The ElGamal Cryptosystem

Eleanor McMurtry

April 8, 2020

1 Mathematical background

References for these notes can be found in [KL14] and [BS15]. We begin by formalising the notion of a public key cryptosystem. Recall that we have two basic objects:

- a *public key* that everybody knows, and is used to *encrypt* messages
- a *secret key* that only you know, and is used to *decrypt* messages

Definition 1.1. Public key cryptosystem
A *public key cryptosystem* is a set of three algorithms:

- $\mathsf{Gen}(n)$: given a security parameter n, outputs a keypair (pk, sk).
- $\mathsf{Enc}_{pk}(m)$: encrypts $m \in M$ (where M is the *message space*) with public key pk.
- $\mathsf{Dec}_{sk}(c)$: decrypts $c \in C$ (where C is the *ciphertext space*) with secret key sk.

We will carry this idea in the back of our minds as we develop the theory needed to build ElGamal. As in all modern cryptography, the theory begins with the concept of a *group*. We will define some basic concepts related to groups and modular arithmetic, which will be the basis of our cryptosystem.

Definition 1.2. Group
A *group* is a set G with an associative operation $\cdot$ such that:

- there exists an *identity* element $e \in G$ such that for all $g \in G$

$$g \cdot e = e \cdot g = g$$

- for all $g \in G$, there exists an *inverse* element $g^{-1} \in G$ such that

$$g \cdot g^{-1} = g^{-1} \cdot g = e$$

Example 1.1. The set of integers $\mathbb{Z}$ with $+$ as the operation and 0 as the identity forms a group, since for all $z \in \mathbb{Z}$, $z + (-z) = 0$.

Example 1.2. The set of integers *modulo* 4, $\mathbb{Z}_4$, is a group with $+$ as the operation and 0 as the identity.

- Remember, "modulo" means "the remainder when divided by".
- $\mathbb{Z}_4 = \{0, 1, 2, 3\}$, and addition looks like:

$+$	**0**	**1**	**2**	**3**
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

Example 1.3. The set of integers modulo 5 with $\times$ as the operation and 1 as the identity, $\mathbb{Z}_5^\times$, forms a group.

- $1 \times 1 = 1 \bmod 5$
- $2 \times 3 = 6 = 1 \bmod 5$
- $3 \times 2 = 6 = 1 \bmod 5$
- $4 \times 4 = 16 = 1 \bmod 5$

So every element has an inverse.

We now turn our attention to a particular kind of group called a *cyclic group*.

Definition 1.3. Order
Given a group G and an element $g \in G$, the smallest positive n such that $g^n = e$ is called the *order* of g.

Example 1.4. 3 has order 4 in $\mathbb{Z}_5^\times$, since

- $3^1 = 3$
- $3^2 = 9 = 4 \bmod 5$
- $3^3 = 27 = 2 \bmod 5$
- $3^4 = 81 = 1 \bmod 5$

Definition 1.4. Generator, cyclic group
Given a group G with n elements, an element of order n is called a *generator* for G. If G has a generator, we say that G is a *cyclic group*.

Example 1.5. $\mathbb{Z}_5^\times$ is a cyclic group with generator 3.

Intuitively, the idea is that raising a generator to higher and higher powers gives you every element of the group. We can now define our cryptosystem.

2 The ElGamal cryptosystem

Definition 2.1. ElGamal cryptosystem
The *ElGamal cryptosystem* is a public key cryptosystem with the algorithms:

- $\mathsf{Gen}(n)$: choose n-bit primes p, q, and an element g of order q in $\mathbb{Z}_p^\times$. Choose a random $x \in \mathbb{Z}_q$ and set $y = g^x$. The public key is $pk = (g, p, q, y)$ and the secret key is $sk = (g, p, q, x)$.
- $\mathsf{Enc}_{pk}(m)$: for $m \in \mathbb{Z}_q$, choose a random $r \in \mathbb{Z}_q$ and output the encryption (g^r, my^r).
- $\mathsf{Dec}_{sk}(c)$: given $c = (g^r, my^r)$, output $my^r/(g^r)^x = my^r/y^r = m$.

In order to divide in the group, we will use the below theorem:

Theorem 2.1. *Fermat's little theorem*
If p is a prime number, then for all $a \in \mathbb{Z}$

$$a^{p-1} = 1 \bmod p$$

As a direct consequence, $1/g = g^{-1} = g^{p-2}$ for $g \in \mathbb{Z}_p^\times$.

3 Security properties of ElGamal

We will provide proofs that ElGamal has the standard security properties for public key cryptosystems (that is, that an adversary has only a small probability of successfully breaking the system). Definitions and proofs below are based on those in [KL14].

We begin with a definition of what a "small" probability of success means.

Definition 3.1. Negligible function
A function $f : \mathbb{N} \to \mathbb{R}$ is *negligible* if for all polynomials $\mathsf{poly}(x)$, there exists $N > 0$ such that for all $x > N$

$$|f(x)| < \frac{1}{\mathsf{poly}(x)}$$

To continue setting the stage, let us define our adversary:

Definition 3.2. Probabilistic polynomial-time adversary
A *probabilistic polynomial-time adversary* (denoted $\mathcal{A}$) is an algorithm (typically interactive) that runs in polynomial time and has access to a randomness source.

We follow with a standard problem that is believed to be computationally difficult to solve (in some groups). We phrase the problem as a *game* between an adversary and a challenger; we will demonstrate that an adversary has a negligible advantage over a coin toss in the game. Intuitively, the goal is to be able to distinguish between the triples (g^a, g^b, g^{ab}) and (g^a, g^b, g^c) for a generator g and uniformly random a, b, c.

Definition 3.3. Decisional Diffie-Hellman (DDH)
Given a cyclic group G of order q, consider the following game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$:

1. $\mathcal{C}$ chooses a, b, c uniformly at random from $\mathbb{Z}_q$, and calculates $x_0 = g^c$ and $x_1 = g^{ab}$.
2. $\mathcal{C}$ sends g^a and g^b to $\mathcal{A}$.
3. $\mathcal{C}$ chooses a random bit i and sends x_b to $\mathcal{A}$.
4. $\mathcal{A}$ outputs a bit b.

$\mathcal{A}$ wins if $b = i$. If for all probabilistic polynomial-time adversaries $\mathcal{A}$, there exists a negligible function negl such that $\Pr[\mathcal{A} \text{ wins}] < \frac{1}{2} + \mathsf{negl}(q)$, we say *the DDH assumption holds in* G_q.

The next definition is what we shall use as a definition of security. The goal is to show that an adversary cannot run a *chosen plaintext attack* (CPA), where by encrypting specific ciphertexts, they can leak information about the secret key and/or other ciphertexts. Clearly if the adversary can decrypt without the key, they can succeed at a CPA; by contraposition, an adversary who cannot succeed at a CPA cannot succeed at general decryption.

Definition 3.4. IND-CPA secure
Given a public key cryptosystem Π with security parameter q, consider the following game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$:

1. $\mathcal{C}$ computes public and secret keys (pk, sk) and sends pk to $\mathcal{A}$, together with oracle access to Enc_{pk}.
2. $\mathcal{A}$ sends a pair of messages m_0, m_1 to $\mathcal{C}$.
3. $\mathcal{C}$ chooses a random bit b, and the ciphertext $\mathsf{Enc}_{pk}(m_b)$ is computed and sent to $\mathcal{A}$.
4. $\mathcal{A}$ outputs a bit b'.

$\mathcal{A}$ wins if $b' = b$. If for all probabilistic polynomial-time adversaries $\mathcal{A}$, there exists a negligible function negl such that $\Pr[\mathcal{A} \text{ wins}] < \frac{1}{2} + \mathsf{negl}(q)$, we say Π is *IND-CPA secure* (indistinguishable-chosen plaintext attack secure).

Theorem 3.1. *If the DDH assumption holds in $\mathbb{G}$, the ElGamal cryptosystem (with security parameter n) is IND-CPA secure.*

Proof. Let $\mathcal{A}$ be a probabilistic polynomial-time adversary for IND-CPA. Suppose that $\mathcal{A}$ can win the IND-CPA game with probability $\frac{1}{2} + \varepsilon(n)$. Consider a DDH adversary B. On input (g^a, g^b, x), it acts as the challenger to $\mathcal{A}$, giving it the alternative encryption oracle $\mathsf{Enc}_B(m) = (g^b, mx)$.

Case 1: if $x = g^c$, then $\mathsf{Enc}_B(m)$ is a uniformly random pair of elements, so $\Pr[\mathcal{A} \text{ wins}] = \frac{1}{2}$.

Case 2: if $x = g^{ab}$, then $\mathsf{Enc}_B(m)$ is a faithful encryption of m with randomness g^b and secret key a, so $\mathcal{A}$ wins with probability $\frac{1}{2} + \varepsilon(n)$.

By outputting the same bit as $\mathcal{A}$, $\mathcal{B}$ wins with probability at most $\frac{1}{2}+\varepsilon(n)$; but since the DDH assumption holds in $\mathbb{G}$, $\varepsilon(n) \leq \mathsf{negl}(q)$, so the ElGamal cryptosystem is IND-CPA secure. □

4 References

References

[BS15] Dan Boneh and Victor Shoup. A graduate course in applied cryptography. *Draft 0.2*, 2015.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. Chapman and Hall/CRC, 2014.