Proxy Re-Encryption based on the Generalized ElGamal encryption scheme

Demba Sow[1] and Mamadou Makhtar LO[2]

Département de Mathématiques et Informatique, FST, UCAD[1]
demba1.sow@ucad.edu.sn[1]
Section Mathématiques Appliquées, UFR SAT, UGB[2]
lo.mamdou-makhtar@ugb.edu.sn[2]

January 6, 2021

Contents

Abstract

1 Introduction

Contributions: Our main aim is ...

-

-

-

Related works: In [GAH05], ...
In [ElG85], ...
In [SS11], ...
In [CS03], ...

Outline: This paper is organized as follows:

- In Section 2, ...
- In Section 3, ...
- In Section 4, ...

2 Recalls

2.1 Proxy Encryption

2.2 The "lite" Cramer-Shoup Encryption

2.3 The Generalized ElGamal Encryption

We give a key generation mechanism and a public key encryption algorithm [SS11], which can be view as a slight modification of ElGamal's schemes [ElG85].

Key generation algorithm. To create a public/private key, we do the following:

- Select a cyclic group G with sufficiently large order d such that $G = \langle g \rangle$.
- Select two random integers r and k sufficiently large such that $2 < k < d$ and r of size half the size of d and compute kd.
- Compute with Euclidean division algorithm, the pair (s, t) such that $kd = rs + t$ where $t = kd \mod s$.
- Compute $\gamma = g^s$ and $\delta = g^t$ in G; Note that $\gamma \neq 1$ and $\delta \neq 1$.

Then public key is $((\gamma, \delta), G)$ and the private key is (r, G).

Encryption algorithm. To encrypt a message with the public key $((\gamma, \delta), d, G)$, we do the following:

- Select a random integer $2 < \alpha < d = \#G$ such that α and $\#G$ are co-prime.
- Compute $c_1 = \gamma^\alpha$ and $\lambda = \delta^\alpha$ in G, hence $c_1 \neq 1$ and $\lambda \neq 1$.
- Transform the message m as an element of G and compute $c_2 = \lambda m$ in G.

The ciphertext is (c_1, c_2).

Decryption algorithm. To decrypt a ciphertext (c_1, c_2) encrypted with the public key $((\gamma, \delta), d, G)$ and knowing the associate secret key (r, G), we just need to compute $c_1^r c_2$.

3 The "lite" Cramer-Shoup variant

3.1 First Attempt

Key Generation

- Compute $n = pq$ such that $p = 2p' + 1$ and $q = 2q' + 1$ are two safe primes. Note that the master secret key is the factorization of $n = pq$.
- Select a random $a \in \mathbb{Z}_{n^2}^*$ and compute a generator g of order $\lambda(n) = 2p'q'$ such that $g = -a^{2n} \mod n^2$.
- Select the "weak" secret key is $x \in [1, n^2/2]$ and compute $h = g^x \mod n^2$.
- The public key is $pk = (g, h, n)$ and the secret key is $sk = (x, n)$.

Encryption To encrypt a message $m \in \mathbb{Z}_n^*$ with $pk = (g, h, n)$.

- Choose a random $r \in [1, n/4]$.
- Compute $T_1 = g^r \mod n^2$ and $T_2 = h^r(1 + mn) \mod n^2$.
- Output the ciphertext (T_1, T_2).

Decryption To decrypt a ciphertext (T_1, T_2).

- If x is known, then the message can be recovered as $m = L(T_2/T_1^x \mod n^2)$, where $L(u) = \frac{u-1}{n}$, for all $u \in \{u < n^2 | u = 1 \mod n\}$.
- If (p, q) are known, then m can be recovered from T_2 by noticing that $T_2^{\lambda(n)} = g^{\lambda(n)xr}(1+m\lambda(n)n) = (1+m\lambda(n)n)$. Thus, given that $gcd(\lambda(n), n) = 1$, m can be recovered as: $L(T_2^{\lambda(n)} \mod n^2)[\lambda(n)]^{-1} \mod n$.

3.2 Second Attempt

To minimize a user's secret storage and thus become key optimal, we present the BBS [MBS98], El Gamal based [ElG85] scheme operating over two groups G_1, G_2 of prime order q with a bilinear map $e : G_1^2 \longrightarrow G_2$. The system parameters are random generators $g \in G_1$ and $Z = e(g, g) \in G_2$.

Key Generation (KG)**.** A user $\mathcal{A}$'s key pair is of the form $pk_a = g^a$, $sk_a = a$.

Re-Encryption Key Generation (RG)**.** A user $\mathcal{A}$ delegates to $\mathcal{B}$ by publishing the re-encryption key $rk_{\mathcal{A}\to\mathcal{B}} = g^{b/a} \in G_1$, computed from $\mathcal{B}$'s public key.

First-Level Encryption (E_1)**.** To encrypt a message $m \in G_2$ under pk_a in such a way that it can only be decrypted by the holder of sk_a, output $c = (Z^{ak}, mZ^k)$.

Second-Level Encryption (E_2)**.** To encrypt a message $m \in G_2$ under pk_a in such a way that it can be decrypted by $\mathcal{A}$ and her delegatees, output $c = (g^{ak}, mZ^k)$.

Re-Encryption (R)**.** Anyone can change a *second-level* ciphertext for $\mathcal{A}$ into a *first-level* ciphertext for $\mathcal{B}$ with $rk_{\mathcal{A}\to\mathcal{B}} = g^{b/a}$. From $c_a = (g^{ak}, mZ^k)$, compute $e(g^{ak}, g^{b/a}) = Z^{bk}$ and publish $c_b = (Z^{bk}, mZ^k)$.

Decryption (D_1)**.** To decrypt a *first-level* ciphertext $c_a = (\alpha, \beta)$ with $sk = a$, compute $m = \beta/\alpha^{1/a}$.

3.3 Third Attempt

Key Generation (KG)**.**

Re-Encryption Key Generation (RG)

First-Level Encryption (E_1)**.**

Second-Level Encryption (E_2)**.**

Re-Encryption (R)**.**

Decryption (D_1, D_2)**.**

4 The Generalized ElGamal variant

4.1 First Attempt

Key Generation

- Compute $n = pq$ such that $p = 2p' + 1$ and $q = 2q' + 1$ are two safe primes. Note that the master secret key is the factorization of $n = pq$.
- Select a random $\mu \in \mathbb{Z}^*_{n^2}$ and compute a generator g of order $\lambda(n) = 2p'q'$ such that $g = -\mu^{2n} \mod n^2$.
- Select random elements $k \in [1, n^2/2]$ and $x \in [1, n^2/4]$.
- Compute $y = \lfloor \frac{k\lambda(n)}{x} \rfloor$ and $z = k\lambda(n) \mod x$ such that $k\lambda(n) = xy + z$.
- Compute $b = g^y \mod n^2$ and $c = g^z \mod n^2$.
- The public key is $pk = (b, c, n)$ and the secret key is $sk = (x, n)$.

Encryption To encrypt a message $m \in \mathbb{Z}^*_n$ with $pk = (b, c, n)$.

- Choose a random $r \in [1, n/4]$.
- Compute $u_1 = b^r \mod n^2$ and $u_2 = c^r(1 + mn) \mod n^2$.
- Output the ciphertext (u_1, u_2).

Decryption To decrypt a ciphertext (u_1, u_2).

- If x is known, then the message can be recovered as $m = L(u_2 u_1^x \mod n^2)$, where $L(v) = \frac{v-1}{n}$, for all $v \in \{v < n^2 | v = 1 \mod n\}$.
- If (p, q) are known, then m can be recovered from u_2 by noticing that $u_2^{\lambda(n)} = g^{\lambda(n)zr}(1 + m\lambda(n)n) = (1 + m\lambda(n)n)$. Thus, given that $gcd(\lambda(n), n) = 1$, m can be recovered as: $L(u_2^{\lambda(n)} \mod n^2)[\lambda(n)]^{-1} \mod n$.

Correctness

- $L(u_2 u_1^x) = \dfrac{u_2 u_1^x - 1}{n} = \dfrac{c^r(1 + mn)b^{rx} - 1}{n} = \dfrac{g^{zr}(1 + mn)g^{xyr} - 1}{n} = \dfrac{g^{r(xy+z)}(1 + mn) - 1}{n} = \dfrac{g^{rk\lambda(n)}(1 + mn) - 1}{n} = \dfrac{(1 + mn) - 1}{n} = m.$

- $$\begin{aligned} L\left(u_2^{\lambda(n)} \mod n^2\right)[\lambda(n)]^{-1} &= \left(\frac{u_2^{\lambda(n)}-1}{n}\right)[\lambda(n)]^{-1} \\ &= \left(\frac{g^{\lambda(n)zr}(1+m\lambda(n)n)-1}{n}\right)[\lambda(n)]^{-1} \\ &= \left(\frac{1+m\lambda(n)n-1}{n\lambda(n)}\right) \\ &= m. \end{aligned}$$

4.2 Second Attempt

Let G_1 and G_2 be two groups of prime order d with a bilinear map $e : G_1^2 \longrightarrow G_2$. The system parameters are random generators $g \in G_1$ and $Z = e(g, g) \in G_2$.

Key Generation (KG)**.** A user $\mathcal{A}$'s key pair is of the form $pk_{\mathcal{A}} = (g^s, g^t)$, $sk_{\mathcal{A}} = q$ where $k \in \mathbb{Z}_p$ and $q \in \mathbb{Z}_p$ are two random elements such that $kd = qs+t$.

Re-Encryption Key Generation (RG)**.** A user $\mathcal{A}$ delegates to $\mathcal{B}$ by publishing the re-encryption key $rk_{\mathcal{A}\to\mathcal{B}} = g^{t'/t} \in G_1$, computed from $\mathcal{B}$'s public key.

First-Level Encryption (E_1)**.** To encrypt a message $m \in G_2$ under $pk_{\mathcal{A}}$ in such a way that it can only be decrypted by the holder of $sk_{\mathcal{A}}$, output $c = (Z^{str}, mZ^{s^2 r})$ where $r \in G_1$ is a random element.

Second-Level Encryption (E_2)**.** To encrypt a message $m \in G_2$ under $pk_{\mathcal{A}}$ in such a way that it can be decrypted by $\mathcal{A}$ and her delegatees, output $c = (g^{tr}, mZ^r)$ where $r \in G_1$ is a random element.

Re-Encryption (R)**.** Anyone can change a *second-level* ciphertext for $\mathcal{A}$ into a *first-level* ciphertext for $\mathcal{B}$ with $rk_{\mathcal{A}\to\mathcal{B}} = g^{t'/t}$. From $c_{\mathcal{A}} = (g^{tr}, mZ^r)$, compute $e(g^{tr}, g^{t'/t}) = Z^{t'r}$ and publish $c_{\mathcal{B}} = (Z^{t'r}, mZ^r)$ where $r \in G_1$ is a random element.

Decryption (D_1)**.** To decrypt a *first-level* ciphertext $c_{\mathcal{A}} = (\alpha, \beta)$ with $sk_{\mathcal{A}} = q$, compute $m = \beta\alpha^{1/q}$.

Correctness

4.3 Third Attempt

Key Generation (KG)**.**

Re-Encryption Key Generation (RG)**.**

First-Level Encryption (E_1)**.**

Second-Level Encryption (E_2)**.**

Re-Encryption (R)**.**

Decryption (D_1, D_2)**.**

Correctness

Conclusion

References

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

[ElG85] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO, IT-31(4)*, volume 4, pages 469–472, 1985.

[GAH05] M. Green G. Ateniese, K. Fu and S. Hohenberger. Improved proxy re-encryptionschemes with applications to secure distributed storage. In *In NDSS*, pages 29–43, 2005.

[MBS98] G. Bleumer Matt Blaze and M. Strauss. Divertible protocols and atomic proxy cryptography. In *In Proceedings of Eurocrypt 1998*, volume 1403, pages 127–144, 1998.

[SS11] Demba Sow and Djiby Sow. A new variant of el gamal's encryption and signatures schemes. *JP Journal of Algebra, Number Theory and Applications*, 20(1):21–39, 2011.