AUTHENTICATED AND SECURE El-GAMAL CRYPTOSYSTEM OVER ELLIPTIC CURVES

Malek Jakob Kakish

Amman Arab University,
Department of Computer Information Systems, P.O.Box 2234, Amman 11953, Jordan
Email: doctor_malek@yahoo.com; malek@aau.edu.jo

ABSTRACT

Information technologies plays a major role in our information society, thus it is important to protect it against many kinds of threats or attacks which may lead to lose of money, or lose of reputation and thus destroy businesses.

The El-Gamal public key cryptosystem over elliptic curves is often used in modern computer and communication technologies; it enables secure communication over public unsecure channels.

This paper introduce the security of the El-Gamal cryptosystem based on elliptic curves, it suggests a modification that can make the cryptosystem more immune against some attacks. This modification solves a weakness in the basic El-Gamal scheme by including the identity parameter of the sender in the encryption process, thus making the cryptosystem immune against man-in-the-middle attack and known *k* parameter attack. The modification described in this paper can in analogy be implemented on the El-Gamal cryptosystem over finite fields. Other important benefit is that the El-Gamal cryptosystem described here can easily be implemented and is very suitable on small and limited devices (e.g. smart cards) due the use of elliptic curves.

This paper also briefly describe some attacks on the El-Gamal cryptosystem and the suitable choice of El-Gamal cryptosystem parameter settings to avoid such attacks.

Keywords: *El-Gamal cryptosystem, elliptic curves, public key cryptosystems, crypto-analysis, finite fields*.

1. INTRODUCTION

The modern technologies we use today are built on security technologies to protect information and systems from attacks and threads and to fulfill our needs to meet the security needs and requirements.

Many of such security systems (e.g. security protocols, algorithms or applications) have been developed are based on standards; such standards are mostly specified from well known standard organizations (e.g. Internet Architecture Board (IAB), Internet Engineering Task Force (IETF), etc.) to ensure compatibility and coordinated work in the development of such systems and to support our security needs on information technology.

A major science that underlies many of the security mechanism is Cryptography (the science of data encryption and decryption). Cryptography [1] enables users to securely store sensitive information or transmit it across insecure networks such that it cannot be read by anyone except the intended recipient. By using such a powerful tool as encryption we gain privacy, authenticity, integrity, and limited access to data.

In Cryptography we differentiate between private key cryptographic systems (also known as conventional cryptography systems) and public key cryptographic systems. Private Key Cryptography, also known as secret-key or symmetric-key encryption, has an old history [1], and is based on using one key for encryption and decryption. In the 1960s many modern private key cryptographic systems where developed that are based on Feistel cipher, e.g. Data Encryption Standard (DES), Triple Data Encryption standards (3DES), Advanced Encryption Standard (AES), The International Data Encryption Algorithm (IDEA), Blowfish, RC5, CAST, etc.

In 1976 Diffie and Hellman [2] published a paper that describes a new concept which was called Public Key Cryptography and is based on using two keys (public and private key). This new concept solved many weaknesses and problems (e.g. key exchange problem) in private key cryptography, since then many public key cryptographic systems were invented (e.g. RSA [4], ElGamal [3], Diffie-Hellman key exchange [2], elliptic curves [5] [6], etc.). The security of such Public key cryptosystems is based on apparently difficulties of some mathematical number theory problems (“also called one way functions”) e.g. the discrete logarithm problem over finite fields and over elliptic curves, the integer factorization problem, the Diffie-Hellman Problem, etc. For more information about Cryptography histories see [1].

The El-Gamal cryptosystem is often used in praxis, such cryptosystem is often an essential part of a whole security system, e.g. El-Gamal is used in Internet security standard protocols Virtual Private Network (VPN), Internet protocol security IPSEC [8], Pretty Good Privacy (PGP), Socket Secure Layer (SSL) to secure transmitted data through public networks and is mostly used in web and email communication systems today. The ElGamal cryptographic algorithm is comparable to the Diffie-Hellman system. Although the inventor, Taher Elgamal [3], did not apply for a patent on his invention, the owners of the Diffie-Hellman patent (US patent 4,200,770) felt this system was covered by their patent.

Due to the widely use of the El-Gamal cryptosystem is it critical to ensure a high level of security, in this paper I introduce an El-Gamal cryptosystem over elliptic curves that is more secure compared with the basic El-Gamal cryptosystem this will make it more difficult for an attacker or cryptanalysis people to break the El-Gamal cryptosystem.

2. PROBLEM FORMULATION

The security of many in the praxis used cryptosystems and protocols are based on the discrete logarithm problem on finite fields and over elliptic curves, this means that if in the future the discrete logarithm problem is efficiently solved then many of these security systems will no longer be secure.

The El-Gamal cryptosystem described in this paper use Z_q^* groups of prime order q (where $q=p^f$, p prime number) over elliptic curves, but the El-Gamal cryptosystem can also be considered on groups of composite integers n [9], in praxis $F^*{}_2{}^m$ are often used because fields elements are binary string containing 0 and 1 and this ensure no waste of bits.

Let p be a prime number, then Z_p denotes the set of integers $\{0, 1, 2, \ldots, p-1\}$, where addition and multiplication are performed modulo p.

DEFINITION: A *Field* is a non empty set F of elements with two operations "+" (called addition) and " $\cdot$" (called multiplication) satisfying the following axioms: for all $a, b, c \in F$,

i. F is closed under + and $\cdot$, i.e., $a + b$ and $a \cdot b$ are in F;

ii. Commutative laws: $a + b = b + a, a \cdot b = b \cdot a$;

iii. Associative laws: $(a + b) + c = a + (b + c),\ a \cdot (b \cdot c) = (a \cdot b) \cdot c;$

iv. Distributive law: $a \cdot (b + c) = a \cdot b + a \cdot c$.

Furthermore, two distinct identity elements 0 and 1 (called the additive and multiplicative identities, respectively) must exist in F satisfying:

v. $a + 0 = a$ for all $a \in F$;

vi. $a \cdot 1 = a$ and $a \cdot 0 = 0$ for all $a \in F$;

vii. $\forall\, a$ in F, there exists an additive inverse element $(-a)$ in F such that $a + (-a) = 0$;

viii. $\forall\, a \neq 0$ in F, there exists a multiplicative inverse element a^{-1} in F such that $a \cdot a^{-1} =1$

DEFINITION: A finite field of prime order p or prime power $q = p^f$ $(f >=1)$ is commonly denoted F_q or $GF(q)$ (Galois field) and because Z_m is a field if and only if m is a prime, we denote the field Z_m by F_m. This is called a *prime field*.

DEFINITION: For $n \geq 1$, let $\varphi(n)$ denote the number of integers in the interval $[1,n]$ which are relatively prime to n. The function φ is called the Euler phi function (or the *Euler totient function*)

DEFINITION: Let $\alpha \in Z_p^*$ .If the order of α is $\varphi(n)$, then α is said to be a *generator* or a *primitive element* of Z_p^*. If Z_p^* has a generator, then Z_p^* is said to be *cyclic*.

DEFINITION: An *Elliptic Curve E* consists of the set of points (X, Y, Z) that satisfy the following homogeneous Weierstrass equation:

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^{\,2}Z + a_4XZ^2 + a_6Z^3$$

Where a_i $(i=1, 2, 3, 4, 5, 6)$ are elements of a field F and with the exception that the triple $(0 ,0, 0)$ is not a point on E.

The field F can be set C (complex numbers), R (real) or Q (rational) or any other finite field F_q we wish. The advantage of using R over E is that we can analyse arithmetic calculation in that field geometrically, this will help use to understand the arithmetic of elliptic curves and why they call them elliptic curves.

If we set $Z= 0$ and substitute $x = X/Z, y = Y/Z$ then we gets the equation:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

The above equation is called the affine Weierstrass equation. If a point *P* satisfy the homogeneous Weierstrass equation and the equation:

$$\frac{\partial F}{\partial X} = \frac{\partial F}{\partial Y} = \frac{\partial F}{\partial Z} = 0$$

Then we call that point singular and we call the Weierstrass equation also singular, note that singular Weierstrass equations are not of interest in the cryptography.

We need now criteria that can help us to determine if a given affine Weierstrass equation singular is or not. The discreminante Δ (field element) is such a tool, which can be defined as follow:

$$\begin{aligned}
&d_2 = a_1^2 + 4a_2 \\
&d_4 = 2a_4 + a_1a_3 \\
&d_6 = a_3^2 \\
&d_8 = a_1^2a_5 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2 \\
&c_4 = d_2^2 - 24d_4 \\
&\Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6 \\
&j(E) = c_4^3 / \Delta
\end{aligned}$$

If Δ = 0, then affine Weierstrass equation is singular, otherwise not singular [5] [6]. We call *j(E)* the *j-invariant* of the elliptic curve *E*. Note that only elliptic curves *E* over finite fields are of interest in cryptography.

DEFINITION: The *characteristic* of a field *F*, often denoted *char(F),* is the smallest positive number *n* such that:

$$\underbrace{1 + 1 + \ldots + 1}_{n} = 0$$

The field is said to have the characteristic zero if this repeated sum never reaches the additive identity.

THE ARITHMETIC OF *E(F$_q$)* GROUPS

The following table shows different finite fields and the corresponding elliptic curve equation classified by the characteristic.

Characteristic of *F*	**Elliptic curve equation *E(x, y)***
1.Char(F) = 2, j(E) ≠ 0	*E(x, y):* $y^2 + xy = x^3 + a_2x^2 + a_6$
2.Char(F) = 2, j(E) = 0	*E(x, y):* $y^2 + a_3y = x^3 + a_4x + a_6$
3.Char(F) = 3, j(E) ≠ 0	*E(x, y):* $y^2 = x^3 + a_2x^2 + a_6$
4.Char(F) = 3, j(E) = 0	*E(x, y):* $y^2 = x^3 + a_4x + a_6$
5.Char(F) > 3	*E(x, y):* $y^2 = x^3 + a_4x + a_6$

Table: **elliptic curves equations and corresponding characteristics**

Such an elliptic curve over a finite field *F* is a plane curve which consists of points that satisfy the elliptic curve equation *E* along with a distinguished point at infinity, denoted *O*. This set together with the "addition" rules as group operation form an *Abelian group*, with the identity element *O*.

DEFINITION: The *discrete logarithm problem* over the elliptic curve *E* is the following: given two points *P* and *Q* in a group that satisfy E, find a number *x* such that *xP = Q*; *x* is called the discrete logarithm of *Q* to the base *P*.

DEFINITION: The *base point C* is also referred to as the generator (often denoted |*E*|, *#E,* and *#E(F$_p$)* in the literature) or subgroup generator (Subgroup order is *#E/h*).

The definition of group of points over elliptic curve *E*:

1. There is a point $O \in E$, such that for all $P \in E$, $P + O = O + P = P$.
2. $–O = O$ (the identity of the group).
3. If $P \neq O$ and $P=(x_1,y_1)$ then $–P$ is $(x_1,-y_1-a_1x_1-a_3)$.
4. If two points on E have same x-coordinate then either $P=Q$ or $P=-Q$.
5. If $Q = -P$, then $P + Q = O$.
6. For two points $P \neq O$ and $Q \neq O$ on E, the addition is defined as follows. Draw the line through P and Q to intersect the curve in a third point; then reflect that point in the x-axis.
7. For two points $P \neq O$ and $Q \neq O$ on E, if $P = Q$, use the tangent line at P. The identity of the group is O, the "point at infinity", which conceptually lies at the top and bottom of every vertical line.

The following figure shows the addition of two points over the elliptic curve E:

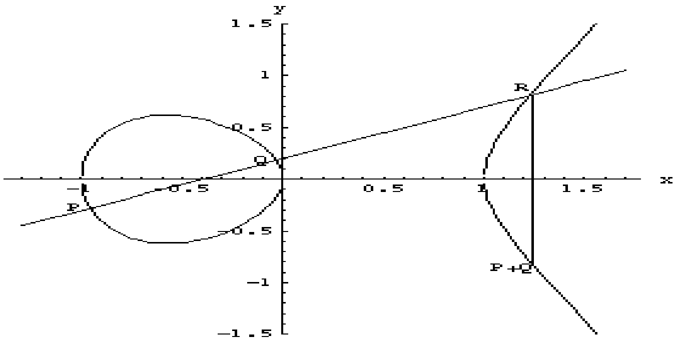

Figure: Addition of two points over elliptic curve E

For more information about elliptic curves in cryptography see [5] [6].

Before we encrypt a text message, we must decode that message into series of numbers that contains the message; here we could use some of the well known codes e.g. ASCII, Unicode.

The process of coding begins with substituting each character in the message with its corresponding numerical code z:

$$0 \leq z \leq a\text{-}1$$

Where a is the number of distinct characters in the selected code.

Next we have to select s such that:

$$a^s < p < a^{s+1}$$

Now we divide our message in blocks of s length, and for each of these blocks we build the following sum:

$$M = \sum_{i=0}^{z-1} z_i a^i$$

It is simple to check that $M < p$, we call M the coding block, it is also an element in F_q.

We now describe the basic version of the El-Gamal cryptosystem that allows communicating parties that have never met before to exchange encrypted messages over unsecure networks.

ALGORITM: The El-Gamal cryptosystem (basic version):
KNOWN: Base point C, user A public key x_AC and user B public key x_BC.
RESULT: User B encrypts a message m for user A, which A decrypts.

1. ENCRYPTION. User B should do the following:
(a) Choose a random integer k, where $1<k<q-1$ and keep it secret
(b) Compute kC
(c) Obtain user A authentic public key $P_A= x_AC$.

(d) Compute kx_AC
(e) Represent the message as an integer m in the interval $[0,p-2]$, m is a point over E.
(f) Compute $c = m + kx_AC$
(g) Send the encrypted message (kC, c) to user A.

2. DECRYPTION. To recover plaintext m from c, user A should do the following:
(a) Multiply own private key x_A with first part of the message (kC), we get: kx_AC.
(b) Use kx_AC and compute the inverse element $-kx_AC$
(c) Recover m: Add $–kx_AC$ to the second part of the message: $–kx_AC + m + kx_AC = m$

The encryption in the above described basic version of the El-Gamal cryptosystem does not contain sender authentication parameter, thus any user X can sent this message and claim to be user B (man-in-the-middle attack), User A can’t be sure if a message m really comes from user B and not from someone else, he doesn’t have any tools to check this. Such problem can be solved if both parties have access to a trusted third party that issues certifications which binds their identity with the corresponding public key or if they use a public distribution of parameter over trusted channels. A list of attacks on El-Gamal cryptosystem can be found in [7].

The security of the El-Gamal cryptosystem is based on the intractability of solving the discrete logarithm problem and the Diffie-Hellman problem; this can be shown in the equation:

$$kx_AC$$

If we can calculate k or x_A in this equation then we can recover message m in the equation:

$$m = –kx_AC + m + kx_AC$$

Here arises the problem than an attacker can recover message m even if he only knows k, thus if k is small chosen it will be an easy task to determine it, this must be taken into consideration when we use El-Gamal cryptosystem.
The actual purpose of k parameter in the encryption process is to produce different encrypted data blocks even if the data blocks contains same information, this will make it more difficult for the crypto-analyst to distinguish between encrypted data blocks.
One potential disadvantage of the ElGamal system is that message expansion by a factor of two takes place during encryption. However, such message expansion is negligible if the cryptosystem is used only for exchange of small messages or secret keys. In praxis public key systems are mostly used to encrypt small amount of data (e.g. symmetric keys, hash values) because they are slow compared with symmetric cryptosystems.

3. AUTHENTICATED EL-GAMAL CRYPTOSYSTEM OVER ELLIPTIC CURVES

The following described authenticated version of El-Gamal cryptosystem provides a solution for the sender authentication problem and the known k parameter in the El-Gamal cryptosystem basic version.

For the below described examples we shall assume the following:

1. Arithmetic operation over F_q, where $char(F_q) > 3$

- Addition of two equal points ($P(x_1,y_1)+P(x_1,y_1) = 2P$): the (x_3, y_3) coordination of the $2P$ point are

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1$$

$$y_3 = -y_1 + \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3)$$

- Addition of two distinct points ($P(x_1,y_1)+Q(x_2,y_2)=Z(x_3,y_3)$, where $P≠Q$): the (x_3, y_3) coordination of the Z point are

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2$$

$$y_3 = -y_1 + \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3)$$

2. We now consider an elliptic curve over the field F_{23}, where the elliptic curve equation $E\!: y^2 = x^3+ax+ b$, if we set $a = 1$ and $b = 0$, then we get the elliptic curve $E\!: y^2 = x^3 + x$. This equation must satisfy the equation $4a^3+27b^2 \neq 0 \bmod p$ to form a group, this is verified. The following 23 points over E that satisfies this equation are:

(0,0) (1,5) (1,18) (9,5) (9,18) (11,10) (11,13) (13,5) (13,18) (15,3) (15,20) (16,8) (16,15) (17,10) (17,13) (18,10) (18,13) (19,1) (19,22) (20,4) (20,19) (21,6) (21,17)

The following algorithm describes the authenticated El-Gamal cryptosystem.

ALGORITM: The El-Gamal cryptosystem (authenticated version):
KNOWN: Elliptic curve E, Base point C, user A public key x_AC and user B public key x_BC.
RESULT: User B encrypts a message m for user A, which A decrypts.

1. ENCRYPTION. User B should do the following:
(a) Choose a random integer k, where $1<k<q-1$ and keep it secret
(b) Compute: kC
(c) Obtain user A authentic public key $P_A= x_AC$.
(d) Compute: kx_AC
(e) Compute: x_Bx_AC
(f) Represent the message m as an integer in the interval $[0,p-2]$ and as a point over E.
(g) Compute: $c = m + kx_AC + x_Bx_AC$
(h) Send the encrypted message (kC, c) to user A.

2. DECRYPTION. To recover plaintext m from c, user A should do the following:
(a) Obtain user B authentic key x_BC
(b) Compute: x_Ax_BC
(c) Compute inverse element: $-x_Ax_BC$
(d) Add $-x_Ax_BC$ to the second part of the message: $m+ kx_AC + x_Bx_AC - x_Ax_BC= m + kx_AC$
(e) Multiple own private key x_A with first part of the message (kC), we get: x_AkC.
(f) Use x_AkC and compute the inverse element $-x_AkC$
(g) Recover m: Add $–kx_AC$ to the second part of the message: $–x_AkC + m + kx_AC = m$

In the following example we assume that the message $m = 16$, and that one can use a simple coding algorithm to transform m into a point on the elliptic curve E.

EXAMPLE: El-Gamal Cryptosystem (authenticated version)
KNOWN: Elliptic curve $E\!: y^2 = x^3 + x$, $C = (19, 22)$, $P_A= (20, 4)$, $P_B = (11, 10)$.
RESULT: User B encrypts a message $m= 16$ for user A, which A decrypts.

1. ENCRYPTION. User B should do the following:
(a) Choose a random integer: $k = 9$
(b) Compute: $kC = 9(19,22) = (15, 3)$
(c) Obtain user A authentic public key $P_A = x_AC = 3(19, 22) = (20, 4)$
(d) Compute: $kx_AC = kP_A = 9(20, 4) = (20, 4)$
(e) Compute: $x_Bx_AC = 5(20, 4) = (15, 20)$
(f) Represent the message m as a point over $E\!: m = (16, 15)$
(g) Compute: $c = m + kx_AC +x_Bx_AC = (16,15)+(20,4)+(15,20)=(18, 13)$
(h) Send the encrypted message $(kC, c) = ((15,3), (18,13))$ to user A.

2. DECRYPTION. To recover plaintext m from c, user A should do the following:
(a) Obtain user B authentic key $x_BC= 5(19, 22) = (11,10)$
(b) Compute: $x_Ax_BC= 3(11, 10)=(15, 20)$
(c) Compute inverse element: $-x_Ax_BC$ = (15, 3)
(d) Compute: $c - x_Ax_BC= (18, 13) + (15, 3) = (19, 22)$
(e) Compute: $x_AkC = 3(15, 3) = (20, 4)$.
(f) Compute inverse element: $-x_AkC = (20, 19)$
(g) Recover m: $–x_AkC + m + kx_AC = (20,19) + (19,22) = (16, 15)$

Below we briefly describe some of the attacks on El-Gamal cryptosystem:

1. Known *k*-parameter attack
The El-Gamal basic version is vulnerable against known *k* parameter, this is because in the equation:

$$c = m + kx_AC$$

Anyone who knows *k* can easily recover the message *m* even if he doesn't know the private key x_A this is due to public key x_AC.

2. Man-In-the-middle attack
The second problem is that *c* doesn't contain any information that can be used to authenticate the sender (man-in-the-middle attack).
The El-Gamal cryptosystem (authenticated version) described above is immune against such kind of attacks because in the equation:

$$c = m + kx_AC + x_Bx_AC$$

Even if an attacker knows *k* he won't be able to recover *m*, but only if he knows the private key x_A, he will be able to recover *m*, thus the private key x_B must be used to compute *c*, this will ensure that the message really comes from user B and not from anybody else.

3. Chosen cipher-text attack
ElGamal encryption is unconditionally malleable (given an encryption of a plaintext m, it is possible to generate another cipher-text which decrypts to *f(m)*, for a known function *f*, without necessarily knowing or learning *m*), and therefore is not secure under chosen cipher-text attack. For example, given an encryption (c_1, c_2) of some (possibly unknown) message *m*, one can easily construct a valid encryption (c_1,$2c_2$) of the message *2m.*

4. Key size attacks
Important for El-Gamal cryptosystem regarding security consideration is the size of the modulus *p* where *p-1* prime factors should be so selected sufficiently large such that factoring is computationally infeasible [16].

Here it is important to mention that the discrete logarithm problem over elliptic curve with small keys is much harder to solve than the discrete logarithm over finite fields, one of the reasons is that not all attacks on finite fields groups can be applied on groups over elliptic curves. This implies that key size for group over elliptic curves can be much smaller that key size for groups on finite fields. The following table shows cryptographic key size recommendation according to the National Institute of Standards and Technology (NIST):

Date	**Finite fields [bits]**	**Elliptic curves [bits]**	**Security**
2007-2010	1024	160	short term
2011-2030	2048	224	middle term
>2030	3072	256	
>>2030	7680	384	long term
>>>2030	15360	512	

Table: NIST recommendations for key size in bits for equivalent levels

5. Attacks on implementation
Some of the attacks on the El-Gamal cryptosystem and RSA cryptosystem implementations can be found in [14].

6. Attacks on the discrete logarithm problem and integer factorization problem
The discrete logarithm problem has been studied for many years but still an efficient solution was not found thus it is considered as being difficult if the parameters are suitably chosen, but if the factors of *p-1* are known or are small integers then the discrete logarithm problem can be easily solved using Pohlig-Hellman Algorithm[11], Pollard rho method [12], number field sieve for discrete logarithms Lenstra and Lenstra [15], and for the integer factorization problem the Index Calculus method [10], the elliptic curve factoring algorithm [17], quadratic sieve [18] and number field sieve [19] for more information see [7].

In 2010, the largest number factored by a general-purpose factoring algorithm was 768 bits long [20] [21] using distributed implementation thus some experts believe that 1024-bit keys may become breakable in the near future so it is currently recommended to use 4096-bit keys for long term security this requires about 270 bits using elliptic curve encryption.

4. CONCLUSION

This paper briefly discussed an authenticated version of the El-Gamal cryptosystem, which use sender private key to authenticate the identity of the sender; this will make the El-Gamal cryptosystem more secure against known *k*-parameter attack and man-in-the-middle attack.

These security improvements are important because El-Gamal cryptosystem is implemented in many internet security standards and protocol, and a weak El-Gamal cryptosystem can make the whole security system compromised.

It should also be mentioned that the El-Gamal cryptosystem over elliptic curve is suitable to be implemented on many small devices (e.g. smart card) where limited processing power and limited memory capacity exist, this is due to the small number of bits required to perform the encryption and decryption process, elliptic curves are considered newly in cryptography and is one of the most researched topic in cryptography.

5. REFERENCES

[1]. D. Kahn, The Code breakers: The comprehensive History of Secret Communication from Ancient to the Internet, Published 1967

[2]. W. Diffie and M. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, 22 (1976) 644-654.

[3]. T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, volume 31, pages 469-472, 1985.

[4]. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. Commun. of the ACM, 21:120-126, 1978.

[5]. N. Koblitz. Elliptic curve cryptosystems. Mathematics of Computation, 48:203-209, 1987

[6]. Menezes Alfred, "Elliptic Curve Public Key Cryptosystem", 1993, 4.eddition, 1997. Kluwer Academic Publishers.

[7]. A. Menezes, P. van Oorscot and S. Vanstone, Handbook of Applied Cryptography, CRC Press, ISBN: 0-8493-8523-7, 1999

[8]. Thayer, R.; Doraswamy, N.; Glenn, R. (November 1998). IP Security Document Roadmap. IETF. RFC 2411. http://tools.ietf.org/html/rfc2411

[9]. E.F. Brickell, K.S. McCurley, An interactive identification scheme based on discrete logarithms and factoring, 5 (1992), 29–39. Journal of Cryptology papers (Volume 1 No.1 – Volume 9 No.3, 1988-1996)

[10]. M. Kraitchik, *Théorie des nombres*, Gauthier--Villards, 1922

[11]. S. Pohlig and M. Hellman, "An improved algorithm for computing logarithm over GF(p) and its cryptographic significance", IEEE Transaction on Information Theory, volume 1462, Springer-Verlag, pages 458-471, 1998

[12]. Pollard J. M., „Monte Carlo Methods for Index Computation (mod p)", Mathematics of Computation 32, 1978, 918-924.

[13]. Ian F. Blake and Theo Garefalakis. On the complexity of the discrete logarithm and diffie- hellman problems. J. Complex., 20(2-3):148{170, 2004}.

[14]. Dan Boneh, Antoine Joux, and Phong Q. Nguyen. Why textbook ElGamal and RSA encryption are insecure. In ASIACRYPT '00: Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security, pages 30{43, London, UK, 2000. Springer-Verlag.

[15]. A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, and J. M. Pollard, "The Number Field Sieve," in A. K. Lenstra and H. W. Lenstra, Jr. (eds.) The Development of the Number Field Sieve, Lecture Notes in Mathematics 1554, Springer-Verlag, New York, pp. 11–42, 1993

[16]. Arjen K. Lenstra and Eric R. Verheul. Selecting cryptographic key sizes. Journal of Cryptology: the journal of the International Association for Cryptologic Research, 14(4):255{293, 2001.

[17]. B. Dixon, A.K. Lenstra, Massively parallel elliptic curve factoring, 183–193.

[18]. C. Pomerance, The quadratic sieve factoring algorithm, 169–182.

[19]. J. Buchmann, J. Loho, J. Zayer, An implementation of the general number field sieve, 159–165

[20]. RSA Laboratories, the RSA Factoring Challenge http://www.rsa.com/rsalabs/node.asp?id=2092

[21]. P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. 26 (1997), 1484–1509. Available at http://www.research.att.com/shor.