SCIENCE CHINA
Information Sciences

• **Supplementary File** •

A CCA Secure Public Key Encryption Scheme Based on Finite Groups of Lie Type

HONG HaiBo[1*], SHAO Jun[1], WANG LiCheng[2], XIE ManDe[1], WEI GuiYi[1], YANG YiXian[2], HAN Song[2] & LIN JianHong[3]

[1]*School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou* 310018*, China;*
[2]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing* 100876*, China;*
[3]*Zhejiang Ponshine Information Technology Co.,Ltd., Hangzhou,* 310000*, China*

Appendix A Security Analysis of Our Proposal

By the techniques used in [1, 2], we can prove that our proposal is secure against the chosen chiphertext attacks in the random oracle model assuming that the NAI problem in the finite group of Lie type is hard.

Theorem 1. The proposal is indistinguishable against adaptively chosen ciphertext attacks (IND-CCA) in the random oracle model assuming that the NAI problem is intractable in the finite group of Lie type.

Proof. If there exists an adversary $\mathcal{A}$, who can break the CCA security of the proposal, then we can build another algorithm $\mathcal{B}$ solving the NAI problem in the finite group of Lie type. That is, given $\Delta_1 = \exp^{r_1 \cdot R} \cdot \exp^{t_1 \cdot T} \in \mathbb{G}$, $\Delta_2 = \exp^{r_2 \cdot R} \cdot \exp^{t_2 \cdot T} \in \mathbb{G}$, and $R, T \in \mathbb{M}$, it aims to output $\Delta = \exp^{(r_1+r_2) \cdot R} \cdot \exp^{(t_1+t_2) \cdot T}$. The details are presented as follows.

Setup: $\mathcal{B}$ sets the public values $R, T, \Sigma = \mathbf{\Delta}_1 = \exp^{r_1 \cdot R} \cdot \exp^{t_1 \cdot T}$, respectively. Clearly, $\mathcal{B}$ has no idea about the corresponding private key $sk = (\exp^{r_1 \cdot R}, \exp^{t_1 \cdot T})$.

Phase 1: $\mathcal{B}$ builds the following oracles.

- Random Oracle $\mathcal{O}_{H_1}$: $\mathcal{A}$ sends a random $I_1 \in \{0,1\}^{\kappa_2+\ell}$ to this oracle, $\mathcal{B}$ firstly searches whether (I_1, α) exists in Table T_{H_1} (that would be empty at the beginning). If it exists, $\mathcal{B}$ returns α to $\mathcal{A}$; otherwise, $\mathcal{B}$ chooses a random value α from $\mathbb{F}_p \times \mathbb{F}_p$, records (I_1, α) into Table T_{H_1}, and sends α to $\mathcal{A}$.
- Random Oracle $\mathcal{O}_{H_2}$: $\mathcal{A}$ sends a random $I_2 \in \mathbb{G}$ to this oracle, $\mathcal{B}$ firstly searches whether (I_2, β) exists in Table T_{H_2} (that would be empty at the beginning). If it exists, $\mathcal{B}$ returns β to $\mathcal{A}$; otherwise, $\mathcal{B}$ chooses a random number β from $\{0,1\}^{\kappa_2}$, records (I_2, β) into Table T_{H_2}, and sends β to $\mathcal{A}$.
- Random Oracle $\mathcal{O}_{H_3}$: $\mathcal{A}$ sends a random $I_3 \in \{0,1\}^{\kappa_2}$ to this oracle, $\mathcal{B}$ firstly searches whether (I_3, γ) exists in Table T_{H_3} (that would be empty at the beginning). If it exists, $\mathcal{B}$ returns γ to $\mathcal{A}$; otherwise, $\mathcal{B}$ chooses a random number γ from $\{0,1\}^{\ell}$, records (I_3, γ) into Table T_{H_2}, and sends γ to $\mathcal{A}$.
- Decryption Oracle $\mathcal{O}_{dec}$: $\mathcal{A}$ sends a ciphertext $C = (C_1, C_2, C_3) \in \{0,1\}^{\kappa_2} \times \mathbb{G} \times \{0,1\}^{\ell}$ to this oracle, $\mathcal{B}$ firstly searches $(\sigma, m, \alpha, \beta, \gamma)$ in tables T_{H_1}, T_{H_2} and T_{H_3}, where $(\alpha_r, \alpha_t) = \alpha = H_1(\sigma||m)$, $C_1 = \beta \oplus \sigma$, $C_2 = \exp^{\alpha_r \cdot R} \cdot \exp^{\alpha_t \cdot T}$, and $C_3 = \gamma \oplus m$. If it exists, $\mathcal{B}$ sends m to $\mathcal{A}$; otherwise, $\mathcal{B}$ sends $\perp$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ sends $\mathcal{B}$ two messages $m_0, m_1 \in \{0,1\}^{\ell}$ with equal bit length. $\mathcal{B}$ computes $C^* = (C_1^*, C_2^*, C_3^*)$ follows:

- Choose random σ^*, β^* from $\{0,1\}^{\kappa_2}$, and compute $C_1^* = \beta^* \oplus \sigma^*$.
- Set $C_2^* = \mathbf{\Delta}_2$.
- Compute $C_3^* = H_3(\sigma^*) \oplus m_b$, where b is a random number from $\{0,1\}$.

At last, $\mathcal{B}$ sends C^* to $\mathcal{A}$ as the challenge ciphertext.

Phase 2: It is almost the same as Phase 1, except that $\mathcal{A}$ can not directly send C^* to the decryption oracle $\mathcal{O}_{dec}$.

Guess: $\mathcal{A}$ outputs the guess b' on b. $\mathcal{B}$ randomly chooses I_2 from Table T_{H_2}, and sets Δ as I_2. If $\mathcal{A}$ can output a correct guess, then I_2 is the correct Δ with probability $1/q_{H_2}$ at least, where q_{H_2} denotes the maximum number of queries to the random oracle $\mathcal{O}_{H_2}$ by $\mathcal{A}$.

* Corresponding author (email: honghaibo1985@163.com)

In analogy with the construction of FullIdent in [1], since q_{H_2} is polynomially bounded, so $\mathcal{B}$ breaks NAI assumption with non-negligible probability $1/q_{H_2}$. Specifically, suppose that $\mathcal{A}$'s advantage in guessing $b' = b$ is ϵ which is non-negligible, then $\mathcal{B}$'s advantage in breaking NAI assumption is about ϵ/q_{H_2} which is also non-negligible. According to the classic conclusion in [1, 2], we have that: if no polynomially bounded adversary has a non-negligible advantage in breaking our scheme, the proposal is indistinguishable against adaptively chosen ciphertext attacks (IND-CCA).

Appendix B Efficiency Analysis

In this section, we would like to analyze the efficiency of our proposal and related security parameters (see Table B1). In particular, we have the followings.

Table B1 Performance and security of our proposal

Operations[1)]			Sizes[2)]				Security	
KeyGen	Enc	Dec	Pk	Sk	Ciphertext	CER[3)]	IA[4)]	Goal/Model
3exp+m	2exp+3m	2exp+5m	$4n^2\kappa_1$	$2n^2\kappa_1$	$\kappa_2 + n^2\kappa_1 + l$	$1 + \frac{\kappa_2 + n^2\kappa_1}{l}$	NAI	IND-CCA2/ROM

[1)]exp/m: Matrix exponential/Matrix multiplication.
[2)]$|\mathbb{G}| = \Theta(2^{n^2\kappa_1})$, $|\mathcal{M}| = \Theta(2^l)$ and $\kappa_1 + 1 - \kappa_2 \leqslant l \leqslant \kappa_2$, where n is the rank of finite groups of Lie type.
[3)]CER: Ciphertext expansion ratio.
[4)]IA: Intractable assumptions.

1. Key generation algorithm requires three matrix exponentials of nilpotent matrices Z, R and T, and the core parameters of the public key (pk) and the secret key (sk) are the tetrad $(Z, R, T, \exp^{rR} \cdot \exp^{tT})$ and the pair $(\exp^{rR}, \exp^{tT})$, respectively. They are $4\lceil \log_2 p^{n^2} \rceil \approx 4n^2\kappa_1$ and $2\lceil \log_2 p^{n^2} \rceil \approx 2n^2\kappa_1$ bit length, respectively. Here, we ignore the part of the parameters to describe $\mathbb{M}, \mathbb{G}, H_1, H_2, H_3$.
2. Encryption algorithm requires two matrix exponentials to compute $\exp^{s_r R}$ and $\exp^{s_t T}$, as well as additional three multiplications to get the final ciphertext. Similarly, the cost for evaluating H_1, H_2 and H_3 is ignored without loss of generality. The bit length of one ciphertext is $\kappa_2 + n^2\kappa_1 + \ell$.
3. Decryption algorithm does not require any matrix exponentials but only two multiplications to get the message, while it needs two matrix exponentials and three multiplications to check the validity of the ciphertext. The cost of evaluating hash functions are still ignored.
4. The ciphertext expansion ratio is $1 + \frac{\kappa_2 + n^2\kappa_1}{l}$, which is closely related to κ_1, κ_2 and the rank n of $\mathbb{G}$. The intractable assumption of our scheme is NAI. Recall the analysis of the NAF problem, the hardness is related to p. Hence, $\kappa_1 = \lceil \log_2 p \rceil$. At last, κ_2 could be set as that in [2].

Furthermore, in order to test the performance of our proposal, we conducted the simulations on a 64-bit Windows 7 operating system PC with an Intel(R) Core(TM) i7-6700 CPU@3.40GHz and 16GB RAM by running Dev.C++ and NTL libraries. Here, we reduced the running time of the hash functions H_1, H_2 and H_3 to the classical SHA256 algorithm. The experiments demonstrated that our platform completed 143 SHA256 operations per millisecond (ms). Thus, the running time of the three hash functions can be neglected considering other steps of our proposal. Ultimately, we obtained relevant data by performing 1000 trials on each parameter (see Table B2).

Table B2 Running time under different parameters

Parameter	n	4	5	6	7	8	9
	p	2^{10}	2^{20}	2^{40}	2^{50}	2^{70}	2^{80}
Time (ms)	KeyGen	3.044	8.265	25.478	47.556	102.886	164.69
	Enc	1.413	5.156	19.151	37.142	84.609	137.917
	Dec	1.639	5.484	19.903	38.295	86.652	140.347

References

1 Boneh D, Franklin M. Identity-based encryption from the Weil pairing [C]. Annual international cryptology conference. Springer, Berlin, Heidelberg, 2001: 213-229.
2 Fujisaki E, Okamoto T. Secure integration of asymmetric and symmetric encryption schemes [C]. Annual International Cryptology Conference. Springer Berlin Heidelberg, 1999: 537-554.