# A COMPARATIVE STUDY OF ELGAMAL BASED DIGITAL SIGNATURE ALGORITHMS

**RAMZI A. HARATY, Lebanese American University, Lebanon, rharaty@lau.edu.lb**
**A. N. EL-KASSAR, Beirut Arab University, Lebanon, ak1@bau.edu.lb**
**BILAL M. SHEBARO, Lebanese American University, Lebanon, bs990645@lau.edu.lb**

## ABSTRACT

A powerful and practical public-key and digital signature scheme was produced by ElGamal. ElGamal public-key and digital signature scheme were modified from the domain of natural integers, **Z**, to the domains of Gaussian integers, **Z**[$i$], and polynomials over finite fields, $F[x]$. We implement the classical and modified ElGamal digital signature scheme to compare and to test their functionality, reliability and security. To test the security of the algorithms we use a famous attack algorithm called Baby-Step-Giant algorithm which works in the domain of natural integers. We enhance the Baby-Step-Giant algorithm to work with the modified ElGamal digital signature algorithms.

**KEYWORDS**: ElGamal digital signature, testing and evaluation.

## 1. INTRODUCTION

The concept of a digital signature was introduced in 1976 by Diffie and Hellman. One of the powerful and practical signature schemes was produced by ElGamal [3] in 1985. El-Kassar et al. [4] and El-Kassar and Haraty [5] modified the ElGamal signature schemes from the domain of natural integers, **Z**, to two principal ideal domains, namely the domain of Gaussian integers, **Z**[$i$]=$\{a+bi \mid a, b \in \mathbf{Z}, i = \sqrt{-1}\}$, and the domain of polynomials over finite fields, $F[x]$, by extending the arithmetic needed for the modifications to these domains. In both cases, it was shown that the same prime modulus used in the classical ElGamal scheme can be used in the new settings to produce larger cyclic groups; hence, the message space, the key space and signature set are enlarged without any additional effort. The larger key space makes the new schemes more secure and harder to break. Moreover, it was shown in both cases that the arithmetic is easy and efficient to apply.

In this paper, we compare and evaluate the classical and modified ElGamal algorithms by implementing and running them on a computer. We investigate the issues of complexity, efficiency and reliability by running the programs with different sets of data. Moreover, comparisons will be done among these different algorithms given the same data as input. In addition, implementation of an attack algorithm will be presented. The attack algorithm consists of subroutines used to verify the signed messages. This is done by applying certain mathematical concepts to find the private key of the signed message. After finding the key, it will be easy to sign the message. A study will be done using the results of running the attack algorithm to compare the security of the different classical and modified signature scheme algorithms. A similar comparison and evaluation was done for the RSA-Based Digital Signature Algorithms [9].

The rest of the paper is organized as follows: section 2 describes the classical technique of ElGamal signature scheme, which depends on the discrete logarithm problem. Then, we present the modifications done on ElGamal signature scheme. In section 3, we deal with the testing procedures and the attack algorithms to evaluate the classical and modified algorithms. Also,

attack programs are run to test the complexity, efficiency and reliability of the different modified algorithms and compare them to the classical one. A conclusion is drawn in section 4.

## 2. CLASSICAL AND MODIFIED ELGAMAL SIGNATURE

### 2.1 Classical ElGamal Signature Scheme

The classical ElGamal signature can be described as follow. Let $p$ be a large odd prime integer and let $\mathbf{Z}_p = \{0,1,2, \ldots, p-1\}$. Then, $\mathbf{Z}_p$ is a ring under addition and multiplication modulo $p$. Since $p$ is prime, $\mathbf{Z}_p$ is actually a field under these operations. Moreover, the multiplicative group of the ring of integers modulo $p$, $\mathbf{Z}^*_p = \{1,2,\ldots, p-1\}$ is a cyclic group generated by some generator $\theta \neq 1$, whose multiplicative order is $p-1$. That is, every element of $\mathbf{Z}^*_p$ is a power of $\theta$. We note that $\mathbf{Z}_p$ is a complete residue system modulo $p$, and $\mathbf{Z}^*_p$ is a reduced residue system modulo $p$. We note that a composite integer $n = 2p^t$ can be used instead of the prime $p$.

The key is generated by selecting a large random prime $p$ and a generator $\theta$ of the multiplicative group $\mathbf{Z}^*_p$. Then we choose randomly an integer $a$, $1 \leq a \leq p-2$, and compute $\theta^a \pmod p$. The public key is $(p, \theta, \theta^a)$ and the private key is $a$.

The signature is generated as follow. First, hash the message $m$ to obtain the hash value $f = h(m)$, where $h$ is a hash function and $m$ is a binary message of arbitrary length. Generate a random secret integer $k$ such that $1 \leq k \leq p-2$ with $\gcd(k, p-1) = 1$. Compute $r = \theta^k \pmod p$ and compute $k^{-1} \bmod (p-1)$. Then, compute $s = k^{-1}\{h(m) - ar\} \bmod (p-1)$. Entity $A$ then sends the signature $m$ and the signature $(r, s)$ to B.

The signature is validated as follow. Obtain $A$'s authentic public key $(p, \theta, \theta^a)$ and verify that $1 \leq r \leq p-1$. Hash the message $m$ and obtain the hash value $f = h(m)$. Then, compute $v_1 = y^r r^s \pmod p$ and $v_2 = \theta^{h(m)} \pmod p$. Accept the signature if $v_1 = v_2$.

### 2.2 ElGamal Signature Scheme in the Domain of Gaussian Integers

For a Gaussian integer $\gamma = a+bi$, let $\delta(\gamma) = a^2+b^2$ be the norm of $\gamma$. Two elements $\alpha$ and $\beta$ in $\mathbf{Z}[i]$ are called associates, denoted by $\alpha \sim \beta$, iff $\alpha = \pm\beta, \pm i\beta$. The Gaussian primes of $\mathbf{Z}[i]$, up to associates, see [7], are of the form: i) $\alpha = 1+i$; ii) $\pi$ and $\bar{\pi}$, where $q = \pi\bar{\pi}$ is a prime of the form $4k+1$; iii) prime $p$ of the form $4k+3$. The domain of Gaussian integers is a factorization domain in which every nonzero element $\gamma$ can be expressed as a product of primes. Let $\eta \in \mathbf{Z}[i]$. and let $G_\eta$ be a complete residue system modulo $\eta$. We define the function $q(\eta)$ to be the number of element in $G_\eta$. For any two elements $\beta$ and $\gamma$ in $\mathbf{Z}[i]$, it is true that $q(\beta\gamma) = q(\beta)q(\gamma)$; see [1]. In [1], Cross gave a full description of the complete residue systems modulo prime powers of Gaussian integers. In particular, when $p$ is a Gaussian prime of the form $4k+3$ and $\pi$ is a factor the odd prime $q = \pi\bar{\pi}$, where $q$ is a prime integer of the form $4k+1$, we have

$$G_\pi = \{a \mid 0 \leq a \leq q-1\}$$

and

$$G_p = \{a+bi \mid 0 \leq a \leq p-1,\ 0 \leq b \leq p-1\}.$$

For a Gaussian integer $\beta$, let $G^*_\beta$ be the elements of $G_\beta$ that are relatively prime to $\beta$, i.e., $\gamma \in G^*_\beta$ iff $\gamma \in G_\beta$ and $\gcd(\gamma,\beta) = 1$. The set $G^*_\beta$ is called a reduced residue system modulo $\beta$. Also, $G^*_\beta$ is the group of units of $G_\beta$. When $\beta$ is a Gaussian prime, $G_\beta$ is a field and $G^*_\beta$ is the set of nonzero elements. The number of elements in any reduced residue system $G^*_\beta$, which is the order of the group of units of $G_\beta$, is constant and is denoted by $\phi(\beta)$. Note that $\phi(\beta)$ is an extension of Euler's phi-function. The value of $\phi(\beta)$ is obtained by using the fact that $\phi(\beta)$ is a multiplicative function and that the value of $\phi(\beta)$, when $\beta$ is a prime power, see [1] or [3], is given by: i) $\phi(\pi^n) = q^{n-1}(q-1)$; ii) $\phi(p^n) = p^n - p^{2n-2} = p^{2n-2}(p^2-1)$; iii) $\phi(\alpha^n) = 2^n - 2^{n-1}$.

A Gaussian integer $\beta$ is said to have a primitive root iff $G^*_\beta$ is cyclic. In such case, any generator $\theta$ of the cyclic group $G^*_\beta$ is called a primitive root of $\beta$. J.T.Cross [1] showed that a

Gaussian integer $\beta$ has a primitive root iff $\beta$ is $1+i$ , $(1+i)^2$, $(1+i)^3$, $\pi^n$, $p$, $\alpha\pi^n$ or $\alpha p$. For a Gaussian prime $\beta$, one can find a generator of $G^*_\beta$ by randomly choosing an element $\theta$ in $G^*_\beta$ and computes $\phi(\beta) = q(\beta)-1 = p_1^{\alpha_1}.p_2^{\alpha_2}...p_k^{\alpha_k}$ . Then $\theta$ is a generator if

$$\theta^{\frac{\phi(\beta)}{p_j}} \not\equiv 1 \pmod{\beta}$$

for all primes $p_j$ dividing $\phi(\beta)$. Otherwise, choose another value for $\theta$. The process is repeated until a generator is found.

Arithmetics in the domain of Gaussian integers can be applied to extend ElGamal signature as follows. First, a Gaussian prime $\beta$ has to be chosen. Note that if $\beta$ is a prime of the form $\pi$, where $\pi\bar{\pi} = p = 4k + 1$, then $G_\beta = G_\pi = \{0, 1, \ldots, p-1\} = \mathbf{Z}_p$. The calculations in this case are identical to ElGamal signature in the domain of integers. Thus, we choose a large Gaussian prime $\beta$ of the form $4k+3$. Note that the number of elements in $G_\beta$ is $q(\beta) = p^2$ and hence $\phi(\beta) = p^2-1$. Thus, the cyclic group used in the Extended ElGamal signature has an order larger than the square of the order of the cyclic group used in the Classical ElGamal signature. This larger setting is obtained with no additional efforts required for finding the prime $p$.

Now, a generator $\theta$ of $G^*_\beta$ is chosen as described previously. Note that there are $\phi(p^2-1)$ generators in $G^*_\beta$. A random positive integer $a$ is then chosen in order to obtain the public-key $(p, \theta, \theta^a)$. Since $a$ is a power of $\theta$, $a$ must be a positive integer less than the order of the group power $G^*_\beta$, which is $p^2 - 1$. This power $a$ is the private key.

To sign a message $m$, *first* hash the message to obtain the hash-value $f = h(m)$. Then select a random secret integer $k$ such that $1 \le k \le p^2 - 2$ with gcd $(k, p^2-1) = 1$ and compute $r = \theta^k \pmod{p}$. Note that $r$ is in $G_\beta = G_p = \{a+bi \mid 0 \le a \le p-1, 0 \le b \le p-1\}$. Next, compute $k^{-1} \bmod (p^2-1)$ and $s = k^{-1}\{h(m)-ak\} \bmod (p^2-1)$. Also, compute $\delta = r^a = y^k \pmod{\beta}$. Send the binary message $m$ and the signature $(r, s, \delta)$.

To validate the signature, obtain the authentic public key $(p, \theta, \theta^a)$ and verify that $r \in G^*_\beta$. Hash the message $m$ and obtain the hash value $f = h(m)$. Compute $v_1 = \delta r^s \pmod{\beta}$ and compute $v_2 = \theta^{h(m)} \pmod{\beta}$. Accept the signature only if $v_1 = v_2$.

## 2.3 ElGamal Signature over Quotient Rings of Polynomials over Finite Fields

The generalized ElGamal signature in the setting of a finite field $F_q$, where $q = p^n$ for an odd prime integer $p$ and a positive integer $n$, is based on working with the quotient ring $\mathbf{Z}_p[x]/\langle h(x)\rangle$, where $h(x)$ is an irreducible polynomial over $\mathbf{Z}_p[x]$. We extend the ElGamal signature to the setting of a finite field. It is well known that $\mathbf{Z}_p[x]/\langle h(x)\rangle$ is a field whose elements are the congruence classes modulo $h(x)$ of polynomials in $\mathbf{Z}_p[x]$ with degree less than $n$. We identify this field by the complete residue system $A(h(x)) = \{a_0+a_1x+...+ a_{n-1}x^{n-1} \mid a\text{'s} \in \mathbf{Z}_p\}$. Note that $\mathbf{Z}_p[x]/\langle h(x)\rangle$ is of order $p^n$ and its nonzero elements form a cyclic group denoted by $U(\mathbf{Z}_p[x]/\langle h(x)\rangle)$ whose order is $\varphi(h(x)) = p^n-1$. Let $\theta(x)$ be a generator of the cyclic group so that every element in $U(\mathbf{Z}_p[x]/\langle h(x)\rangle)$ is a power of $\theta(x)$.

The ElGamal public-key signature is also extended in the setting of the cyclic group of the finite quotient ring $\mathbf{Z}_p[x]/\langle f(x)\rangle$, where $f(x)$ is a reducible polynomial of degree $n$. El-Kassar et [4] obtained a characterization for which $U(\mathbf{Z}_p[x]/\langle f(x)\rangle)$ is cyclic. A particular case of interest is when $f(x) = h(x)^2$, where $h(x)$ is linear and $U(\mathbf{Z}_p[x])$ is cyclic and isomorphic to $\mathbf{Z}_{p-1} \times \mathbf{Z}_p$. Hence, we can say that the extension of the ElGamal scheme in this case applies to the group of units of the ring $\mathbf{Z}_p[x]/\langle x^2\rangle$, of order $\varphi(x^2) = p^2-p$. We note that $U(\mathbf{Z}_p[x]/\langle x^2\rangle) = \{c+dx \mid 1 \le c \le p-1, 0 \le d \le p-1\}$.

Entity A should select an odd prime $p$ and a polynomial $f(x)$ of degree $n$, $f(x)$ is irreducible or $f(x) = x^2$. Then A computes $\varphi(f(x))$ and finds a generator $\theta(x)$ of $U(\mathbf{Z}_p[x]/\langle f(x)\rangle)$. Next A selects a

random integer $a$, $1 \le a \le \varphi(f(x))-1$ and finds $y(x) = \theta(x)^a \pmod{f(x)}$. The public-key is $(p, f(x), \theta(x), y(x))$ and the private-key is $a$.

To generate a signature of a message, we represent the message as a polynomial $m(x)$ and select a random secret integer $k$, $1 \le k \le \varphi(f(x))-1$ such that $\gcd(k, \varphi(f(x))-1) = 1$. Compute $h(m(x))$, $r(x) = \theta(x)^k \pmod{f(x)}$, $k^{-1} \pmod{\varphi(f(x))-1}$, $s = k^{-1}\{h(m(x)) - a.k\} \pmod{\varphi(f(x))-1}$, and $\delta(x) = r(x)^a \pmod{f(x)}$. Send $(r(x), s, \delta(x))$.

To verify that the message $m(x)$, obtain the authentic public key $(p, f(x), \theta(x), y(x))$ and make sure that $r(x) \in U(\mathbf{Z}_p[x]/\langle f(x)\rangle)$, otherwise reject the signature. Compute $v_1(x) = \delta(x) r(x)^s \pmod{f(x)}$. Compute $h(m(x))$ and $v_2(x) = \theta(x)^{h(m(x))} \pmod{f(x)}$. Accept the signature only if $v_1(x) = v_2(x)$.

## 3. TESTING AND EVALUATION

### 3.1 ElGamal Based Digital Signature Algorithm

Using Mathematica 5.0 functions and an additional abstract algebra library, we have written programs for the following algorithms:

1.Classical ElGamal.
2.Classical ElGamal with $n$ of the form $2p^t$.
3.ElGamal with Gaussian numbers.
4.ElGamal with irreducible polynomials.
5.ElGamal with reducible polynomials.

The various procedures were compared as follows:

a-A total of 25 runs of the various algorithms were conducted. In each run, a 20-digit random prime integer $p$ of the form 4k+3 was generated.

b-The same prime $p$ was used for all algorithms.

c-For each method a public key was generated by finding a generator $\theta$, a random integer a, and computing $\theta^a$.

d-Using the public key $(\theta, \theta^a)$, the same message $m$ = 12345678 was signed by all algorithms to obtain the signature $(r, s, \delta)$.

e-The verification algorithms were then used to verify the signature.

f-All algorithms used the built-in Mathematica functions for modular arithmatic and for finding powers modulo an integer, Gaussian integer or a polynomial over $\mathbf{Z}_p$.

g-The running times of the algorithm (Key generation, signature, verification) for each method were recorded.

Note that the cyclic groups used, and their corresponding orders, are:

1.Classical ElGamal: $\mathbf{Z}_p^*$ of order p.
2.Classical ElGamal with $n$ of the form $2p^2$: $\mathbf{Z}_n^*$ of order $p^2-p$.
3.ElGamal with Gaussian integers: $G_p^*$ of order $p^2-1$.
4.ElGamal with reducible polynomials: $U(\mathbf{Z}_p[x]/\langle x^2\rangle)$ of order $p^2-p$.
5.ElGamal with irreducible polynomials: $U(\mathbf{Z}_p[x]/\langle x^2+ax+b\rangle)$ of order $p^2-1$.

Except for the classical ElGamal in the setting of the cyclic group $\mathbf{Z}_p^*$, all cyclic groups used have comparable sizes. Hence, we expect the algorithms in the first case to be much faster. A different prime $p$ having 40 digits could have been used for that case; but this would have been equivalent to case 2.

After running the programs, it was clear that these programs have applied the ElGamal signature scheme in the correct way. All the programs have generated a public and private key with different mathematical concepts. Then a message is signed using the signature scheme and is sent to a verification procedure which verify the signature. After running and comparing the programs without considering the time of generating the irreducible polynomial, we observe the following:

a- All programs are reliable; they can sign and verify any signature.

b- The complexity for each of the algorithms is $O(n^2)$.

c- The reducible polynomial signature scheme is reliable but took more time to generate a key and to sign a message. This does not mean that it is inefficient because it is more secure than the other algorithms.

d- The irreducible polynomial program in the setting $\mathbf{Z}_p[x]/\langle x^2+ax+b\rangle$ worked well but requires more time. The encryption and decryption execution time for the irreducible polynomial scheme is slightly more than that of the reducible case.

e-The key generation time for the irreducible polynomial scheme is considerably more than that of the other methods. This is due to the fact that an irreducible polynomial must be generated before a generator θ is found. On average, it took about 0.4 sec to generate a quadratic irreducible polynomial for a 20-digit prime number.

By comparing the average execution time of these algorithms including the time taken to find an irreducible polynomial in the key generating algorithm, we observe the following:

a- The time for generating the key depends on finding the generator θ and not the prime $p$. The time for generating a prime number is negligible. The average time need for generating a 100-digit (recommended size) random prime is approximately 0.1 sec.

b- It took more time to find the key in the case of polynomials. This will not be a problem if common system-wide parameters are used. In such a case, all entities may elect to use the same cyclic group $G$ and generator φ. Also, once a generator φ for a given prime $p$ is found, all other generators can be easily obtained.

c- The time needed to encrypt and to decrypt the message for the classical, modified $2p^t$ and Gaussian is better than the time needed for the polynomials. However, the time is very reasonable even for larger primes.

d- Overall, the Gaussian integers methods performed very well. The algorithms can be made more efficient by modifying Mathematica built-in functions to take advantage of the arithmetic modulo the Gaussian prime p of the form $4k+3$.

e- The key generation time in the case of Gaussian integers is less than that of the modified $2p^t$ method. This is due to the fact that the number of generators is $\varphi(p(p-1))$, is almost always more than that in $G_p^*$, which is $\varphi(p^2-1)$. In fact, among the first 200,000 primes, there are only 7 primes $p$ of the form $4k+3$ for which $\varphi(p^2-1)>\varphi(p(p-1))$.

f-The reducible polynomial method is little slower but provide more security. The irreducible polynomial method is not recommended since it is as secure as the reducible case but requires more time especially in finding the key.

### 3.2 Attack Algorithm

In order to attack any protocol that uses ElGamal signature scheme we have to solve the discrete logarithm problem. We enhanced the Exhaustive search and Baby-step giant-step algorithms to work with the modified algorithms.

To test the security of the algorithms, we implemented and applied the attack schemes to the classical and modified signature algorithms. The ElGamal algorithm using irreducible polynomials was not tested since the attack time would be equivalent to that of the reducible polynomial case. For the exhaustive search algorithm, a random 3-digit prime $p$ of the form $4k+3$ was generated and a public key was obtained for each the four methods using the same prime.

Attacking the ElGamal schemes using Baby-step giant-step algorithm, a random 4-digit prime $p$ of the form $4k+3$ was generated and a public key was obtained for each the four methods using the same prime. After running these attack algorithms, we observed the following:

a- All the attack programs are reliable so they can forge any message by finding the private key.

b- The $2p^t$ algorithm is stronger than the classical algorithm because we have an unknown power t.

c- The Gaussian algorithm is stronger than the classical algorithm. The attack algorithm for Gaussian integers required more time than that of the $2p^2$ algorithm.

d- The most difficult algorithm to attack is in the polynomial domain. This is due to the fact that mathematically it is complex and needs considerable computing time to perform arithmetic modulo a given polynomial.

## 4. CONCLUSIONS

In this work, we presented the classic ElGamal signature scheme and four modifications. We implemented these algorithms and tested their efficiency, reliability, and security. The results obtained showed that all the algorithms applied the ElGamal signature scheme correctly and generated public and private key using different mathematical concepts. Messages were then signed using the signature scheme and were sent to a verification procedure which verifies the signature.

We also built attack scenarios directly aimed at solving the discrete logarithm problem that these algorithms utilize. We modified the Baby-step Giant-step algorithm to handle the modified algorithms. We observed that the classical ElGamal scheme is the weakest to attack and one of the modified methods should be used. The ElGamal scheme in the multiplicative group $\mathbf{Z}_n^*$, where $n = 2p^2$, is the easiest to apply and the weakest among the modified method. The ElGamal scheme in the domain of Gaussian is superior to that of $\mathbf{Z}_n^*$ since it requires less time to generate a key, about the same time to sign and verify a signature, and is more secure. The ElGamal scheme in the setting of a finite field has a disadvantage in finding an irreducible polynomial. The reducible polynomial scheme was the most challenging to attack due to the mathematical complexity of arithmetic modulo a polynomial.

## 5. REFERENCES


[1] J. T. Cross, "The Euler's φ-function in the Gaussian Integers", American Mathematics Monthly Vol. 90, 1983, pp. 518-528.

[2] W. Diffie, and M. E. Hellman,, "New directions in cryptography", IEEE Transaction on Information Theory, IT-22, 1978, pp. 472-492.

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

[4] A. N. El-Kassar, Chihadi H., and D. Zentout, "Quotient rings of polynomials over finite fields with cyclic group of units", Proc. International Conference on Research Trends in Science and Technology, Beirut, Lebanon, 2002, pp. 257-266.

[5] A. N. El-Kassar and R. Haraty, "ElGamal public-key cryptosystem using reducible polynomials over a finite field", Proc. ISCA 13th International Conference on Intelligent and Adaptive Systems and Software Engineering, ISCA 2004, Nice, France, 2004, 189-194.

[6] A. J. Menezes, P.C. Van Oorshot, and S. A. Vanstone, Handbook of Applied Cryptography, CRC press, 1997.

[7] R. Haraty and A. N. El-Kassar, "A Comparative Study of ElGamal Based Cryptographic Algorithms", Journal of Theoretical and Applied Computing, Vol. 12, 2005.

[8] R. Haraty and A. N. El-Kassar, "El Gamal Public-Key Cryptosystem in Multiplicative Groups of Quotient Rings of Polynomials over Finite Fields", Journal of Computer Science and Information Systems. Vol. 2, 2005.

[9] R. Haraty, A. N. El-Kassar, and B. Shebaro, "A Comparative Study of RSA-Based Digital Signature Algorithms", Journal of Mathematics and Statistics Science, Vol. 2, 2006.

[10] R. Haraty and A. N. El-Kassar, "Attacking El Gamal Based Cryptographic Algorithms Using Pollard's Rho Algorithm", Proc. ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2005). Cairo, Egypt, 2005