Yang LU, Ji-guo LI, 2019. Constructing pairing-free certificateless public key encryption with keyword search. *Frontiers of Information Technology & Electronic Engineering*, 20(8):1049-1060. https://doi.org/10.1631/FITEE.1700534

Constructing pairing-free certificateless public key encryption with keyword search

Key words: Searchable public key encryption; Certificateless public key encryption with keyword search; Bilinear pairing; Computational Diffie-Hellman problem

Corresponding author: Yang LU
E-mail: luyangnsd@163.com

ORCID: http://orcid.org/0000-0003-4860-8384

Motivation

- In practice, cryptographic operations are often performed on some devices with constrained resources, such as smart phone and personal digital assistant (PDA). Due to the limited computation performance or battery power, only the lightweight or power-saving cryptographic schemes can be employed on these devices.
- The previous certificateless encryption with keyword search (CLEKS) schemes have to be based on the costly bilinear pairing.
- In practical implementations, bilinear pairing is less efficient than other common cryptographic operations and will greatly increase the overload of the systems. Therefore, a CLEKS scheme without using bilinear pairing would be more attractive in terms of computational cost.

Main idea

- Our CLEKS scheme is constructed over an elliptic curve group.
- To avoid the costly pairing operation, we design the partial key generation algorithm based on the Schnorr signature scheme and the keyword encryption algorithm based on the hashed ElGamal encryption scheme.

The proposed scheme

1. Setup(λ): Generate public system parameters psp={q, G, P, P_{pub}, H_1, H_2, H_3} and the master key mk=s.
2. PartialKeyGen(psp, mk, ID_U): Generate a partial public key PPK_U= xP and a partial private key $PSK_U = x+sH_1(ID_U)$ for user U.
3. UserKeyGen(psp, ID_U, PPK_U, PSK_U): Generate a full private key SK_U= (SK_{U1}, SK_{U2})=($x+sH_1(ID_U)$, y) and a full public key PK_U=(PK_{U1}, PK_{U2})=(xP, yP) for user U.
4. Encrypt(psp, w, ID_U, PK_U): Generate a keyword ciphertext C_w= (C_{w1}, C_{w2})=(rP, $H_3(rH_2(w)(PK_{U1}+PK_{U2}+H_1(ID_U)P_{pub}))$).
5. Trapdoor(psp, w, ID_U, SK_U): Generate a keyword trapdoor T_w=(SK_{U1}+SK_{U2})$H_2(w)$.
6. Test(psp, C_w, $T_{w'}$): If $C_{w2}=H_3(T_{w'}C_{w1})$, output 1; else, output 0.

Security conclusion

- **Theorem 2** (IND-CKA security) The proposed CLEKS scheme achieves IND-CKA security under the complexity assumption of the CDH problem in the random oracle model.

Efficiency comparison

- The experimental results show that our scheme enjoys better performance than the previous pairing-based CLEKS schemes.

Table 4 Simulation results of the compared CLEKS schemes

Scheme	Computation cost (ms)			Communication cost (bit)	
	Encrypt	Trapdoor	Test	Ciphertext size	Trapdoor size
PCPY-CLEKS	172.92	28.60	32.79	768	1536
ZLA-CLEKS	34.94	51.03	80.15	2048	2048
IORA-CLEKS	34.94	9.42	59.22	1536	512
Ours	6.85	0.22	2.21	1024	160

Three extensions

- Multi-receiver CLEKS scheme
- Conjunctive-keyword CLEKS scheme
- Designated server CLEKS scheme

Conclusions

- We have presented a pairing-free CLEKS scheme and formally proved its security under the complexity assumption of the CDH problem in the random oracle model.
- Compared with the previous pairing-based CLEKS schemes, our scheme enjoys obvious advantages in both computation performance and communication bandwidth.
- We have further presented three extensions of the proposed CLEKS scheme.
- In our future research, we may consider building a provably secure CLEKS scheme without random oracles. In addition, it would be interesting to devise CLEKS schemes that support authorized keyword search.