CCA Secure Unidirectional PRE with Key Pair in the Standard Model without Pairings

Anass Sbai, C. Drocourt, Gilles Dequen

CCA secure unidirectional PRE with key pair in the standard model without pairings

Anass Sbai[1], Cyril Drocourt [1] and Gilles Dequen [1]
[1] *MIS Laboratory, University of Picardie Jules Verne*
{*anass.sbai, cyril.drocourt, dequen.gilles*}*@u-picardie.fr*

Keywords: Proxy Re-Encryption, Unidirectional, Chosen Ciphertext Attack, Cramer-Shoup, Standard model.

Abstract: Secure Data sharing has become an ubiquitous need. One way of pursuing it is to use Proxy Re-Encryption (PRE), which allows delegation of decryption rights selectively. This work tackles the problem of designing a Proxy Re-Encryption that is unidirectional and CCA-secure in the standard model without pairings. In (Zhang et al., 2013) they propose a solution that makes the Cramer-Shoup encryption scheme publicly verifiable and use their result to construct a CCA secure PRE in the standard model. However, we show that their scheme is vulnerable against adaptive chosen ciphertexts attacks. Then we propose a new construction based on Cramer-Shoup crypto-system (Cramer and Shoup, 1998), that is CCA secure without pairings nor random oracle.

1 INTRODUCTION

Proxy Re-Encryption (PRE) is a very useful tool that transforms ciphers intended for Alice into new ciphers that can be decrypted by Bob. Thus, it allows the delegation of the decryption rights on Alice's data, only for the intended recipients (we will also refer to Alice as the delegator and Bob as the delegate). The first scheme was proposed by Blaze, Bleumer, and Strauss (Blaze et al., 1998) whose goal was to avoid that the data must be recovered, decrypted then encrypted with the delegate's key. And thus, relying on a semi-trusted proxy that converts the ciphers using re-encryption keys created by the delegator. The major disadvantage of their scheme is that Alice's delegation to Bob automatically allowed Bob's delegation to Alice, what will later be called bidirectional PRE. This property is due to the fact that re-encryption keys were created using the private keys of the two actors.

In terms of security, such as PKE (Public Key Encryption), we can assess the secrecy of the schemes on three levels:

- IND-CPA (indistinguishability under chosen plaintexts attacks) , where we give the attacker access to an encryption oracle. He has the possibility to query plaintexts of his choice and get the corresponding ciphertexts. Then comes the challenge where he generates two messages with the same length and sends it to the challenger who will then chose randomly to encrypt one of them. The scheme is broken if the adversary is able to guess which of the two messages has been encrypted with a non-negligible probability.
- IND-CCA-1 (indistinguishability under chosen ciphertexts attack), here the attacker has access to an encryption and a decryption oracle. He can send decryption queries as much as he can before the challenge. The later remains the same as in IND-CPA game and the scheme is broken if the adversary guesses which of the two messages has been encrypted with a non-negligible probability.
- IND-CCA-2 (indistinguishability under adaptive chosen ciphertexts attack), the game runs the same as in the IND-CCA-1, but in addition the attacker can send decryption queries to the oracle after the challenge except for the challenge ciphertext.

The main difference in the indistinguishability game between PKE and PRE is that the adversary has access to a re-encryption oracle, thus the proxy should not learn any information about the message during the re-encryption process. We give a more formal definition in section 5. For the rest of this paper, a CCA secure scheme will stand for the IND-CCA-2 security notion.

The construction of the BBS (Blaze et al., 1998) PRE achieves CPA security. In (Ateniese et al., 2006) the authors formalizes the properties and security requirements of PRE that we define in section 2 and propose the first unidirectional scheme. Since then, several works have been published concerning

PRE. The first functional system of an Identity-based Proxy Re-Encryption (IB-PRE) using pairing that is CPA secure was proposed in (Green and Ateniese, 2007). (Canetti and Hohenberger, 2007) proposes the first bidirectional CCA secure PRE scheme where he proves the security of his scheme using the UC framework (Universal Composability framework (Canetti, 2001)). In (Deng et al., 2008), the authors deal with the open problem presented by Canetti concerning the construction of a CCA secure PRE without pairing. (Ateniese et al., 2009) formalizes the notion of key privacy which means that using the re-encryption key we cannot recover the identity of both the delegate and the delegator. He shows why the previous systems were not key-private and proposes a new re-encryption system considered as the first unidirectional PRE that is key private. Their construction is single-use CPA secure. (Chow et al., 2010) has demonstrated the possibility of conducting a CCA attack on the Shao's system (Shao and Cao, 2009) and shows how to fix the issue. They proposed their own scheme without using pairing and relying only on ElGamal and the Schnorr signature. (Selvi et al., 2017) find a flaw in the security proof of Chow's construction and propose to fix it. The system is unidirectional CCA secure in the random oracle model and was implemented in (Sbai et al., 2019).

The security proofs in this model are founded on the existence of an ideal hash function that guarantees uniformly-random outputs which in practice, cannot be instantiated. And there is no proof that a random oracle can exist. Nevertheless, many schemes base their security proofs on random oracle and are used in practice e.g RSA-OAEP. Thus, without showing any vulnerability so far. But still, it is recommended to have a scheme which is proved secure under standard cryptographic assumptions e.g discrete logarithm problem or other. This does not mean that we cannot use hash function, as long as the proofs rely on the assumption of collision resistance or second preimage and not as ideal hash function. Until now, the only unidirectional PRE scheme that has been claimed to be CCA secure without relying on pairing nor random oracles is (Zhang et al., 2013). Where the authors propose a PVPKE (Public Verifiable Public Key Encryption) and use their result to construct a PRE scheme.

In this paper, we first evince that the proposal of (Zhang et al., 2013) is not CCA-secure. Then, we deal with the open problem of constructing a PRE scheme with CCA-security in the standard model without relying on pairing, based only on a DVPKE (Designated Verifier Public Key Encryption) which is Cramer Shoup cryptosystem. And thus, by giving the proxy a private and public key pair, that allows him to check the validity of ciphertexts. Unlike (Wei et al., 2010) we do not need to include a new signature scheme and new key pairs. In this case, it amounts to the same problem where all efficient signature schemes relies on random oracle or pairing. Our method shows that the property of designated verifier in the Cramer Shoup encryption is sufficient to construct a CCA secure PRE, by setting the proxy as peer. We show later in the paper its benefits and inconveniences. We explain our construction and prove its CCA-security. This work was inspired by (Chow et al., 2010) and (Wang et al., 2009) cryptosystems.

2 PRELIMINARIES

2.1 PRE definition

Usually a PRE scheme can be defined as a tuple ζ : $\{Setup, KGen, RkGen, Encrypt, ReEncrypt, Decrypt\}$ where :

- $Setup(1^{\lambda})$: takes as input a security parameter λ and generates the scheme parameters denoted *params*.
- $KGen(params)$: take as input the scheme parameters and generate the pair public/private key (Pk, Sk).
- $RkGen(Sk_a, Pk_b)$: in the case of unidirectional PRE, it takes as input a's private key denoted as Sk_a and b's public key Pk_b to generate the re-encryption key $Rk_{a\to b}$.
- $Encrypt(m, Pk_a) = C_a$: is the encryption function.
- $ReEncrypt(C_a, Rk_{a\to b}) = C_b$: is the re-encryption function.This can be either deterministic or probabilistic.
- $Decrypt(C, Sk) = M$: is the decryption function.

In some cases, we can find two more functions used for encryption and decryption in which the cipher cannot be re-encrypted that we call non-transformable ciphertexts, so that only the owner of the private key can decrypt. There are also schemes like (Wei et al., 2010) and (Purushothama et al., 2013) that uses key pairs for the proxy. The definition remains the same, but some properties can be missed especially the transparency. In (Ateniese et al., 2006) Ateniese gives a more formal definition for PRE and defines concretely the properties such that :

- *Unidirectional*: Delegation of decryption rights from Alice to Bob does not allow Alice to decrypt Bob's cipher.

- *Non-interactive*: The re-encryption key can be generated by Alice without interacting with Bob, and thus using only Bob's public key.
- *Transparent*: Or invisible, meaning that the delegate cannot distinguish between an encrypted message and a re-encrypted message.
- *Key-optimal*: The size of Bob's secret storage must remain unchanged, no matter how many delegations he accepts.
- *Original access*: The sender can decrypt any re-encrypted message of which he was originally the owner.
- *Collusion-safe*: If the proxy and Bob collude, they should not get Alice's secret key.
- *Non-transitive*: The proxy cannot re-delegate re-encryption rights. (e.g from $Rk_{a\to b}$ and $Rk_{b\to c}$ the proxy cannot calculate $Rk_{a\to c}$)
- *Non-transferable*: The proxy and delegates cannot redefine decryption rights. (e.g from $Rk_{a\to b}$ and Pk_c and Sk_b we cannot calculate $Rk_{a\to c}$)
- *Temporary*: Bob can decipher the messages received from Alice only at a certain point in time.

2.2 Cramer-Shoup encryption scheme (Cramer and Shoup, 1998)

It is the first efficient asymmetric encryption scheme that fulfills CCA security in the standard model. This under the assumption that we have a universal one-way hash function, and the Decisional Diffie-Hellman Problem is hard in the underlying group. Assume we have a group $\mathbb{G}$ with large prime order q, the plaintexts are elements of $\mathbb{G}$ and the key generation process as follow : Choose g_1, g_2 from $\mathbb{G}$ and $x_i \xleftarrow{\$} \mathbb{Z}_q$, $i \in \{1,2,..,5\}$ then compute $c = g_1^{x_1} \times g_2^{x_2}$, $d = g_1^{x_3} \times g_2^{x_4}$, $h = g_1^{x_5}$. Next choose a hash function H from the family of universal one-way hash functions. Set the private key as $Sk = (x_1,x_2,x_3,x_4,x_5)$ and the public key $Pk = (q,g_1,g_2,H,c,d,h)$. To encrypt a message the sender must choose $r \xleftarrow{\$} \mathbb{Z}_q$ and compute $u_1 = g_1^r$, $u_2 = g_2^r$, $e = m \times h^r$, $\alpha = H(u_1,u_2,e)$, $v = c^r \times d^{r\times\alpha}$. The ciphertext is (u_1,u_2,e,v), to be decrypted the first step is to verify the validity of the ciphertext by computing $v = u_1^{x_1+\alpha\times x_3} \times u_2^{x_2+\alpha\times x_4}$. If this equality does not hold reject the decryption request else compute and return $m = \frac{e}{u_1^{x_5}}$.

3 Analysis of PVPKE by (Zhang et al., 2013)

The trick that helps to create a CCA secure PRE is the public verifiability of ciphertexts. the first step for the proxy will be to check the validity of the ciphertext before its re-encryption. As an example, (Chow et al., 2010) relies on schnorr signature with a slight modification to get the public verifiability. This makes also ElGamal encryption CCA secure. The schnorr signature used is a sort of NIZK (Non Interactive Zero-Knowledge) proof obtained from the Fiat and Shamir transformation on the interactive schnorr identification scheme. This transformation leads to the use of random oracle in the security model.

As for (Zhang et al., 2013), their scheme is based on Cramer-Shoup encryption. In order to make it publicly verifiable, the authors opted for the use of composite order groups. Thus, based on the problem of factorization of large prime numbers, they can compute using some elements of the private key a values in $\mathbb{Z}_{\phi(N)}$ that can be used for verification by raising it in exponent in $\mathbb{Z}_N$ while keeping $\phi(N)$ hidden. The scheme is CPA secure but not CCA as they had claimed. We review the scheme due to (Zhang et al., 2013), and show how to achieve an adaptive chosen ciphertext attack below.

3.1 Review of the scheme

- $KGen()$:
  - Let p, q , p' and q' be big primes such that $p = 2\times p'+1, q = 2\times q'+1$ and $N = p\times q$
  - Choose g_1, g_2 from $\mathbb{Z}_N$ such that $g_i^{\phi(N)} \equiv 1 \bmod N$ (i = 1, 2)
  - Choose $b \xleftarrow{\$} \mathbb{Z}_{\phi(N)}$ and $x_i \xleftarrow{\$} \mathbb{Z}_{\phi(N)}$ (i = 1, 2, 3, 4, 5)
  - Choose a hash function $H : \{0,1\}^* \to \mathbb{Z}_N$.
  - Compute $x'_i \equiv b \times x_i \bmod \phi(N)$ (i = 1, 2, 3, 4).
  - Compute $c = g_1^{x_1} \times g_2^{x_2}$, $d = g_1^{x_3} \times g_2^{x_4}$, $h = g_1^{x_5}$
  - Return $Sk = (p',q',x_1,x_2,x_3,x_4,x_5)$ and $Pk = (N,g_1,g_2,H,b,x'_1,x'_2,x'_3,x'_4,c,d,h)$
- $Encrypt(Pk,m)$:
  - Choose $r \xleftarrow{\$} \mathbb{Z}_{\mathbb{N}}$
  - Compute $u_1 = g_1^r$, $u_2 = g_2^r$.
  - Compute $e = m \times h^r \bmod N$, $\alpha = H(u_1,u_2,e)$, $v = c^r \times d^{r\times\alpha} \bmod N$
  - Return $C = (u_1,u_2,e,v)$
- $Decrypt(Sk,C)$:

- Compute $\alpha = H(u_1, u_2, e)$ and test if $v^b = u_1^{x'_1+\alpha\times x'_3} \times u_2^{x'_2+\alpha\times x'_4} \bmod N$
- If the condition does not hold return "reject" else return $m = \frac{e}{u_1^{x_5}}$

3.2 Weakness in the PVPKE scheme of (Zhang et al., 2013)

In this section we demonstrate that their PVPKE is not CCA2-secure. This imples that its use to design the PRE is also not secure. We can easily prove it based on IND-CCA2 game .

Recall that the game can be seen as two phases, the first one gives the attacker access to a fixed public key and to decryption oracles. The adversary can submit a large amount of decryption queries without any restriction. Then it comes the challenge which concerns the distinction between two ciphers created by the challenger. Those ciphers correspond to two messages chosen by the attacker and encrypted under the same public key, e.g the attacker sends m_0 & m_1 and receives : $C^* = (u_1, u_2, e, v)$ which is the encryption of m_i with $i \xleftarrow{\$} \{0,1\}$. In the second phase the adversary can submit decryption queries to the oracle except for the challenge $C^* = (u_1, u_2, e, v)$.

The attack consists of computing an invalid but uniformly distributed ciphertext C' such that $C' \neq C^*$ and still the decryption oracle will not reject the request as the verification will pass. The invalid ciphertext could be constructed this way $C' = (u_1^b, u_2^b, e^b, v' = u_1^{(x'_1+x'_3\times\alpha)} \times u_2^{(x'_2+x'_4\times\alpha)})$. The decryption oracle will verify the signature : $v'^b = u_1'^{(x'_1+x'_3\times\alpha)} \times u_2'^{(x'_2+x'_4\times\alpha)} = (u_1^{(x'_1+x'_3\times\alpha)} \times u_2^{(x'_2+x'_4\times\alpha)})^b$ which is valid . Thus the decryption will return $m' = e'/u_1'^{x_5} = m^b \times h^{r\times b}/g_1^{x'_5\times r} = m^b$. Now the attacker has only to test if $m_0^b = m^b$ or $m_1^b = m^b$ and win the challenge.

The other schemes proposed in (Zhang et al., 2013), could also be broken by the same attack.

4 Our construction based on the Cramer-Shoup scheme

As we have demonstrated in the latter section, the public verifiability for the Cramer-Shoup in (Zhang et al., 2013) scheme is not secure. In order to deflect this issue, (Wei et al., 2010) they consider the delegator, the proxy and the delegate as peers, having their own encryption public/private keys and sign/verify keys. Thus the proxy cannot modify the challenge ciphertext and other outside adversaries cannot modify the original and the re-encrypted ciphertext. Nevertheless their system is not fully CCA secure, since no verification is made on the validity of the ciphertexts by the proxy. The idea of using key pairs at the proxy level seemed interesting to us. For our case we use encryption public/private keys which permit the proxy to verify the validity of original ciphertexts and for the delegate to test the validity of re-encrypted ciphertexts. It can be seen as constraining in terms of flexibility and transparency, but is rather advantageous in the sense that we can easily detect malicious proxies with their public keys. In addition to checking the validity of the ciphertexts and re-encrypted ciphertexts, we can also check the well-formness of the re-encryption keys which decreases the damage of DDos attacks. The scheme is proved CCA-secure under DDH assumption in the next section.

4.1 The proposed scheme :

- $Setup(1^\lambda)$:
  - Let $\mathbb{G}$ be a group of prime order q, such that the bit-length of q is the security parameter λ. Choose random elements $g_1, g_2 \in \mathbb{G}$ and two universal one way hash function $H_1 : \{0,1\}^* \to \mathbb{Z}_q$ and $H_2 : \mathbb{G}^2 \to \mathbb{Z}_q^*$. The parameters are $params : (\mathbb{G}, q, g_1, g_2, H_1, H_2)$
- $KGen(params)$:
  - Let us denote (Sk_a, Pk_a) the couple of private/public key associated to the user 'a'. Pick $Sk_a = (x_i : i \in \{1,2,...,7\})$ where $x_i \xleftarrow{\$} \mathbb{Z}_q$ and set $Pk_a = (c, d, h_1, h_2)$ such that $c = g_1^{x_1} \times g_2^{x_2}$, $d = g_1^{x_3} \times g_2^{x_4}$,$h_1 = g_1^{x_5+H_1(c,d)\times x_6}$, $h_2 = g_1^{x_7}$
- $Encrypt(m, Pk_a)$:
  - For non-transformable ciphertexts :

    Choose $r \xleftarrow{\$} \mathbb{Z}_q$
    Compute $u_1 = g_1^r$, $u_2 = g_2^r$
    Compute $e = h_{a2}^r \times m$, $\alpha = H_2(u_1, u_2, e)$
    Compute $v = c_a^r \times d_a^{r\times\alpha}$
    Output $C_a = (u_1, u_2, e, v)$
  - For transformable ciphertext : add the proxy public key Pk_p as input :

    Choose $r \xleftarrow{\$} \mathbb{Z}_q$
    Compute $u_1 = g_1^r$, $u_2 = g_2^r$
    Compute $e = h_{a1}^r \times m$, $\alpha_1 = H_2(u_1, u_2, e)$
    Compute $v = c_a^r \times d_a^{r\times\alpha_1}$, $\alpha_2 = H_2(\alpha_1, v)$
    Compute $v_p = c_p^r \times d_p^{r\times\alpha_2}$
    Output $C_a = (u_1, u_2, e, v, v_p)$

- $RkGen(Sk_a, Pk_b, Pk_p)$:
  - On input user a's private key Sk_a and user b's public key Pk_b and the proxy public key Pk_p :

    Choose $j \xleftarrow{\$} \mathbb{Z}_q$, $k \xleftarrow{\$} \mathbb{Z}_q$

    Compute $rk = \frac{x_{a_5} + H_1(c,d) \times x_{a_6}}{k}$

    Compute $u'_1 = g_1^j$, $u'_2 = g_2^j$

    Compute $e' = h_{b1}^j \times k$, $\alpha'_1 = H_2(u'_1, u'_2, e')$

    Compute $v' = c_b^j \times d_b^{j \times \alpha'_1}$, $\alpha'_2 = H_2(\alpha'_1, v')$, $v'_p = c_p^j \times d_p^{j \times \alpha'_2}$

    Return $Rk_{a \to b} = (rk, u'_1, u'_2, e', v', v'_p)$
- $ReEncrypt(Rk_{a \to b}, C_a, Pk_p, Sk_p)$:
  - On input a re-encryption key, a transformable ciphertext and the proxy public/private key pair:

    Test if $v_p = u_1^{(x_{p_1} + x_{p_3} \times \alpha_2)} \times u_2^{(x_{p_2} + x_{p_4} \times \alpha_2)}$ & $v'_p = u_1'^{(x_{p_1} + x_{p_3} \times \alpha'_2)} \times u_2'^{(x_{p_2} + x_{p_4} \times \alpha'_2)}$

    Choose $\omega \xleftarrow{\$} \mathbb{Z}_q$

    Compute $\beta = u_1^{rk}$, $u''_1 = g_1^\omega$, $u''_2 = g_2^\omega$

    Compute $e'' = e$, $\alpha'' = H_2(u''_1, u''_2, e'', \beta)$

    Compute $v'' = c_b^\omega \times d_b^{\omega \times \alpha''}$

    Output $C_b = (\beta, u'_1, u'_2, e', v', u''_1, u''_2, e'', v'')$
- $Decrypt(sk_b, C_b)$:
  - If $C_b = (u_1, u_2, e, v)$

    Test if $v = u_1^{(x_{b_1} + x_{b_3} \times \alpha)} \times u_2^{(x_{b_2} + x_{b_4} \times \alpha)}$

    Compute $m = \frac{e}{u_1^{x_{b_7}}}$
  - If $C_b = (u_1, u_2, e, v, v_p)$

    Test if $v = u_1^{(x_{b_1} + x_{b_3} \times \alpha_1)} \times u_2^{(x_{b_2} + x_{b_4} \times \alpha_1)}$

    Compute $m = \frac{e}{u_1^{(x_{b_5} + H_1(c,d) \times x_{b_6})}}$
  - If $C_b = (\beta, u'_1, u'_2, e', v', u''_1, u''_2, e'', v'')$

    Test if $v'' = u_1''^{(x_{b_1} + x_{b_3} \times \alpha'')} \times u_2''^{(x_{b_2} + x_{b_4} \times \alpha'')}$ & $v' = u_1'^{(x_{b_1} + x_{b_3} \times \alpha')} \times u_2'^{(x_{b_2} + x_{b_4} \times \alpha')}$

    Compute $k = \frac{e'}{u_1'^{(x_{b_5} + H_1(c,d) \times x_{b_6})}}$

    Compute $m = \frac{e''}{\beta^k}$

4.2 Correctness and security analysis :

- The correctness of decryption for original ciphertext (transformable or non transformable) is trivial since it is the same as in Cramer-Shoup. Correctness of decryption for re-encrypted ciphertexts can be viewed as follow :

$$m = \frac{e''}{\beta^k} = \frac{e''}{u_1'^{rk \times k}} = \frac{m \times g_1^{r \times (x_5 + H_1(c,d) \times x_6)}}{g_1^{\frac{r \times (x_5 + H_1(c,d) \times x_6)}{k} \times k}}$$

- Intuitively, we can check the IND-CCA security of our scheme as follow :
  - For original non-transformable ciphertexts, they're a Cramer-Shoup ciphers which is proved in (Cramer and Shoup, 1998) as CCA-secure under DDH assumption and second preimage.
  - With original transformable ciphers, encryption is almost the same as Cramer-Shoup. However, we compute a v_p so that the proxy could verify the validity of the ciphertext. In the IND CCA-2 game we give the challenger access to both secret keys of the proxy and the delegator so that he can verify the validity of v_p. Otherwise, it does not effect on the security of the scheme since the v_p is computed with another public key, thus it will be linearly independent of v even if we use the same random coin.
  - Re-encryption keys generation was inspired by the work of (Chow et al., 2010), where even if k is leaked which was chosen randomly to compute $rk = \frac{x_5 + H_1(c,d) \times x_6}{k}$, eg. as the proxy and the delegate collude, only $x_5 + H_1(c,d) \times x_6$ could be computed. This linear combination prevents from finding x_5 and x_6 due to the fact that there are as many possible solutions as the cardinal of the group $\mathbb{G}$ to which x_i belong. And thus no information on the private keys is revealed which make the scheme collusion resistant.
  - Re-encrypted ciphertexts are two different Cramer-Shoup ciphers, the first one is used to decrypt the substitution key k created by the delegator which is primordial for the decryption of the second cipher as we saw before in the correctness.

5 Proof of security

We first give the definition of unidirectional single-hop PRE-CCA game following the model of (Canetti and Hohenberger, 2007). We take into account the changes proposed by the authors for unidirectional schemes, since the formal model was intended to bidirectional PRE. We have made changes related to the addition of proxy key pairs :

Let λ be a security parameter. Let $\mathscr{A}$ be an oracle TM representing the adversary. The game consists in an execution of $\mathscr{A}$ with the following oracles. They can be invoked several times in any order, subject to the constraints below:

$\mathcal{O}KGen$: For uncorrupted users return Pk , where $(Pk,Sk) \leftarrow KGen(params)$. For corrupted users return Pk and Sk , where $(Pk,Sk) \leftarrow KGen(params)$

$\mathcal{O}RkGen$: On input Pk_a , Pk_b and Pk_p, the re-encryption key generation algorithm outputs $Rk_{a\rightarrow b}$. We reject the query if it's a re-encryption key generation between a corrupted and uncorrupted key.

$\mathcal{O}Encryption$: For non transformable ciphertext, on input a message m, the output is $C = (u_1,u_2,e,v)$. For original transformable ciphertext, output $C = (u_1,u_2,e,v,v_p)$.

$\mathcal{O}Challenge$: This oracle can be queried only once. On input, (Pk^*,m_0,m_1). where Pk^* is called the challenge key, the oracle chooses a bit $b \xleftarrow{\$} \{0,1\}$ and returns the challenge ciphertext $C = Enc(Pk,m_b)$. (As we note later, the challenge key must be uncorrupted for $\mathscr{A}$ to win).

$\mathcal{O}ReEncryption$: On input (Pk_a,Pk_b,C_a), if Pk_b is corrupted or $a = b$ returns $\perp$. Otherwise it returns C_b

$\mathcal{O}Decryption$: On input (Pk,C), if Pk was not generated before returns $\perp$. Else returns $Decrypt(C,Sk)$

$\mathcal{O}Decision$: This oracle can also be queried only once. On input b' : If $b' = b$ and the challenge key pk^* is not corrupted , then outputs 1 else outputs 0.

We say that $\mathscr{A}$ wins the PRE-CCA game with advantage ε, if the probability over the random choices of $\mathscr{A}$ and the oracles, that the decision oracle is invoked and outputs 1, is at least $1/2+\varepsilon$

Theorem 1. *Our scheme is secure against adaptive chosen ciphertexts attack assuming that (1) the hash functions H_1,H_2 are chosen from a universal one-way family, and (2) the Diffie-Hellman decision problem is hard in the group $\mathbb{G}$.*

We give our scheme's formal proof based on (Wang et al., 2009) and (Chow et al., 2010) proofs as follows :

Assume the external adversarie's algorithm $\mathscr{B}$ breaking the IND-CCA2 property of the scheme ,we use $\mathscr{B}$ to construct algorithm $\mathscr{A}$ distinguish a four tuple (g_1,g_2,u_1,u_2) from $\mathbb{G}$ is a DDH tuple or not. Oralce queries from $\mathscr{B}$ are handled by $\mathscr{A}$ as following:

- **Query to** $\mathcal{O}KGen$: If user A is corrupted , $\mathscr{A}$ randomly chooses $Sk_a = (x_{a_i}) \xleftarrow{\$} \mathbb{Z}_q$ for $(i = 1,2,...,7)$, computes $Pk_a = (g_1,g_2,c_a = g_1^{x_{a_1}} \times g_2^{x_{a_2}}, d_a = g_1^{x_{a_3}} \times g_2^{x_{a_4}}, h_{a_1} = g_1^{x_{a_5}+H_1(c,d)\times x_{a_6}}, h_{a_2} = g_1^{x_{a_7}})$ returns Sk_a,Pk_a which is an identical distribution to the real distribution of real private and public key. For uncorrupted user B, randomly chooses $Sk_b = (x_{b_i}) \xleftarrow{\$} \mathbb{Z}_q$ for $(i = 1,2,...,9)$, computes $Pk_b = (g_1,g_2,c_b = g_1^{x_{b_1}} \times g_2^{x_{b_2}}, d_b = g_1^{x_{b_3}} \times g_2^{x_{b_4}}, h_{b_1} = g_1^{x_{b_5}+H_1(c,d)\times x_{b_6}} \times g_2^{x_{b_8}}, h_{a_2} = g_1^{x_{a_7}} \times g_2^{x_{b_9}})$ and returns Pk_b. Assuming that $g_2 = g_1^w$ the output has an identical distribution to the real distribution of real public key. Which gives us a perfect simulation.
- **Query to** $\mathcal{O}RkGen$: On input Pk_a,Pk_b and Pk_p if one of A and B is corrupted we reject the query. Else $\mathscr{A}$ outputs $Rk_{a\rightarrow b} = (rk \xleftarrow{\$} \mathbb{Z}_{\phi(N)}), \mathscr{C}_{a_1}, v'_p)$ which is indistinguishable with $Rk_{i\rightarrow j} = (\frac{(x_5+H_1(c,d)\times x_6)}{k}, \mathscr{C}_{i_1}, v'_p)$
- **Query to** $\mathcal{O}Encryption$: For a non transformable encryption, given a message m, the encryption algorithm returns $C = (u_1,u_2,e,v) = (g_1^r, g_2^r, u_1^{x_7} \times u_2^{x_9} \times m, c^r \times d^{r\times\alpha})$ where $r \xleftarrow{\$} \mathbb{Z}_q$. This is a perfect simulation as in Cramer-Shoup encryption scheme. For an original transformable encryption, if the users are uncorrupted the encryption algorithm returns $C = (u_1,u_2,e,v,v_p) = (g_1^r, g_2^r, u_1^{(x_5+H_1(c,d)\times x_6)} \times u_2^{x_8} \times m, c^r \times d^{r\times\alpha_1}, c'^r \times d'^{r\times\alpha_1})$ where $r \xleftarrow{\$} \mathbb{Z}_q$ and v_p is computed with a random public key. Else, it output $\perp$. This is also a perfect simulation. We will show below one cannot construct a valid tuple (u_1,u_2,e,v,v_p) by itself with (g_1,g_2,u_1,u_2) being not a DDH tuple, relying on the same method used in Cramer-Shoup Encryption.
- **Query to** $\mathcal{O}ReEncryption$: On input $Pk_a,Pk_b,C_a = (u_1,u_2,e,v,v_p)$ from user i to user j, search in the RkGen list an item including i and j. If it does not exist run the querying to $\mathcal{O}RkGen$. Then the proxy verifies ciphertext's validity by testing, if $v_p \neq u_1^{x_{p_1}} \times u_2^{x_{p_2}} \times u_1^{x_{p_3}\times\alpha_2} \times u_1^{x_{p_4}\times\alpha_2}$ return $\perp$. Else, return $C_j = (\beta,u'_1,u'_2,e',v',u''_1,u''_2,e'',v'') = (u_1^{(x_5+H_1(c,d)\times x_6)}, u'_1,u'_2,e',v',u''_1,u''_2,e'',v'')$ which include two Cramer-Shoup ciphers and have the same distribution as for $(u'^k_1,u'_1,u'_2,e',v',u''_1,u''_2,e'',v'')$. Thus the real output and simulated output are indistinguishable. So this is also a perfect simulation.
- **Query to** $\mathcal{O}Decryption$: In the real decryption, given a re-encrypted ciphertext $C = (\beta,u'_1,u'_2,e',v',u''_1,u''_2,e'',v'')$, the decryption al-

gorithm runs as follows. It first computes $\alpha_1'' = H_2(u_1'', u_2'', e'', \beta)$, and tests if $v'' = u_1^{''(x_1+x_3\times\alpha_1'')} \times u_2^{''(x_2+x_4\times\alpha_1'')}$ & $v' = u_1^{'(x_1+x_3\times\alpha_1')} \times u_2^{'(x_2+x_4\times\alpha_1')}$. If this condition does not hold, the decryption algorithm outputs $\perp$, otherwise, it computes $k = \frac{e'}{u_1^{'(x_5+H_1(c,d)\times x_6)}}$ and outputs $m = \frac{e''}{\beta^k}$. In our simulation, on input C_j from user i to j , $\mathscr{B}$ first verifies ciphertexts's validity. If it's invalid ciphertexts return $\perp$,else computes $k = \frac{e'}{u_1^{'(x_{j_5}+H_1(c,d)\times x_{j_6})} \times u_2^{'x_{j_8}}}$ then outputs $m = \frac{e''}{\beta^k} = \frac{e''}{u_1^{1/(x_{j_5}+H_1(c,d)\times x_{j_6})}}$

As in Cramer-Shoup encryption, if (g_1, g_2, u_1, u_2) is a DDH tuple, our simulated decryption is a perfect decryption. For original ciphertexts the same proof holds as for Cramer-Shoup encryption, there is a slightly difference in the proof for original transformable ciphertexts which we will explain below. And the simulated decryption is also a perfect decryption.

Lemma 2. *If (g_1, g_2, u_1, u_2) is not a DDH tuple, the $\mathscr{O}Decryption$ will reject all invalid ciphertexts, except with negligible probability.*

The proof of this lemma is the same as (Cramer and Shoup, 1998), the only difference is that in $\mathscr{O}Decryption$ simulation for transformable ciphertexts, the adversary must solve these equations :

$x_1 + w\times x_2 = log_{g_1} c \bmod q$
$x_3 + w\times x_4 = log_{g_1} d \bmod q$
$x'_{p_1} + w\times x'_{p_2} = log_{g_1} c_p \bmod q$
$x'_{p_3} + w\times x'_{p_4} = log_{g_1} d_p \bmod q$
$r_1x_1 + r_2\alpha_1x_3 + r_1wx_2 + r_2\alpha_1wx_4 = log_{g_1} v \bmod q$
$r_1x_{p_1} + r_2\alpha_2x_{p_3} + r_1wx_{p_2} + r_2\alpha_2wx_{p_4} = log_{g_1} v_p \bmod q$

Which are linearly independent, thus our simulation is perfect for the external adversary. Unless the proxy reveals his private key. If $\mathscr{A}$ can break our re-encryption scheme, $\mathscr{B}$ can solve the DDH problem in $\mathbb{G}$. Thus we prove our theorem.

6 Conclusion

In this paper, we point out that the schemes in (Zhang et al., 2013) are not CCA-secure , we show how an adversary could distinguish between two ciphers in the IND-CCA2 game. Also, we present a construction of unidirectional proxy re-encryption scheme without bilinear pairing in the standard model. Our scheme is proven CCA-secure in the standard model under decisional Diffie-Hellman assumption and second preimage resistance of the chosen universal one way hash family. We do not consider efficiency, but rather and above all to come up with a solution for one of the two open problems left by (Deng et al., 2008). As a perspective, we will implement the proposed algorithm in order to compare it with other related works, and try to design more efficient schemes.

ACKNOWLEDGEMENTS

We would like to thank Damien Vergnaud for the valuable discussions and for his constructive comments about the flaw, which leads us to find the concrete attack in (Zhang et al., 2013) . This work is supported by ADEME on the VertPom project.

REFERENCES

Ateniese, G., Benson, K., and Hohenberger, S. (2009). Key-private proxy re-encryption. In *Cryptographers' Track at the RSA Conference*, pages 279–294. Springer.

Ateniese, G., Fu, K., Green, M., and Hohenberger, S. (2006). Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC)*, 9(1):1–30.

Blaze, M., Bleumer, G., and Strauss, M. (1998). Divertible protocols and atomic proxy cryptography. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 127–144. Springer.

Canetti, R. (2001). Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE.

Canetti, R. and Hohenberger, S. (2007). Chosen-ciphertext secure proxy re-encryption. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 185–194. ACM.

Chow, S. S., Weng, J., Yang, Y., and Deng, R. H. (2010). Efficient unidirectional proxy re-encryption. In *International Conference on Cryptology in Africa*, pages 316–332. Springer.

Cramer, R. and Shoup, V. (1998). A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Annual International Cryptology Conference*, pages 13–25. Springer.

Deng, R. H., Weng, J., Liu, S., and Chen, K. (2008). Chosen-ciphertext secure proxy re-encryption without pairings. In *International Conference on Cryptology and Network Security*, pages 1–17. Springer.

Green, M. and Ateniese, G. (2007). Identity-based proxy re-encryption. In *International Conference on Applied Cryptography and Network Security*, pages 288–306. Springer.

Purushothama, B., Shrinath, B., and Amberker, B. (2013). Secure cloud storage service and limited proxy re-encryption for enforcing access control in public cloud. *International Journal of Information and Communication Technology*, 5(2):167–186.

Sbai, A., Drocourt, C., and Dequen, G. (2019). Pre as a service within smart grid cities. In *16th International Conference on Security and Cryptography*.

Selvi, S. S. D., Paul, A., and Pandurangan, C. (2017). A provably-secure unidirectional proxy re-encryption scheme without pairing in the random oracle model. In *International Conference on Cryptology and Network Security*, pages 459–469. Springer.

Shao, J. and Cao, Z. (2009). Cca-secure proxy re-encryption without pairings. In *International Workshop on Public Key Cryptography*, pages 357–376. Springer.

Wang, a. X., Wu, W., and Yang, X. (2009). On ddos attack against proxy in re-encryption and re-signature. *Engineering College of Chinese, PR China*.

Wei, P., Wang, X. A., and Yang, X. (2010). Proxy re-encryption schemes with proxy having its own public/private keys. In *2010 2nd International Workshop on Database Technology and Applications*, pages 1–4. IEEE.

Zhang, M., Wang, X. A., Li, W., and Yang, X. (2013). Cca secure publicly verifiable public key encryption without pairings nor random oracle and its applications. *JCP*, 8(8):1987–1994.