Hardening the ElGamal Cryptosystem in the Setting of the Second Group of Units

Ramzi Haraty, AbdulNasser ElKassar, and Suzan Fanous
Department of Computer Science and Mathematics, Lebanese American University, Lebanon

Abstract: *The Elgamal encryption scheme is best described in the setting of any finite cyclic group. Its classic case is typically presented in the multiplicative group Z_p^* of the ring of integers modulo a prime p and the multiplicative groups $F_{2^m}^*$ of finite fields of characteristic two. The Elgamal cryptosystem was modified to deal with Gaussian integers, and extended to work with group of units of $Z_p[x]/<x^2>$. In this paper, we consider yet another extension to the Elgamal cryptosystem employing the second group of units of Z_n and the second group of units of $Z_2[x]/<h(x)>$, where h(x) is an irreducible polynomial. We describe the arithmetic needed in the new setting, and present examples, proofs and algorithms to illustrate the applicability of the proposed scheme. We implement our algorithms and conduct testing to evaluate the accuracy, efficiency and security of the modified cryptographic scheme.*

Keywords: *Second group of units of $z_n$and $z_n[x]/<h(x)>$, elgamal cryptosystem, and baby step giant step attack algorithm.*

Received April 24, 2012; accepted October 25, 2012; published online February 26, 2014

1. Introduction

The Classical Elgamal Public Key encryption scheme is perhaps one of the most popular and widely used cryptosystems. It is described in the setting of the multiplicative group Z_p^*. The multiplicative group of $Z_p^* = \{1, 2, ..., p-1\}$ is a cyclic group generated by a generator from the group [9]. The following algorithms show how the Elgamal cryptosystem functions:

Algorithm 1: Key Generation

A should do the following:
1. *Generate a large random prime p and find the generator α of Z_p^*.*
2. *Select a random integer a, $1 \le a \le p-2$ and compute α^a mod p.*
3. *A's public key is (p, α, α^a) and A's private key is a.*

Algorithm 2: Encryption

B should do the following:
1. *Obtain A's authentic public key (p, α, α^a).*
2. *Represent the message as an integer m in the range {0, 1, 2, ..., p−1}.*
3. *Select a random integer k, where $2 \le k \le p-2$.*
4. *Compute $\gamma \equiv \alpha^k \bmod p$ and $\delta \equiv m.(\alpha^a)^k \bmod p$.*
5. *Send the ciphertext $c = (\gamma, \delta)$ to A.*

Algorithm 3: Decryption

A should do the following:
1. *Use the private key a to compute $\gamma^{-p-a} \bmod p$.*
2. *Compute $c \equiv \gamma^{-a}.\delta \bmod p$ to recover the message m.*

However, the Elgamal cryptosystem can be generalized to work in any finite cyclic group *G*. The security of the scheme is based on the intractability of the Discrete Logarithm Problem [12] in the group *G*. *G* should be carefully chosen so that the group operations in *G* would be relatively easy to apply for efficiency. Moreover, the Discrete Logarithm Problem in *G* should be computationally infeasible.

Cross [3] gave a classification of all Gaussian integers β such that the group of units of the quotient ring $Z[i]/<\beta>$ is cyclic. So, one may consider Elgamal public-key cryptosystem using the cyclic group of units of $Z[i]/<\beta>$, where $\beta = 1+i$, $(1+i)^2$, $(1+i)^3$, p, $(1+i)\,p$, π^n, $(1+i)\,\pi^n$, p is a prime integer of the form $4k+3$, and π is a Gaussian prime with $|\pi|^2$ is a prime integer of the form $4k+1$.

In [6], the authors described the second group of units of Z_n, denoted by $U^2(Z_n)$, and characterized the cases for n that make $U^2(Z_n)$ cyclic. They applied the Elgamal scheme in $U^2(Z_n)$ (in the cases where it is cyclic).

The authors in [10] determined the structure of the group of units of the quotient ring $F_q[x]/<f(x)>$, where $f(x)$ is a polynomial over a finite field F_q of order q. Using this decomposition, a characterization of the quotient ring of polynomials over finite fields with cyclic group of units was given. In [8], this classification was applied to extend Elgamal scheme to the settings of the group of units of $Z_p[x]/<x^2>$ and $Z_2[x]/<h(x)>$, where $h(x)$ is a

product of irreducible polynomials whose degrees are pairwise relatively prime.

The purpose of this paper is to use the characterization of the quotient ring of polynomials over finite fields and the characterization of the second group of units in order to apply the Elgamal scheme in the second group of units of $Z_2[x]/<h(x)>$, where $h(x)$ is an irreducible polynomial of degree n.

The rest of the paper is organized follows: Section 2 summarizes the second group of units of Z_n and describes the construction of $U^2(Z_n)$. In section 3, the construction of the second group of units of $Z_2[x]/<h(x)>$, where $h(x)$ is an irreducible polynomial, is illustrated. Section 4 investigates the Elgamal scheme in this setting. Section 5 tests and evaluates the modified algorithms. Finally, section 6 concludes the paper.

2. The Elgamal Cryptosystem over the Second Group of Units of Z_n

Before discussing the extended Elgamal cryptosystem, we present the following theorems and definitions that are necessary for our work.

- *Theorem 1*: *Fundamental Theorem of Abelian Groups*, every finite Abelian group is a direct product of cyclic groups of prime power order [1].
  Let R be a finite commutative ring with identity. By the fundamental theorem of finite Abelian groups, the group of units, $U(R)$, is isomorphic to the direct product of cyclic groups, say $U(R) \cong Z_{n1} \times Z_{n2} \times \ldots \times Z_{ni}$. Hence, the multiplicative group $U(R)$ supports a ring structure by defining the operations $\oplus$ and $\otimes$ that make $(U(R), \oplus, \otimes)$ a ring isomorphic to the direct sum $Z_{n1} \oplus Z_{n2} \oplus \ldots \oplus Z_{ni}$. The ring $Z_{n1} \oplus Z_{n2} \oplus \ldots \oplus Z_{ni}$ is denoted as R^2.

- *Definition 1*: The second group of units of R is defined as the group of units of the ring $U(R)$ [6]: $U^2(R) = U(U(R)) \cong U(R^2)$.

  The authors considered the problem of determining the values of n that make $U^2(Z_n)$ cyclic. The result was as follows:

- *Theorem 2*: Let p and q be odd prime integers and α be a positive integer. Then, $U^2(Z_n)$ is cyclic iff one of the following is true:

  1. $n = 2^{\alpha}.3.p$, where $\alpha = 1$, 2 or 3.
  2. $n = 15$.
  3. $n = 3.p$, where $p = 4k+3$ and $2k+1 = q^{\alpha}$.
  4. $n = 2.3^{\alpha}$, $2^2.3^{\alpha}$, $2^3.3^{\alpha}$ or $2^4.3$.
  5. $n = 2$, 4, 8, or 16.
  6. $n = 5$ or $2p^{\alpha}+1$, where $2p^{\alpha}+1$ is a prime integer.
  7. $n = 3^{\alpha}$.

The results can be classified into two cases:

- *Case 1*: Both $U(Z_n)$ and $U^2(Z_n)$ are cyclic, as known $U(Z_n)$ is cyclic when $n = 2$, 4, p^{α} or $2p^{\alpha}$, where $\alpha \geq 1$ and p is an odd prime integer. So out of the previous seven cases for n, only the following belong to case 1:
  1. $n = 2.3^{\alpha}$, since n is in the form of $2p^{\alpha}$.
  2. $n = 2$ or 4.
  3. $n = 5$ or $2p^{\alpha}+1$, where $2p^{\alpha}+1$ is a prime integer.
  4. $n = 3^{\alpha}$, since n is a power of an odd prime.

- *Case 2*: $U^2(Z_n)$ is cyclic, whereas $U(Z_n)$ is not cyclic,
  1. $n = 2^{\alpha}.3.p$, where $\alpha = 1$, 2 or 3.
  2. $n = 15$.
  3. $n = 3.p$, where $p = 4k+3$ and $2k+1 = q^{\alpha}$.
  4. $n = 2^2.3^{\alpha}$, $2^3.3^{\alpha}$ or $2^4.3$.
  5. $n = 8$, or 16.

To construct the second group of units $U^2(Z_n)$, we follow these steps:

1. Form the group of units $U(Z_n) = \{a \in Z_n : \gcd(a,n) = 1\}$. The order of $U(Z_n)$ is $\varphi(n)$.
2. Find a generator r of $U(Z_n)$.
3. Write $U(Z_n)$ in the form $\{r^0, r^1, \ldots, r^{\varphi(n)-1}\}$.
4. Find $U^2(Z_n) = \{r^i \ mod \ n : \gcd(i, \varphi(n)) = 1\}$. Note that the order of $U^2(Z_n)$ is $\varphi(\varphi(n))$.

For example, let $n = 23$. Then $U(Z_{23}) = \{1, 2, 3, \ldots, 22\}$ and $\varphi(23) = 22$. A generator of $U(Z_{23})$ is $r = 5$ as shown in Table 1.

Table 1. Elements of $U(Z_{23})$.

$U(Z_{23})$	1	2	3	4	5	6	7	8	9	10	11
5^i	5^0	5^2	5^{16}	5^4	5^1	5^{18}	5^{19}	5^6	5^{10}	5^3	5^9

$U(Z_{23})$	12	13	14	15	16	17	18	19	20	21	22
5^i	5^{20}	5^{14}	5^{21}	5^{17}	5^8	5^7	5^{12}	5^{15}	5^5	5^{13}	5^{11}

$U^2(Z_{23}) = \{5^i : \gcd(i, 22) = 1\}$
$= \{5^1, 5^{19}, 5^3, 5^9, 5^{21}, 5^{17}, 5^7, 5^{15}, 5^5, 5^{13}\}$.

Reducing the powers modulo 23, we have:

$$U^2(Z_{23}) = \{5, 10, 20, 17, 11, 21, 19, 15, 7, 14\}.$$

The order of $U^2(Z_{23})$ is $\varphi(\varphi(23)) = 10$.

The operations of the ring $(U(Z_n), . , \otimes)$ are defined as follows [4]:

1. Addition operation: $x.y = xy \bmod n$.
2. Multiplication operation: $x \otimes y = x^{\log_r y} (\bmod n)$, where r is the generator of $U(Z_n)$.
3. The power operation: Let θ be an element of $U^2(Z_n)$ such that $\theta = r^k$ and r is a generator of $U(Z_n)$. Then $\theta^N = (r)^{k^N}$.

The identity of the second group of units of Z_n is r, where r is the generator of $U(Z_n)$.
For the construction of $U^2(Z_n)$ in case 2, consider:

- *Lemma 1*: Let n and m be any two positive integers, then $Z_{mn} \cong Z_m \times Z_n$ iff gcd$(m,n) = 1$.
  Hence, for $n = n_1. n_2 \ldots n_i$, $Z_n \cong Z_{n_1} \times Z_{n_2} \times \ldots \times Z_{n_i}$ iff $n_1, n_2, \ldots, n_i$ are pairwise relatively prime.
- *Lemma 2*:
  1. $U(Z_2) \cong \{0\}$.
  2. $U(Z_4) \cong Z_2$.
  3. $U(Z_{2^k}) \cong Z_2 \oplus Z_{2^{k-2}}$, for $k \geq 3$.
  4. $U(Z_{p^n}) \cong Z_{p^{n-1}} \oplus Z_{p-1}$, where p is an odd prime.
- *Theorem 3*: If $R = R_1 \oplus R_2 \oplus \ldots \oplus R_i$, then $U(R) \cong U(R_1) \times U(R_2) \times \ldots \times U(R_i)$.
  For the case where $n = 3p$,
  $$U(Z_n) \cong U(Z_3) \times U(Z_p) \cong Z_2 \times Z_{p-1}.$$
  Hence, $R^2 = Z_2 \oplus Z_{p-1}$ and
  $$U^2(R) \cong U(R^2) = U(Z_2 \oplus Z_{p-1}) \cong U(Z_2) \times U(Z_{p-1}) \cong U(Z_{p-1}).$$
- *Definition 2*: (Isomorphism Functions)
  1. The function f: $U(Z_{mn}) \rightarrow U(Z_m) \times U(Z_n)$ defined by $f(a) = (a \bmod m, a \bmod n)$, for all $a \in U(Z_{mn})$, is an isomorphism whenever $(m,n) = 1$.
  2. The function f_1: $U(Z_n) \rightarrow Z_{\varphi(n)}$ defined by $f_1(a) = \log_r a \bmod n$, for all $a \in Z_{\varphi(n)}$, is an isomorphism whenever $U(Z_n)$ is a cyclic and r is a generator of $U(Z_n)$.

The group $U^2(Z_n)$ can be constructed as follows:

1. Construct $U(Z_n)$, $U(Z_3)$ and $U(Z_p)$.
2. Find $G = \{ (a \pmod 3), a \pmod p) : a \in U(Z_n)$.
3. Find a generator r of $U(Z_3)$ and a generator r_1 of $U(Z_p)$.
4. Write G in the form:
   $$\left\{ \left(r^t \pmod 3, r_1^{t_1} \pmod p \right) : 0 \leq t \leq 1 \text{ and } 0 \leq t_1 \leq p-2 \right\}.$$
5. Form the set $Z_2 \oplus Z_p - 1 = \{(t, t_1)\}$ and find G_1, the set of its invertible elements. Note that (a, b) is invertible in $Z_2 \oplus Z_p - 1$ iff $a = 1$ (invertible in Z_2) and b is invertible in $Z_p - 1$.
6. $U^2(Z_n)$ is the set of elements in $U^2(Z_n)$ corresponding to elements in G_1.

For example, let $p = 11$. Then, $U(Z_{33}) \cong U(Z_3) \times U(Z_{11}) \cong Z_2 \oplus Z_p - 1$, where $U(Z_{33}) = \{1, 2, 4, 5, 7, 8, 10, 13, 14, 16, 17, 19, 20, 23, 25, 26, 28, 29, 31, 32\}$. Now, $U(Z_{11}) = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ has 2 as a generator. Similarly, 2 is a generator for $U(Z_3) = \{1, 2\}$. The isomorphism between $U(Z_{33})$, $U(Z_3) \times U(Z_{11})$ and $Z_2 \oplus Z_{10}$ is depicted in Table 2. The invertible elements in $Z_2 \oplus Z_{10}$ are: $\{(1,1), (1,3), (1,9), (1,7)\}$. Then, $U^2(Z_{33}) = \{2, 8, 17, 29\}$.

Define the operations on $U^2(Z_n)$ as follows. Multiplication in $(U^2(Z_n), \bullet)$ is defined by $a \bullet b = f^{-1}(f(a) \otimes f(b))$,
where f is the isomorphism f: $U(Z_n) \rightarrow U(Z_3) \times U(Z_p)$ in Definition 2, and

$$f(a) \otimes f(b) = (a \bmod 3, a \bmod p) \otimes (b \bmod 3, b \bmod p) = ((a \bmod 3) \otimes_3 (b \bmod 3), (a \bmod p) \otimes_p (b \bmod p)),$$
$x \otimes_3 y = x^{\log_r y}$ and $x \otimes_p y = x^{\log_{r1} y}$.

Let $a \in U^2(Z_n)$ and let N be positive integer. Then, $a^N = f^{-1}(r^{t^N}, r_1^{t_1^N})$. The identity of $U^2(Z_n)$ is the element $a = f^{-1}(r, r_1)$.

Table 2. Isomorphism between $U(Z_{33})$ and $U(Z_3) \times U(Z_{11})$.

$U(Z_{33})$	$U(Z_3) \times U(Z_{11})$	$(2^1, 2^1)$	$Z_2 \oplus Z_{10}$	$U(Z_{33})$	$U(Z_3) \times U(Z_{11})$	$Z_2 \oplus Z_{10}$
1	(1,1)	$(2^0,2^0)$	(0,0)	17	(2,6)	(1,9)
2	(2,2)	$(2^1,2^1)$	(1,1)	19	(1,8)	(0,3)
4	(1,4)	$(2^0,2^2)$	(0,2)	20	(2,9)	(1,6)
5	(2,5)	$(2^1,2^4)$	(1,4)	23	(2,1)	(1,0)
7	(1,7)	$(2^0,2^7)$	(0,7)	25	(1,3)	(0,8)
8	(2,8)	$(2^1,2^3)$	(1,3)	26	(2,4)	(1,2)
10	(1,10)	$(2^0,2^5)$	(0,5)	28	(1,6)	(0,9)
13	(1,2)	$(2^0,2^1)$	(0,1)	29	(2,7)	(1,7)
14	(2,3)	$(2^1,2^8)$	(1,8)	31	(1,9)	(0,6)
16	(1,5)	$(2^0,2^4)$	(0,4)	32	(2,10)	(1,5)

2.1. Elgamal Cryptosystem over $U^2(Z_n)$ for Case 1.

Algorithm 4: Generator of $U^2(Z_n)$

1. Find a generator $\theta 1$ of $U(Z_n)$.
2. Write the order of $U^2(Z_n)$ as $p_1^{\alpha_1}.p_2^{\alpha_2} \ldots p_i^{\alpha_i}$.
3. Select a random integer s, $0 \leq s \leq \varphi(n)-1$, $(s, \varphi(n)) = 1$.
4. For j = 1 to i, do:
  4.1. Compute $\theta 1^{N/p_j} \bmod n$.
  4.2. If $\theta 1^{s(N/p_j)} \bmod n = \theta 1$, then go to step 3.
5. Return s.

For key generation, entity A does the following:

Algorithm 5: Key Generation

1. Find a generator $\theta 1$ of $U(Z_n)$.
2. Find s using Algorithm 1.
3. Compute the order of $U^2(Z_n)$ using $\varphi(\varphi(n))$.
4. Select a random integer a, $2 \leq a \leq \varphi(\varphi(n))-1$, and compute $f = s^a (\text{mod } \varphi(n))$.
5. A's public key is (n, $\theta 1$, s, f) and A's private key is a.

B encrypts a message m for A using the algorithm below:

Algorithm 6: Encryption

1. B obtains A's authentic public key (n, $\theta 1$, s, f).
2. Represent the message as an integer in $U^2(Z_n)$.
3. Select a random integer k, $2 \leq k \leq \varphi(\varphi(n))-1$.
4. Compute $q = s^k (\text{mod } \varphi(n))$, $r = f^k (\text{mod } \varphi(n))$, $\gamma = \theta^k = \theta 1^q \pmod n$ and $\delta \equiv m^r \pmod n$.
5. Send the ciphertext $c = (q, \delta)$ to A.

To recover the plaintext m from c, A should do the following:

Algorithm 7: Decryption

1. Use the private key a to compute b = φ(φ(n)) – a.
2. Recover the message by computing t = q^b(mod φ(n)) and δ^t (mod n).

Theorem 4 proves that the formula $\gamma^{-a}.\delta \pmod n$ allows the recovery of the message *m*. The proof is for case 1.

- *Theorem 4*: Given a generator θ of $U^2(Z_n)$ such that $\theta = \theta 1^s$, where $\theta 1$ is a generator of $U(Z_n)$. Let $\gamma \equiv \theta^r \pmod n$ and $\delta \equiv m.(\theta^a)^k \pmod n$. If $s \in U^2(Z_n)$ such that: $s \equiv \gamma^{-a}.\delta \pmod n$, then $s = m$.

For example, consider the case where $n = 3^3$. Select the generators $\theta 1 = 5$ and $\theta = 2$ so that $s = 11$. Entity A selects $a = 3$ and calculates $s^a = 2^3 \equiv (5^{11})^3 \equiv 11 \pmod{27}$ and $f = 11^3 \equiv 17 \pmod{\varphi(27)}$. A's public key is ($n$=27, $\theta 1$=5, s=11, f=17). To encrypt $m = 5$, B selects an integer $k = 4$ and finds $q = s^k = 11^4 \equiv 7 \pmod{\varphi(27)}$, $r = f^k = 17^4 \equiv 1 \pmod{\varphi(27)}$, and $\delta \equiv 5^r = 5 \pmod{27}$. B sends ($s^k = 7$, $\delta = 5$) to A. Finally, A computes $b = \varphi(\varphi(n)) - a = 2$ and $t = q^b \pmod{\varphi(n)} = 1$. Then A finds $\delta^t = 5^1 \pmod{27} = 5 = m$.

2.2. Elgamal Cryptosystem over $U^2(Z_n)$) for Case 2

Algorithm 8: Generator of $U^2(Z_{3p})$

1. Find a generator θ1 of $U(Z_n)$.
2. Write the order of $U^2(Z_n)$ as $\varphi(\varphi(n)) = p_1^{\alpha_1}.p_2^{\alpha_2}...p_i^{\alpha_i}$.
3. Select a random integer s, $0 \le s \le \varphi(p)-1$, (s , φ(p)) = 1.
4. For j = 1 to i, do:
  4.1 Compute $\theta 1^{N/p_j} \bmod p$.
  4.2 If $\theta 1^{s(N/p_j)} \bmod p = \theta 1$, then go to step 3.
5. Use the Chinese Remainder Theorem to find θ, and s by solving the system of congruencies: x ≡ 2 (mod 3) and $x \equiv \theta 1^s$ (mod p).
6. Return s.

To generate the key when $n = 3.p$, A uses the following:

Algorithm 9: Key Generation

1. Find a generator θ1 of $U(Z_n)$.
2. Find s using Algorithm 1.
3. Compute the order of $U^2(Z_n)$ using φ(φ(p)).
4. Select a random integer a, $2 \le a \le \varphi(\varphi(p))-1$, and compute $f = s^a \pmod{\varphi(n)}$.
5. A's public key is (n, θ1, s, f) and A's private key is a.

B encrypts a message *m* for A using the algorithm below:

Algorithm 10: Encryption

1. B obtains A's authentic public key (n, θ1, s, f).
2. Represent the message as an integer in $U^2(Z_p)$.
3. Select a random integer k, $2 \le k \le \varphi(\varphi(p))-1$.
4. Compute $q = s^k \pmod{\varphi(n)}$, $r = f^k \pmod{\varphi(p)}$, $\gamma = \theta^k = \theta 1^q \pmod n$ and $\delta \equiv m^r \pmod n$.
5. Send the ciphertext c = (q, δ) to A.

To recover the plaintext *m* from *c*, A should do the following:

Algorithm 11 (Decryption)

1. Use the private key a to compute b = φ(φ(p)) – a.
2. Recover the message by computing t = q^b(mod φ(p)) and δ^t (mod p).

- *Theorem 5*: Let $n = 3.p$ and let θ be a generator of $U^2(Z_n)$ such that $\theta = (2, \theta 1^s)$, where $\theta 1$ is a generator of $U(Z_p)$. Let $m \in U(Z_p)$. Let $\gamma \equiv \theta^r \pmod n$ and $\delta \equiv m.(\theta^a)^k \pmod n$. If $s \in U(Z_p)$ such that $s \equiv \gamma^{-a}.\delta \pmod n$, then $s = m$.

For example, let $p = 11$. Then, $\varphi(n) = 20$, $\theta = 29$ is a generator of $U^2(Z_{33})$, $\theta 1 = 2$ is a generator of $U(Z_{11})$ with $s = 7$. If A uses $a = 3$, then $f = s^a = 7^3 \equiv 3 \pmod{\varphi(11)}$. A's public key is ($p$=11, $\theta 1$=2, s=7, f=3). To encrypt the message $m = 5$, B selects a random integer $k = 2$ and finds $s^k = 7^2 \equiv 9 \pmod{\varphi(11)}$, $\gamma \equiv (2, \theta 1^{s^k} \bmod 11) = (2,6)$, and $\delta \equiv 5.(2,6)$. Finally, A computes $b = 1$ and $\gamma^1.\delta \equiv (2,6).5.(2,6) \equiv 5.(2,2^9.2^9) \equiv 5.(2,2) \equiv 5$.

3. Elgamal Cryptosystem over $U^2(Z_2[x]/<h(x)>)$

Let $h(x)$ be an irreducible polynomial of degree n. Then, $Z_2[x]/<h(x)> = \{a_0+a_1x+\ldots+a_{n-1}x^{n-1}: a_0, \ldots, a_{n-1} \in Z_2\}$ is a field. The order of $Z_2[x]/<h(x)>$ is 2^n and its non-zero elements form a cyclic group $U(Z_2[x]/<h(x)>)$ of order $\varphi(h(x)) = 2^n-1$, see [7] for more details.

- *Theorem 6*: $U^2(Z_2[x]/<h(x)>)$ is cyclic iff one of the following conditions is satisfied:
  1. $2^n = q^\alpha+1$ where q is an odd prime and $\alpha > 0$.
  2. $2^n = q+1$, where q is a Mersenne prime [2].

To construct $U^2(Z_2[x]/<h(x)>)$:

1. Find a generator r of the group of units $U(Z_2[x]/<h(x)>)$
2. Write $U(Z_2[x]/<h(x)>) = \{r^0, r^1, \ldots r^{\varphi(h(x))-1}\}$.
3. Find $U^2(Z_2[x]/<h(x)>) = \{r^i : \gcd(i, 2^n-1) = 1\}$.
4. Write $U^2(Z_2[x]/<h(x)>) = \{x : x \equiv r^i \bmod h(x)\}$.

For example, let $h(x) = 1+x+x^3$. Then, $Z_2[x]/<1+x+x^3> = \{0, 1, x, 1+x, x^2, 1+x^2, x+x^2, 1+x+x^2\}$, and $U(Z_2[x]/<h(x)>) = \{1, x, 1+x, x^2, 1+x^2, x+x^2, 1+x+x^2\}$. Using the fact that $1+x+x^3 = 0$ in $Z_2[x]/<1+x+x^3>$, we have $1 = (1+x)^0$, $1+x = (1+x)^1$, $1+x^2 = (1+x)^2$, $x^2 = (1+x)^3$, $1+x+x^2 = (1+x)^4$, $x = (1+x)^5$, $x+x^2 = (1+x)^6$, and $r = 1+x$ is a generator. Hence, $U^2(Z_2[x]/<h(x)>) = \{x, 1+x, x^2, 1+x^2, x+x^2, 1+x+x^2\}$.

Next, the Elgamal scheme is extended to the setting of the second group of units of $U^2(Z_2[x]/<h(x)>)$.

Algorithm 12: Generator of $U^2(Z_2[x]/<h(x)>)$

1. *Find a generator $\theta 1(x)$ of $U(Z_2[x]/<h(x)>)$.*
2. *Write the order of $U^2(Z_2[x]/<h(x)>)$* **as** *$\varphi(h(x)) = p_1^{\alpha_1}.p_2^{\alpha_2}...p_i^{\alpha_i}$.*
3. *Select a random integer s, $0 \le s \le \varphi(h(x)) - 1$, such that (s, $\varphi(h(x)) = 1$.*
4. *For j = 1 to i, do:*
   4.1 Compute $\theta 1(x)^{N/Pj}$ mod h(x).
   4.2 If $\theta 1(x)^{s^{(N/p_j)}}$ mod h(x) = $\theta 1(x)$, then go to step 3.
5. *Return s.*

In order to generate the corresponding public and private keys, entity A follows the steps below:

Algorithm 13: Key Generation

1. *Select an irreducible polynomial h(x) of degree n.*
2. *Find a generator $\theta 1(x)$ of $U(Z_2[x]/<h(x)>)$.*
3. *Find s using Algorithm 12.*
4. *Compute $\varphi(\varphi(h(x)))$, the order of $U^2(Z_2[x]/<h(x)>)$.*
5. *Select a random integer a, $0 \le a \le \varphi(\varphi(h(x))) - 1$, and compute $f = s^a \pmod{\varphi(h(x))}$.*
6. *A's public key is (n, h(x), $\theta 1(x)$, s, f) and A's private key is a.*

B encrypts a message $m(x)$ for A using the algorithm below:

Algorithm 14: Encryption

1. *B obtains A's authentic public key (n, h(x), $\theta 1(x)$, s, f).*
2. *Represent a message m(x) in $U^2(Z_2[x]/<h(x)>)$.*
3. *Select a random integer k, where $0 \le k \le \varphi(\varphi(h(x))) - 1$.*
4. *Compute $q = s^k \pmod{\varphi(h(x))}$, $r = s^k \pmod{\varphi(h(x))}$, $\gamma(x) = \theta(x)^k = \theta 1(x)^q \pmod{h(x)}$, and $\delta(x) = m(x)^r \pmod{h(x)}$.*
5. *Send the ciphertext $c = (q, \delta(x))$ to A.*

To recover $m(x)$ from c, A should do the following:

Algorithm 15: Decryption

1. *Use the private key a to compute $b = \varphi(\varphi(h(x))) - a$.*
2. *Recover the message by computing $t = q^b \pmod{\varphi(h(x))}$ and $\delta(x)^k \pmod{h(x)}$.*

- *Theorem 7*: Let $\theta(x) = \theta 1(x)^s$ be a generator of $U^2(Z_2[x]/<h(x)>)$, where $\theta 1(x)$ is the generator of $U(Z_2[x]/<h(x)>)$, and let $\gamma(x) \equiv \theta(x)^k \pmod{h(x)}$, and $\delta(x) = m(x).(\theta(x)^a)^k \pmod{h(x)}$. If $s(x) \in U^2(Z_2[x]/<h(x)>)$ such that $s(x) \equiv \gamma(x)^{-a}.\delta(x) \pmod{h(x)}$, then $s(x) = m(x)$.

For example, consider $h(x) = 1+x+x^3$. A selects the generator $\theta(x) = x$, $\theta 1(x) = 1+x$, $s = 5$ and $a = 3$. Then A computes $f = s^a = 5^3 \equiv 5 \pmod{\varphi(7)}$. Then A's public key is ($n = 3$, $h(x) = 1+x+x^3$, $\theta 1(x) = 1+x$, $s = 5$, $f = 5$). To encrypt the message $m(x) = x^2$, B selects a random integer k=3 and computes, $r=s^k = 5^3 \equiv 5 \pmod{\varphi(7)}$, $q=f^k \pmod 6 \equiv 5$, and $\delta(x)= m(x)^q \pmod{1+x+x^3} \equiv 1+x$. B sends $(r, \delta(x))$.

To decrypt the message A computes $b = 6 - 3 = 3$ and $t = r^b = 5^3 \bmod 6 = 5$. Finally, A computes $\delta(x)^t = (1+x)^3 \pmod{1+x+x^3} \equiv x^2$.

4. Testing and Evaluation

The modified Elgamal cryptosystem was tested and evaluated by implementing the modified algorithms. We used Mathematica 7.0 as a programming language and an HP computer with 1.73 GHZ CPU and 1014 MB RAM.

Using Mathematica 7.0, we have written programs for the following algorithms:

1. Elgamal with n in the form 2.3^{α}.
2. Elgamal with n in the form 4.3^{α}.
3. Elgamal with n in the form 3^{α}.
4. Elgamal with n in the form $3.p$.
5. Elgamal with n in the form $3.2^{\alpha}.p$.
6. Elgamal with n in the form $2.p^{\alpha}+1$.
7. Elgamal over the polynomial case.

After running the programs, it was clear that all the programs have generated a public and private key. A message is encrypted and is sent to a decryption scheme which recovered the message.

Table 3 and Figure 3 show the results obtained after running the Mathematica programs for 100 times. For more information on these programs, readers are referred to [5].

Table 4. Elgamal evaluation.

Algorithm	Key Generation	Encryption	Decryption
n=3^a	972.158	5116.84	2801.47
n=2.3^a	1941.4	10302.2	5669.9
n=2p^a+1	2.4	2.35	7736.1
n=4.3^a	2651.25	14160.9	2.13333
n=3.p	2.3	6.81421*10^(-15)	6.81421*10^(-15)
n=3.4.p	2.35	0.75	5.14996*10^(-15)
Poly case	937.5	20.4	3.15

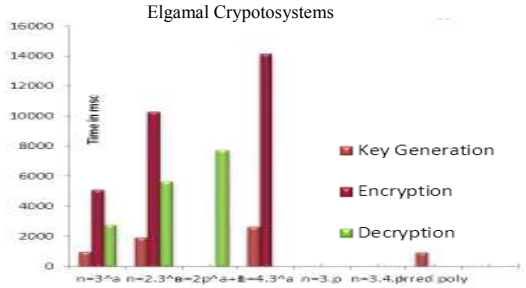

Figure 1. Testing the modified Elgamal cryptosystems.

Comparing these algorithms with one another, one can conclude the following:

1. All programs are reliable. They can encrypt and decrypt any message.
2. For the irreducible polynomial case, it took considerable time to find an irreducible polynomial of high degree. Moreover, it takes more time to generate the public and private key than to decrypt or encrypt a message. Finally, it takes a considerable time to find some of the elements of $U(Z_2[x]/<h(x)>)$ when the degree of $h(x)$ is high.

3. For cases 2, 3 and 4, the time needed for encryption is more than that needed for generating public and private keys and for decryption. But it works well even for a=10000.
4. Cases 5, 6 and 7 are more efficient than others since they work for every prime number, even for primes consisting of 13 digits and they require less time for encryption, decryption and key generation.
5. The case where $n = 3p$ is the most efficient since it takes the least time for encryption, key generation and decryption.
6. In case 7, the time needed for decryption was much more than that needed for key generation and encryption.

4.1. The Discrete Logarithm Problem

The security of the El-gamal cryptosystem depends on the intractability of the Discrete Logarithm Problem: Let G be a finite cyclic group of order n. Let a be a generator of G, and $\beta \in G$. The Discrete Logarithm of β to the base a, denoted by $log_\alpha \beta$, is the unique integer x, $0 \le x \le n-1$, such that $\beta = a^x$.

To attack the modified Elgamal cryptosystem, we have to solve the Discrete Logarithm Problem. The most popular attack algorithm is the Baby-Step Giant-Step algorithm [11]:

Let $m = \lceil \sqrt{n} \rceil$, where n is the order of a. If $\beta = a^x$, then one can write $x = im + j$, where $0 \le i, j < m$. Hence, $a^x = a^{im} a^j$, which implies $\beta(a^{-m})^i = a^j$.

Algorithm 16: Baby-Step Giant-Step algorithm
Input: a generator of a cyclic group G of order n, and an element $\beta \in G$

Output: the discrete logarithm $x = log_\alpha \beta$.

1. *Set* $m = \lceil \sqrt{n} \rceil$.
2. *Construct a table with entries* (j, a^j) *for* $0 \le j < m$. *Sort this table by the second component.*
3. *Compute* a^{-m} *and set* $\gamma \leftarrow \beta$.
4. *For i from 0 to m-1 do the following:*
   4.1 Check if γ *is the second component of some entry in the table,*
   4.2 If $\gamma = a^j$ *then return x = im + j,*
   4.3 Set $\leftarrow \gamma . a^{-m}$.

Below is a list of the implemented attack algorithms:

1. Baby giant with $n = 2.3^\alpha$.
2. Baby giant with $n = 4.3^\alpha$.
3. Baby giant with $n = 3^\alpha$.
4. Baby giant with $n = 3.p$.
5. Baby giant with $n = 3.2^\alpha.p$.
6. Baby giant with $n = 2.p^\alpha+1$.
7. Irreducible polynomial baby giant.

In order to attack any protocol that uses Elgamal public key encryption scheme we have to solve the discrete logarithm problem. We enhanced the baby step giant step algorithm to work with the modified algorithms. Table 5 and Figure. 2 show the results of running the programs 100 times in each case.

Table 5. Baby step giant step attack.

Algorithm	Time needed in atto sec
n=3^a	3.04119)
n=2.3^a	3.07689
n=2p^a+1	4.34042
n=4.3^a	3.20547
n=3.p	5.0357
n=3.4.p	2.96404)
Poly case	11.3508

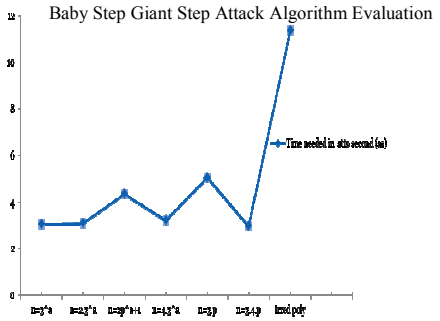

Figure 2. The baby step giant step algorithm evaluation.

After running these attack algorithms, we observed the following:

1. All the attack programs are reliable so that they can hack an encrypted message by finding the private key.
2. In all the cases, the time needed to attack the modified cryptosystem is approximately the same.
3. The most difficult to attack is the irreducible polynomial case. This is due to the fact that mathematically it is complex and needs considerable computing time to find the modulus of a given polynomial with respect to a certain irreducible polynomial.
4. We were not able to run the programs on large values of p and large powers since it would take a considerable time to generate some of the elements of the second group of units in each case.

5. Conclusions

In this paper, we extended the Elgamal cryptosystem using the second group of units. We presented algorithms for the extended cryptosystem in the setting of $U^2(Z_n)$ and provided numerical examples to illustrate the proposed scheme. We provided algorithms for the case of the second group of units of $Z_2[x]/<h(x)>$ where $h(x)$ is an irreducible polynomial. We also provided proofs that the proposed scheme does really recover the plaintext from the ciphertext. We also conducted testing and evaluation of the proposed scheme.

As for future work, we are currently investigating the case of $n = 8.3^{\alpha}$ and 16.3^{α} where different algorithms would be used. This due to the fact that 4 is relatively prime with neither 2.3^{α} nor 4.3^{α}. We can also considering the cases where $h(x) = x^2$ and when $h(x)$ is a product of irreducible polynomials whose degrees are pairwise relatively prime and using time stamps [13].

References

[1] Beachy J. and Blair W., *Abstract Algebra,* Waveland Press, USA, 1996.

[2] Caldwell C. and Honaker J., *Prime Curious! The Dictionary of Prime Number Trivia*, Caldwell and Honaker, 2009.

[3] Cross J., "The Euler's φ-Function in the Gaussian Integers," *American Mathematical Monthly*, vol. 90, no.8, pp. 518 - 528, 1983.

[4] El-Kassar A., Chihadi H., and Zentout D., "Quotient Rings of Polynomials over Finite Fields with Cyclic Group of Units," *in Proceedings of the International Conference on Research Trends in Science and Technology*, Beirut, Lebanon, pp. 257 - 266. 2002.

[5] El-Kassar A. and Haraty R., "Elgamal Public-Key Cryptosystem Using Reducible Polynomials over a Finite Field," *in Proceedings of the 13th International Conference on Intelligent & Adaptive Systems and Software Engineering*, Nice, France, pp. 189 - 194, 2004.

[6] El-Kassar A., Haraty R., Awad Y., and Debnath N., "Modified RSA in the Domains of Gaussian Integers and Polynomials Over Finite Fields," *in Proceedings of the ISCA 18th International Conference on Computer Applications in Industry and Engineering*, Hawaii, USA, 2005.

[7] Gallian J., *Contemporary Abstract Algebra*, Houghton Mifflin Company, Boston, 1998.

[8] Haraty R., El-Kassar A., and Shibaro B., "A Comparative Study of RSA Based Digital Signature Algorithms," *Journal of Mathematics and Statistics*, vol. 2, no. 1, pp. 354 - 359, 2006.

[9] Menezes A., Van-Oorshot J., and Vanstone P., *Handbook of Applied Cryptography*, Boca Raton, CRC Press, 1997.

[10] Smith J. and Gallian J., "Factoring Finite Factor Rings," *Mathematics Magazine*, vol. 58, no. 2, pp. 93 - 95, 1985.

[11] Stein A. and Teske E., "Optimized Baby Step-Giant Step Methods," *Journal of the Ramanujan Mathematical Society*, vol. 20, no. 1, pp. 1 - 32, 2005.

[12] Stinson D., *Cryptography: Theory and Practice*, London, CRC Press, 2006.

[13] VNKrishna A., "TimStamp Based ECC Encryption and Decryption," *the International Arab Journal of Information Technology*, vol. 11, pp. 276 - 280, no. 3, 2014.

Ramzi Haraty is an associate professor of Computer Science in the Department of Computer Science and Mathematics at the Lebanese American University in Beirut, Lebanon. He is also the academic and internship coordinator for Middle East Program Initiative's Tomorrow Leader's program. He received his B.S. and M.S. degrees in Computer Science from Minnesota State University - Mankato, Minnesota, and his Ph.D. in Computer Science from North Dakota State University - Fargo, North Dakota. His research interests include database management systems, artificial intelligence, and multilevel secure systems engineering. He has well over 110books, book chapters, journal and conference paper publications. He supervised over 110 dissertations, theses and capstone projects. He is a member of the Association of Computing Machinery, Institute of Electronics, Information and Communication Engineers, and the International Society for Computers and Their Applications.

Abdul-Nasser El-Kassar is the Chairperson of the Information Technology and Operation Management Department, School of Business, Lebanese American University, Beirut, Lebanon. He was awarded a PhD in Mathematics from University of Louisiana - Lafayette. His research areas include security and cryptography, production planning and inventory control, abstract algebra, and number theory. He has over forty publications in refereed journals and a similar number in refereed conference proceeding.

Suzan Fanous received her Master of Science degree in Computer Science from the Lebanese American University – Beirut, Lebanon. Her research interests include cryptography and computer security.