc}(m, s) = (u, v) = (g^s, y^s m) \in G_q \times G_q$$

Decryption of a ciphertext $(u, v) \in G_q \times G_q$ is achieved by using the private key x to compute

$$\text{Dec}(u, v) = u^{-x} v = (g^s)^{-x} y^s m = (g^x)^{-s} y^s m = y^{-s} y^s m = m$$

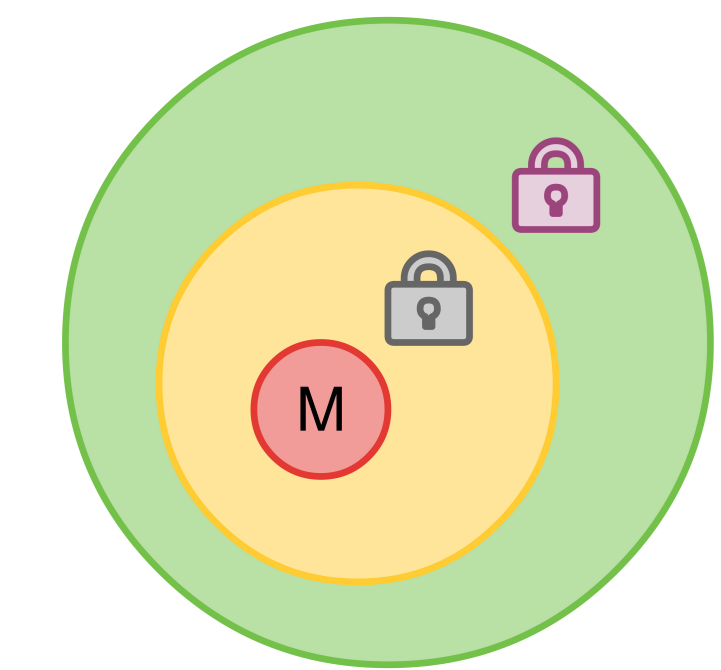
The El Gamal cryptosystem possesses a homomorphic property. This means that for two messages $m_1, m_2 \in G_q$ and random numbers $s_1, s_2 \in \mathbb{Z}_q$

$$\text{Enc}(m_1, s_1) \cdot \text{Enc}(m_2, s_2) = \text{Enc}(m_1 m_2, s_1 + s_2)$$

By choosing one of the messages to be the identity element, $m_2 = 1$, and letting the other one being any message $m_1 = m$, one has obtained the ability to reencrypt a particular ciphertext without knowing the original plaintext nor the randomness.

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

This property of a cryptosystem is necessary if it should be used in a reencryption mix-net.



CONCLUSION

Using the cryptographic constructions of El Gamal cryptography it is possible to create a scheme for electronic voting by using mix-nets. From the use of zero-knowledge proofs the Verificatum mix-net is a verifiably secure implementation. The need for verification of correct execution is essential to the use of computers in voting applications. It is therefore of great importance that the documentation describing implementation of a verifier is correct and accessible.

The specification document for the verifier includes all the information required to write the verifier. However, the structure could be improved for greater readability. The document could also benefit from an improved overall description of the Verificatum mix-net so that one easier can get a better understanding of what approach to take when implementing the verifier.