Methods of Public-Key Cryptography

Émilie Wheeler

December 10, 2012

Contents

1 Introduction **2**

2 Cryptosystems based on Elementary Number Theory **3**

2.1 Elementary Number Theory Background 3

2.1.1 The Euler Function and Primitive Roots 3

2.1.2 Important Algorithms 4

2.1.3 A Key Observation 5

2.2 Diffie-Hellman Key Exchange 6

2.3 ElGamal Protocol 8

2.4 RSA Cryptosystem 10

2.4.1 Proof of Proposition 2.13 12

3 Cryptosystems based on Elliptic Curves **16**

3.1 Elliptic Curve Background 16

3.2 Elliptic Diffie-Hellman Key Exchange 18

3.3 Elliptic ElGamal Protocol 19

3.4 Elliptic Curve Variation on the RSA Cryptosystem 22

4 Conclusion **23**

5 References **24**

1 Introduction

Secret codes have been around for thousands of years, the earliest form being observed in non-standard hieroglyphs carved into monuments of the Old Kingdom of Egypt circa 1900 BC. For some reason or another, humans have always been desperate for a means of secure communication, in which their secret message cannot be intercepted and interpreted by adversaries. The practice and study of techniques for secure communication in the presence of these adversaries is called cryptography. The ancient Greeks and Romans knew of ciphers and cryptography, but the latter's true claim to fame came thousands of years later, during the first and second World Wars. Many countries used cryptographic methods to exchange secret information over non-secure radio waves. The science of attempting to decrypt these secret messages is called cryptanalysis. Since WWII, cryptography and cryptanalysis have come a long way, with technological and mathematical advances leading to a vast array of cryptographic methods and protocols. Modern cryptography can be divided into two large branches: Private-Key Cryptography and Public-Key Cryptography.

Private-key cryptography, also known as symmetric-key cryptography, is a method in which the two parties wishing to communicate over a non-secure channel first agree on a key k, which they keep secret. To send a plaintext message m to the other party, one encrypts m by using the encryption algorithm E and the shared key k, to obtain the ciphertext $c{:=}E(k;m)$, which is sent to the other party. The second party uses the decryption algorithm D and the same key k to recover the plaintext $m{:=}D(k;c)$. The encryption and decryption algorithms E and D are publicly known, which means that anyone can decrypt the ciphertext if he or she knows the key k. Therefore, k must remain secret. The biggest problem with symmetric cryptography is exactly how the two parties can agree on a shared key k in a secure and efficient way.

In 1976, Whitfield Diffie and Martin E. Hellman published their paper entitled *New Directions in Cryptography*, and introduced the notion of Public-Key Cryptography (or asymmetric cryptography), which describes a solution to this problem. Their paper proposes that it is possible for two parties to exchange secret messages over a public channel and using publicly known algorithms. Public-key cryptography uses a public key (known by all) for encryption and a private key (known only by one party) for decryption. Below is a proper definition of public-key cryptosystems taken from [3].

Definition 1.1: A cryptosystem consisting of a set of enciphering transformations $\{E_e\}$ and a set of deciphering transformations $\{D_d\}$ is called a *Public-Key Cryptosystem* or an *Asymmetric Cryptosystem* if, for each pair (e,d), the enciphering key e, called the *public key*, is made publicly available, while the deciphering key d, called the *private key*, is kept secret. The cryptosystem must satisfy the property that it is computationally infeasible to compute d from e.

In the following report, I will present examples of public-key cryptography as well as the reasoning for their security.

2 Cryptosystems based on Elementary Number Theory

2.1 Elementary Number Theory Background

To help with the description of the specific cryptographic protocols in this report, I will first present some important number theory notions.

2.1.1 The Euler Function and Primitive Roots

Definition 2.1: The *Euler function* $\phi : \mathbb{N} \to \mathbb{N}$ is a mapping associating to each positive integer n the number $\phi(n)$ of elements of $\mathbb{Z}_n$ (integers modulo n) relatively prime to n, i.e. $\phi(n)$ is the number of integers $k \in \mathbb{Z}_n$ for which $gcd(n,k) = 1$.

The following are facts about ϕ:
• For a prime p and $k \geq 1, \phi(p^k) = p^{k-1}(p-1)$.
• For integers m, n with $gcd(m,n) = 1, \phi(mn) = \phi(m)\phi(n)$.

Using the above, we can prove that if $n = p_1^{a_1} p_2^{a_2} \cdots p_r^{a_r}$, where p_i are distinct primes and $a_i > 0$, then

$$\phi(n) = p_1^{a_1-1}(p_1-1)p_2^{a_2-1}(p_2-1)\cdots p_r^{a_r-1}(p_r-1).$$

Definition 2.2: Given an integer a and a positive integer n with $gcd(a, n)$=1, the *multiplicative order* of a (mod n) is defined to be the smallest positive integer k such that $a^k \equiv 1 \pmod{n}$.

Definition 2.3: A *primitive root* modulo an integer n is an element a such that $a^{\phi(n)} \equiv 1 \pmod{n}$ but no smaller power of a is congruent to 1 (mod n).

Theorem 2.4: There is a primitive root modulo any prime p. In particular, the group $\mathbb{Z}_p^*$ is cyclic.

The proof of this theorem can be found in [2], as well as a proof for the following:
• If there are any primitive roots (mod n), then there are exactly $\phi(\phi(n))$ of them.

For example, the powers of 3 mod 7 are

$$3^1 \equiv 3, 3^2 \equiv 2, 3^3 \equiv 6, 3^4 \equiv 4, 3^5 \equiv 5, 3^6 \equiv 1 \pmod{7}$$

so that 3 is a primitive root of 7.

Definition 2.5: *Carmichael's lambda-function* $\lambda(n)$ is defined to be the least number m such that $a^m \equiv 1 \pmod{n}$ for all a such that $gcd(a, n) = 1$.

The following are facts about λ:
•$\lambda(N)$ always divides $\phi(N)$ (but it may be strictly smaller).
• For p prime, $\lambda(p) = p - 1$.
• If $n = p_1^{a_1} p_2^{a_2} \cdots p_r^{a_r}$, where p_i are distinct primes and $a_i > 0$, then

$$\lambda(n) = lcm\{\lambda(p_1^{a_1}), \cdots, \lambda(p_r^{a_r})\}.$$

Note: The multiplicative order of a primitive root mod p is $\lambda(p) = p - 1$.

2.1.2 Important Algorithms

Theorem 2.6: (The Division Algorithm) If $a \in \mathbb{N}$ and $b \in \mathbb{Z}$, then there exist unique integers $q, r \in \mathbb{Z}$ with $0 \leq r < a$, and $b = aq + r$.

The proof of this theorem, as well as the next, can be found in [6].

Theorem 2.7: (The Euclidean Algorithm) Let $a, b \in \mathbb{Z}$ $(a \geq b > 0)$, and set $a = r_{-1}, b = r_0$. By repeatedly applying the Division Algorithm, we get $r_{j-1} = r_j q_{j+1} + r_{j+1}$ with $0 < r_{j+1} < r_j$ for all $0 \leq j < n$, where n is the least nonnegative number such that $r_{n+1} = 0$, in which case $gcd(a, b) = r_n$.

By reversing the Euclidean algorithm calculation, if $gcd(a, b) = 1$, we find that a has a multiplicative inverse mod b, i.e. $1 = \lambda a + \mu b$. That is, $\lambda a \equiv 1 \pmod{b}$.

2.1.3 A Key Observation

We will use the following observation to prove Theorem 2.12 in Section 2.4.

Observation 2.8: Suppose that N is the product of two distinct primes. Then, from any one of the following pieces of information, we can compute the others:
(1) the prime factors of N;
(2) $\phi(N)$;
(3) $\lambda(N)$.

Proof. Suppose we know (1), i.e. we know primes p and q such that $N = pq$. Then we can calculate $\phi(N) = (p-1)(q-1)$ and

$$\begin{aligned} \lambda(N) &= lcm(p-1, q-1) \\ &= \frac{(p-1)(q-1)}{gcd(p-1, q-1)}, \end{aligned}$$

where we can find $gcd(p-1, q-1)$ by using the Euclidean algorithm.

Now suppose we know (2), i.e. we know

$$\begin{aligned} \phi(N) &= (p-1)(q-1) \\ &= pq - p - q + 1 \\ &= N - (p+q) + 1 \end{aligned}$$

$$\Rightarrow p + q = N - \phi(N) + 1$$

Let's say $p + q = N - \phi(N) + 1 = \alpha$ for convenience.

$$\begin{aligned} & p + q = \alpha \\ & \Rightarrow p = \alpha - q \\ & \Rightarrow N = pq = (\alpha - q)q = \alpha q - q^2 \\ & \Rightarrow q^2 - \alpha q + N = 0 \\ & \Rightarrow q^2 - (N - \phi(N) + 1)q + N = 0, \end{aligned}$$

for which we can find the roots using the quadratic formula. Substitute q into $p = \alpha - q$ to find p. Knowing p and q, we can calculate $\lambda(N)$ as above.

Now suppose we know (3), i.e. we know $\lambda(N)$ and N.
Without loss of generality, suppose p is the larger prime factor. Then $\lambda(N) = lcm(p-1, q-1)$ is a multiple of $p-1$, and divides $\phi(N)$. Let $r \equiv N \pmod{\lambda(N)}$ be the remainder on dividing N by $\lambda(N)$. Then

- $N - \phi(N) \equiv r \pmod{\lambda(N)}$, since $\lambda(N) | \phi(N)$; and
- $N - \phi(N) = p + q - 1 < 2\lambda(N)$, since $\lambda(N) \geq p - 1 > q$ (assuming that $N > 6$).

So $N - \phi(N) = r$ or $N - \phi(N) = r + \lambda(N)$. We can solve the quadratic for each of these two possible values of $\phi(N)$; one of them will give us the factors of N. (Since p and q must be real, distinct roots.)

□

We will use these notions in the following sections of this report.

2.2 Diffie-Hellman Key Exchange

In their 1976 paper, Diffie and Hellman presented a method of key agreement over an insecure channel in which the two parties never had to meet. The shared key which results from the exchange is for use in a symmetric cipher. The following is the first proposed protocol in modern cryptography.

Suppose Alice wants to send a secret message to Bob. Let p be a sufficiently large prime, such that it is infeasible to compute discrete logarithms in ${\mathbb{Z}_p}^*$. Let g be a primitive root in ${\mathbb{Z}_p}^*$. These two values are publicly known, so an eavesdropper, Eve, has access to them.

First, Alice chooses a secret integer a at random, such that $0 \leq a \leq p-2$. Alice then computes $A \equiv g^a \pmod{p}$. Alice sends A to Bob. Meanwhile, Bob chooses a secret integer b at random, such that $0 \leq b \leq p-2$, and computes $B \equiv g^a \pmod{p}$. Bob sends this B to Alice. Receiving B, Alice computes $B^a \pmod{p}$, using her secret integer a. With A, Bob computes $A^b \pmod{p}$, using his secret integer b. Thus the shared secret value is

$$B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \pmod{p}$$

This is the key to be used in symmetric ciphers in order to send and receive messages over an insecure communication channel.

Table 1 provides a clear overview of the protocol.

Table 1: Diffie-Hellman Key Exchange

	Steps to follow	Alice		Eve		Bob
1.	A trusted party chooses and publishes a prime p and a primitive root g.	p, g		p, g		p, g
2.	Alice chooses a secret integer a.	a				
3.	Alice calculates $A \equiv g^a \pmod{p}$ and sends A to Bob.	$\downarrow$ $A \equiv g^a \pmod{p}$	--→	A	--→	A
4.	Bob chooses a secret integer b.					b
5.	Bob calculates $B \equiv g^b \pmod{p}$ and sends B to Alice.	B	←--	B	←--	$\downarrow$ $B \equiv g^b \pmod{p}$
6.	Alice and Bob compute the shared secret key k using their secret integers a and b.	$k \equiv B^a \pmod{p}$				$k \equiv A^b \pmod{p}$

Definition 2.9: The following problem is known as the *Discrete Logarithm Problem (DLP)*: Given g, A, and a prime p such that $A \equiv g^a \pmod{p}$, find a.

This problem is believed to be at least as difficult as factorisation, although it is not known to be in P nor in NP-complete. (See [15].) If the order of $g \pmod{p}$ is small, i.e. there are only a few distinct powers of $g \pmod{p}$, then a can be found by exhaustive search. Therefore, in order to make the problem hard, and ensure the security of the cryptosystem, we should take the order of g to be as large as possible, which is the reason we take g to be a primitive root mod p in the above protocol (g is an element of order $\lambda(p) = p-1$).

The eavesdropper Eve knows: p, g, $A \equiv g^a \pmod{p}$, and $B \equiv g^b \pmod{p}$. If Eve can solve the DLP on $A \equiv g^a \pmod{p}$ (or respectively on $B \equiv g^b \pmod{}$

p)), then she can find a (resp. b), and thus can compute the shared key $B^a \pmod p$ (resp. $A^b \pmod p$).

Definition 2.10: Let p be a prime and g be an integer. The *Diffie-Hellman Problem (DHP)* is the problem of computing $g^{ab} \pmod p$ from $g^a \pmod p$ & $g^b \pmod p$.

It is not known whether an algorithm that efficiently solves the DHP can also be used to solve the DLP.

2.3 ElGamal Protocol

In his 1985 paper entitled *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, Taher Elgamal described an asymmetric key encryption algorithm based on the Diffie-Hellman key exchange. Contrary to Diffie-Hellman, which simply produces a shared secret key, the ElGamal protocol proposes a method to transmit messages over an insecure channel.

First, Bob chooses a prime p, a primitive root $g \pmod p$, and an integer $a \in \{1, \cdots, p-2\}$, where a is **random**. Bob then computes $h \equiv g^a \pmod p$. So Bob's public key is (p, g, h). (Bob keeps a secret.)

Now, if Alice wants to send a plaintext message x to Bob, encoded as an integer in the range $\{1, \cdots, p-1\}$, she chooses a number $k \in \{1, \cdots, p-1\}$ at **random**. (k is called the *ephemeral key.*) Alice then computes $y_1 \equiv g^k \pmod p$, and $y_2 \equiv xh^k \pmod p$. Alice sends the ciphertext (y_1, y_2) to Bob.

Bob, receiving this ciphertext pair, can decipher the message by computing $(y_1)^{-a} \equiv (g^k)^{-a} \equiv (g^k)^{p-1-a} \pmod p$, since Bob knows a, y_1 was sent by Alice, and g is a primitive root mod p, i.e. $g^{p-1} \equiv 1 \pmod p$. He can then find x by computing

$$(g^k)^{p-1-a} y_2 \equiv 1^k \cdot (g^k)^{-a} x h^k \equiv (g^k)^{-a} x (g^a)^k \equiv x g^{ak-ak} \equiv x \pmod p$$

The exponent $\{p-1-a\}$ is positive and non-zero because $1 \leq a \leq p-2$. Therefore, g being chosen as a primitive root mod p facilitates the computation of x. Note that x could also be calculated by using Euclid's algorithm on h^k.

$$h^k \to \text{[Euclid's algorithm]} \to (h^k)^{-1}y_2 \equiv (h^k)^{-1}xh^k \equiv x \pmod{p}$$

Table 2 demonstrates the steps of the protocol.

Table 2: ElGamal Cryptosystem

	Steps to follow	Alice		Eve		Bob
1.	*Key Creation:* Bob chooses prime p, primitive root g (mod p), and random **secret** integer $1 \le a \le p-2$.					p, g, a ↓
2.	Bob computes $h \equiv g^a \pmod{p}$.					$h \equiv g^a \pmod{p}$
3.	Bob publishes (p, g, h).	p, g, h	⇠--	p, g, h	⇠--	p, g, h
4.	*Encryption:* Alice chooses plaintext $1 \le x \le p-1$.					
5.	Alice chooses random ephemeral key $1 \le k \le p-1$.	x, k ↓				
6.	Alice uses Bob's public key to compute $y_1 \equiv g^k \pmod{p}$ and $y_2 \equiv xh^k \pmod{p}$.	$y_1 \equiv g^k \pmod{p}$ $y_2 \equiv xh^k \pmod{p}$				
	Alice sends (y_1, y_2) to Bob.	y_1, y_2	--⇢	y_1, y_2	--⇢	y_1, y_2
7.	*Decryption:* Bob computes $(g^k)^{-a} \pmod{p}$ using y_1 and a.					$(y_1)^{p-1-a}$ $\equiv (g^k)^{-a} \pmod{p}$
8.	Bob finds x using the above and y_2.					↓ $(g^k)^{-a}y_2$ $\equiv (g^k)^{-a}x(g^a)^k$ $\equiv x \pmod{p}$

Eve knows p, g, h, y_1 and y_2. If Eve solves the Discrete Logarithm Problem (DLP) on $h \equiv g^a \pmod{p}$, then she can find a, and hence $h^k \equiv (y_1)^a \pmod{p}$. Knowing h^k, she can decipher x. If Eve solves the DLP on $y_1 \equiv g^k \pmod{p}$, then she can find k, and hence h^k. Again, she could decipher x. Hence, the security of the ElGamal protocol depends on the difficulty of the DLP.

Proposition 2.11: Fix a prime p and primitive root g to use for ElGamal encryption. Suppose that Eve has access to an oracle that decrypts arbitrary ElGamal ciphertexts encrypted using arbitrary ElGamal public keys. Then she can use the oracle to solve the Diffie-Hellman Problem.

The proof of this proposition is given in [4].

This proposition shows that the ElGamal system is secure if one assumes the Diffie-Hellman Problem is hard.

2.4 RSA Cryptosystem

In 1977, the MIT team of computer scientists Ron Rivest and Adi Shamir, and mathematician Leonard Adleman described an algorithm for public-key cryptography based on the presumed difficulty of the factorization of large integers. This cryptosystem, named RSA after the creators, is widely used and consists of the following three steps:

Key creation begins with one user, say Bob, choosing two large, random primes $p \neq q$ of roughly the same size, and calculating $n = pq$ and $\phi(n) = (p-1)(q-1)$. Bob then chooses e such that $gcd(e, \phi(n)) = 1$. (e is called the *encryption exponent.*) Bob publishes n and e.

In the encryption step, Alice chooses a plaintext encoded as an integer m in the range $\{1, \cdots, n-1\}$ that she wants to send to Bob. Using Bob's public key (n, e), Alice computes $c \equiv m^e \pmod{n}$. Alice sends the ciphertext c to Bob.

In the final step, decryption, Bob computes $d \equiv e^{-1} \pmod{\phi(n)}$ using the Euclidean algorithm. This is possible because $gcd(e, \phi(n)) = 1$. Bob then uses this d (called the *decryption exponent*) to compute $m \equiv c^d \pmod{n}$.

Table 3 demonstrates the steps clearly.

Table 3: RSA Cryptosystem

	Steps to follow	Alice		Eve		Bob
1.	*Key Creation:* Bob chooses large, distinct primes p and q and computes $n = pq$ and $\phi(n)$.					p, q $\downarrow$ $n = pq$, $\phi(n) = (p-1)(q-1)$
2.	Bob chooses encryption exponent e such that $gcd(e, \phi(n)) = 1$. Bob publishes (n, e).	n, e	←--	n, e	←--	e $\downarrow$ n, e
3.	*Encryption:* Alice chooses plaintext m.	m				
4.	Alice uses Bob's public key to compute $c \equiv m^e \pmod{n}$. Alice sends c to Bob.	$\downarrow$ $c \equiv m^e \pmod{n}$	--→	c	--→	c
5.	*Decryption:* Bob computes $d \equiv e^{-1} \pmod{\phi(n)}$.					$d \equiv e^{-1} \pmod{\phi(n)}$ $\downarrow$
6.	Bob computes m using d.					$m \equiv c^d \pmod{n}$

Eve knows n, e and c. So if Eve knows d, she can compute m from the fact that $m \equiv c^d \pmod{n}$. Obtaining d requires knowledge of $\phi(n)$, since d is the inverse of e mod $\phi(n)$. Knowledge of $\phi(n)$ requires in turn knowledge of p and q. If n is large, no good algorithms for finding factors p and q exist as of yet, thus the security of the RSA, at this time, is guaranteed.

Theorem 2.12: The secret components of the RSA protocol for each user are (p, q), $\phi(n)$ and d. If Eve obtains any one of these three values, she can calculate the others. Thus, the security of the system for that user will be destroyed.

Proof. Suppose Eve knows (p, q). Then she can calculate $\phi(n) = (p-1)(q-1)$. With $\phi(n)$, she can find $d \equiv e^{-1} \pmod{\phi(n)}$.

If Eve knows $\phi(n)$, she can compute d. From Observation 2.8, Eve can also find (p, q).

Finally, if Eve knows $d \equiv e^{-1} \pmod{\phi(n)}$, she can compute (p, q), and thus $\phi(n)$, in the following way:

(Denote $\phi(n)$ by ϕ for convenience.) We know that $ed \equiv 1 \pmod{\phi}$. Hence $\exists k \in \mathbb{Z}$ such that $ed - 1 = k\phi$. Euler's theorem states that if n and a are relatively prime integers (i.e. $gcd(n, a) = 1$), then $a^{\phi} \equiv 1 \pmod{n}$. Therefore, $a^{k\phi} \equiv a^{ed-1} \equiv 1^k \equiv 1 \pmod{n}$, for all a relatively prime to n, i.e. $\forall a \in \mathbb{Z}_n^*$ (a is invertible mod n). Let $ed - 1 = 2^s t$, where t is an odd integer. Therefore $a^{2^s t} \equiv 1 \pmod{n}$. We need the following proposition:

Proposition 2.13: $\exists\ 1 \leq i \leq s$ such that
• $a^{2^{i-1}t} \not\equiv \pm 1 \pmod{n}$ for exactly half of $a \in \mathbb{Z}_n^*$; and
• $a^{2^i t} \equiv 1 \pmod{n}$ for all $a \in \mathbb{Z}_n^*$.

The proof will be given in Section 2.4.1. Using this proposition, the proof of Theorem 2.12 can be completed as follows: We therefore have $kn = a^{2^i t} - 1 = (a^{2^{i-1}t} - 1)(a^{2^{i-1}t} + 1) \neq 0$
$\Rightarrow$ If we take $gcd(n, a^{2^{i-1}t} - 1)$, this will be a non trivial factor p or q of n.

So Eve simply has to repeatedly select random $a \in \mathbb{Z}_n^*$ and check if an $i \in [1, s]$ satisfying the above claim exists. The expected number of trials before a non-trivial factor of n is obtained is 2. Knowing p and q, Eve can calculate ϕ. □

This shows that these three values, (p, q), $\phi(n)$ and $d \equiv e^{-1} \pmod{\phi}$, **must remain secret** in order for the cryptosystem to remain secure.

Remark 2.14: The cryptosystem also relies on the difficulty of factoring integers, which means that if there is a significant advance in that direction, the RSA cryptosystem may be compromised. For example, using a *quantum computer*, Peter Shor discovered an algorithm in 1994 that solves the prime number factorization problem in polynomial time. Shor's algorithm takes only $\mathcal{O}(b^3)$ time and $\mathcal{O}(b)$ space on b-bit number inputs. In 2001, the first seven-qubit quantum computer ran Shor's algorithm and factored the number 15. Some believe quantum computers will never reach a level in which the security of the RSA is at risk, however if the technology in quantum computers does advance, serious readjustments to the RSA cryptosystem will have to be made.

2.4.1 Proof of Proposition 2.13

We give the proof in several steps.

Step 1: We prove that $s > 0$. If $s = 0$, then taking $(-1) \in \mathbb{Z}_n^*$,

$$(-1)^{de-1} \equiv (-1)^{2^0 t} \equiv (-1)^t \equiv (-1) \pmod{n},$$

since t is odd. But $a^{2^s t} \equiv 1 \pmod{n}$. This is a contradiction. Therefore $s > 0$.

Step 2: Let us choose $0 \leq i \leq s$ as the smallest possible value such that $a^{2^i t} \equiv 1 \pmod{n}$. By the same argument as in Step 1, $i > 0$.

Step 3: We identify two isomorphisms and their mappings of $+1$ and -1. From the Chinese Remainder Theorem, we have $\mathbb{Z}_n^* \simeq \mathbb{Z}_p^* \times \mathbb{Z}_q^* \simeq \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$, where $\mathbb{Z}_n^*$, $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$ are multiplicative groups, and $\mathbb{Z}_{p-1}$ and $\mathbb{Z}_{q-1}$ are additive groups.

Consider the isomorphisms $\mathbb{Z}_n^* \xrightarrow{\Phi_1} \mathbb{Z}_p^* \times \mathbb{Z}_q^* \xrightarrow{\Phi_2} \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$. Φ_1 maps the identity element of $\mathbb{Z}_n^*$, 1, to the identity element of $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$, $(1, 1)$. Φ_2 maps this element $(1, 1)$ to the identity element of $\mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$, which is $(0, 0)$, since $\mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$ is additive.

$$\mathbb{Z}_n^* \xrightarrow{\Phi_1} \mathbb{Z}_p^* \times \mathbb{Z}_q^* \xrightarrow{\Phi_2} \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$$

$$1 \mapsto (1,1) \mapsto (0,0)$$

We also have $\Phi_1(-1) = (-1,-1)$ by uniqueness. Φ_2 maps $(-1,-1)$ to an element $(x,y) \in \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$ such that

$$2x \equiv 0 \pmod{(p-1)}, \tag{1}$$

since $(-1)^2 = 1$, the identity element of $\mathbb{Z}_p^*$, and multiplication in $\mathbb{Z}_p^*$ corresponds to addition in $\mathbb{Z}_{p-1}$. Therefore,

$$x = 0 \text{ or } x = \tfrac{p-1}{2}$$

Similarly for y. Hence,

$$y = 0 \text{ or } y = \tfrac{p-1}{2}$$

But Φ_2 is an isomorphism (in particular, injective), so:

$$-1 \mapsto (-1,-1) \mapsto (\tfrac{p-1}{2}, \tfrac{p-1}{2})$$

Now, $\forall a \in \mathbb{Z}_n^*$, we have $a^{2^i t} \equiv 1 \pmod{n}$, the identity element in $\mathbb{Z}_n^*$.

So $\forall(\alpha,\beta) \in \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$, we have $(2^i t\alpha, 2^i t\beta) = (0,0)$, the identity element of $\mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$, since, as mentioned above, multiplication in $\mathbb{Z}_n^*$ corresponds to addition in $\mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$.

Step 4: We define 3 maps using the above isomorphisms. Set $m = 2^i t$. The map

$$\tilde{f} : \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1} \xrightarrow{x \mapsto mx} \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$$

is thus the zero map. But i is the smallest value such that $a^{2^i t} \equiv 1 \pmod{n}$, $\forall a \in \mathbb{Z}_n^*$, as seen in Step 2, therefore for $\frac{m}{2} = 2^{i-1}t$, the map

$$f : \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1} \xrightarrow{x \mapsto \frac{m}{2}x} \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$$

is not the zero map.

We define $f : \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1} \xrightarrow{x \mapsto \frac{m}{2}x} \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$, and $g : \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1} \xrightarrow{x \mapsto 2x} \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$.
So $g \circ f(x) = g(f(x)) = g(\frac{m}{2}x) = \frac{m}{2}2x = mx = (0,0), \forall x \in \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$.
We observe that $ker(g) = \{(\frac{p-1}{2}, 0), (\frac{p-1}{2}, \frac{q-1}{2}), (0, \frac{q-1}{2}), (0,0)\}$ which we can deduce from (1).

Step 5: We prove the following claim:

Claim: At least one of the following statements is true:
(a) $\exists x = (\alpha, 0) \in \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$ such that $f(x) \neq (0,0)$
(b) $\exists x = (0, \beta) \in \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$ such that $f(x) \neq (0,0)$.

Proof of the Claim: If both (a) and (b) are false, then

$$\begin{aligned} f(\alpha, \beta) &= f((\alpha, 0) + (0, \beta)) \text{ [addition in } \mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}] \\ &= f(\alpha, 0) + f(0, \beta) \text{ [} f \text{ is a group homomorphism]} \\ &= (0,0) + (0,0) \\ &= (0,0) \end{aligned}$$

But f is not the zero map. This is a contradiction.

Case 1: Suppose (a) is true and (b) is false. Then

$$\begin{aligned} f(\alpha, \beta) &= f((\alpha, 0) + (0, \beta)) \\ &= f(\alpha, 0) + f(0, \beta) \\ &= f(\alpha, 0) + (0,0) \\ &= f(\alpha, 0) \\ &= (\frac{m}{2}\alpha, 0) \end{aligned}$$

If α is even,

$$\begin{aligned} \frac{m}{2}\alpha &= \frac{m}{2}(2k) \\ &= mk, \end{aligned}$$

for some $k \in \mathbb{Z}$, is divisible by $\phi(n) = (p-1)(q-1)$ and thus $(p-1)$. So we have

$$\tfrac{m}{2}\alpha = 0 \text{ in } \mathbb{Z}_{p-1}.$$

If α is odd,

$$\begin{aligned}
\frac{m}{2}\alpha &= \frac{m}{2}(2k+1), \text{ for some } k \in \mathbb{Z}, \\
&= mk + \frac{m}{2} \\
&= 0 + \frac{m}{2} \text{ in } \mathbb{Z}_{p-1} \\
&\neq 0 \\
&\Rightarrow \frac{m}{2}\alpha = \frac{p-1}{2}
\end{aligned}$$

by the previous argument (1). Thus,

$$f(\mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}) = \{(0,0), (\tfrac{p-1}{2}, 0)\}.$$

From group theory, we have: if $f : G \longrightarrow H$ is a surjective group homomorphism between finite groups, then $|f^{-1}(\{x\})| = \frac{|G|}{|H|}$, for $x \in H$. Hence,

$$|f^{-1}((0,0))| = |f^{-1}((\tfrac{p-1}{2}, 0))|.$$

Therefore, exactly half of the elements of $\mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$ satisfy $f(x) = (\frac{p-1}{2}, 0)$. These x's correspond to elements $a \in \mathbb{Z}_p^* \times \mathbb{Z}_q^*$ such that $a^{\frac{m}{2}} \not\equiv \pm 1 \pmod{n}$.

Case 2: Suppose (a) is false and (b) is true. Similar to Case 1.

Case 3: Suppose both (a) and (b) are true.

$$\begin{aligned}
f(\alpha, \beta) &= f((\alpha, 0) + (0, \beta)) \\
&= f(\alpha, 0) + f(0, \beta) \\
&\neq (0,0) \\
&\Rightarrow f(\mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}) = \{(0,0), (0, \frac{q-1}{2}), (\frac{p-1}{2}, 0), (\frac{p-1}{2}, \frac{q-1}{2})\}.
\end{aligned}$$

Thus, as above,

$$|f^{-1}((0,0))| = |f^{-1}((0, \tfrac{q-1}{2}))| = |f^{-1}((\tfrac{p-1}{2}, 0))| = |f^{-1}((\tfrac{p-1}{2}, \tfrac{q-1}{2}))|.$$

Notice that $f^{-1}((\frac{p-1}{2}, 0)) \cup f^{-1}((0, \frac{q-1}{2}))$ corresponds to the set of $a \in \mathbb{Z}_p^* \times \mathbb{Z}_q^*$ such that $a^{\frac{m}{2}} \not\equiv \pm 1 \pmod{n}$, and therefore exactly half of the elements of $\mathbb{Z}_{p-1} \times \mathbb{Z}_{q-1}$ satisfy $f(x) = (\frac{p-1}{2}, 0)$ or $f(x) = (0, \frac{q-1}{2})$.

3 Cryptosystems based on Elliptic Curves

In 1985, Neal Koblitz and Victor S. Miller independently suggested using the algebraic structure of elliptic curves over finite fields in public-key cryptosystems. Similar to Diffie-Hellman and ElGamal, elliptic curve cryptosystems are based on the infeasibility of solving the discrete logarithm problem of a random point on the elliptic curve with respect to a publicly-known base point. The following section presents the basic notions of elliptic curves as well as an introduction to elliptic curve cryptosystems.

3.1 Elliptic Curve Background

Definition 3.1: Let $\mathbb{F}$ be a field. Assume that $char(\mathbb{F}) \neq 2$. For cryptographic purposes, we take $\mathbb{F}$ to be finite (e.g. $\mathbb{F} = \mathbb{F}_p = \mathbb{Z}_p$ for a prime p). An *elliptic curve* over $\mathbb{F}$ is the set of points $(x, y) \in \mathbb{F}$ that satisfy an equation of the form

$$E : y^2 = x^3 + \alpha x + \beta \tag{2}$$

where α and β are constants, together with one additional point, the *point at infinity*, ∞. The constants α and β must satisfy $4\alpha^3 + 27\beta^2 \neq 0$. (2) is known as the *Weierstrass equation* for an elliptic curve. We have:

$$E(\mathbb{F}_p) = \{(x, y) \mid x, y \in \mathbb{F}_p \text{ satisfy } y^2 = x^3 + \alpha x + \beta\} \cup \{\infty\}$$

Addition in Elliptic Curves: In order to "add" two points P and Q on an elliptic curve E to produce a third point R, we start by drawing a line L through points P and Q. L intersects the elliptic curve E at 3 points, P, Q and R (possibly ∞). We reflect R across the x-axis (i.e. we multiply its y coordinate by (-1)) to get the point R'. (Note that the reflection along the x-axis of ∞ is ∞.) R' is the "sum of P and Q", denoted $R' = P \oplus Q$. (See Figure 1.) To add a point P to itself, we take L as the tangent line to E at point P. (See Figure 2.)

Theorem 3.2: Let E be an elliptic curve. Then the addition law on E has the following properties:

(a)	$P \oplus \infty = \infty \oplus P = P$	for all $P \in E$.	[Identity]
(b)	$P \oplus (-P) = \infty$	for all $P \in E$.	[Inverse]
(c)	$(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$	for all $P, Q, R \in E$.	[Associative]
(d)	$P \oplus Q = Q \oplus P$	for all $P, Q \in E$.	[Commutative]

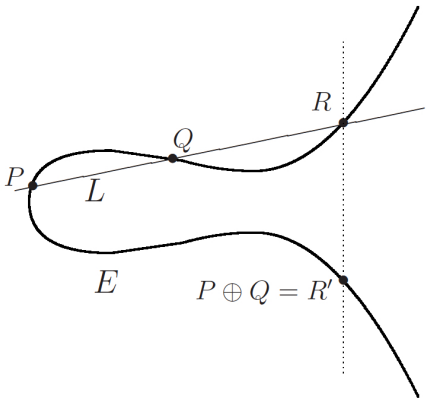


Figure 1: Addition of $P \oplus Q$, $P \neq Q$

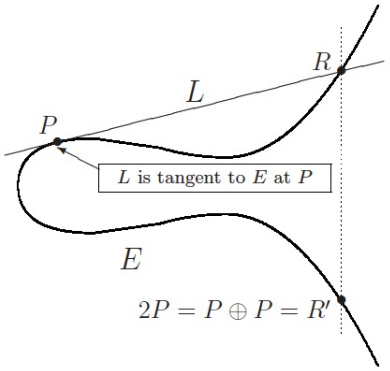


Figure 2: Addition of $P \oplus P$

Therefore, the addition law makes the points of E into an abelian group. The proof of this theorem is given in [4].

Multiplication in Elliptic Curves: Elliptic curve point multiplication is the operation of successively adding a point P along an elliptic curve E to itself. For a scalar n and $P = (x, y) \in E$:

$$n \cdot P = \underbrace{P \oplus P \oplus P \oplus \cdots \oplus P}_{\text{n times}}$$

The Double-and-Add Algorithm: In order to calculate $Q = n \cdot P$ for a large n, we can use the "Double-and-Add" algorithm as follows:
• Write n in binary expansion, i.e. $n = n_0 + n_1 \cdot 2 + n_2 \cdot 4 + n_3 \cdot 8 + \cdots + n_r \cdot 2^r$, $n_i \in \{0, 1\}$. (Assume $n_r = 1$.)

• Compute $Q_0 = P, Q_1 = 2 \cdot Q_0, Q_2 = 2 \cdot Q_1, \cdots, Q_r = 2 \cdot Q_{r-1}$. $(Q_i = 2^i \cdot P)$
• Compute $n \cdot P = n_0 \cdot Q_0 + n_1 \cdot Q_1 + n_2 \cdot Q_2 + \cdots + n_r \cdot Q_r$, which is simple because $n_i \in \{0, 1\}$.
($\Rightarrow$ The total time to compute $n \cdot P$ is at most $2r$ point operations in $E(\mathbb{F}_p)$.)
This algorithm and its proof are given in [4].

A fundamental result in the theory of elliptic curves is that elliptic curves have many points. This is obtained from Hasse's theorem:

Theorem 3.3: (Hasse's Theorem) Let $q = p^n$ and E be an elliptic curve over $\mathbb{F}_q$. Then the *order* (i.e. number of points) of $E(\mathbb{F}_q)$ is

$$\#E(\mathbb{F}_q) = q + 1 - t_q$$

where the *trace of Frobenius* t_q satisfies $|t_q| \leq 2\sqrt{q}$.

The proof of Hasse's theorem is given in [7].

3.2 Elliptic Diffie-Hellman Key Exchange

The following is a variation on the Diffie-Hellman Key Exchange using elliptic curves.

First, Alice and Bob agree on a large prime p, an elliptic curve $E(\mathbb{F}_p)$ and a point $P \in E(\mathbb{F}_p)$. These values are made public. Then, Alice chooses a secret integer n_A and computes the point on the elliptic curve $Q_A = n_A \cdot P \in E(\mathbb{F}_p)$. (She can do this using the "Double-and-Add" algorithm.) Bob respectively chooses n_B and computes $Q_B = n_B \cdot P \in E(\mathbb{F}_p)$. Alice and Bob exchange their respective Q_A and Q_B. Finally, using their secret values n_A and n_B, they compute the shared secret key

$$n_A \cdot Q_B = n_A(n_B \cdot P) = n_A n_B \cdot P = n_B(n_A \cdot P) = n_B \cdot Q_A$$

This is illustrated in Table 4.

Definition 3.4: Let E be an elliptic curve over the finite field $\mathbb{F}_p$ and let P and Q be points in $E(\mathbb{F}_p)$. The *Elliptic Curve Discrete Logarithm Problem* (ECDLP) is the problem of finding an integer n such that $Q = n \cdot P$.

Table 4: Elliptic Diffie-Hellman Key Exchange

	Steps to follow	Alice		Eve		Bob
1.	A trusted party chooses and publishes a prime p, an elliptic curve $E(\mathbb{F}_p)$ and a point $P \in E(\mathbb{F}_p)$.	$p \Rightarrow E(\mathbb{F}_p)$, $P \in E(\mathbb{F}_p)$		$p, E(\mathbb{F}_p), P$		$p, E(\mathbb{F}_p), P$
2.	Alice chooses a secret integer n_A.	n_A				
3.	Alice calculates $Q_A = n_A \cdot P$ and sends Q_A to Bob.	$\downarrow$ $Q_A = n_A \cdot P$	$\dashrightarrow$	Q_A	$\dashrightarrow$	Q_A
4.	Bob chooses a secret integer n_B.					n_B
5.	Bob calculates $Q_B = n_B \cdot P$ and sends Q_B to Alice.	Q_B	$\dashleftarrow$	Q_B	$\dashleftarrow$	$\downarrow$ $Q_B = n_B \cdot P$
6.	Alice and Bob compute the shared secret key k using their secret integers n_A and n_B.	$k = n_A \cdot Q_B$				$k = n_B \cdot Q_A$

So the eavesdropper Eve knows $p, E(\mathbb{F}_p), P, Q_A$ and Q_B. If Eve can solve the ECDLP on $Q_A = n_A \cdot P$, she can find n_A, and also on $Q_B = n_B \cdot P$, she can find n_B. With these two values, she can then compute $k = n_A n_B \cdot P$.

The fastest known algorithm to solve the ECDLP in $E(\mathbb{F}_p)$ takes approximately $\sqrt{p}$ steps (see [4]), which makes the ECDLP appear much more difficult than the DLP, whose fastest known algorithm (a number field sieve algorithm) takes approximately $e^{(\log p)^{1/3}(\log(\log p))^{2/3}}$ time (see [14]).

Definition 3.5: Let $E(\mathbb{F}_p)$ be an elliptic curve over a finite field and let $P \in E(\mathbb{F}_p)$. The *Elliptic Curve Diffie-Hellman Problem* (ECDHP) is the problem of computing the value of $n_A n_B \cdot P$ from the known values of $n_A \cdot P$ and $n_B \cdot P$.

Remark 3.6: It is possible for Alice and Bob to only send each other the x-coordinate of their respective Q_A, Q_B, and use only the x-coordinate of $k = n_A n_B \cdot P$ as the shared secret key, since the y-coordinate is related to x by $y_{Q_i}^2 = x_{Q_i}^3 + \alpha x_{Q_i} + \beta$ in $\mathbb{F}_p$ ($i \in \{A, B\}$) and contains little additional information. This allows them to send fewer bits over the insecure channel, and thus reduces the risk of an eavesdropper Eve to be able to decrypt the secret key.

3.3 Elliptic ElGamal Protocol

An elliptic curve variation on the ElGamal cryptosystem was also developed as follows ([7]).

Suppose Bob wants to send a message M to Alice. Alice and Bob first agree on a large prime p, an elliptic curve E over $\mathbb{F}_p$, and a point P in $E(\mathbb{F}_p)$. Alice then chooses a private key n_A and computes $Q_A = n_A \cdot P$ in $E(\mathbb{F}_p)$. She sends this Q_A to Bob. Bob then chooses a plaintext M encoded as a point on $E(\mathbb{F}_p)$ and an ephemeral key k (a scalar) at random. Bob then computes the points $C_1 = k \cdot P$ and $C_2 = M + k \cdot Q_A$ in $E(\mathbb{F})$, and sends this ciphertext pair (C_1, C_2) to Alice. Alice decrypts the ciphertext by computing

$$\begin{aligned} C_2 - n_A C_1 &= (M + k \cdot Q_A) - n_A(k \cdot P) \\ &= M + k(n_A \cdot P) - (n_A k) \cdot P \\ &= M + (n_A k) \cdot P - (n_A k) \cdot P \\ &= M \in E(\mathbb{F}_p). \end{aligned}$$

Table 5 illustrates the protocol clearly.

Table 5: Elliptic ElGamal Cryptosystem

	Steps to follow	Alice		Eve		Bob
1.	*Key Creation:* A trusted party chooses prime p, an elliptic curve E over $\mathbb{F}_p$, and a point $P \in E(\mathbb{F}_p)$, and makes these public.	$p, E(\mathbb{F}_p), P$		$p, E(\mathbb{F}_p), P$		$p, E(\mathbb{F}_p), P$
2.	Alice chooses a secret private key n_A.	n_A $\downarrow$				
3.	Alice computes $Q_A = n_A \cdot P$ in $E(\mathbb{F}_p)$. Alice sends Q_A to Bob.	$Q_A = n_A \cdot P$ Q_A	--→	Q_A	--→	Q_A
4.	*Encryption:* Bob chooses plaintext $M \in E(\mathbb{F}_p)$.					
5.	Bob chooses random ephemeral key k, a scalar.					M, k $\downarrow$
6.	Bob uses Alice's public key to compute $C_1 = k \cdot P \in E(\mathbb{F}_p)$ and $C_2 = M + k \cdot Q_A \in E(\mathbb{F}_p)$. Bob sends (C_1, C_2) to Alice.					$C_1 = k \cdot P$ $C_2 = M + k \cdot Q_A$
		C_1, C_2	←--	C_1, C_2	←--	C_1, C_2
7.	*Decryption:* Alice uses n_A to compute $C_2 - n_A \cdot C_1 = M \in E(\mathbb{F}_p)$.	$C_2 - n_A \cdot C_1 = M$				

So Eve can obtain $p, E(\mathbb{F}_p), P, Q_A, C_1$ and C_2. Therefore, if she can solve the ECDLP on $Q_A = n_A \cdot P$, she can find n_A and thus calculate $C_2 - n_A \cdot C_1 = M$. Or, if she could solve the ECDLP on $C_1 = k \cdot P$, she could find k and compute $C_2 - k \cdot Q_A = M$. However, if Eve cannot solve the ECDLP, it is infeasible for her to compute M, as shown in [7].

A natural question which we have not yet addressed is how to encode a message as a point on an elliptic curve. We will present two methods proposed by Neal Koblitz, one probabilistic and one non-probabilistic.

Probabilistic Representation of a Message on $E(\mathbb{F}_p)$:
Suppose E is an elliptic curve over $\mathbb{F}_p$ given by $y^2 = x^3 + \alpha x + \beta$, and let M be a message expressed as a number $0 \leq M \leq p/100$. Let $x_j = 100M + j$ for $0 \leq j < 100$. For $j = 0, 1, 2, \ldots, 99$, compute $s_j = x_j^3 + \alpha x_j + \beta$. If $s_j^{(p-1)/2} \equiv 1 \pmod{p}$, then s_j is a square mod p, so we do not have to try any more values of j. We take the square root of s_j as our y-coordinate for M, and the x_j used as our x-coordinate. When $p \equiv 3 \pmod{4}$, Lagrange showed that a square root of s_j is given by $y_j \equiv s_j^{\frac{(p+1)}{4}} \pmod{p}$. So we take $M = (x_j, y_j)$ as above. The point (x_j, y_j) is on E. Since s_j is essentially a random element of $\mathbb{F}_p^*$, which is cyclic of even order, the probability that s_j is a square is about $1/2$, and thus the probability of not finding a point for M after trying 100 values of j is about 2^{-100}.

Non-Probabilistic Representation of a Message on $E(\mathbb{F}_{p^n})$:
Suppose p is arbitrary (e.g., 2) and $n = 2n'$ is even. Suppose the plaintext is an integer m, $0 \leq m < p^{n'}$ written in the form $m = m_0 + m_1 p + \cdots + m_{n'-1} p^{n'-1}$, $0 \leq m_j < p$; and let $\{b_0, \ldots, b_{n'-1}\}$ be a convenient vector space basis of $\mathbb{F}_{p^{n'}}$ over $\mathbb{F}_p$. Set $x(m) = m_0 b_0 + m_1 b_1 + \cdots + m_{n'-1} b_{n'-1}$, and let $y(m) \in \mathbb{F}_{p^n}$ be either solution of the quadratic equations

$$y^2 = x^3 + \alpha x + \beta \ (\alpha, \beta \in \mathbb{F}_{p^n}); \text{ or}$$
$$y^2 + \gamma xy + \delta y = x^3 + \alpha x + \beta \ (\alpha, \beta, \gamma, \delta \in \mathbb{F}_{p^n}),$$

defining points on E. There may not necessarily exist a solution $y(m)$ in $\mathbb{F}_{p^{n'}}$, but there is guaranteed to be a solution $y(m)$ in $\mathbb{F}_{p^n}$, since $n = 2n'$ and therefore $\mathbb{F}_{p^n}$ is an extension of $\mathbb{F}_{p^{n'}}$. So we set $P_m = (x(m), y(m)) \in E$. Even though such a solution $y(m)$ is guaranteed to exist, the most efficient algorithms for solving quadratic equations over finite fields are probabilistic, such as the algorithm described above.

Koblitz has proposed other probabilistic methods as well, which can be found in [8].

3.4 Elliptic Curve Variation on the RSA Cryptosystem

In 1991, K. Koyama, U. Maurer, T. Okamoto and S. Vanstone proposed an elliptic curve analogue of the RSA cryptosystem, which was named the KMOV cryptosystem after its creators. KMOV is based on elliptic curves $E(\mathbb{Z}_n)$, where $n = pq$ for two distinct primes p, q.

First, we must introduce new notation and a lemma.

Notation: Let $p \geq 3$ be a prime. For an elliptic curve over the finite field $\mathbb{F}_p$, we will use the following notation for the Weierstrass equation:

$$E_p(a, b) : y^2 = x^3 + ax + b, \qquad a, b \in \mathbb{F}_p, \qquad 4a^3 + 27b^2 \neq 0,$$

Lemma 3.7: Let $p \geq 3$ be a prime satisfying $p \equiv 2 \pmod 3$ and $0 < b < p$. Then $E_p(0, b)$ is a cyclic group of order

$$\#E_p(0, b) = p + 1. \tag{3}$$

The proof of this lemma is given in [9].

We will now describe the KMOV protocol.

In the first step, key creation, Bob chooses two large primes $p \neq q$ such that $p \equiv q \equiv 2 \pmod 3$, and computes $n = pq$. Bob also computes $N_n = lcm(\#E_p(0, b), \#E_q(0, b)) = lcm(p + 1, q + 1)$ (by (3)) (He could use $N_n = (p + 1)(q + 1)$ in place of $lcm(p + 1, q + 1)$). Bob then chooses integers e, d such that $gcd(e, N_n) = 1$ and $ed \equiv 1 \pmod{N_n}$. Bob publishes his public key (n, e) and keeps private key $d, (p, q, \#E_p(0, b), \#E_q(0, b), N_n)$ secret.

Next, in the encryption step, Alice represents the plaintext message $(m_1, m_2) \in \mathbb{Z}_n \times \mathbb{Z}_n$ as a point $M \in E_n(0, b)$, where $E : y^2 = x^3 + b \pmod n$, and where $b = m_2^2 - m_1^3 \pmod n$. (Note that b does not have to be computed for the purpose of the cryptosystem, since addition over elliptic curves is independent of b.) Alice computes $C = e \cdot M = (c_1, c_2) \in E_n(0, b)$ and sends the ciphertext C to Bob.

In the final step, decryption, Bob computes the point

$$M = d \cdot C = d \cdot (e \cdot M) = (ed) \cdot M = M \in E_n(0, b).$$

Table 6 summarizes the KMOV cryptosystem.

Table 6: KMOV Cryptosystem

	Steps to follow	Alice		Eve		Bob
1.	*Key Creation:* Bob chooses large, distinct primes p and q such that $p \equiv q \equiv 2 \pmod 3$ and computes $n = pq$.					p, q $\downarrow$ $n = pq$
3.	Bob computes N_n $= lcm(\#E_p(0,b), \#E_q(0,b))$ $= lcm(p+1, q+1)$ (by (3)).					N_n $= lcm(\#E_p(0,b), \#E_q(0,b))$ $= lcm(p+1, q+1)$
2.	Bob chooses $e, d \in \mathbb{Z}$ such that $gcd(e, N_n) = 1$ and $ed \equiv 1 \pmod{N_n}$. Bob publishes (n, e).	n, e	⇠--	n, e	⇠--	e, d $ed \equiv 1 \pmod{N_n}$ $\downarrow$ n, e
3.	*Encryption:* Alice encodes her message $(m_1, m_2) \in \mathbb{Z}_n \times \mathbb{Z}_n$ as a point $M \in E_n(0,b)$, where $E : y^2 = x^3 + b \pmod n$; $b = m_2^2 - m_1^3 \pmod n$.	(m_1, m_2) $\downarrow$ $M \in E_n(0,b)$ $\downarrow$				
4.	Alice computes $C = e \cdot M$ $= (c_1, c_2)$. Alice sends C to Bob.	$C = e \cdot M$	--⇢	C	--⇢	C
5.	*Decryption:* Bob computes $M = d \cdot C$ using private key d.					$M = d \cdot C$

Hence Eve knows the public key (n, e) and the ciphertext C. As in the RSA cryptosystem, the difficulty of solving the order $\#E(\mathbb{F}_p)\#E(\mathbb{F}_q)$ as well as the difficulty of solving the secret key d are computationally equivalent to factoring a composite number n (see [10]).

Other elliptic curve variations on the RSA cryptosystem include the Demytko cryptosystem ([11]), Meyer and Müller's cryptosystem ([12]), and the Paillier-Galbraith encryption scheme ([13]).

4 Conclusion

In conclusion, we have described six public-key cryptographic protocols and examined the reasoning for their security. The Diffie-Hellman key exchange is based on the infeasibility of computing discrete logarithms in $\mathbb{Z}_p^*$, and produces a shared secret key which can be used for decryption in a symmetric cryptosystem. The ElGamal cryptosystem provides a method to exchange

messages over an insecure channel, and is also based on the infeasibility of solving the Discrete Logarithm Problem. These two protocols are thus secure, since no algorithm currently exists which solves the DLP in polynomial time. The RSA cryptosystem is widely used to exchange messages between parties. Unlike Diffie-Hellman and ElGamal, RSA is based on the difficulty of factoring large integers. This cryptosystem is also secure, because no efficient algorithms exist which factor such numbers. Next, we explored the elliptic curve analogues of these three protocols. The Elliptic Diffie-Hellman Key Exchange, instead of taking the powers of public keys, takes the point multiplication along the elliptic curve. It is secured by the Elliptic Curve Discrete Logarithm Problem, which appears more difficult than the DLP. The Elliptic ElGamal protocol is also identical to its non-elliptic curve counterpart, using elliptic curve point multiplication and addition as opposed to exponents and multiplication in $\mathbb{Z}_p$, and is also guaranteed by the hardness of the ECDLP. We also looked at the KMOV cryptosystem, an elliptic curve variation on the RSA protocol. KMOV depends on the prime factorization of n (for elliptic curve $E(\mathbb{Z}_n)$), and again uses the point multiplication as exponents. While the prime factorization problem is difficult, certain attacks have been proposed on the KMOV protocol, proving that it is insecure under certain conditions. (For an example of such an attack, see [9].) This makes the KMOV protocol much less widely implemented than the RSA. All of these cryptographic protocols (with perhaps the exception to the KMOV protocol) remain secure so long as no algorithmic advances are made in the areas of the DLP or prime factorization. Until then, they will continue to be implemented in order to make sharing secret messages over insecure channels not only possible, but efficient and resolute.

5 References

[1] Cameron, Peter J. *Notes on Cryptography.* Queen Mary, University of London, London, 2003.

[2] Stein, William. *Elementary Number Theory: Primes, Congruences, and Secrets.* Springer, London, 2008.

[3] Mollins, Richard. *RSA and Public-Key Cryptography.* CRC Press LLC, 2003.

[4] Hoffstein, Jeffrey, Pipher, Jill, Silverman, J.H. *An Introduction to Mathematical Cryptography.* Springer, 2008.

[5] Biggs, N. L. *An Introduction to Information Communication and Cryptography.* Springer-Verlag London Limited, 2008.

[6] Mollins, Richard. *Fundamental Number Theory with Applications.* CRC Press LLC, 1998.

[7] Washington, Lawrence. *Elliptic Curves: Number Theory and Cryptography.* CRC Press LLC, 2003.

[8] Koblitz, Neal. *Elliptic Curve Cryptosystems.* Mathematics of Computation, Vol. 48, No. 177, 1987, 203-209.

[9] Nitaj, A. *A New Attack on the KMOV Cryptosystem.* Université de Caen, Laboratoire de Mathématiques Nicolas Oresme, France, 2011.

[10] Koyama, K., Maurer, U.M., Okamoto, T., Vanstone, S.A. *New public-key schemes based on elliptic curves over the ring* $\mathbb{Z}_n$. Advances in Cryptology - Crypto'91, Lecture Notes in Computer Science, Springer-Verlag, pp. 252-266.

[11] Demytko, N. *A new elliptic curve based analogue of RSA.* Advances in Cryptology - EUROCRYPT '93, Springer-Verlag, New York, 1994, 40-49.

[12] Meyer, B., Müller, V. *A public key cryptosystem based on elliptic curves over* $\mathbb{Z}_n$ *equivalent to factoring.* Advances in Cryptology - EUROCRYPT '96, LNCS 1070, 1996, 49-59.

[13] Galbraith, S. *Elliptic Curve Paillier Schemes.* J. of Cryptology, 15, no. 2, 2002, 129-138.

[14] Holden, Joshua. *A Tour of Public Key Cryptography (and of Number Theory).* Rose-Hulman Institute of Technology, Lecture - March 17, 2001.

[15] Tazawa, S. *Integer Factorization and Discrete Logarithm problem are neither in P nor NP-complete.* University of Tokyo, Japan, July 12, 2012.