Re-randomization Attack on the Certificateless Encryption Scheme proposed by Guo et al.

Nobuyuki Sugio[1*], Keita Emura[2] and Toshihiro Ohigashi[3]

[1*]Information and Computer Science, Hokkaido University of Science, 15-4-1, Maeda 7-jo, Sapporo, Hokkaido, 006-8585, Japan.
[2]Faculty of Electrical, Information and Communication Engineering, Institute of Science and Engineering, Kanazawa University, Kakuma, Kanazawa, Ishikawa, 920-1192, Japan.
[3]Department of Information and Telecommunication Engineering, Tokai University, 2-3-23 Takanawa, Minato-ku, Tokyo, 108-8619, Japan.

*Corresponding author(s). E-mail(s): sugio-n@hus.ac.jp;
Contributing authors: k-emura@se.kanazawa-u.ac.jp;
ohigashi@tokai.ac.jp;

Abstract

Guo, Li, and Qin proposed a lightweight certificateless encryption (CLE) scheme designed for IoT environments (*Discover Computing*, 2025). This paper demonstrates that the proposed scheme does not achieve CCA security, contrary to the authors' claim. Specifically, we identify two critical points. First, since the ciphertext retains a multiplicative ElGamal structure, it can always be re-randomized using arbitrary randomness. Second, based on this property, an adversary can transform a challenge ciphertext into another valid ciphertext of the same plaintext, and then query the decryption oracle with the transformed ciphertext to recover the challenge plaintext. This attack exploits a definitional gap in the CCA game, where only direct decryption queries on the challenge ciphertext are prohibited. In this work, we formalize the attack procedure and verify its validity based on implementation.

Keywords: Certificateless Encryption, Re-randomization Attack, CCA, Bilinear Map, IoT Security

1 Introduction

Certificateless Encryption (CLE) [1] was proposed as a new framework proposed by Al-Riyami and Paterson. In CLE, each user obtains a partial private key from the Private Key Generator (PKG) and generates the remaining part independently to form a complete private key. As a result, the PKG does not fully know the user's private key, thus eliminating the key escrow problem. At the same time, the certificate-free design reduces the burden of certificate management. Consequently, CLE has attracted increasing attention in recent cryptographic research and is particularly well-suited for scenarios such as IoT and cloud environments, where lightweight and flexible key management is required.

Guo, Li, and Qin proposed a new CLE-based scheme [2] (hereafter referred to as the GLQ scheme) and claimed that it achieves CCA security. The GLQ scheme was specifically designed for IoT environments, emphasizing efficiency and ease of implementation.

In this work, we focus on the ciphertext structure of the GLQ scheme, which retains a multiplicative ElGamal-like form. This feature enables straightforward ciphertext re-randomization, leaving potential weaknesses in the game-based definition of CCA security. In particular, an adversary may exploit the current definition, which only prohibits reuse of the identical challenge ciphertext, to mount a re-randomization attack. The main contributions of this work are as follows:

- We revisit the security of the GLQ scheme and construct a concrete re-randomization attack that challenges the claimed CCA security.
- Through implementation, we evaluate the computational cost and runtime of the proposed attack, demonstrating that the theoretical vulnerability can also pose serious risks in practical environments.

The remainder of this paper is organized as follows. Section 2 presents the ciphertext generation part of GLQ scheme, and Section 3 describes the re-randomization attack. Section 4 provides an implementation-based evaluation. Section 5 discusses the GLQ scheme. Finally, Section 6 concludes the paper.

2 Ciphertext Generation of the GLQ Scheme

This section extracts only the ciphertext-generation component of the GLQ scheme [2], which is necessary for the re-randomization attack described in Section 3.

Let p be a prime and $\mathbb{G}_1, \mathbb{G}_T$ be cyclic groups of order p with generator $g \in \mathbb{G}_1$. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ be a bilinear map satisfying

$$e(g_1^a, g_2^b) = e(g_1, g_2)^{ab} \qquad \text{for all } a, b \in \mathbb{Z}_p.$$

Let $H_1 : \mathsf{ID} \to \mathbb{G}_1$ be a hash function.

Let DU and DO denote the identifiers of the data user and the data owner, respectively. Given a valid public key $\mathsf{pk}_{\mathsf{DU}} \in \mathbb{G}_T$ for data user DU, the data owner DO forms a ciphertext $\mathsf{CT} = (c_1, c_2, c_3)$ as follows. Let the plaintext be $\mathsf{M} \in \mathbb{G}_T$. The data owner

DO selects a random $t \in \mathbb{Z}_p$ and computes

$$\begin{aligned} c_1 &= \mathsf{M} \cdot \mathsf{pk}_{\mathsf{DU}}^t, \\ c_2 &= g^t, \\ c_3 &= H_1(\mathsf{DU})^t. \end{aligned} \tag{1}$$

The ciphertext is $\mathsf{CT} = (c_1, c_2, c_3)$.

The data user DU holds a private key $\mathsf{SK}_{\mathsf{DU}} = (\mathsf{sk}_{\mathsf{DU},1}, \mathsf{sk}_{\mathsf{DU},2})$, which is derived from the master secret and the identity information of DU during the key generation phase of the GLQ scheme. For any $x \in \mathbb{Z}_p$, the following Eq. (2) holds by the GLQ scheme's key relations between public key $\mathsf{pk}_{\mathsf{DU}}$ and private key $\mathsf{SK}_{\mathsf{DU}} = (\mathsf{sk}_{\mathsf{DU},1}, \mathsf{sk}_{\mathsf{DU},2})$.

$$\mathsf{pk}_{\mathsf{DU}}^x = \frac{e(\mathsf{sk}_{\mathsf{DU},1}, g^x)}{e(\mathsf{sk}_{\mathsf{DU},2}, H_1(\mathsf{DU})^x)} \tag{2}$$

Using a private key $\mathsf{SK}_{\mathsf{DU}} = (\mathsf{sk}_{\mathsf{DU},1}, \mathsf{sk}_{\mathsf{DU},2})$, the data user DU recovers the plaintext M from the ciphertext $\mathsf{CT} = (c_1, c_2, c_3)$ as follows:

$$\begin{aligned} c_1 \cdot \frac{e(\mathsf{sk}_{\mathsf{DU},2}, c_3)}{e(\mathsf{sk}_{\mathsf{DU},1}, c_2)} &= \mathsf{M} \cdot \mathsf{pk}_{\mathsf{DU}}^t \cdot \frac{e(\mathsf{sk}_{\mathsf{DU},2}, H_1(\mathsf{DU})^t)}{e(\mathsf{sk}_{\mathsf{DU},1}, g^t)} \\ &= \mathsf{M} \cdot \mathsf{pk}_{\mathsf{DU}}^t \cdot \frac{1}{\mathsf{pk}_{\mathsf{DU}}^t} \\ &= \mathsf{M} \end{aligned} \tag{3}$$

3 Re-randomization Attack on the GLQ Scheme

3.1 Re-randomization operator

For any $s \in \mathbb{Z}_p$, define the re-randomization operator

$$\mathsf{ReRand}_s(c_1, c_2, c_3) \;:=\; \big(c_1 \cdot \mathsf{pk}_{\mathsf{DU}}^s,\; c_2 \cdot g^s,\; c_3 \cdot H_1(\mathsf{DU})^s\big).$$

If $\mathsf{CT} = (c_1, c_2, c_3)$ is a valid ciphertext for the data user DU and plaintext M, then $\mathsf{CT}' = \mathsf{ReRand}_s(\mathsf{CT})$ is a different ciphertext (i.e., $\mathsf{CT}' \neq \mathsf{CT}$ for $s \neq 0$) that encrypts the same plaintext M for the same data user DU.

3.2 Correctness

We verify that CT' decrypts to the same plaintext M. From Eq. (2) it follows that

$$\mathsf{pk}_{\mathsf{DU}}^{t+s} \cdot \frac{e(\mathsf{sk}_{\mathsf{DU},2}, H_1(\mathsf{DU})^{\,t+s})}{e(\mathsf{sk}_{\mathsf{DU},1}, g^{\,t+s})} = \mathsf{pk}_{\mathsf{DU}}^{t+s} \cdot \frac{1}{\mathsf{pk}_{\mathsf{DU}}^{t+s}} = 1.$$

Using Eq. (3) and $\mathsf{CT}' = (\mathsf{M} \cdot \mathsf{pk}_{\mathsf{DU}}^{t+s},\ g^{\,t+s},\ H_1(\mathsf{DU})^{\,t+s})$, we obtain

$$\begin{aligned}
\mathsf{M}' &= \left(\mathsf{M} \cdot \mathsf{pk}_{\mathsf{DU}}^{t+s}\right) \cdot \frac{e(\mathsf{sk}_{\mathsf{DU},2}, H_1(\mathsf{DU})^{t+s})}{e(\mathsf{sk}_{\mathsf{DU},1}, g^{t+s})} \\
&= \mathsf{M} \cdot \left(\mathsf{pk}_{\mathsf{DU}}^{t+s} \cdot \frac{e(\mathsf{sk}_{\mathsf{DU},2}, H_1(\mathsf{DU})^{t+s})}{e(\mathsf{sk}_{\mathsf{DU},1}, g^{t+s})}\right) \\
&= \mathsf{M} \cdot \left(\mathsf{pk}_{\mathsf{DU}}^{t+s} \cdot \frac{1}{\mathsf{pk}_{\mathsf{DU}}^{t+s}}\right) \\
&= \mathsf{M}.
\end{aligned}$$

3.3 Breaking CCA of the GLQ Scheme

The following procedure describes how an adversary $\mathcal{A}$ can break the CCA security of the GLQ scheme in interaction with a challenger $\mathcal{C}$:

1. $\mathcal{A}$ prepares two challenge plaintexts M_0 and M_1.
2. $\mathcal{C}$ randomly selects $b \in \{0, 1\}$ and encrypts M_b to produce the challenge ciphertext CT.
3. $\mathcal{A}$ receives CT.
4. $\mathcal{A}$ chooses $s \neq 0$ and computes $\mathsf{CT}' = \mathsf{ReRand}_s(\mathsf{CT})$.
5. $\mathcal{A}$ submits CT' to the decryption oracle and obtains M'_b.
6. By comparing M'_b with M_0 and M_1, $\mathcal{A}$ can always determine b correctly.

In the CCA game, decryption queries for $(\mathsf{DU}, \mathsf{CT})$ are forbidden, but $\mathsf{CT}' \neq \mathsf{CT}$, so the query on CT' is allowed. As a result, adversary $\mathcal{A}$ can always guess b with probability 1. Therefore, the GLQ scheme does not achieve CCA security.

4 Evaluation

This section evaluates the computational cost and runtime of the re-randomization attack described in Section 3.

4.1 Computational Cost

The adversary's work in this attack consists only of a single re-randomization shown in Eq. (4) and one query to the decryption oracle.

$$\mathsf{CT}' = \mathsf{ReRand}_s(\mathsf{CT}) = \left(c_1 \cdot \mathsf{pk}_{\mathsf{DU}}^{s},\ c_2 \cdot g^{s},\ c_3 \cdot H_1(\mathsf{DU})^{s}\right) \quad (s \in \mathbb{Z}_p) \tag{4}$$

No bilinear-pairing evaluation is required on the adversary side. Thus the adversary's computations reduce to the following group operations:

- One exponentiation in $\mathbb{G}_T$ ($\mathsf{pk}_{\mathsf{DU}}^{s}$) and one multiplication in $\mathbb{G}_T$ ($c_1 \cdot \mathsf{pk}_{\mathsf{DU}}^{s}$).
- Two exponentiations in $\mathbb{G}_1$ (g^s and $H_1(\mathsf{DU})^s$) and two multiplications in $\mathbb{G}_1$ ($c_2 \cdot g^s$ and $c_3 \cdot H_1(\mathsf{DU})^s$).

4.2 Experimental Results

For evaluation, we implemented the scheme in a symmetric pairing environment (Type-1) and measured practical runtimes. All experiments were performed on Ubuntu 24.04.2 LTS with an AMD Ryzen^{TM} 5800X (8 cores), 128 GiB RAM, and 9.0 TB storage. We used the PBC library [3] with the 128-bit security parameter set from [4]. Table 1 reports averaged runtimes. Measured times are averages over 10,000 runs. Pairing and group initialization were included in the measured timings. Compiler optimizations were disabled and default library settings were used.

Table 1 Average execution time over 10,000 runs using PBC (Type-1, a128.param)

Operation	**PBC (Type-1)**
Setup	9,08 ms
PSKeyGen ($\times 2^*$)	51,21 ms
SKeyGen ($\times 2^*$)	67,55 ms
PKVerify ($\times 2^*$)	1,91 ms
Encrypt	26,06 ms
Decrypt	10,91 ms
Re-randomize	26,09 ms
Decryption oracle	10,91 ms

※ PSKeyGen, SKeyGen, and PKVerify are conducted two times for data owner DO and data user DU.

The measured attack cost confirms the theoretical assessment: an adversary performs the re-randomization attack with three exponentiations and three multiplications. The runtime of this attack is nearly identical to that of Eq. (1) and remains practical from a performance perspective. Because the adversary does not require pairing evaluations, the attack remains fast regardless of pairing type used in the GLQ implementation.

5 Discussion

As demonstrated by our re-randomization attack, the GLQ scheme fails to achieve the claimed CCA security. The core of this weakness lies in its ciphertext structure, which retains a multiplicative ElGamal-like form. Because of this algebraic property, any ciphertext can be re-randomized into another valid ciphertext that encrypts the same plaintext for the same recipient. This structural malleability implies that the scheme's security cannot be restored through minor modifications, such as altering the hash functions or adding additional verification steps. In essence, the vulnerability is inherent to the scheme's fundamental design.

The results presented in this paper indicate that repairing the GLQ scheme would require a complete redesign rather than a patch. A new construction must eliminate the direct multiplicative relationship between ciphertext components and the recipient's

public key, and should instead incorporate non-malleable transformations or tightly bound signatures that prevent algebraic manipulation.

Given these considerations, we argue that further exploration of the GLQ scheme in its current form has limited merit. Any future design should start from first principles, taking into account both the theoretical and implementation-level insights revealed by our analysis. Therefore, we do not pursue this direction further in this paper.

6 Conclusion

In this paper, we revisited the security of the GLQ certificateless encryption scheme, which was originally proposed as a lightweight and secure solution for IoT environments. Contrary to the authors' claim of achieving CCA security, we demonstrated through a concrete re-randomization attack that the scheme is fundamentally insecure against adaptive chosen-ciphertext adversaries. Our analysis revealed that the ciphertext of the GLQ scheme preserves a multiplicative ElGamal-like structure, allowing any valid ciphertext to be transformed into another valid ciphertext of the same plaintext for the same recipient. This algebraic property enables an adversary to re-randomize the challenge ciphertext and submit the transformed one to the decryption oracle, thereby trivially recovering the original plaintext with probability one.

Theoretical analysis and implementation results consistently confirmed that the attack is both efficient and practical. In particular, the adversary-side computation requires only a few exponentiations and multiplications in the underlying groups, without invoking any bilinear pairing evaluations. Experimental evaluation using the PBC library under the 128-bit security parameter set further demonstrated that the runtime of the re-randomization operation is nearly identical to that of the original encryption process, making the attack feasible even in constrained IoT environments.

Acknowledgements. This work was supported by JSPS KAKENHI Grant Number JP25K15119.

Declarations

- Funding:
  This work was supported by JSPS KAKENHI Grant Number JP25K15119.
- Conflict of interest:
  Not applicable.
- Ethics approval and consent to participate:
  Not applicable.
- Consent for publication:
  We will appropriately provide a consent for publication.
- Data availability:
  We will provide our source code and data on reasonable request.
- Materials availability:
  We wrote all of the materials in this paper.
- Code availability:
  We will provide our source code and data on reasonable request.

- Clinical trial number:
  Not applicable.
- Author contribution:
  Nobuyuki Sugio wrote the whole part of this paper and conducted an implementation for evaluations. Keita Emura verified the correctness of our re-randomization attack and reviewed the whole part of this paper. Toshihiro Ohigashi helped to develop our implementation environment and provided a 128-bit security PBC parameters (a128.param [4]).

References

[1] Sattam S. Al-Riyami and Kenneth G. Paterson: Certificateless Public Key Cryptography, Proc. of Advances in Cryptology, ASIACRYPT 2003, LNCS, Vol. 2894, pp. 452–473, (2003).

[2] Hongzhi Guo, Xiangrong Li, and Haowen Qin: A lightweight and secure certificateless encryption scheme for smart city IoT security, *Discover Computing*, Vol. 28, Article number 104, (2025).

[3] PBC: Pairing-Based Cryptography (online), https://crypto.stanford.edu/pbc/

[4] Keita Emura, Katsuhiko Ito, and Toshihiro Ohigashi: Secure-channel free searchable encryption with multiple keywords: A generic construction, an instantiation, and its implementation, *Journal of Computer and System Sciences*, Vol. 114, pp. 107–125, (2020).