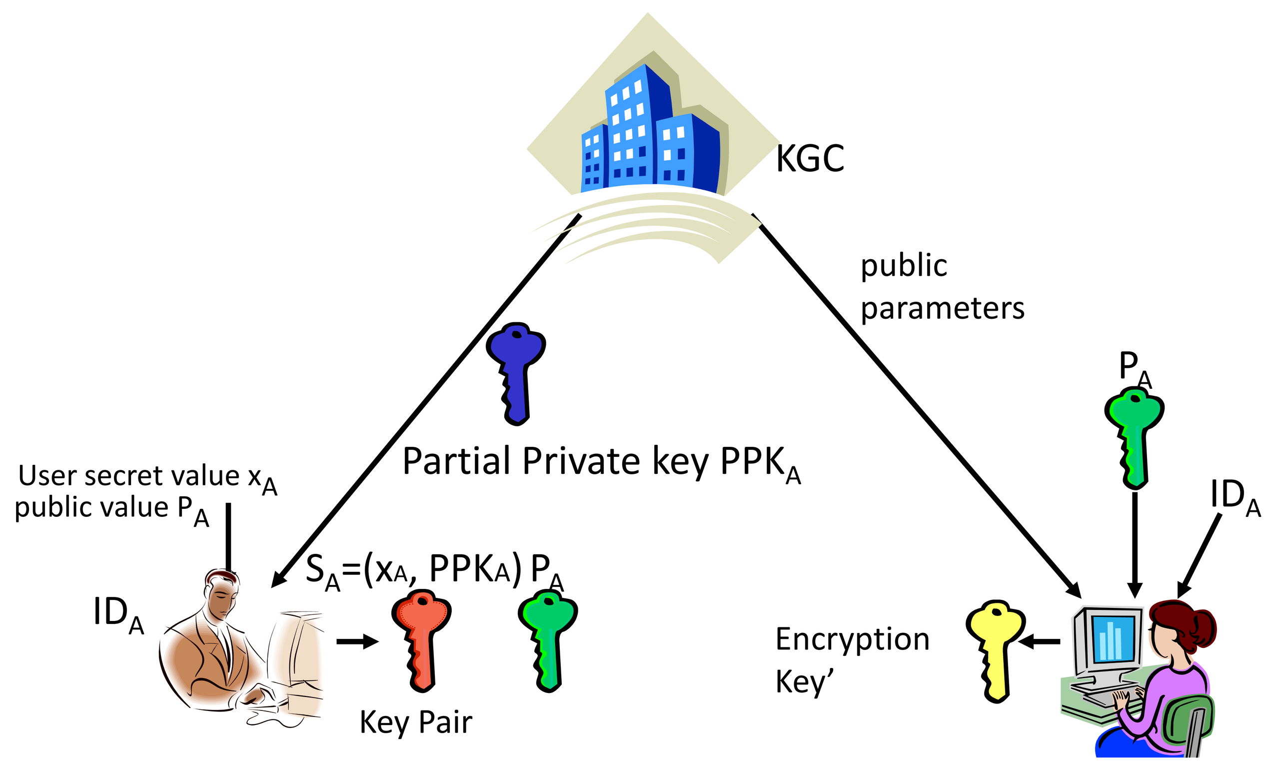
An Efficient Certificateless Cryptography Scheme without Pairing

Seung-Hyun Seo, Mohamed Nabeel, Xiaoyu Ding, Elisa Bertino
Purdue University

1. Certificateless Public Key Cryptography

- CL-PKE: Certificateless Public Key Encryption



- Goals of CL-PKE
  1) To solve the certificate management problem of traditional PKC
  2) To solve the key escrow problem of ID based PKC

2. Mediated CL-PKE without pairing

- mCL-PKE: Mediated Certificateless Public Key Encryption

User

2. SetPrivateKey

$z_A \in \mathbb{Z}_q^*$

3. SetPublicKey

$U_A = g^{z_A}$

7. User-Decrypt

Compute $C_1^{z_A}$
$C_1^{z_A} = g^{r \cdot z_A} = g^{z_A \cdot r} = U_A^{\,r}$

Parse M' and σ' from $M'||\sigma' = H_4(C_1^{z_A}) \oplus C_2'$

Compute $r' = H_3(M', \sigma', ID_A, U_A)$ and $g^{r'}$

Check whether $g^{r'} = C_1$

KGC

1. SetUp

Pick a generator g of $\mathbb{Z}_p^*$ with order q.

Select $x \in \mathbb{Z}_q^*$ uniformly at random and compute $y = g^x$.
Choose cryptographic hash functions $H_1 : \{0,1\}^* \times \mathbb{Z}_p^* \to \mathbb{Z}_q^*$, $H_2 : \{0,1\}^* \times \mathbb{Z}_p^* \times \mathbb{Z}_p^* \to \mathbb{Z}_q^*$, $H_3 : \{0,1\}^* \to \mathbb{Z}_q^*$, $H_4 : \mathbb{Z}_p^* \to \{0,1\}^{n+k_0}$, $H_5 : \mathbb{Z}_p^* \to \{0,1\}^{n+k_0}$, and $H_6 : \mathbb{Z}_p^* \times \{0,1\}^{n+k_0} \times \mathbb{Z}_p^* \times \{0,1\}^{n+k_0} \to \mathbb{Z}_q^*$, where n, k_0 are the bit-length of a plaintext and a random bit string, respectively.

4. SEM-Key Extract

selects $s_0, s_1 \in \mathbb{Z}_q^*$

$w_0 = g^{s_0}$, $w_1 = g^{s_1}$, $d_0 = s_0 + xH_1(ID_A, w_0)$, $d_1 = s_1 + xH_2(ID_A, w_0, w_1)$. KGC sets d_0 as the SEM-key for A. After A proves the knowledge of the secret value z_A such that $U_A = g^{z_A}$, KGC sets (U_A, w_0, w_1, d_1) as the A's public keys.

SEM

6. SEM-Decrypt

Check that ID_A is a legitimate user whose key has not been revoked.

Compute $C_1^{d_0}$.
$C_1^{d_0} = g^{r \cdot d_0} = g^{r \cdot (s_0 + xH_1(ID_A, w_0))}$
$= g^{r \cdot s_0} \cdot g^{r \cdot xH_1(ID_A, w_0)} = w_0^{\,r} \cdot y^{r \cdot H_1(ID_A, w_0)}$

Compute $C_2 \oplus H_5(C_1^{d_0})$
$C_2 \oplus H_5(C_1^{d_0})$
$= (M||\sigma) \oplus H_4(U_A^{\,r}) \oplus H_5(w_0^r \cdot y^{H_1(ID_A, w_0) \cdot r}) \oplus H_5(C_1^{d_0})$
$= (M||\sigma) \oplus H_4(U_A^{\,r}) \oplus H_5(w_0^r \cdot y^{H_1(ID_A, w_0) \cdot r}) \oplus H_5(w_0^r \cdot y^{H_1(ID_A, w_0) \cdot r})$
$= (M||\sigma) \oplus H_4(U_A^{\,r})$

Check whether $C_3 = H_6(U_A, C_2 \oplus H_5(C_1^{d_0}), C_1, C_2)$.

Encryptor

5. Encrypt

Check whether $g^{d_1} = w_1 \cdot y^{H_2(ID_A, w_0, w_1)}$.
If the checking result is not valid, encryption algorithm must be aborted.

Choose $\sigma \in \{0,1\}^{k_0}$ and compute $r = H_3(M, \sigma, ID_A, U_A)$.

Compute $C_1 = g^r$.

Compute $C_2 = (M||\sigma) \oplus H_4(U_A^{\,r}) \oplus H_5(w_0^r \cdot y^{H_1(ID_A, w_0) \cdot r})$.

Compute $C_3 = H_6(U_A, (M||\sigma) \oplus H_4(U_A^r), C_1, C_2)$.

- Drawbacks of previous work
  1) Inefficient pairing based approach
  2) Weak Security – CPA(Chosen Plaintext Attack), Partial decryption attack

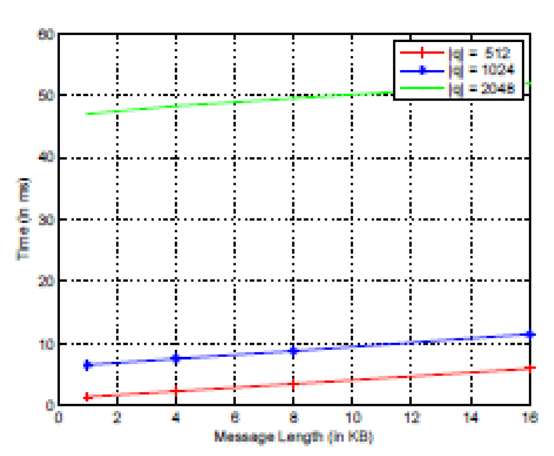
- Key features of our mCL-PKE without pairings
  1) Instantaneous revocation of compromised public keys using Security Mediator(SEM)
  2) Solution of the key escrow problem and certificate management problem based on CL-PKC
  3) Efficiency based on pairing-free approach
  4) Security against CCA (Chosen Ciphertext Attack) and Partial decryption attack

3. Experimental Results

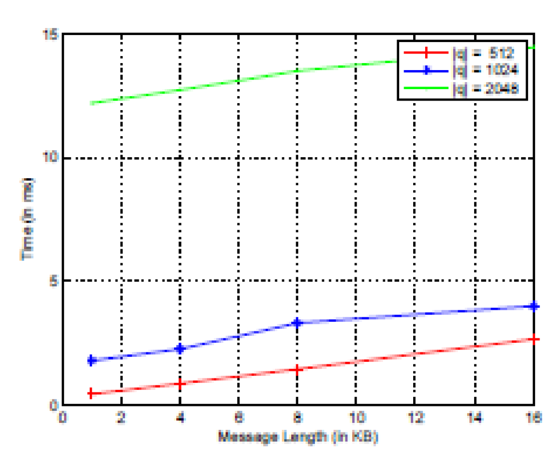
- The experimental environment

CPU	Memory	OS	Program Lang.	Library
Intel Core™ i5-2430 CPU @ 2.40GHZ	8 GBytes memory	32 bits GNU Linux kernel	C/C++	NTL library version 5.5.2

- Encryption and decryption times of the mCL-PKE for different message size

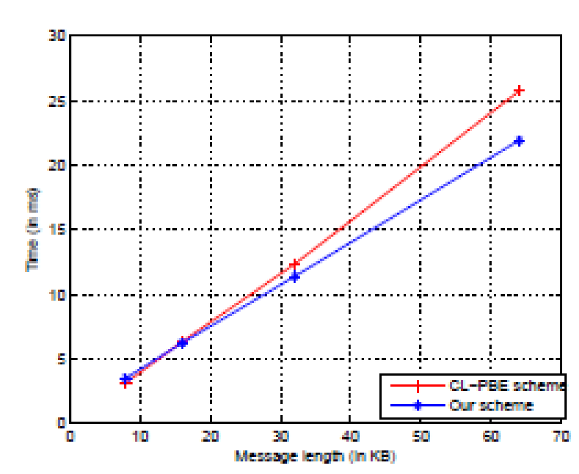


(a) mCL-PKE Encryption

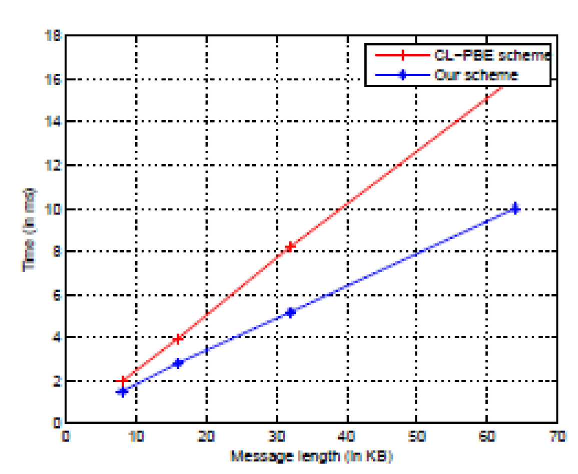


(b) mCL-PKE Decryption

- Performance comparison with a recent pairing based scheme



(c) Encryption Comparison

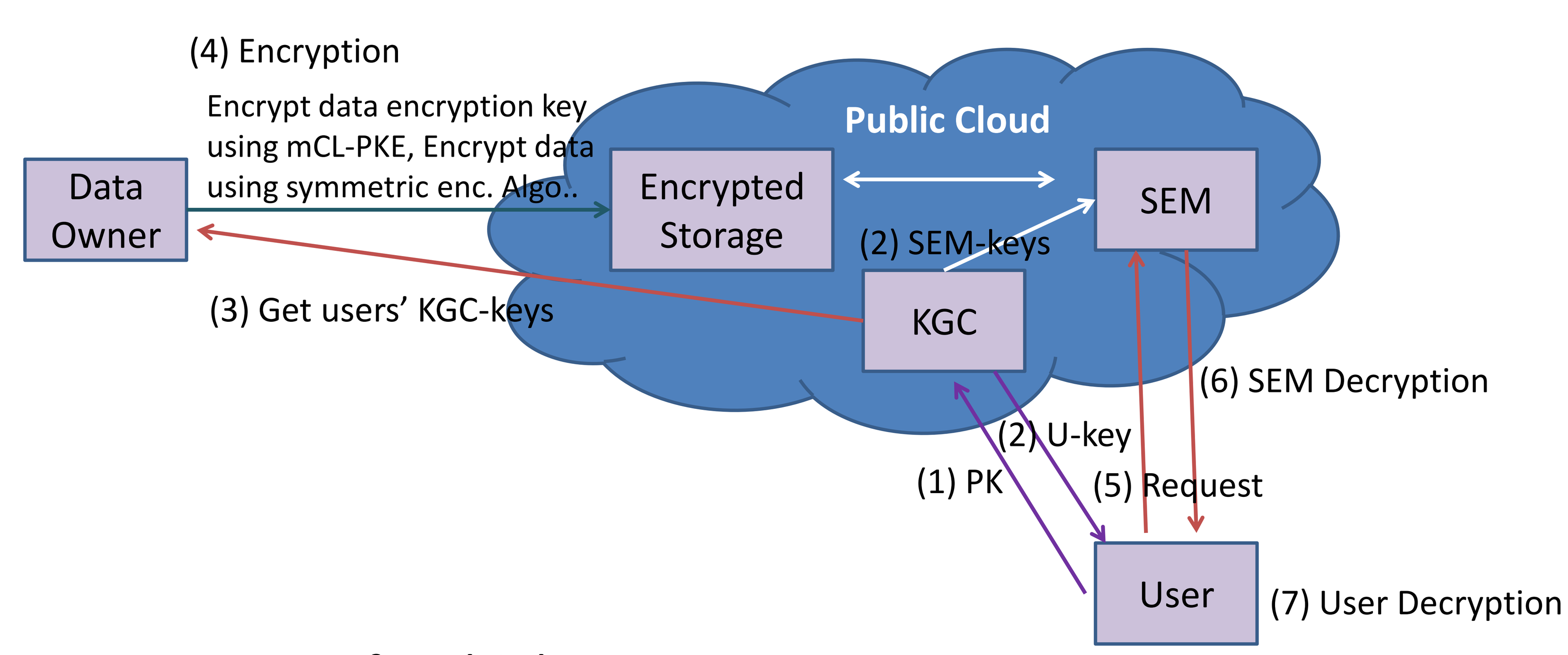


(d) Decryption Comparison

4. Discussions and Future Work

Application Scenario

- Secure data sharing for public cloud computing services



- In case of multiple users,
  → bottleneck problem:
  : The data owner must encrypt the same data encryption key multiple times.