Toward Practical Anonymous Rerandomizable RCCA Secure Encryptions*

Rui Xue and Dengguo Feng

State Key Laboratory of Information Security,
Institute of Software, Chinese Academy of Sciences
{rxue,feng}@is.iscas.ac.cn

Abstract. Replayable adaptively chosen ciphertext attack (RCCA) security is a relaxation of popular adaptively chosen ciphertext attack (CCA) security for public key encryption system. Unlike CCA security, RCCA security allows modifying a ciphertext into a new ciphertext of the same message. One of the open questions is that if there exists a perfectly rerandomizable RCCA secure encryption [4]. Prabhakaran and Rosulek recently answered this question affirmatively [14]. The scheme they proposed (PR scheme for short) is composed of a double-strands Cramer-Shoup schemes that involves as many as 56 exponents in encryption and 65 exponents in decryption, and 55 exponents operations during rerandomization.

We present a practical perfectly rerandomizable RCCA secure encryption system in this paper. The system constitutes of two layers of encryptions. One layer carries message, the other layer carries a random quantity used to hiding the message in previous layer. This random quantity in the encryption also works as correlation between the two parts of encryption such that they are formed in a prescribed way. The proposed construction dramatically reduces the complexities, compared with PR scheme, to 15 exponents in encryption, 6 exponents decryption as well as 16 exponents operations in rerandomization.

Besides the practical feature, our scheme is also the first *receiver anonymous*, perfectly rerandomizable RCCA secure encryption, which settles an open question in [14]. The scheme is secure under DDH assumption.

1 Introduction

The popular standard for public key encryption security is the security against adaptive chosen ciphertext attack (CCA). Doleve, Dwork and Naor constructed the first scheme secure against CCA based on standard primitive [9]. Cramer and

* Supported by National Basic Research Program of China 973 Program Grant 2007CB311202, and supported by the National High Technology Research and Development Program of China (863 program) Grant 2006AA01Z427. The first author is also supported by the China Scholarship Council, and by the Natural Science Foundation of China Grant 60773029.

S. Qing, H. Imai, and G. Wang (Eds.): ICICS 2007, LNCS 4861, pp. 239–253, 2007.

Shoup gave the first practical CCA secure system based on ElGamal cryptosystem [5]. It is then intensively investigated afterwards [15,6,13]. Although CCA security is preferred in most of cryptographic applications, it is, however, also viewed too strong in some scenarios. For example, rerandomizable encryption is used in mixnets [10] with applications to voting anonymization. The nonmalleability of CCA secure system prevents such kind of system being adopted in these applications. A relaxed but strong enough security in this case is desired.

The relaxed security definitions for encryption system appeared in Krawczyk [12], Shoup [16] and An et al. [1]. The notions are called, loose cipher-unforgeability, benign malleability, and generalized CCA security respectively. Later, Canetti et al. [4] systematically developed the relaxed notion about CCA security. All notions mentioned above are proved being equivalent to the notion of publicly detectable RCCA in [4]. Many properties and schemes are setting up or constructed in that paper. One of the open questions remaining solved is the existence of so called "(perfectly) rerandomizable replayable CCA schemes".

The replayable CCA (RCCA in short) security is the same as CCA security, except no guarantees are given against adversaries that just try to modify a ciphertext into a new one with the *same* plaintext. This relaxation allows modifying a ciphertext into a new ciphertext provided that the later is decrypted into the same plaintext as the former. The *(perfectly) rerandomizable* RCCA security augments an encryption system with an algorithm to alter a ciphertext c of a message m into a new ciphertext c' that is computationally indistinguishable from the fresh ciphertext of m.

Groth [11] investigated the rerandomization of RCCA secure cryptosystem. Groth's scheme is proved only to be generic RCCA secure. Prabhakaran and Rosulek [14] constructed the first perfectly rerandomizable replayable CCA secure encryption system (briefly, PR scheme). The PR scheme is an extension of Cramer-Shoup encryption called Double-Strands Cramer-Shoup scheme. One strand of a variant of Cramer-Shoup encryption is used for carrying the message, the other one for helping rerandomization. The two strands of ciphertexts are correlated with shared random masks. In order to prevent the two strands being combined in abnormal ways, some perturbation of the exponents of the message-carrying strand is performed. As a result, the encryption system needs as many as 28 elements for the public key and 30 elements for the private key. The encryption alone involves more than 50 exponentiation operations (see Table1 in Section 3.3 for detailed complexity). The system is secure under the DDH assumption. The notion of key-privacy [3] is modified to denote receiver anonymous property. The PR scheme is not an anonymous rerandomizable RCCA secure scheme. The existence of perfectly rerandomizable RCCA secure encryption is posed as an open question.

Since rerandomizable RCCA security is weaker than CCA security in requirement, it is intuitively plausible and desirable to construct a rerandomizable RCCA scheme that is at least as efficient as CCA secure one (if not more efficient). The main contribution in this paper is a practical receiver anonymous rerandomizable RCCA secure encryption.

The scheme in this paper bears some similarity with PR scheme in that it constitutes of two layers of encryptions. However, ours is not double-strands of Cramer-Shoup like encryptions. One layer is a variant of Cramer-Shoup encryption. But the other is just an ElGamal encryption. The first layer has the functionality of carrying the message like in PR scheme but in a completely different way. The second layer instead is used only to carry a random quantity that is used to hide the message in the other layer. The quantity is also the correlation string between the two layers. The rerandomization of the ciphertext makes use of the rerandomizability of ElGamal encryption. Since two layers in our scheme are both ElGamal type of encryptions, the scheme is also receiver anonymous according to the result by Bellare et al. in [3]. Thus, our construction is not only a practically, perfectly rerandomizable but also receiver anonymous RCCA secure encryption scheme, which settle an open question in [14].

The organization of the paper : After giving out notations and definitions in Section 2, we present a publicly rerandomizable scheme and its secretly rerandomizable version in Section 3. The receiver anonymity and perfectly rerandomization of them are also indicated in Section 3. Section 4 is devoted to the proof of RCCA security of the basic scheme. The last section is our conclusion and future works.

2 Preliminaries

2.1 Some Notations

A function $f(k)$ is a *negligible function* in k if it always holds that $f(k) < 1/k^c$ for any $0 < c \in \mathbb{Z}$ for sufficiently large k. We use $\mathtt{negl}(k)$ to denote a negligible function in k, or just $\mathtt{negl}$ if k is obvious from contexts. If F is finite set, then the notation "$x \stackrel{R}{\leftarrow} F$" denotes the act of choosing x uniformly from F. Notation "$u \leftarrow f(x)$" denotes the act of assigning the value $f(x)$ to u. The notation $\Pr[x_1 \stackrel{R}{\leftarrow} S_1; x_2 \stackrel{R}{\leftarrow} S_2; \ldots; x_m \stackrel{R}{\leftarrow} S_m : p(x_1, \ldots, x_m)]$ denotes the probability that $p(x_1, \ldots, x_m)$ will be true after the ordered execution of the probabilistic assignments $x_1 \stackrel{R}{\leftarrow} S_1; x_2 \stackrel{R}{\leftarrow} S_2; \ldots; x_m \stackrel{R}{\leftarrow} S_m$.

2.2 Security of Public Key Encryption

We model all algorithms including adversaries as probabilistic polynomial time (PPT) Turing machines. The public-key cryptosystem is defined as usual with three algorithms (K, E, D). Where K is key generation algorithm, E the encryption algorithm, and D the decryption algorithm. The rerandomizable encryption system is a public-key encryption system augmented with an additional rerandomization algorithm *Rand*.

The security of a public key encryption system against CCA and its variants is defined as follows.

Definition 1. *Let (K, E, D) be an encryption system. For any PPT algorithm A if the following probability is negligibly close to $1/2$:*

$$\Pr\left[(pk, sk) \leftarrow K(1^k); (m_0, m_1) \leftarrow A^{\mathcal{O}_1}; b \overset{R}{\leftarrow} \{0,1\}; c^* \leftarrow E(m_b) : A^{\mathcal{O}_2}(pk, c^*) = b\right].$$

where $\mathcal{O}_1$ runs exactly as decryption algorithm D, and $\mathcal{O}_2$ run exactly as decryption algorithm D except in the following cases:

1. *When a query to $\mathcal{O}_2$ is c^*, $\mathcal{O}_2$ will return $\perp$. Then, the cryptosystem is said secure against chosen ciphertext attack (CCA).*
2. *When a query to $\mathcal{O}_2$ has plaintext in $\{m_0, m_1\}$, $\mathcal{O}_2$ will return* `test`*. Then, the cryptosystem is said secure against replayable chosen ciphertext attack (RCCA).*
3. *When a query to $\mathcal{O}_2$ has plaintext in $\{m_0, m_1\}$, $\mathcal{O}_2$ will return $\perp$. Then, the cryptosystem is said secure against weak replayable chosen ciphertext attack (WRCCA).*

2.3 Rerandomizable Encryption

There are two kinds of randomization algorithms defined in [4]: a *public rerandomization algorithm PRand* takes a ciphertext c and public key in system as input and turns out a well-formed new ciphertext c' with identical distribution to that of c such that both are decrypted as the same plaintext. *A secret randomization algorithm Srand* is the same as public randomization algorithm except taking only a ciphertext as input. A public-key encryption system augmented with a public (secret) randomization algorithm is called a *publicly (secretly) rerandomizable encryption system*. We some times call secretly rerandomizable encryption just as *rerandomizable encryption* as in [14].

For a rerandomizable encryption scheme $(K, E, D, Rand)$, any PPT adversary $\mathcal{A}$, following experiment is called *perfect rerandomization attack* experiment (PRA) in [14].

Stage 1. Pick $(pk, sk) \leftarrow K(1^k)$. The public keys pk is given to $\mathcal{A}$.
Stage 2. Adversary $\mathcal{A}$ gets access to decryption oracle and rerandomization oracle $D_{sk}(\cdot), Rand(\cdot)$.
Stage 3. Adversary submits a message m to encryption oracles. Challenger picks $b \overset{R}{\leftarrow} \{0,1\}$ at random and
If $b = 0$ **then** pick r_1, r_2 at random, $c_1 \leftarrow E(pk, r_1)$ and $c_2 \leftarrow Rand(c_1, r_2)$
else $c_1 \leftarrow E(pk, r_1)$ and $c_2 \leftarrow E(m, r_2)$
return (c_1, c_2)
Where random strings r_1, r_2 are chosen uniformly from defined ranges in encryption scheme, and $E(m, r_1)$ indicates the ciphertext of m encrypted using random string r_1.
Stage 4. Adversary $\mathcal{A}$ continues to get access to decryption and randomization oracles $D_{sk}(\cdot), Rand(\cdot)$.
Stage 5. Adversary $\mathcal{A}$ outputs a bit b'.

The advantage of adversary $\mathcal{A}$ in PRA experiment is $\Pr[b' = b] - \frac{1}{2}$.

Definition 2 (Perfectly Rerandomizable Encryption). *A rerandomizable encryption system $(K, E, D, Rand)$ is said perfectly rerandomizable if for any PPT algorithm $\mathcal{A}$, the advantage of any PPT algorithm $\mathcal{A}$ in PRA experiment is negligible in security parameter k.*

Loosely speaking, a (perfectly) rerandomizable encryption guarantees the rerandomization to a ciphertext c will return a new ciphertext c' of the same message that is indistinguishable from any fresh encryption of original message.

Let $\mathcal{E} = (K, E, D, Rand)$ be a rerandomizable encryption system. Let R be the range of random strings used in encryption. The following is a direct conclusion from the definition above.

Lemma 1. *If for any $r_1, r_2 \in R$, we have $Rand(E(m, r_1), r_2) = E(m, f(r_1, r_2))$, where $f(r_1, r_2)$ is a random variable in R. The cryptosystem $\mathcal{E}$ is a perfectly rerandomizable encryption system, if for any two independent strings r_1, r_2 chosen uniformly from R, $f(r_1, r_2)$ is uniformly distributed on R.*

2.4 Decisional Diffie-Hellman Assumption

The security of scheme in this paper depends on the hardness of Decisional Diffie-Hellman (DDH) problem. For any group G of prime order q, a tuple (g_1, g_2, g_3, g_4) in G^4 is a Diffie-Hellman (DH) tuple if there is a $r \in \mathbb{Z}_q$ such that $(g_1, g_2, g_3, g_4) = (g_1, g_2, g_1^r, g_2^r)$. The *decisional Diffie-Hellman assumption* holds in G if any PPT algorithm cannot tell DH tuples from uniformly chosen tuples from G^4 except with negligible probability. Let $\texttt{Rand}$ is the event that the tuple (g_1, g_2, g_3, g_4) is chosen from distribution of random tuples, and $\texttt{DH}$ the event that the tuple is chosen from distribution of Diffie-Hellman tuples. The advantage $\texttt{Adv}_{\mathcal{A}}^{DDH}$ of PPT algorithm $\mathcal{A}$ telling the DH tuples from random tuples is

$$\texttt{Adv}_{\mathcal{A}}^{DDH} = |\Pr[\mathcal{A}(g_1, g_2, g_3, g_4) = 1 \mid \texttt{DH}] - \Pr[\mathcal{A}(g_1, g_2, g_3, g_4) = 1 \mid \texttt{Rand}]|$$

The probability is over the coin toss of $\mathcal{A}$ and the randomness of the chosen of (g_1, g_2, g_3, g_4).

As in [14], the scheme in this paper employs two groups $G, \hat{G}$ with special relationship: G is with order q, and $\hat{G}$ order p. Where group $\hat{G}$ is a subgroup in $\mathbb{Z}_q^*$. This is exemplified by the following groups: Let $p, 2p+1, 4p+3$ (first kind *Cunningham chain* of length 3 [2]) be a primes chain. Group $\mathcal{QR}_p$ denotes the quadratic residue modulo p. Then $\hat{G} = \mathcal{QR}_{2p+1}$ and $G = \mathcal{QR}_{4p+3}$ are desired groups. It is widely believed that DDH assumption holds in quadratic residue modulo a safe prime. It is conjectured there are infinite many Cunningham chain (see Prabhakaran and Mike Rosulek [14] for details).

3 Rerandomizable Encryption Schemes

We present three randomizable encryption schemes in this section. The first two are publicly randomizable encryption systems, and the last one is a secretly

randomizable. The first publicly rerandomizable system is our basic scheme. We will only give out the security proof for the first scheme in next section. The proof for the variants are essentially the same, which are omitted.

3.1 Publicly Rerandomizable Encryption System (PRE)

The publicly rerandomizable encryption system consists of four algorithms (`KeyGen`, `Enc`, `Dec`, `PRand`):

Key Generation. Taking security parameter k as input, algorithm `KeyGen` chooses a cyclic group G of prime order q and a subgroup $\hat{G} \subseteq \mathbb{Z}_q^*$ of order p (see the last paragraph of Section 2.4) such that DDH problem is hard in G and $\hat{G}$. Where the length of p is k. It then chooses $g_1, g_2 \leftarrow G$, $\hat{g} \leftarrow \hat{G}$ and $x, y, a, b, a', b' \leftarrow \mathbb{Z}_q, \lambda \leftarrow \mathbb{Z}_p$ at random. To choose a collision resistant Hash function $H : \hat{G} \times G \to \mathbb{Z}_q$. The public key pk and secret key sk are defined as

$$pk := (\hat{g}, \hat{e} = \hat{g}^{\lambda};\ g_1, g_2, h = g_1^x g_2^y, c = g_1^a g_2^b, d = g_1^{a'} g_2^{b'}, H),$$
$$sk := (\lambda;\ x, y, a, b, a', b')$$

Encryption. Given a message $m \in G \backslash \{1_G\}$, the encryption algorithm `Enc` runs as follows. First it chooses $r, s \leftarrow \mathbb{Z}_q, t \leftarrow \hat{G}, \gamma \leftarrow \mathbb{Z}_p$ at random. It then computes

$$u_0 \leftarrow \hat{g}^{\gamma}, w_0 \leftarrow \hat{e}^{\gamma} t, \alpha \leftarrow H(m),$$
$$u_1 \leftarrow g_1^r, v_1 \leftarrow g_2^r, w_1 \leftarrow h^r m^t,$$
$$u_2 \leftarrow g_1^s, v_2 \leftarrow g_2^s, w_2 \leftarrow (c^{\alpha t} d^t)^s.$$

The ciphertext is $C = (u_0, w_0; u_1, v_1, w_1, u_2, v_2, w_2)$.

Decryption. Given a ciphertext $C = (u_0, w_0; u_1, v_1, w_1, u_2, v_2, w_2)$ and sk , the decryption algorithm `Dec` first computes

$$t \leftarrow w_0 / u_0^{\lambda}, \qquad \overline{m} \leftarrow w_1 / (u_1^x v_1^y), \qquad m \leftarrow \overline{m}^{1/t}, \qquad \alpha \leftarrow H(m)$$

To check if $\overline{m} \neq 1_G \wedge w_2 \stackrel{?}{=} u_2^{(a\alpha + a')t} v_2^{(b\alpha + b')t}$. If it holds, then the algorithm outputs m; otherwise it outputs $\perp$.

Rerandomization. Given ciphertext $C = (u_0, w_0; u_1, v_1, w_1, u_2, v_2, w_2)$ and public key pk. The public rerandomization algorithm `PRand` selects $r', s', t' \stackrel{R}{\leftarrow} \mathbb{Z}_q, \gamma' \stackrel{R}{\leftarrow} \mathbb{Z}_p$ at random and then computes

$$C' = (u_0 \hat{g}^{\gamma'}, w_0 \hat{e}^{\gamma'} t';\ u_1^{t'} g_1^{r'}, v_1^{t'} g_2^{r'}, w_1^{t'} h^{r'}, u_2^{s'}, v_2^{s'}, w_2^{s' t'})$$

It outputs C' as replayed ciphertext.

This completes the description of the cryptosystem. We first verify that decryption of honestly constructed ciphertext C will yield correct value. Since $u_1 = g_1^r, v_1 = g_2^r$, we have

$$v_1 = u_1^w,\ u_1^x v_1^y = g_1^{rx} g_2^{ry} = (g_1^x g_2^y)^r = h^r \text{ and } \overline{m} = w_1 / h^r = w_1 / (u_1^x v_1^y).$$

Similarly, let $t = w_3/u_3^{\lambda}$, $m = \overline{m}^{1/t}$ and $\alpha = H(m)$,

$$\begin{aligned} u_2^{(a\alpha+a')t} v_2^{(b\alpha+b')t} &= g_1^{(a\alpha+a')ts} g_2^{(b\alpha+b')ts} = (g_1^{a\alpha ts} g_2^{b\alpha ts})(g_1^{a'ts} g_2^{b'ts}) \\ &= c^{ts\alpha} d^{ts} = (c^{\alpha t} d^t)^s = w_2. \end{aligned}$$

If $m \neq 1_G$, the decryption test will pass, and the output will be m.

Next, the rerandomized ciphertext is

$$\begin{aligned} C' &= (u_0 \hat{g}^{\gamma'}, w_0 \hat{e}^{\gamma'} t'; u_1^{t'} g_1^{r'}, v_1^{t'} g_2^{r'}, w_1^{t'} h^{r'}, u_2^{s'}, v_2^{s'}, w_2^{s't'}) \\ &= (\hat{g}^{\gamma+\gamma'}, \hat{e}^{\gamma+\gamma'} t\, t'; g_1^{rt'+r'}, g_2^{rt'+r'}, h^{rt'+r'} m^{t\,t'}, g_1^{ss'}, g_2^{ss'}, (c^{\alpha t} d^t)^{ss't'}) \\ &= (\hat{g}^{\overline{\gamma}}, \hat{e}^{\overline{\gamma}} \overline{t}; g_1^{\overline{r}}, g_2^{\overline{r}}, h^{\overline{r}} m^{\overline{t}}, g_1^{\overline{s}}, g_2^{\overline{s}}, (c^{\alpha \overline{t}} d^{\overline{t}})^{\overline{s}}) \end{aligned}$$

Where

$$\begin{aligned} \overline{\gamma} &\equiv \gamma + \gamma' \mod p & \qquad \overline{r} &\equiv rt' + r' \mod q \\ \overline{s} &\equiv ss' \mod q & \qquad \overline{t} &\equiv t\,t' \mod q \end{aligned} \tag{1}$$

It is again a well formed ciphertext for message m that can be decrypted successfully. Here is the right place to indicate the perfect rerandomizability and anonymity of our scheme:

Proposition 1. *The rerandomizable encryption* (`KeyGen`, `Enc`, `Dec`, `PRand`) *is a perfectly reranzomizable encryption.*

Proof. It is easy to see that for any given γ, random variable $\overline{\gamma} \equiv \gamma + \gamma' \mod p$ is uniformly distributed on $\mathbb{Z}_p$ if γ' is uniformly distributed. Similarly to see from (1) that $\overline{r}, \overline{s}$ and $\overline{t}$ are all uniformly distributed in $\mathbb{Z}_q$, whenever r', t', s' are independently and uniformly chosen from $\mathbb{Z}_q$. This shows the perfect rerandomness according to Lemma 1. ✦

Since the encryption constructed above is two layers of ElGamal type of encryption, it is a conclusion in [3] that ElGamal encryption is key-privacy and hence receiver anonymity in our sense.

Proposition 2. *The encryption* (`KeyGen`, `Enc`, `Dec`, `PRand`) *is receiver anonymous.*

3.2 Secretly Rerandomizable Encryption System

Publicly rerandomizable encryption system in last subsection needs public key for randomization algorithm to make a randomization to the purported ciphertext. This is not desired in some applications like universal mixnets [10]. Where each mixing server will permute many ciphertexts from different sources to different receivers. The public keys are either unavailable to mix servers or unintended to mix servers.

We expand the publicly rerandomizable encryption systems in last subsection to obtain a secretly rerandomizable encryption system (briefly SRE) (or rerandomizable RCCA encryption system [14]). SRE system consists of four

algorithms (SKeyGen, SEnc, SDec, SRand). The algorithms SKeyGen and SDec behave in the same way respectively as KeyGen and Dec in last subsection. The other two are as follows.

Encryption. Given a message $m \in G$, the encryption algorithm SEnc runs as follows. First it chooses $r, s \leftarrow \mathbb{Z}_q, t \leftarrow \hat{G}, \gamma \leftarrow \mathbb{Z}_p$ at random. It then computes

$$\overline{u}_0 \leftarrow \hat{g}^{\gamma'}, \overline{w}_0 \leftarrow \hat{e}^{\gamma'}, u_0 \leftarrow \hat{g}^{\gamma}, w_0 \leftarrow \hat{e}^{\gamma} t,$$
$$\overline{u}_1 \leftarrow g_1^{r'}, \overline{v}_1 \leftarrow g_2^{r'}, \overline{w}_1 \leftarrow h^{r'}, u_1 \leftarrow g_1^{r}, v_1 \leftarrow g_2^{r}, w_1 \leftarrow h^{r} m^{t},$$
$$\alpha \leftarrow H(m), u_2 \leftarrow g_1^{s}, v_2 \leftarrow g_2^{s}, w_2 \leftarrow (c^{\alpha} d)^{ts}.$$

The ciphertext is $C = (\overline{u}_0, \overline{w}_0, u_0, w_0; \overline{u}_1, \overline{v}_1, \overline{w}_1, u_1, v_1, w_1, u_2, v_2, w_2)$.

Rerandomization. Given ciphertext C denoted as above. The secretly rerandomization algorithm SRand selects $r_1, r_1', s_2, t' \leftarrow \mathbb{Z}_q, \gamma_0, \gamma_0' \leftarrow \mathbb{Z}_p$ at random and then computes C' as

$$(\overline{u}_0^{\gamma_0'}, \overline{w}_0^{\gamma_0'}, u_0 \overline{u}_0^{\gamma_0}, w_0 \overline{w}_0^{\gamma_0} t'; \overline{u}_1^{r_1'}, \overline{v}_1^{r_1'}, \overline{w}_1^{r_1'}, u_1^{t'} \overline{u}_1^{r_1}, v_1^{t'} \overline{v}_1^{r_1}, w_1^{t'} \overline{w}_1^{r_1}, u_2^{s_2}, v_2^{s_2}, w_2^{t' s_2})$$

It outputs C' as the replayed ciphertext.

The rerandomization of C does not take public key pk as input, rather, only the information from C is used. This shows that it is indeed a secretly randomizable encryption.

3.3 Complexity

There are two secretly rerandomizable schemes in the literature and they appeared in [11] and [14] respectively. We compare the efficiency of our scheme in Section 3.2 with them in the following:

Groth [11] proposed the first secretly rerandomizable encryption system with *weak* RCCA security. The scheme there used $O(k)$ group elements to encode a k-bit message. The public key and secret key is of length $O(k)$.

The scheme proposed by Prabhakaran and Rosulek in [14] as a first secretly rerandomizable system with RCCA security used 40 elements from $\hat{G}$ and 14 elements from G in their ciphertext. Public key there uses 13 group elements, and secret key uses 20 group elements.

The secretly rerandomizable encryption scheme in this paper uses 4 elements from $\hat{G}$ and 9 elements from G in a ciphertext. The public key here uses 7 group elements, and secret key uses 5 group elements.

The number of operations in the group performed by our scheme is summarized and compared to Groth's scheme and PR's scheme in Table 1. It is easy to see that ours is the most practical scheme and much more efficient in the length of ciphertext and operations needed during encryption and decryption as well as rerandomization.

Table 1. The comparison of the number of group operations performed by our scheme with those by existing RCCA schemes. Where k is the number of bits in plaintext.

Schemes	Functions	Exponentiations	Multiplications
Groth's Scheme	`Enc`	$O(k)$	$O(k)$
	`Dec`	$O(k)$	$O(k)$
	`Rand`	$O(k)$	0
PR's Scheme	`Enc`	56	18
	`Dec`	65	62
	`Rand`	55	47
Our Scheme	`Enc`	15	4
	`Dec`	6	4
	`Rand`	16	6

4 Proof of RCCA Security

In this section we will prove the publicly rerandomizable RCCA security for scheme in Section 3.1. The system is a combination of two cryptosystems: one is an expansion of Cramer-Shoup encryption system, another is ElGamal encryption system with public key $pk_1 = (\hat{g}, \hat{e})$ and secret key $sk_1 = \lambda$. To denote it as Γ. The RCCA attack to the system forms an attack to the ElGamal encryption Γ. This attack is called indirect RCCA attack to Γ. We show that under indirect RCCA attack, ElGamal system Γ remains secure. To be more faithful to the proof of our main theorem, we formalize the following indirect attack game:

Stage 1. Adversary queries a key generation oracle. The key generation oracle computes $pk = (\hat{g}, \hat{e} = \hat{g}^{\lambda}), sk = \lambda$ and responds with pk.

Stage 2. The adversary makes a sequence of indirect decryption oracle. For each indirect decryption query, the adversary will submits a pair of messages (m, c). Where $m \in G \backslash \{1_G\}$ and c a cipher encrypted with pk. Here c could be any ill formed ciphertext without using pk. The decryption oracle decrypt c to get $t \in \hat{G}$ and responds with $m^{1/t}$.

Stage 3. The adversary submits a message $m_0 \in G \backslash \{1_G\}$, and the encryption oracle chooses $t_0 \in \hat{G}$ and $\gamma \leftarrow \mathbb{Z}_p$ at random, and responds with $\psi^* = (m_0^{t_0}, (\hat{g}^{\gamma}, \hat{e}^{\gamma} t_0))$

Stage 4. The adversary continues to make calls to indirect decryption oracle, subject only to the restriction that the submitted should not the same as the target message ψ^*.

Stage 5. The adversary outputs a value $\hat{t} \in \hat{G}$.

We say the adversary succeeded in indirect attack game if $\hat{t} = t_0$.

Lemma 2. *Under the assumption of DDH problem is hard in G and $\hat{G}$, the encryption system Γ is secure with respect to the indirect attack. Specifically, for any PPT algorithm A, adversary A's has negligible success probability in indirect attack game.*

The proof is followed from definition and omitted due to lack of spaces.

We now show the RCCA security of the scheme.

Theorem 1 (RCCA security). *Under the assumption of DDH problem is hard in the group G and that the hash H is a collision resistant hash function, the system is a rerandomized RCCA secure encryption system.*

Proof. The rerandomization of the system is straightforward to see. The proof for RCCA security share the similarity to that for original Cramer-Shoup cryptosystem [5,7] but with different arguments in details. We reduce the security of system to the security of DDH problem assuming the collision resistance of hash function H.

For any PPT algorithm $\mathcal{A}$ with success probability $\Pr_{\mathcal{A}}[\mathtt{Succ}]$ during RCCA attack to the real cryptosystem, we construct an algorithm $\hat{\mathcal{A}}$ to distinguish DDH tuples from random tuples as follow.

The input for $\hat{\mathcal{A}}$ is a tuple (g_1, g_2, g_3, g_4) that is either a random tuple meaning that each entry g_i is independently chosen from G uniformly, or a DH tuple meaning that there is a random $r \in \mathbb{Z}_q$ such that $g_3 = g_1^r, g_4 = g_2^r$. The algorithm $\hat{A}$ runs the following experiment

$\underline{\hat{\mathcal{A}}(g_1, g_2, g_3, g_4)}$

$x, y, a, b, a', b' \leftarrow \mathbb{Z}_q, \hat{g} \leftarrow \hat{G}, \lambda \leftarrow \mathbb{Z}_p$

$\hat{e} = \hat{g}^{\lambda}; h = g_1^x g_2^y, c = g_1^a g_2^b, d = g_1^{a'} g_2^{b'}$

$pk = (\hat{g}, \hat{e}; g_1, g_2, h, c, d, H),\ \ sk = (\lambda; x, y, a, b, a', b')$

$(m_0, m_1, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}_1}(pk),$ where $1_G \neq m_0 \neq m_1 \neq 1_G$

$\mathbf{b} \leftarrow \{0,1\},\ s \leftarrow \mathbb{Z}_q,\ t^* \leftarrow \hat{G}, \gamma \leftarrow \mathbb{Z}_p,\ \ \alpha^* := H(m_{\mathbf{b}})$

$C^* = (\hat{g}^{\gamma}, \hat{e}^{\gamma} t^*;\ g_3, g_4, g_3^x g_4^y m_{\mathbf{b}}^{t^*}, g_3^s, g_4^s, (g_3^{a\alpha^* + a'} g_4^{b\alpha^* + b'})^{t^* s})$

$\mathbf{b}' \leftarrow \mathcal{A}^{\mathcal{O}_2}(pk, C, \sigma)$

If $\mathbf{b} = \mathbf{b}'$ then output 1

otherwise output 0

Where H is the collision resistant hash function used in the cryptosystem and the string σ is state information of $\mathcal{A}$ that will be transformed to the second decryption enquiry phase.

The two oracles are decryption oracles. The oracle $\mathcal{O}_1$, whenever recieve a query, behaves exactly the same as in the cryptosystem using knowledge of sk to decrypt the query. The oracle $\mathcal{O}_2$ runs just same as the decryption algorithm except when the query has plaintext in $\{m_0, m_1\}$. When the plain text for the query to oracle $\mathcal{O}_2$ is in $\{m_0, m_1\}$ then oracle $\mathcal{O}_2$ will outputs `test`.

Firstly, we have the following result:

Lemma 3. *If the adversary $\hat{\mathcal{A}}$ gets DH tuple, $\mathcal{A}$'s view of the experiment is the same as that in an attack execution to the real encryption scheme.*

Proof of Lemma 3. When the input tuple (g_1, g_2, g_3, g_4) to $\hat{\mathcal{A}}$ is random DH tuple, there exists a $r \in \mathbb{Z}_q$ such that $g_2 = g_1^{\delta}, g_3 = g_1^{r}, g_4 = g_1^{\delta r} = g_2^{r}$. The challenging ciphertext in the experiment is

$$\begin{aligned} C^* &= (\hat{g}^{\gamma}, \hat{e}^{\gamma} t^*;\ g_3, g_4, g_3^{x} g_4^{y} m_{\mathbf{b}}^{t^*}, g_3^{s}, g_4^{s}, (g_3^{a\alpha^*+a'} g_4^{b\alpha^*+b'})^{t^* s}) \\ &= (\hat{g}^{\gamma}, \hat{e}^{\gamma} t^*;\ g_1^{r}, g_2^{r}, (g_1^{x} g_2^{y})^{r} m_{\mathbf{b}}^{t}, g_1^{rs}, g_2^{rs}, (g_1^{a\alpha^*+a'} g_2^{b\alpha^*+b'})^{t^* rs}). \end{aligned}$$

Where $\alpha^* = H(m_{\mathbf{b}})$. Which is a well formed ciphertext for message $m_{\mathbf{b}}$ encrypted with public key pk. The oracles in the experiment run exact in the same way as in the real attacking executions. This shows the view of adversary $\mathcal{A}$ in this case is the same as that in attacking in the real system. This ends the proof of Lemma. ✧

Let $\mathtt{Rand}$ be the event that the tuple (g_1, g_2, g_3, g_4) is chosen from distribution of random tuples, and $\mathtt{DH}$ the event that the tuple is chosen from distribution of Diffie-Hellman tuples.

Lemma 3 implies that conditioned the input is DH tuples, the success probability $\Pr_{\mathcal{A}}[\mathtt{Succ}]$ of adversary $\mathcal{A}$ attacking the encryption system in RCCA game is same as the probability of algorithm $\hat{\mathcal{A}}$ outputs 1. That is,

$$\Pr_{\mathcal{A}}[\mathtt{Succ}] = \Pr[\hat{\mathcal{A}} = 1 \mid \mathtt{DH}] \tag{2}$$

However, the DDH intractability assumption implies

$$|\Pr[\hat{\mathcal{A}} = 1 \mid \mathtt{DH}] - \Pr[\hat{\mathcal{A}} \mid \mathtt{Rand}]| = \mathtt{negl}(k) \tag{3}$$

where k is the security parameter. This is the case since algorithm $\hat{\mathcal{A}}$ is a PPT algorithm as $\mathcal{A}$ is.

The advantage of adversary attacking real system is, using equation (2) and (3),

$$\begin{aligned} \left|\Pr_{\mathcal{A}}[\mathtt{Succ}] - \frac{1}{2}\right| &= \left|\Pr[\hat{\mathcal{A}} = 1 \mid \mathtt{DH}] - \frac{1}{2}\right| \\ &\leq \left|\Pr[\hat{A} = 1 \mid \mathtt{DH}] - \Pr[\hat{A} = 1 \mid \mathtt{Rand}]\right| + \left|\Pr[\hat{A} = 1 \mid \mathtt{Rand}] - \frac{1}{2}\right| \\ &= \left|\Pr[\hat{A} = 1 \mid \mathtt{Rand}] - \frac{1}{2}\right| + \mathtt{negl}(k) \end{aligned} \tag{4}$$

This, together with the conclusion (5) in following Lemma 4, will show that the advantage of $\mathcal{A}$ attacking real system is negligible. Since $\mathcal{A}$ is any such adversary, that will complete the proof of the theorem. It remains to prove the following:

Lemma 4. *If algorithm $\hat{\mathcal{A}}$ gets random tuple as input, algorithm $\mathcal{A}$ has no information about the bit $\mathbf{b}$ chosen by $\hat{\mathcal{A}}$ except with negligible probability. That is,*

the probability that adversary $\mathcal{A}$ correctly guesses bit $\mathbf{b}$ *chosen by* $\hat{\mathcal{A}}$ *is negligibly close to* $1/2$*. Hence,*

$$\left|\Pr[\hat{A} = 1 \mid \mathtt{Rand}] - \frac{1}{2}\right| = \mathtt{negl}(k) \tag{5}$$

Proof of Lemma4. For a random tuple $(g_1, g_2, g_3, g_4) \in G^4$, there exists δ, r, γ uniformly in $\mathbb{Z}_q$ such that

$$(g_1, g_2, g_3, g_4) = (g_1, g_1^{\delta}, g_1^{r}, g_1^{\eta})$$

It is with overwhelming probability that $\eta \neq r\delta \pmod q$ and $\delta \neq 0 \pmod q$ except with negligible probability. We suppose $\eta \neq r\delta$ in the following proof. Let $r' \in \mathbb{Z}_q$ satisfy $g_4 = g_2^{r'}$, then $r' \neq r$.

We call a ciphertext $C = (u_0, w_0; u_1, v_1, w_1, u_2, v_2, w_2)$ *valid* if and only if $\log_{g_1} u_1 = \log_{g_2} v_1$. The following claim says, adversary $\mathcal{A}$ might be able to get useful information about $\mathbf{b}$ only if $\mathcal{A}$ queries with invalid ciphertext.

Claim 1. *If the oracles $\mathcal{O}_1, \mathcal{O}_2$ reject all the invalid queries by A, then the distribution of* $\mathbf{b}$ *independent on the view of adversary A during the attack.*

Proof of the Claim 1. We show this by showing that any valid ciphertext query will not be able to give $\mathcal{A}$ further information about (x, y).

Before attack, $\mathcal{A}$'s view about x, y is from pk. To be exactly from h. Even A might solve discrete logarithm, A can only know that

$$\log_{g_1} h = x + \delta y\,. \tag{6}$$

A linear combination about x, y.

We will see that adversary cannot get any information about x, y from the challenge ciphertext $C^* = (u_0^*, w_0^*; u_1^*, v_1^*, w_1^*, u_2^*, v_2^*, w_2^*)$. From which $\mathcal{A}$ gets

$$\log_{g_1}(w_1^*/m_{\mathbf{b}}^{t^*}) = rx + r'\delta y \tag{7}$$

Where $\delta = \log_{g_1} g_2$, $u_1^* = g_1^r$ and $v_1^* = g_2^{r'}$. By Lemma 2 and randomness of (g_1, g_2, g_3, g_4), $w_1^*/m_{\mathbf{b}}^t$ is uniformly distributed in G in the view of adversary. Equation systems (6) and (7) will give out a uniformly distributed point (x, y) on the line (6). Thus adversary $\mathcal{A}$ cannot obtain any further information about (x, y).

If, however, adversary queries only valid ciphertext $(u_0, w_0; u_1, v_1, w_1, u_2, v_2, w_2)$. Then $\log_{g_1} u_1 = \log_{g_2} v_1 = \overline{r}$, and $u_1^x v_1^y = g_1^{\overline{r}x} g_2^{\overline{r}y} = h^{\overline{r}}$. Hence, $\overline{r}\log h = \overline{r}x + \overline{r}\delta y$. This is a linear dependent relations of equation (6). No further information about x, y can be obtained. This ends the proof of Claim 1. ✧

Claim 2. *The oracles $\mathcal{O}_1$, $\mathcal{O}_2$ will output $\perp$ for all invalid queries that is not a rerandomization of challenge meassage C^*, except with negligible probability.*

Proof of the Claim 2. Consider the first invalid query ciphertext

$$C = (u_0, w_0; u_1, v_1, w_1, u_2, v_2, w_2)$$

to oracles. If it comes from the rerandomization of the challenge ciphertext C^*, adversary will get reply `test`. We will only consider invalid queries that is not a rerandomization of C^* in the following.

Let $\overline{m} = w_1/(u_1^x v_1^y)$, $t = w_0/u_0^\lambda$. As the first invalid query, by the result shown in the proof of last claim, adversary cannot get the value $\overline{m}$ except with negligible possibility. This is because (x, y) is now a random point (on line (6)). This further implies that $\mathcal{A}$ cannot generate correct value α' such that $\alpha' = H(\overline{m}^{1/t})$ except with negligible probability due to the collision resistance property of H. If $\log_{g_1} u_2 = \log_{g_2} v_2 = s'$, the ciphertext C cannot pass the decryption test $w_2^{(ts')^{-1}} = c^{H(\overline{m}^{1/t})} d$ except with negligible probability.

Now we are in the case that adversary queries C with $\hat{r} = \log_{g_1} u_2 \neq \log_{g_2} v_2 = \hat{r}'$. Let $m := \overline{m}^{1/t}$ and $\alpha = H(m)$. Lemma 2 ensures that adversary will choose t such that $m^t \neq m_{\mathbf{b}}^{t^*}$ with overwhelming probability (here comes the restriction of $m_{\mathbf{b}} \neq 1_G$). This shows $\alpha^* = H(m_{\mathbf{b}}) \neq \alpha$ except with negligible probability.

We investigate the probability of C passing the test

$$u_2^{(a\alpha+a')t} v_2^{(b\alpha+b')t} = w_2. \tag{8}$$

We consider the distribution of (a, b, a', b') conditioned the view of adversary A. Before attacking, from public key pk, adversary knows at most (even if $\mathcal{A}$ can solve discrete logarithms) that

$$\log_{g_1} c = a + \delta b \tag{9}$$

$$\log_{g_1} d = a' + \delta b' \tag{10}$$

The further information might be obtained from the challenge ciphertext C^*, where

$$\log_{g_1} w_2^* = r_1(\alpha^* a + a')t^* + r_1'\delta(\alpha^* b + b')t^*. \tag{11}$$

Where $r_1 = rs \pmod q$ and $r_1' = r's \pmod q$. We have $r_1 \neq r_1'$ since $r \neq r'$.

Adversary queries the invalid ciphertext C and passes the test (8) if and only if it satisfies

$$\log_{g_1} w_2 = \hat{r}(\alpha a + a')t + \hat{r}'\delta(\alpha b + b')t. \tag{12}$$

Recall that we have now $\delta \neq 0$, $\alpha' \neq \alpha^*$, $r_1 \neq r_1'$, and $\hat{r} \neq \hat{r}'$ except with negligible probability. Rewriting of equations (9), (10), (11), and (12) gives

$$\begin{pmatrix} \log_{g_1} c \\ \log_{g_1} d \\ \log_{g_1} w_2^* \\ \log_{g_1} w_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \delta & 0 & 0 \\ 0 & 0 & 1 & \delta \\ r_1 t^* \alpha^* & r_1' t^* \alpha^* \delta & r_1 t^* & r_1' t^* \delta \\ \hat{r} t \alpha & \hat{r}' t \alpha \delta & \hat{r} t & \hat{r}' t \delta \end{pmatrix}}_{:=M} \begin{pmatrix} a \\ b \\ a' \\ b' \end{pmatrix} \tag{13}$$

The matrix M is with determinant $\det M = t\, t^* \delta^2 (\alpha - \alpha^*)(r_1 - r_1')(\hat{r} - \hat{r}') \neq 0$ with all but negligible probability. For each guess value of w_2, there will be a unique tuple (a, b, a', b') satisfying the equations. Since w_2 could be any element

in group G, the first invalid query that is able to pass decryption test with negligible probability.

The reject of an invalid decryption query will rule out one possibility of tuple (a, b, a', b'). The n'th invalid decryption query will pass the test with probability only at most $1/(q-n-1)$. Suppose there are totally n many invalid decryption query, one of them will pass the test with probability at most $n/(q-n-1)$. Since algorithm $\mathcal{A}$ is a PPT algorithm, the number of invalid queries n will be bounded up by a polynomial, while $|q| \geq k$ and q is exponential in k. Therefore, $n/(q-n-1)$ is negligible in k.

In other words, the decryption oracles will reject all invalid queries except with negligible probability. This accomplishes the proof of Claim 2 and Lemma 4. ✧

This also finishes the proof of Theorem 1. □

To combine the conclusions in Proposition 1, Proposition 2, and Theorem 1, we have

Theorem 2. *The encryption system in Section 3.1 is an anonymous, perfectly rerandomizable RCCA secure system. And the encryption system in Section 3.2 is a secretly anonymous, perfectly rerandomizable RCCA secure system.*

5 Conclusion

While the PR scheme is the first perfectly rerandomizable RCCA secure scheme, their construction is not receiver anonymous as stated by its authors in [14]. We present the first receiver anonymous, perfectly rerandomizable RCCA secure encryptions. The constructions inherit the double-strands feature from PR scheme in Prabhakaran and Rosulek [14]. It constitutes of two layers of encryptions: one layer to carry message, the other layer to carry a random quantity to hiding the message in previous layer. New constructions dramatically reduce the complexities compared with PR scheme. They are plausible to be applied in the scenarios like mix-nets [10,8], where the rerandomization of encryption are required and the encryption with weaker re-encryption are not enough.

Our final aim is to construct an efficient perfectly rerandomizable RCCA secure encryption. Though the constructions in this paper are much efficient than PR scheme, they are far from efficient than desired. Some other properties like universal composability of proposed constructions need further investigation and we plan to do it as a part of our future work.

Acknowledgement

Most of this work was done while Rui Xue was at UIUC. He thanks Manoj Prabhakaran for his warm hospitality. We thank Manoj Prabhakaran and Mike Rosulek for discussions about RCCA secure encryption from which this work originated.

References

1. An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer, Heidelberg (2002)
2. Andersen, J.K., Weisstein, E.W.: Cunningham chain. From MathWorld CA Wolfram Web (2005), http://www.cs.umd.edu/jkatz/gradcrypto2/scribes.html
3. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 566–582. Springer, Heidelberg (2001)
4. Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing chosen-ciphertext security. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 565–582. Springer, Heidelberg (2003)
5. Cramer, Shoup.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, Springer, Heidelberg (1998)
6. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
7. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal of Computing 33, 167–226 (2003)
8. Danezis, G.: Breaking four mix-related schemes based on universal re-encryption. In: Katsikas, S.K., Lopez, J., Backes, M., Gritzalis, S., Preneel, B. (eds.) ISC 2006. LNCS, vol. 4176, pp. 46–59. Springer, Heidelberg (2006)
9. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography. In: Awerbuch, B. (ed.) Proceedings of the 23rd Annual ACM Symposium on the Theory of Computing, New Orleans, LS, pp. 542–552. ACM Press, New York (1991)
10. Golle, P., Jakobsson, M., Juels, A., Syverson, P.: Universal re-encryption for mixnets. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 163–178. Springer, Heidelberg (2004)
11. Groth, J.: Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 152–170. Springer, Heidelberg (2004)
12. Krawczyk, H.: The order of encryption and authentication for protecting communications (or: How secure is SSL?). In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 310–331. Springer, Heidelberg (2001)
13. Lindell, Y.: A simpler construction of CCA2-secure public-key encryption under general assumptions. In: Biham, E. (ed.) EUROCRPYT 2003. LNCS, vol. 2656, pp. 241–254. Springer, Heidelberg (2003)
14. Prabhakaran, M., Rosulek, M.: Rerandomizable RCCA encryption. In: Menezes, A. (ed.) CRYPTO 2007. 27th Annual International Cryptology Conference, Santa Barbara, CA, USA. LNCS, vol. 4622, pp. 517–534. Springer, Heidelberg (2007)
15. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: FOCS 1999. 40th Annual Symposium on Foundations of Computer Science, Washington - Brussels - Tokyo, oct 1999, pp. 543–553. IEEE Computer Society Press, Los Alamitos (1999)
16. Shoup, V.: A proposal for an ISO standard for public key encryption (version 2.1) (December 20 2001), http://www.shoup.net/papers/