Non-interactive Evaluation of Encrypted Elementary Functions

Liang Chen\(^1\)\(^2\), Chengmin Gao\(^2\), Yong Xu\(^1\)\(^+\)
1 School of Computer Science & Engineering, South China University of Technology, Guangzhou, China
2 Department of Computer, Guangdong Police College, Guangzhou, China
+ Corresponding author

Abstract

Mobile code can be considered composing of functions. Sander et al implemented non-interactive evaluation of encrypted functions (non-interactive EEF) based on homomorphism. But their scheme is limited to encrypting polynomials in positive integer domain. There is no homomorphism that can securely implement non-interactive evaluation of encrypted elementary functions (non-interactive EEF) in real domain now. In this paper, addition and multiplication homomorphism in real domain based on a modified ElGamal are proposed. Then using Taylor series we expand elementary functions approximately into polynomials, which can be encrypted and computed by the proposed homomorphism non-interactivey. Then we implement non-interactive EEEF in real domain.

Key words: mobile code security; Taylor series; non-interactive evaluation of encrypted elementary functions; homomorphism encryption.

1. Introduction

Mobile code means moving code across the nodes of a network. It is technologies and design paradigms fitted for large scale distributed applications and is characterized by configurability, scalability, and customizability\(^\[1\]\). The promising applications of mobile code exist in active networks, e-commerce based on mobile agent, computing grid and so on. The security threats and current resolutions of mobile code are classified in \[2]\[3\].

It is believed that it is impossible to prevent agent from being tampered without tamper-resistant hardware. Sander and Tschudin\(^\[4]\[5]\) proposed software protection scheme of mobile code. Their scheme is implemented by homomorphism. The main idea of their scheme is: they deem mobile code is composed of functions, if the functions are encrypted and still can be executed then the confidentiality of mobile code would be protected. The difficult of the scheme is to find a cryptosystem to encrypt function and get back the right value of the function simply from the encrypted result. They succeed in non-interactive evaluation of encrypted polynomials with algebraic homomorphism. But their scheme is limited to encrypting polynomials in integer domain.

There are no existed homomorphisms\(^\[6]\[7]\[8]\) can be employed to encrypted elementary functions and implement non-interactive EEEF in real domain. In this paper, we present a modified ElGamal, based on which addition and multiplication homomorphism (AH-MH) in real domain are presented. They can encrypt elementary functions and implement non-interactive EEF in real domain.

2. Modified EIGamal

2.1. Real number expressed by integer

A random \(R_i\) is introduced to express real with positive integer and enforce the security of modified ElGamal. \(R_i = r + 2^w r_i + r_i'\). Let \(r = 2^w r\), \(r_i\) is an integer and can be computed according to definition 2, \(r_i'\) is a random integer of \(w\) bits, which can resist known plaintext attacks on modified ElGamal. \(w\) should be large enough in practice to resist brute force attacks. Plaintext \(M = \text{sign}(M)I.D\) is a real, \(\text{sign}(M)\) is the negative or positive sign of \(M\), \(I\) is positive integer portion and \(D\) is positive decimal fraction. \(\Delta \geq 0\) is precision.

Definition 1 \(\Delta (M_i) = M_i - \Gamma\). \(M_0\) is a positive real number. \(\Gamma\) is practical error.

Definition 2

1) If \(I \neq 0\) and \(D \neq 0\). Then \(R_i = r + 2^w r_i + r_i'\), and

\[
0 \leq \frac{r_i'}{r} < \frac{I2^w}{2^w + w} \leq \Delta \quad \text{...(1)}
\]

\[
0, D < \frac{I(2^w r_i')}{r} \leq 0, D + \Delta \quad \text{...(2)}
\]

2) If \(I \neq 0\) and \(D = 0\). Then \(r_i = 0\), \(R_i = r + r_i'\), and Equation (1) should be satisfied.

3) If \(I = 0\) and \(D \neq 0\). Then \(R_i = 2^w r_i + r_i'\), \(IR_i = R_i\), and
\[ 0 \leq \frac{r_i}{r} < \frac{2^w}{2^{2w} + 1} \leq \Delta \quad \cdots \cdots \cdots \cdots \cdots (3) \]

\[ 0.0 \leq \frac{2^w r_i}{r} = \frac{r_i}{r} \leq 0.0 + \Delta \quad \cdots \cdots \cdots \cdots \cdots (4) \]

4) \( I=0 \) and \( D=0 \). Then \( r_i = 0 \), \( R_i = r_i \), \( IR_i = R_i \), and Equation (3) should be satisfied.

**Theorem 1** \[ M = \text{sign}(M) \Delta D \text{ and } R_i = r_i + \frac{2^w r_i + 1}{r} \] which satisfy definition 2. \( \Delta \) is precision defined in definition 2. \( \Gamma_{\text{max}} \) is the maximum error permitted.

\[ 0 \leq \Gamma \leq \Gamma_{\text{max}} \]. Then \( |M| = \Delta \left( \frac{IR_i}{r} \right) \), \( \Delta = \frac{\Gamma_{\text{max}}}{2} \).

**Theorem 2** \( M_i = \text{sign}(M_i) I_{D_1} \), \( M_2 = \text{sign}(M_2) I_{D_2} \), \( R_i = r_i + 2^w r_i + r_i \), and \( R_i = r_i + 2^w r_i + r_i \), which satisfy definition 2. \( \Delta \) is precision defined in definition 2. \( \Gamma_{\text{max}} \) is the maximum error permitted. \( 0 \leq \Gamma \leq \Gamma_{\text{max}} \)

Then \( |M_1 M_2| = \Delta \left( \frac{R_i I_1 R_i I_2}{r^2} \right) \), \( \Delta = \frac{\Gamma_{\text{max}}}{2(I_1 D_1 + I_2 D_2)} \).

### 2.2. Modified ElGamal

We modify ElGamal into algorithm 1 and enable it to encrypt real number.

**Algorithm 1** (Modified ElGamal). Choose a large prime \( p \), a generator \( g \) \((g)p\) of a cyclic group \( Z_p^* \), pick a random \( x \in Z_p^* \), compute \( y = g^x \mod p \). \( R_i, r_i = r_i + 2^w r_i + r_i \) and \( R_i, r_i = r_i + 2^w r_i + r_i \) according to the requirements of error \( \Gamma_{\text{max}} \) and security. Public key: \( (p, g, y, u, w, \Gamma_{\text{max}}) \). Private key: \( x \).

Choose random \( k \in Z_p^* \), \( r_i \in Z_p^* \). Plaintext \( M = \text{sign}(M) I D \). Compute \( r_i \) from (2), (4) or \( r_i = 0 \), compute \( R_i = r_i + 2^w r_i + r_i \) and \( R_i, r_i = r_i + 2^w r_i + r_i \) according to definition 2. \( 0 \leq R_i, r_i \leq \frac{p-1}{2} \) should be satisfied.

**Encryption:**

1) \( M > 0 \), \( E_{k,R}(M) = (a, b) = (g^k \mod p, y^k R_i \mod p) \)

2) \( M > 0 \), \( E_{k,R}(M) = (a, b) = (g^k \mod p, y^k (p-R_i) \mod p) \)

**Decryption:**

1) \( 0 \leq b(a')^1 \mod p \leq \frac{p-1}{2}, \Delta = \text{Det}(\frac{b(a')^{-1}}{r}) \)

2) \( \Delta \leq \frac{b(a')^{-1}}{r} \mod p, \Delta \leq \text{Det}(\frac{b(a')^{-1}}{r}) \)

**Proof:**

1) \( M > 0, E_{k,R}(M) = (a, b) = (g^k \mod p, y^k R_i \mod p) \)

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

\[ \Delta = \frac{b(a')^{-1}}{r} \mod p, \Delta = \frac{b(a')^{-1}}{r} \mod p \]

**Security:** ElGamal could not resist known plaintext attack when same \( k \) is used to encrypt plaintext but the modified ElGamal could do. \( r_i \) should be large enough to resist brute force attacks. Given \( R_i = r_i + 2^w r_i + r_i \), \( R_i = r_i + 2^w r_i + r_i \), \( M_i \geq 0 \), \( E_{k,R}(M_i) = (a, b) = (g^k \mod p, y^k R_i \mod p) \), \( M_2 \geq 0 \), \( E_{k,R}(M_2) = (a, b) = (g^k \mod p, y^k R_2 \mod p) \). \( i \) is known but \( R_i = r_i + 2^w r_i + r_i \) is not known, so that adversary could not computer the inverse of \( R_i, l \), he she could not get \( y^k \) from \( (R_i, l)^{y^k (R_i, l) \mod p} \) and then could not
get $R_1 I_2$ form $y^k b_1 R_2 I_2 \mod p$.

3. Algebraic homomorphism

We present and prove addition homomorphism and multiplication homomorphism as follows:

$|M_1| = \text{Delt}(\frac{R_1 I_1}{r}) = I_1 D_1$, $|M_2| = \text{Delt}(\frac{R_2 I_2}{r}) = I_2 D_2$.

In order to save the pages, we omit modp since then.

3.1. Addition homomorphism (AH)

**Definition 3** $E_{k,R}(M_i) = (a_i, b_i)$, $E_{k,R_2}(M_2) = (a_2, b_2)$.

$E_{k,R_1}(M_i) \oplus E_{k,R_2}(M_2) = (a, b) = (a_1 = a_2, b_1 + b_2)$

**Theorem 3** Addition homomorphism (AH). Given $0 \leq R_1 I_1 \leq \frac{p - 1}{4}$, $0 \leq R_2 I_2 \leq \frac{p - 1}{4}$, Then $E_{k,R}(M_1 + M_2) = E_{k,R_1}(M_1) \oplus E_{k,R_2}(M_2)$ and does not reveal $M_1$ and $M_2$.

**Corollary 1** Multi-Addition homomorphism (m-AH). Given $0 \leq R_1 I_1 \leq \frac{p - 1}{2n}$, $i = 1, \ldots, m \geq 2$. Then $E_{k,R} \left( \sum_{i=1}^{m} M_i \right) = \bigoplus E_{k,R}(M_i)$ and does not reveal $M_i$.

3.2. Multiplication homomorphism (MH)

**Definition 4** $E_{k,R_1}(M_i) = (a_i, b_i) = (g^{k_i} b_1)$, $E_{k,R_2}(M_2) = (a_2, b_2)$.

$E_{k,R_1}(M_i) \circ E_{k,R_2}(M_2) = (a, b) = (a_1 = a_2, b_1 b_2)$

**Theorem 4** Multiplication homomorphism (MH). Given $0 \leq R_1 I_1 \leq \sqrt{\frac{p - 1}{2}}$, $0 \leq R_2 I_2 \leq \sqrt{\frac{p - 1}{2}}$, $k = k_1 + k_2$.

Then $E_{k,R}(M_1 M_2) = E_{k,R_1}(M_i) \circ E_{k,R_2}(M_2)$ and does not reveal $M_1$ and $M_2$.

**Corollary 2** Multi-Multiplication homomorphism (m-MH). Given $0 \leq R_1 I_1 \leq \sqrt{\frac{p - 1}{2n}}$, $i = 1, \ldots, m \geq 2$.

Then $E_{k,R} \left( \prod_{i=1}^{m} M_i \right) = \bigotimes E_{k,R}(M_i)$ and does not reveal $M_i$.

4. Non-interactive EEEF in real domain

Sander et al’s scheme, which can implement non-interactive EEF, is limited to encrypting polynomials in integer domain. In this section, we present scheme and homomorphism to implement non-interactive EEEF in real domain.

4.1. Encrypt polynomials in real domain

**Theorem 5** AH and MH can implement non-interactive evaluation of encrypted polynomials $(p(x) = \sum_{i=0}^{n} c_i x^i)$ in real domain.

Proof: Given $c_i = \text{sign} (c_i) R_i D_i$, $c_i \in \{\text{Real number}\}$, $i \in [0, m]$.

1) Alice encrypts $c_i$ into $E_{k,R_1}(c_i) = (a_i, b_i)$.

Then she encrypts polynomial:

$E_{k,R_1}(p(x)) = (a_1 b_1 + b_1 x + \ldots + b_m x^m) = (a_1, \sum_{i=0}^{m} b_i x^i)$

2) Alice sends $E_{k,R_1}(p(x)) = (a_1, \sum_{i=0}^{m} b_i x^i)$ to Bob.

3) Bob computes $x^i$ at his input $x$ and encrypts $x^i$ into $E_{k,R_2}(x^i) = (a_2, d_i)$. Bob computes $(a_1 a_2, b_1 d_i) = E_{k,R_1}(c_i) \circ E_{k,R_2}(x^i) = E_{k,R}(c_i x^i) = (a, e_i)$.

Bob computes $a = \sum_{i=0}^{m} c_i$ and $b = \sum_{i=0}^{m} b_i x^i$.

4) Bob sends $(a, \sum_{i=0}^{m} e_i) = E_{k,R}(\sum_{i=0}^{m} c_i x^i) = E_{k,R}(p(x))$ to Alice.

5) Alice decrypts $(a, \sum_{i=0}^{m} e_i) = E_{k,R}(\sum_{i=0}^{m} c_i x^i) = E_{k,R}(p(x))$ and obtains $\sum_{i=0}^{m} c_i x^i = p(x)$.

Elementary functions $f(x)$ in real domain can be expanded into Taylor series $p(x) = \sum_{i=0}^{m} c_i x^i$ in their interval of convergence. $p(x)$ can be encrypted by algebraic homomorphism in real domain, such as AH and MH. Then the elementary functions $f(x)$ will be encrypted indirectly and the non-interactive EEEF would be implemented by AH, PH and Theorem 5.

4.2. Taylor series of elementary functions

Basic elementary functions include power function $(y = x^u, u$ is a constant), exponential function $(y = a^x, a > 0, a \neq 1)$, logarithm function $(y = \log^x, a > 0, a \neq 1)$, trigonometric function $(y = \sin x, y = \cos x, y = \tan x, y = \cot x, y = \sec x, y = \csc x)$, inverse trigonometric function $(y = \arcsin x, y = \arccos x)$. We expand basic elementary functions into Taylor series as follows:

1) Power function $y = x^u, (u$ is a constant). If $u$ is not an integer $f(x) = x^u$. 

f(x)=1+u(x-1)+\ldots+\frac{u(u-1)\ldots(u-n+1)}{n!}(x-1)^n+\ldots
=1+c_1(x-1)+c_2(x-1)^2+\ldots+c_n(x-1)^n+\ldots=p(x-1), x\in (0,2).

2) Exponential function y=a^x, (a>0,a \neq 1).

f(x)=1+\ln x+\left(\frac{(\ln a)^2}{2!}\right)x^2+\ldots+\left(\frac{(\ln a)^n}{n!}\right)x^n+\ldots
=1+c_1x+c_2x^2+\ldots+c_nx^n+\ldots=p(x), x\in (-\infty, +\infty) .

3) Logarithm function y=\log_a z, (a>0,a \neq 1).

\[ f(x) = \frac{1}{\ln a} \left( (x-1) - \frac{1}{2} (x-1)^2 + \ldots + \frac{(-1)^{n-1}}{n} (x-1)^n + \ldots \right) \]
\[ = c_1(x-1) + c_2(x-1)^2 + \ldots + c_n(x-1)^n + \ldots = p(x-1), x \in (0,2). \]

4) Trigonometric function y=\sin x, y=\cos x.

\[ f(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots + \frac{(-1)^{n-1}}{(2n-1)!} x^{2n-1} + \ldots \]
\[ = c_1(x-1) + c_2(x-1)^2 + \ldots + c_n(x-1)^n + \ldots = p(x), x \in (-\infty, +\infty). \]

y=\cos x and other trigonometric functions can be expanded into Taylor series similarly.

5) Inverse Trigonometric function y=\arcsin x

\[ f(x) = x + \frac{1}{2} \cdot 3 \cdot 5 \ldots \frac{1}{2(n-1)} x^{2n-1} + \ldots \]
\[ = c_1x + c_2x^3 + \ldots + c_nx^n + \ldots = p(x), x \in [-1,1]. \]

y=\arcsin x and other inverse trigonometric functions can be expanded into Taylor series similarly.

In mathematics, an elementary function is a function built from a finite number of exponentials, logarithms, constants, one variable, trigonometric functions and their inverses, and power functions through composition and combinations using the four elementary operations (+ – \times \div). Elementary functions and their inverses, trigonometric functions built from a finite number of exponentials, logarithms, constants, one variable, trigonometric function can be expanded into Taylor series similarly.

References