Unidirectional Chosen-Ciphertext Secure Proxy Re-encryption

Benoît Libert[1] and Damien Vergnaud[2]

[1] UCL Crypto Group
Place du Levant, 3
1348 Louvain-la-Neuve
Belgium
[2] École Normale Supérieure – C.N.R.S. – I.N.R.I.A.
Département d'informatique, 45 rue d'Ulm
75230 Paris CEDEX 05
France

Abstract. In 1998, Blaze, Bleumer, and Strauss proposed a cryptographic primitive called *proxy re-encryption*, in which a proxy transforms – without seeing the corresponding plaintext – a ciphertext computed under Alice's public key into one that can be opened using Bob's secret key. Recently, an appropriate definition of chosen-ciphertext security and a construction fitting this model were put forth by Canetti and Hohenberger. Their system is *bidirectional*: the information released to divert ciphertexts from Alice to Bob can also be used to translate ciphertexts in the opposite direction. In this paper, we present the first construction of *unidirectional* proxy re-encryption scheme with chosen-ciphertext security in the standard model (i.e. without relying on the random oracle idealization), which solves a problem left open at CCS'07. Our construction is efficient and requires a reasonable complexity assumption in bilinear map groups. Like the Canetti-Hohenberger scheme, it ensures security according to a relaxed definition of chosen-ciphertext introduced by Canetti, Krawczyk and Nielsen.

Keywords: proxy re-encryption, unidirectionality, chosen-ciphertext security, standard model.

1 Introduction

The concept of proxy re-encryption (PRE) dates back to the work of Blaze, Bleumer, and Strauss in 1998 [5]. The goal of such systems is to securely enable the re-encryption of ciphertexts from one key to another, without relying on trusted parties. Recently, Canetti and Hohenberger [12] described a construction of proxy re-encryption providing chosen-ciphertext security according to an appropriate definition of the latter notion for PRE systems. Their construction is *bidirectional*: the information to translate ciphertexts from Alice to Bob can also be used to translate from Bob to Alice. This paper answers the question of how to secure unidirectional proxy re-encryption schemes against chosen-ciphertext attacks – at least in the sense of a natural extension of the Canetti-Hohenberger definition to the unidirectional case – while keeping them efficient.

R. Cramer (Ed.): PKC 2008, LNCS 4939, pp. 360–379, 2008.

Background. In a PRE scheme, a proxy is given some information which allows turning a ciphertext encrypted under a given public key into one that is encrypted under a different key. A naive way for Alice to have a proxy implementing such a mechanism is to simply store her private key at the proxy: when a ciphertext arrives for Alice, the proxy decrypts it using the stored secret key and re-encrypts the plaintext using Bob's public key. The obvious problem with this strategy is that the proxy learns the plaintext and Alice's secret key.

In 1998, Blaze, Bleumer and Strauss [5] (whose work is sometimes dubbed BBS) proposed the first proxy re-encryption scheme, where the plaintext and secret keys are kept hidden from the proxy. It is based on a simple modification of the ElGamal encryption scheme [17]: let $(\mathbb{G}, \cdot)$ be a group of prime order p and let g be a generator of $\mathbb{G}$; Alice and Bob publish the public keys $X = g^x$ and $Y = g^y$ (respectively) and keeps secret their discrete logarithms x and y. To send a message $m \in \mathbb{G}$ to Alice, a user picks uniformly at random an integer $r \in \mathbb{Z}_p$ and transmits the pair (C_1, C_2) where $C_1 = X^r$ and $C_2 = m \cdot g^r$. The proxy is given the re-encryption key $y/x \mod p$ to divert ciphertexts from Alice to Bob via computing $(C_1^{y/x}, C_2) = (Y^r, m \cdot g^r)$.

This scheme is efficient and semantically secure under the Decision Diffie-Hellman assumption in $\mathbb{G}$. It solves the above mentioned problem since the proxy is unable to learn the plaintext or secret keys x or y. Unfortunately, Blaze *et al.* pointed out an inherent limitation: the proxy key y/x also allows translating ciphertexts from Bob to Alice, which may be undesirable in some situations. They left open the problem to design a proxy re-encryption method without this restriction. Another shortcoming of their scheme is that the proxy and the delegatee can collude to expose the delegator's private key x given y/x and y.

In 2005, Ateniese, Fu, Green and Hohenberger [2,3] showed the first examples of *unidirectional* proxy re-encryption schemes based on bilinear maps. Moreover, they obtained the *master key security* property in that the proxy is unable to collude with delegatees in order to expose the delegator's secret. The constructions [2,3] are also efficient, semantically secure assuming the intractability of decisional variants of the Bilinear Diffie-Hellman problem [7].

These PRE schemes only ensure chosen-plaintext security, which seems definitely insufficient for many practical applications. Very recently, Canetti and Hohenberger [12] gave a definition of security against chosen ciphertext attacks for PRE schemes and described an efficient construction satisfying this definition. In their model, ciphertexts should remain indistinguishable even if the adversary has access to a re-encryption oracle (translating adversarially-chosen ciphertexts) and a decryption oracle (that "undoes" ciphertexts under certain rules). Their security analysis takes place in the standard model (without the random oracle heuristic [4]). Like the BBS scheme [5], their construction is *bidirectional* and they left as an open problem to come up with a chosen-ciphertext secure unidirectional scheme.

Related Work. Many papers in the literature – the first one of which being [26] – consider applications where data encrypted under a public key pk_A should eventually be encrypted under a different key pk_B. In proxy encryption schemes

[22,15], a receiver Alice allows a delegatee Bob to decrypt ciphertexts intended to her with the help of a proxy by providing them with shares of her private key. This requires delegatees to store an additional secret for each new delegation. Dodis and Ivan [15] notably present efficient proxy encryption schemes based on RSA, the Decision Diffie-Hellman problem as well as in an identity-based setting [28,7] under bilinear-map-related assumptions.

Proxy re-encryption schemes are a special kind of proxy encryption schemes where delegatees only need to store their own decryption key. They are generally implemented in a very specific mathematical setting and find practical applications in secure e-mail forwarding or distributed storage systems (e.g. [2,3]).

From a theoretical point of view, the first positive obfuscation result for a complex cryptographic functionality was recently presented by Hohenberger, Rothblum, shelat and Vaikuntanathan [21]: they proved the existence of an efficient program obfuscator for a family of circuits implementing re-encryption.

In [19], Green and Ateniese studied the problem of identity-based PRE and proposed a unidirectional scheme that can reach chosen-ciphertext security. Their security results are presented only in the random oracle model. Besides, the recipient of a re-encrypted ciphertext needs to know who the original receiver was in order to decrypt a re-encryption.

Our contribution. In spite of the recent advances, the "*holy grail for proxy re-encryption schemes – a unidirectional, key optimal, and CCA2 secure scheme – is not yet realized*" [20]. This paper aims at investigating this open issue.

We generalize Canetti and Hohenberger's work [12] and present the first construction of chosen-ciphertext secure *unidirectional* proxy re-encryption scheme in the standard model. Our system is efficient and requires a reasonable bilinear complexity assumption. It builds on the unidirectional scheme from [2,3] briefly recalled at the beginning of section 3. The technique used by Canetti-Hohenberger to acquire CCA-security does not directly apply to the latter scheme because, in a straightforward adaptation of [12] to [2], the validity of translated ciphertexts cannot be publicly checked. To overcome this difficulty, we need to modify (and actually randomize) the re-encryption algorithm of Ateniese *et al.* so as to render the validity of re-encrypted ciphertexts publicly verifiable.

Whenever Alice delegates some of her rights to another party, there is always the chance that she will either need or want to revoke those rights later on. In [2,3], Ateniese *et al.* designed another unidirectional PRE scheme that allows for temporary delegations: that is, a scheme where re-encryption keys can only be used during a restricted time interval. We construct such a scheme with temporary delegation and chosen-ciphertext security.

The paper is organized as follows: we recall the concept of unidirectional proxy re-encryption and its security model in section 2.1. We review the properties of bilinear maps and the intractability assumption that our scheme relies on in section 2.2. Section 3 describes the new scheme, gives the intuition behind its construction and a security proof. Section 4 finally shows an adaptation with temporary delegation.

2 Preliminaries

2.1 Model and Security Notions

This section first recalls the syntactic definition of unidirectional proxy re-encryption suggested by Ateniese *et al.* [2,3]. We then consider an appropriate definition of chosen-ciphertext security for unidirectional PRE schemes which is directly inferred from the one given by Canetti and Hohenberger [12] in the bidirectional case. Like [12], we consider security in the *replayable* CCA sense [13] where a harmless mauling of the challenge ciphertext is tolerated.

Definition 1. *A (single hop) unidirectional PRE scheme consists of a tuple of algorithms* $(\mathsf{Global\text{-}setup}, \mathsf{Keygen}, \mathsf{ReKeygen}, \mathsf{Enc}_1, \mathsf{Enc}_2, \mathsf{ReEnc}, \mathsf{Dec}_1, \mathsf{Dec}_2)$*:*

- $\mathsf{Global\text{-}setup}(\lambda) \rightarrow \mathsf{par}$*: this algorithm is run by a trusted party that, on input of a security parameter* λ*, produces a set* par *of common public parameters to be used by all parties in the scheme.*
- $\mathsf{Keygen}(\lambda, \mathsf{par}) \rightarrow (sk, pk)$*: on input of common public parameters* par *and a security parameter* λ*, all parties use this randomized algorithm to generate a private/public key pair* (sk, pk)*.*
- $\mathsf{ReKeygen}(\mathsf{par}, sk_i, pk_j) \rightarrow R_{ij}$*: given public parameters* par*, user* i*'s private key* sk_i *and user* j*'s public key* pk_j*, this (possibly randomized) algorithm outputs a key* R_{ij} *that allows re-encrypting second level ciphertexts intended to* i *into first level ciphertexts encrypted for* j*.*
- $\mathsf{Enc}_1(\mathsf{par}, pk, m) \rightarrow C$*: on input of public parameters* par*, a receiver's public key* pk *and a plaintext* m*, this probabilistic algorithm outputs a first level ciphertext that cannot be re-encrypted for another party.*
- $\mathsf{Enc}_2(\mathsf{par}, pk, m) \rightarrow C$*: given public parameters* par*, a receiver's public key* pk *and a plaintext* m*, this randomized algorithm outputs a second level ciphertext that can be re-encrypted into a first level ciphertext (intended to a possibly different receiver) using the appropriate re-encryption key.*
- $\mathsf{ReEnc}(\mathsf{par}, R_{ij}, C) \rightarrow C'$*: this (possibly randomized) algorithm takes as input public parameters* par*, a re-encryption key* R_{ij} *and a second level ciphertext* C *encrypted under user* i*'s public key. The output is a first level ciphertext* C' *re-encrypted for user* j*. In a single hop scheme,* C' *cannot be re-encrypted any further. If the well-formedness of* C *is publicly verifiable, the algorithm should output '*`invalid`*' whenever* C *is ill-formed w.r.t.* X_i*.*
- $\mathsf{Dec}_1(\mathsf{par}, sk, C) \rightarrow m$*: on input of a private key* sk*, a first level ciphertext* C *and system-wide parameters* par*, this algorithm outputs a message* $m \in \{0,1\}^*$ *or a distinguished message '*`invalid`*'.*
- $\mathsf{Dec}_2(\mathsf{par}, sk, C) \rightarrow m$*: given a private key* sk*, a second level ciphertext* C *and common public parameters* par*, this algorithm returns either a plaintext* $m \in \{0,1\}^*$ *or '*`invalid`*'.*

Moreover, for any common public parameters par*, for any message* $m \in \{0,1\}^*$ *and any couple of private/public key pair* (sk_i, pk_i)*,* (sk_j, pk_j) *these algorithms should satisfy the following conditions of correctness:*

$$\mathsf{Dec}_1(\mathsf{par}, sk_i, \mathsf{Enc}_1(\mathsf{par}, pk_i, m)) = m; \quad \mathsf{Dec}_2(\mathsf{par}, sk_i, \mathsf{Enc}_2(\mathsf{par}, pk_i, m)) = m;$$
$$\mathsf{Dec}_1(\mathsf{par}, sk_j, \mathsf{ReEnc}(\mathsf{par}, \mathsf{ReKeygen}(\mathsf{par}, sk_i, pk_j), \mathsf{Enc}_2(\mathsf{par}, pk_i, m))) = m.$$

To lighten notations, we will sometimes omit to explicitly write the set of common public parameters par, taken as input by all but one of the above algorithms.

CHOSEN-CIPHERTEXT SECURITY. The definition of chosen-ciphertext security that we consider is naturally inspired from the bidirectional case [12] which in turn extends ideas from Canetti, Krawczyk and Nielsen [13] to the proxy re-encryption setting. For traditional public key cryptosystems, in this relaxation of Rackoff and Simon's definition [27], an adversary who can simply turn a given ciphertext into another encryption of the same plaintext is *not* deemed successful. In the game-based security definition, the attacker is notably disallowed to ask for a decryption of a re-randomized version of the challenge ciphertext. This relaxed notion was argued in [13] to suffice for most practical applications.

Our definition considers a challenger that produces a number of public keys. As in [12], we do not allow the adversary to adaptively determine which parties will be compromised. On the other hand, we also allow her to adaptively query a re-encryption oracle and decryption oracles. A difference with [12] is that the adversary is directly provided with re-encryption keys that she is entitled to know (instead of leaving her adaptively request them as she likes). We also depart from [12], and rather follow [2,3], in that we let the target public key be determined by the challenger at the beginning of the game. Unlike [2,3], we allow the challenger to reveal re-encryption keys R_{ij} when j is corrupt for honest users i that differ from the target receiver. We insist that such an enhancement only makes sense for *single-hop* schemes like ours (as the adversary would trivially win the game if the scheme were multi-hop).

Definition 2. *A (single-hop) unidirectional PRE scheme is replayable chosen-ciphertext secure (RCCA) at level* 2 *if the probability*

$$\begin{aligned}\Pr[(pk^\star, sk^\star) \leftarrow \mathsf{Keygen}(\lambda), \{(pk_x, sk_x) \leftarrow \mathsf{Keygen}(\lambda)\}, \{(pk_h, sk_h) \leftarrow \mathsf{Keygen}(\lambda)\},\\ \{R_{x\star} \leftarrow \mathsf{ReKeygen}(sk_x, pk^\star)\},\\ \{R_{\star h} \leftarrow \mathsf{ReKeygen}(sk^\star, pk_h)\}, \{R_{h\star} \leftarrow \mathsf{ReKeygen}(sk_h, pk^\star)\},\\ \{R_{hx} \leftarrow \mathsf{ReKeygen}(sk_h, pk_x)\}, \{R_{xh} \leftarrow \mathsf{ReKeygen}(sk_x, pk_h)\},\\ \{R_{hh'} \leftarrow \mathsf{ReKeygen}(sk_h, pk_{h'})\}, \{R_{xx'} \leftarrow \mathsf{ReKeygen}(sk_x, pk_{x'})\},\\ (m_0, m_1, St) \leftarrow \mathcal{A}^{\mathcal{O}_{1\text{-}dec}, \mathcal{O}_{renc}}\big(pk^\star, \{(pk_x, sk_x)\}, \{pk_h\}, \{R_{x\star}\}, \{R_{h\star}\},\\ \{R_{\star h}\}, \{R_{xh}\}, \{R_{hx}\}, \{R_{hh'}\}, \{R_{xx'}\}\big),\\ d^\star \xleftarrow{R} \{0,1\}, C^\star = \mathsf{Enc}_2(m_{d^\star}, pk^\star), d' \leftarrow \mathcal{A}^{\mathcal{O}_{1\text{-}dec}, \mathcal{O}_{renc}}(C^\star, St) :\\ d' = d^\star]\end{aligned}$$

is negligibly (as a function of the security parameter λ) close to $1/2$ *for any PPT adversary $\mathcal{A}$. In our notation, St is a state information maintained by $\mathcal{A}$ while $(pk^\star, sk^\star)$ is the target user's key pair generated by the challenger that also chooses other keys for corrupt and honest parties. For other honest parties, keys are subscripted by h or h' and we subscript corrupt keys by x or x'. The adversary is given access to all re-encryption keys but those that would allow re-encrypting from the target user to a corrupt one. In the game, $\mathcal{A}$ is said to have advantage*

ε if this probability, taken over random choices of $\mathcal{A}$ and all oracles, is at least $1/2 + \varepsilon$. Oracles $\mathcal{O}_{1\text{-}dec}, \mathcal{O}_{renc}$ proceed as follows:

Re-encryption $\mathcal{O}_{renc}$: *on input (pk_i, pk_j, C), where C is a second level ciphertext and pk_i, pk_j were produced by* Keygen*, this oracle responds with '**invalid**' if C is not properly shaped w.r.t. pk_i. It returns a special symbol $\perp$ if pk_j is corrupt and $(pk_i, C) = (pk^\star, C^\star)$. Otherwise, the re-encrypted first level ciphertext $C' = \mathsf{ReEnc}(\mathsf{ReKeygen}(sk_i, pk_j), C)$ is returned to $\mathcal{A}$.*

First level decryption oracle $\mathcal{O}_{1\text{-}dec}$: *given a pair (pk, C), where C is a first level ciphertext and pk was produced by* Keygen*, this oracle returns '**invalid**' if C is ill-formed w.r.t. pk. If the query occurs in the post-challenge phase (a.k.a. "guess" stage as opposed to the "find" stage), it outputs a special symbol $\perp$ if (pk, C) is a* Derivative *of the challenge pair $(pk^\star, C^\star)$. Otherwise, the plaintext $m = \mathsf{Dec}_1(sk, C)$ is revealed to $\mathcal{A}$.* Derivatives *of $(pk^\star, C^\star)$ are defined as follows.*

> *If C is a first level ciphertext and $pk = pk^\star$ or pk is another honest user, (pk, C) is a* Derivative *of $(pk^\star, C^\star)$ if $\mathsf{Dec}_1(sk, C) \in \{m_0, m_1\}$.*

Explicitly providing the adversary with a second level decryption oracle is useless. Indeed, ciphertexts encrypted under public keys from $\{pk_h\}$ can be re-encrypted for corrupt users given the set $\{R_{hx}\}$. Besides, second level encryptions under $pk^\star$ can be translated for other honest users using $\{R_{\star h}\}$. The resulting first level ciphertext can then be queried for decryption at the first level.

Security of first level ciphertexts. The above definition provides adversaries with a second level ciphertext in the challenge phase. An orthogonal definition of security captures their inability to distinguish first level ciphertexts as well. For *single-hop* schemes, the adversary is granted access to *all* re-encryption keys in this definition. Since first level ciphertexts cannot be re-encrypted, there is indeed no reason to keep attackers from obtaining all honest-to-corrupt re-encryption keys. The re-encryption oracle thus becomes useless since all re-encryption keys are available to $\mathcal{A}$. For the same reason, a second level decryption oracle is also unnecessary. Finally, Derivatives of the challenge ciphertext are simply defined as encryptions of either m_0 or m_1 for the same target public key $pk^\star$. A unidirectional PRE scheme is said RCCA-secure at level 1 if it satisfies this notion.

Remark 1. As in [12], we assume a static corruption model. Proving security against adaptive corruptions turns out to be more challenging. In our model and the one of [12], the challenger generates public keys for all parties and allows the adversary to obtain private keys for some of them. This does not capture a scenario where adversaries generate public keys on behalf of corrupt parties (possibly non-uniformly or as a function of honest parties' public keys) themselves. We also leave open the problem of achieving security in such a setting.

Remark 2. A possible enhancement of definition 2 is to allow adversaries to adaptively choose the target user at the challenge phase within the set of honest players. After having selected a set of corrupt parties among n players at the

beginning, the adversary receives a set of n public keys, private keys of corrupt users as well as corrupt-to-corrupt, corrupt-to-honest and honest-to-honest re-encryption keys. When she outputs messages (m_0, m_1) and the index $i^\star$ of a honest user in the challenge step, she obtains an encryption of $m_{d^\star}$ under $pk_{i^\star}$ together with all honest-to-corrupt re-encryption keys R_{ij} with $i \neq i^\star$.

In this setting, a second level decryption oracle is also superfluous for schemes (like ours) where second level ciphertexts can be publicly turned into first level encryptions of the same plaintext for the same receiver. The scheme that we describe remains secure in this model at the expense of a probability of failure for the simulator that has to foresee which honest user will be attacked with probability $O(1/n)$.

Master secret security. In [2], Ateniese *et al.* define another important security requirement for unidirectional PRE schemes. This notion, termed *master secret security*, demands that no coalition of dishonest delegatees be able to pool their re-encryption keys in order to expose the private key of their common delegator. More formally, the following probability should be negligible as a function of the security parameter λ.

$$\begin{aligned}\Pr[(pk^\star, sk^\star) \leftarrow \mathsf{Keygen}(\lambda),\ & \{(pk_x, sk_x) \leftarrow \mathsf{Keygen}(\lambda)\},\\ & \{R_{\star x} \leftarrow \mathsf{ReKeygen}(sk^\star, pk_x)\},\\ & \{R_{x\star} \leftarrow \mathsf{ReKeygen}(sk_x, pk^\star)\},\\ & \gamma \leftarrow A(pk^\star, \{(pk_x, sk_x)\}, \{R_{\star x}\}, \{R_{x\star}\})\\ & : \gamma = sk^\star]\end{aligned}$$

At first glance, this notion might seem too weak in that it does not consider colluding delegatees who would rather undertake to produce a new re-encryption key $R_{\star x'}$ that was not originally given and allows re-encrypting from the target user to another malicious party x'. As stressed in [2] however, *all* known unidirectional PRE schemes fail to satisfy such a stronger notion of security. It indeed remains an open problem to construct a scheme withstanding this kind of *transfer of delegation* attack.

The notion of RCCA security at the first level is easily seen to imply the master secret security and we will only discuss the former.

2.2 Bilinear Maps and Complexity Assumptions

Groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p are called *bilinear map groups* if there is a mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following properties:

1. bilinearity: $e(g^a, h^b) = e(g, h)^{ab}$ for any $(g, h) \in \mathbb{G} \times \mathbb{G}$ and $a, b \in \mathbb{Z}$;
2. efficient computability for any input pair;
3. non-degeneracy: $e(g, h) \neq 1_{\mathbb{G}_T}$ whenever $g, h \neq 1_{\mathbb{G}}$.

We shall assume the intractability of a variant of the Decision Bilinear Diffie-Hellman problem.

Definition 3. *The* 3**-Quotient Decision Bilinear Diffie-Hellman** *assumption (3-QDBDH) posits the hardness of distinguishing* $e(g,g)^{b/a}$ *from random given* $(g, g^a, g^{(a^2)}, g^{(a^3)}, g^b)$. *A distinguisher* $\mathcal{B}$ (t,ε)*-breaks the assumption if it runs in time* t *and*

$$\big|Pr[\mathcal{B}(g, g^a, g^{(a^2)}, g^{(a^3)}, g^b, e(g,g)^{b/a}) = 1 | a, b \xleftarrow{R} \mathbb{Z}_p^*] - Pr[\mathcal{B}(g, g^a, g^{(a^2)}, g^{(a^3)}, g^b, e(g,g)^z) = 1 | a, b, z \xleftarrow{R} \mathbb{Z}_p^*]\big| \geq \varepsilon.$$

The 3-QDBDH problem is obviously not easier than the (q-DBDHI) problem [6] for $q \geq 3$, which is to recognize $e(g,g)^{1/a}$ given $(g, g^a, \ldots, g^{(a^q)}) \in \mathbb{G}^{q+1}$. Dodis and Yampolskiy showed that this problem was indeed hard in generic groups [16]. Their result thus implies the hardness of 3-QDBDH in generic groups.

Moreover, its intractability for any polynomial time algorithm can be classified among *mild* decisional assumptions (according to [11]) as its strength does not depend on the number of queries allowed to adversaries whatsoever.

2.3 One-Time Signatures

As an underlying tool for applying the Canetti-Halevi-Katz methodology [14], we need one-time signatures. Such a primitive consists of a triple of algorithms $\mathsf{Sig} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ such that, on input of a security parameter λ, $\mathcal{G}$ generates a one-time key pair (ssk, svk) while, for any message M, $\mathcal{V}(\sigma, svk, M)$ outputs 1 whenever $\sigma = \mathcal{S}(ssk, M)$ and 0 otherwise.

As in [14], we need strongly unforgeable one-time signatures, which means that no PPT adversary can create a new signature for a previously signed message (according to [1]).

Definition 4. $\mathsf{Sig} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ *is a strong one-time signature if the probability*

$$Adv^{\mathsf{OTS}} = \Pr\big[\ (ssk, svk) \leftarrow \mathcal{G}(\lambda); (M, St) \leftarrow \mathcal{F}(svk); \sigma \leftarrow \mathcal{S}(ssk, M); (M', \sigma') \leftarrow \mathcal{F}(M, \sigma, svk, St) : \mathcal{V}(\sigma', svk, M') = 1 \wedge (M', \sigma') \neq (M, \sigma)\ \big],$$

where St *denotes the state information maintained by* $\mathcal{F}$ *between stages, is negligible for any PPT forger* $\mathcal{F}$.

3 The Scheme

Our construction is inspired from the first unidirectional scheme suggested in [2,3] where second level ciphertexts $(C_1, C_2) = (X^r, m \cdot e(g,g)^r)$, that are encrypted under the public key $X = g^x$, can be re-encrypted into first level ciphertexts $(e(C_1, R_{xy}), C_2) = (e(g,g)^{ry}, m \cdot e(g,g)^r)$ using the re-encryption key $R_{xy} = g^{y/x}$. Using his private key y s.t. $Y = g^y$, the receiver can then obtain the message.

The Canetti-Hohenberger method for achieving CCA-security borrows from [14,10,23] in that it appends to the ciphertext a checksum value consisting of an element of $\mathbb{G}$ raised to the random encryption exponent r. In the security proof, the simulator uses the publicly verifiable validity of ciphertexts in groups equipped with bilinear maps. Unfortunately, the same technique does not directly apply to secure the unidirectional PRE scheme of [2] against chosen-ciphertext attacks. The difficulty is that, after re-encryption, level 1 ciphertexts have one component in the target group $\mathbb{G}_T$ and pairings cannot be used any longer to check the equality of two discrete logarithms in groups $\mathbb{G}$ and $\mathbb{G}_T$. Therefore, the simulator cannot tell apart well-shaped level 1 ciphertexts from invalid ones.

The above technical issue is addressed by having the proxy replace C_1 with a pair $(C_1', C_1'') = (R_{xy}^{1/t}, C_1^t) = (g^{y/(tx)}, X^{rt})$, for a randomly chosen "blinding exponent" $t \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ that hides the re-encryption key in C_1', in such a way that all ciphertext components but C_2 remain in $\mathbb{G}$. This still allows the second receiver holding y s.t. $Y = g^y$ to compute $m = C_2/e(C_1', C_1'')^{1/y}$. To retain the publicly verifiable well-formedness of re-encrypted ciphertexts however, the proxy needs to include X^t in the ciphertext so as to prove the consistency of the encryption exponent r w.r.t. the checksum value.

Of course, since the re-encryption algorithm is probabilistic, many first level ciphertexts may correspond to the same second level one. For this reason, we need to tolerate a harmless form of malleability (akin to those accepted as reasonable in [1,13,29]) of ciphertexts at level 1.

3.1 Description

Our system is reminiscent of the public key cryptosystem obtained by applying the Canetti-Halevi-Katz transform [14] to the second selective-ID secure identity-based encryption scheme described in [6][1].

Like the Canetti-Hohenberger construction [12], the present scheme uses a strongly unforgeable one-time signature to tie several ciphertext components altogether and offer a safeguard against chosen-ciphertext attacks in the fashion of Canetti, Halevi and Katz [14]. For simplicity, the description below assumes that verification keys of the one-time signature are encoded as elements from $\mathbb{Z}_p^*$. In practice, such verification keys are typically much longer than $|p|$ and a collision-resistant hash function should be applied to map them onto $\mathbb{Z}_p^*$.

Global-setup(λ)**:** given a security parameter λ, choose bilinear map groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2^\lambda$, generators $g, u, v \stackrel{R}{\leftarrow} \mathbb{G}$ and a strongly unforgeable one-time signature scheme $\mathsf{Sig} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$. The global parameters are

$$\mathsf{par} := \{\mathbb{G}, \mathbb{G}_T, g, u, v, \mathsf{Sig}\}.$$

Keygen(λ)**:** user i sets his public key as $X_i = g^{x_i}$ for a random $x_i \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$.

[1] It was actually shown in [24] that, although the security of the underlying IBE scheme relies on a rather strong assumption, a weaker assumption such as the one considered here was sufficient to prove the security of the resulting public key encryption scheme.

ReKeygen(x_i, X_j)**:** given user i's private key x_i and user j's public key X_j, generate the unidirectional re-encryption key $R_{ij} = X_j^{1/x_i} = g^{x_j/x_i}$.

Enc$_1(m, X_i, \mathsf{par})$**:** to encrypt a message $m \in \mathbb{G}_T$ under the public key X_i at the first level, the sender proceeds as follows.

1. Select a one-time signature key pair $(ssk, svk) \xleftarrow{R} \mathcal{G}(\lambda)$ and set $C_1 = svk$.
2. Pick $r, t \xleftarrow{R} \mathbb{Z}_p^*$ and compute

$$C_2' = X_i^t \quad C_2'' = g^{1/t} \quad C_2''' = X_i^{rt} \quad C_3 = e(g,g)^r \cdot m \qquad C_4 = (u^{svk} \cdot v)^r$$

3. Generate a one-time signature $\sigma = \mathcal{S}(ssk, (C_3, C_4))$ on (C_3, C_4).

The ciphertext is $C_i = \big(C_1, C_2', C_2'', C_2''', C_3, C_4, \sigma\big)$.

Enc$_2(m, X_i, \mathsf{par})$**:** to encrypt a message $m \in \mathbb{G}_T$ under the public key X_i at level 2, the sender conducts the following steps.

1. Select a one-time signature key pair $(ssk, svk) \xleftarrow{R} \mathcal{G}(\lambda)$ and set $C_1 = svk$.
2. Choose $r \xleftarrow{R} \mathbb{Z}_p^*$ and compute

$$C_2 = X_i^r \qquad C_3 = e(g,g)^r \cdot m \qquad C_4 = (u^{svk} \cdot v)^r$$

3. Generate a one-time signature $\sigma = \mathcal{S}(ssk, (C_3, C_4))$ on the pair (C_3, C_4).

The ciphertext is $C_i = \big(C_1, C_2, C_3, C_4, \sigma\big)$.

ReEnc(R_{ij}, C_i)**:** on input of the re-encryption key $R_{ij} = g^{x_j/x_i}$ and a ciphertext $C_i = (C_1, C_2, C_3, C_4, \sigma)$, check the validity of the latter by testing the following conditions

$$e(C_2, u^{C_1} \cdot v) = e(X_i, C_4) \tag{1}$$
$$\mathcal{V}(C_1, \sigma, (C_3, C_4)) = 1. \tag{2}$$

If well-formed, C_i is re-encrypted by choosing $t \xleftarrow{R} \mathbb{Z}_p^*$ and computing

$$C_2' = X_i^t \qquad C_2'' = R_{ij}^{1/t} = g^{(x_j/x_i)t^{-1}} \qquad C_2''' = C_2^t = X_i^{rt}$$

The re-encrypted ciphertext is

$$C_j = \big(C_1, C_2', C_2'', C_2''', C_3, C_4, \sigma\big).$$

If ill-formed, C_i is declared '`invalid`'.

Dec$_1(C_j, sk_j)$**:** the validity of a level 1 ciphertext C_j is checked by testing if

$$e(C_2', C_2'') = e(X_j, g) \tag{3}$$
$$e(C_2''', u^{C_1} \cdot v) = e(C_2', C_4) \tag{4}$$
$$\mathcal{V}(C_1, \sigma, (C_3, C_4)) = 1 \tag{5}$$

If relations (3)-(5) hold, the plaintext $m = C_3 / e(C_2'', C_2''')^{1/x_j}$ is returned. Otherwise, the algorithm outputs '`invalid`'.

$\mathbf{Dec}_2(C_i, sk_i)$: if the level 2 ciphertext $C_i = (C_1, C_2, C_3, C_4, \sigma)$ satisfies relations (1)-(2), receiver i can obtain $m = C_3/e(C_2, g)^{1/x_i}$. The algorithm outputs '`invalid`' otherwise.

Outputs of the re-encryption algorithm are perfectly indistinguishable from level 1 ciphertexts produced by the sender. Indeed, if $\tilde{t} = tx_i/x_j$, we can write

$$C_2' = X_i^t = X_j^{\tilde{t}} \qquad C_2'' = g^{(x_j/x_i)t^{-1}} = g^{\tilde{t}^{-1}} \qquad C_3''' = X_i^{rt} = X_j^{r\tilde{t}}.$$

As in the original scheme described in [2], second level ciphertexts can be publicly turned into first level ciphertexts encrypted for the same receiver if the identity element of $\mathbb{G}$ is used as a re-encryption key.

In the first level decryption algorithm, relations (3)-(5) guarantee that re-encrypted ciphertexts have the correct shape. Indeed, since $C_4 = (u^{C_1} \cdot v)^r$ for some unknown exponent $r \in \mathbb{Z}_p$, equality (4) implies that $C_2''' = C_2'^r$. From (3), it comes that $e(C_2'', C_2''') = e(X_j, g)^r$.

We finally note that first level ciphertexts can be publicly re-randomized by changing (C_2', C_2'', C_3'') into $(C_2'^s, C_2''^{1/s}, C_3'''^s)$ for a random $s \in \mathbb{Z}_p^*$. However, the pairing value $e(C_2'', C_2''')$ remains constant and, re-randomizations of a given first level ciphertext are publicly detectable.

3.2 Security

For convenience, we will prove security under an equivalent formulation of the 3-QDBDH assumption.

Lemma 1. *The 3-QDBDH problem is equivalent to decide whether T equals $e(g,g)^{b/a^2}$ or a random value given $(g, g^{1/a}, g^a, g^{(a^2)}, g^b)$ as input.*

Proof. Given $(g, g^{1/a}, g^a, g^{(a^2)}, g^b)$, we can build a 3-QDBDH instance by setting $(y = g^{1/a}, y^A = g, y^{(A^2)} = g^a, y^{(A^3)} = g^{(a^2)}, y^B = g^b)$, which implicitly defines $A = a$ and $B = ab$. Then, we have $e(y,y)^{B/A} = e(g^{1/a}, g^{1/a})^{(ab)/a} = e(g,g)^{b/a^2}$. The converse implication is easily established and demonstrates the equivalence between both problems. □

Theorem 1. *Assuming the strong unforgeability of the one-time signature, the scheme is RCCA-secure at level 2 under the 3-QDBDH assumption.*

Proof. Let $(A_{-1} = g^{1/a}, A_1 = g^a, A_2 = g^{(a^2)}, B = g^b, T)$ be a modified 3-QDBDH instance. We construct an algorithm $\mathcal{B}$ deciding whether $T = e(g,g)^{b/a^2}$ out of a successful RCCA adversary $\mathcal{A}$.

Before describing $\mathcal{B}$, we first define an event F_{OTS} and bound its probability to occur. Let $C^\star = (svk^\star, C_2^\star, C_3^\star, C_4^\star, \sigma^\star)$ denote the challenge ciphertext given to $\mathcal{A}$ in the game. Let F_{OTS} be the event that $\mathcal{A}$ issues a decryption query for a first level ciphertext $C = (svk^\star, C_2', C_2'', C_2''', C_3, C_4, \sigma)$ or a re-encryption query $C = (svk^\star, C_2, C_3, C_4, \sigma)$ where $(C_3, C_4, \sigma) \neq (C_3^\star, C_4^\star, \sigma^\star)$ but $\mathcal{V}(\sigma, svk, (C_3, C_4)) = 1$. In the "find" stage, $\mathcal{A}$ has simply no information on

$svk^\star$. Hence, the probability of a pre-challenge occurrence of F_{OTS} does not exceed $q_O \cdot \delta$ if q_O is the overall number of oracle queries and δ denotes the maximal probability (which by assumption does not exceed $1/p$) that any one-time verification key svk is output by $\mathcal{G}$. In the "guess" stage, F_{OTS} clearly gives rise to an algorithm breaking the strong unforgeability of the one-time signature. Therefore, the probability $\Pr[F_{\mathsf{OTS}}] \leq q_O/p + Adv^{\mathsf{OTS}}$, where the second term accounts for the probability of definition 4, must be negligible by assumption.

We now proceed with the description of $\mathcal{B}$ that simply halts and outputs a random bit if F_{OTS} occurs. In a preparation phase, $\mathcal{B}$ generates a one-time signature key pair $(ssk^\star, svk^\star) \leftarrow \mathcal{G}(\lambda)$ and provides $\mathcal{A}$ with public parameters including $u = A_1^{\alpha_1}$ and $v = A_1^{-\alpha_1 svk^\star} \cdot A_2^{\alpha_2}$ for random $\alpha_1, \alpha_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$. Observe that u and v define a "hash function" $F(svk) = u^{svk} \cdot v = A_1^{\alpha_1(svk - svk^\star)} \cdot A_2^{\alpha_2}$. In the following, we call HU the set of honest parties, including user $i^\star$ that is assigned the target public key $pk^\star$, and CU the set of corrupt parties. Throughout the game, $\mathcal{A}$'s environment is simulated as follows.

- *Key generation*: public keys of honest users $i \in HU \backslash \{i^\star\}$ are defined as $X_i = A_1^{x_i} = g^{a x_i}$ for a randomly chosen $x_i \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$. The target user's public key is set as $X_{i^\star} = A_2^{x_{i^\star}} = g^{(x_{i^\star} a^2)}$ with $x_{i^\star} \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$. The key pair of a corrupt user $i \in CU$ is set as $(X_i = g^{x_i}, x_i)$, for a random $x_i \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, so that (X_i, x_i) can be given to $\mathcal{A}$. To generate re-encryption keys R_{ij} from player i to player j, $\mathcal{B}$ has to distinguish several situations:
  - If $i \in CU$, $\mathcal{B}$ knows $sk_i = x_i$. Given X_j, it simply outputs X_j^{1/x_i}.
  - If $i \in HU \backslash \{i^\star\}$ and $j = i^\star$, $\mathcal{B}$ returns $R_{ii^\star} = A_1^{x_{i^\star}/x_i} = g^{x_{i^\star} a^2/(a x_i)}$ which is a valid re-encryption key.
  - If $i = i^\star$ and $j \in HU \backslash \{i^\star\}$, $\mathcal{B}$ responds with $R_{i^\star j} = A_{-1}^{x_i/x_{i^\star}} = g^{(a x_i/(x_{i^\star} a^2))}$ that has also the correct distribution.
  - If $i, j \in HU \backslash \{i^\star\}$, $\mathcal{B}$ returns $R_{ij} = g^{x_j/x_i} = g^{(a x_j)/(a x_i)}$.
  - If $i \in HU \backslash \{i^\star\}$ and $j \in CU$, $\mathcal{B}$ outputs $R_{ij} = A_{-1}^{x_j/x_i} = g^{x_j/(a x_i)}$ which is also computable.

- *Re-encryption* queries: when facing a re-encryption query from user i to user j for a second level ciphertext $C_i = (C_1, C_2, C_3, C_4, \sigma)$, $\mathcal{B}$ returns '`invalid`' if relations (1)-(2) are not satisfied.
  - If $i \neq i^\star$ or if $i = i^\star$ and $j \in HU \backslash \{i^\star\}$, $\mathcal{B}$ simply re-encrypts using the re-encryption key R_{ij} which is available in either case.
  - If $i = i^\star$ and $j \in CU$,
    - If $C_1 = svk^\star$, $\mathcal{B}$ is faced with an occurrence of F_{OTS} and halts. Indeed, re-encryptions of the challenge ciphertext towards corrupt users are disallowed in the "guess" stage. Therefore, $(C_3, C_4, \sigma) \neq (C_3^\star, C_4^\star, \sigma^\star)$ since we would have $C_2 \neq C_2^\star$ and $i \neq i^\star$ if $(C_3, C_4, \sigma) = (C_3^\star, C_4^\star, \sigma^\star)$.
    - We are thus left with the case $C_1 \neq svk^\star$, $i = i^\star$ and $j \in CU$. Given

$C_2^{1/x_{i^\star}} = A_2^r$, from $C_4 = F(svk)^r = (A_1^{\alpha_1(svk-svk^\star)} \cdot A_2^{\alpha_2})^r$, $\mathcal{B}$ can compute

$$A_1^r = (g^a)^r = \left(\frac{C_4}{C_2^{\alpha_2/x_{i^\star}}}\right)^{\frac{1}{\alpha_1(svk-svk^\star)}}. \quad (6)$$

Knowing g^{ar} and user j's private key x_j, $\mathcal{B}$ picks $t \xleftarrow{R} \mathbb{Z}_p^*$ to compute

$$C_2' = A_1^t = g^{at} \qquad C_2'' = A_{-1}^{x_j/t} = (g^{1/a})^{x_j/t} \qquad C_2''' = (A_1^r)^t = (g^{ar})^t$$

and return $C_j = (C_1, C_2', C_2'', C_3''', C_3, C_4, \sigma)$ which has the proper distribution. Indeed, if we set $\tilde{t} = at/x_j$, we have $C_2' = X_j^{\tilde{t}}$, $C_2'' = g^{1/\tilde{t}}$ and $C_2''' = X_j^{r\tilde{t}}$.

- *First level decryption* queries: when the decryption of a first level ciphertext $C_j = (C_1, C_2', C_2'', C_2''', C_3, C_4, \sigma)$ is queried under a public key X_j, $\mathcal{B}$ returns '`invalid`' if relations (3)-(5) do not hold. We assume that $j \in HU$ since $\mathcal{B}$ can decrypt using the known private key otherwise. Let us first assume that $C_1 = C_1^\star = svk^\star$. If $(C_3, C_4, \sigma) \neq (C_3^\star, C_4^\star, \sigma^\star)$, $\mathcal{B}$ is presented with an occurrence of F_{OTS} and halts. If $(C_3, C_4, \sigma) = (C_3^\star, C_4^\star, \sigma^\star)$, $\mathcal{B}$ outputs $\perp$ which deems C_j as a Derivative of the challenge pair $(C^\star, X_{i^\star})$. Indeed, it must be the case that $e(C_2'', C_2''') = e(g, X_j)^r$ for the same underlying exponent r as in the challenge phase. We now assume $C_1 \neq svk^\star$.
  - If $j \in HU \backslash \{i^\star\}$, $X_j = g^{ax_j}$ for a known $x_j \in \mathbb{Z}_p^*$. The validity of the ciphertext ensures that $e(C_2'', C_2''') = e(X_j, g)^r = e(g, g)^{arx_j}$ and $C_4 = F(svk)^r = g^{\alpha_1 ar(svk-svk^\star)} \cdot g^{a^2 r\alpha_2}$ for some $r \in \mathbb{Z}_p$. Therefore,
  $$e(C_4, A_{-1}) = e(C_4, g^{1/a}) = e(g, g)^{\alpha_1 r(svk-svk^\star)} \cdot e(g, g)^{ar\alpha_2} \quad (7)$$
  and
  $$e(g, g)^r = \left(\frac{e(C_4, A_{-1})}{e(C_2'', C_2''')^{\alpha_2/x_j}}\right)^{\frac{1}{\alpha_1(svk-svk^\star)}} \quad (8)$$
  reveals the plaintext m since $svk \neq svk^\star$.
  - If $j = i^\star$, we have $X_j = g^{(x_{i^\star} a^2)}$ for a known exponent $x_{i^\star} \in \mathbb{Z}_p^*$. Since $e(C_2'', C_2''') = e(X_{i^\star}, g)^r = e(g, g)^{a^2 rx_{i^\star}}$ and
  $$e(C_4, g) = e(g, g)^{\alpha_1 ar(svk-svk^\star)} \cdot e(g, g)^{a^2 r\alpha_2},$$
  $\mathcal{B}$ can first obtain
  $$\gamma = e(g, g)^{ar} = \left(\frac{e(C_4, g)}{e(C_2'', C_2''')^{\alpha_2/x_{i^\star}}}\right)^{\frac{1}{\alpha_1(svk-svk^\star)}}.$$
  Together with relation (7), γ in turn uncovers
  $$e(g, g)^r = \left(\frac{e(C_4, A_{-1})}{\gamma^{\alpha_2/x_{i^\star}}}\right)^{\frac{1}{\alpha_1(svk-svk^\star)}}$$
  and the plaintext $m = C_3/e(g, g)^r$.

In the "guess" stage, $\mathcal{B}$ must check that m differs from messages m_0, m_1 involved in the challenge query. If $m \in \{m_0, m_1\}$, $\mathcal{B}$ returns $\perp$ according to the replayable CCA-security rules.

- *Challenge*: when she decides that the first phase is over, $\mathcal{A}$ chooses messages (m_0, m_1). At this stage, $\mathcal{B}$ flips a coin $d^\star \xleftarrow{R} \{0,1\}$ and sets the challenge ciphertext as

$$C_1^\star = svk^\star \quad C_2^\star = B^{x_{i^\star}} \quad C_3^\star = m_{d^\star} \cdot T \quad C_4^\star = B^{\alpha_2}$$

and $\sigma = \mathcal{S}(ssk^\star, (C_3, C_4))$.

Since $X_{i^\star} = A_2^{x_{i^\star}} = g^{x_{i^\star} a^2}$ and $B = g^b$, $C^\star$ is a valid encryption of $m_{d^\star}$ with the random exponent $r = b/a^2$ if $T = e(g,g)^{b/a^2}$. In contrast, if T is random in $\mathbb{G}_T$, $C^\star$ perfectly hides $m_{d^\star}$ and $\mathcal{A}$ cannot guess $d^\star$ with better probability than $1/2$. When $\mathcal{A}$ eventually outputs her result $d' \in \{0,1\}$, $\mathcal{B}$ decides that $T = e(g,g)^{b/a^2}$ if $d' = d^\star$ and that T is random otherwise. □

Theorem 2. *Assuming the strong unforgeability of the one-time signature, the scheme is RCCA-secure at level* 1 *under the* 3*-QDBDH assumption.*

Proof. The proof is very similar to the one of theorem 1. Given a 3-QDBDH instance $(A_{-1} = g^{1/a}, A_1 = g^a, A_2 = g^{(a^2)}, B = g^b, T)$, we construct an algorithm $\mathcal{B}$ that decides if $T = e(g,g)^{b/a^2}$.

Before describing $\mathcal{B}$, we consider the same event F_{OTS} as in the proof of theorem 1 except that it can only arise during a decryption query (since there is no re-encryption oracle). Assuming the strong unforgeability of the one-time signature, such an event occurs with negligible probability as detailed in the proof of theorem 1. We can now describe our simulator $\mathcal{B}$ that simply halts and outputs a random bit if F_{OTS} ever occurs. Let also $C^\star = (C_1^\star, C_2'^\star, C_2''^\star, C_2'''^\star, C_3^\star, C_4^\star, \sigma^\star)$ denote the challenge ciphertext at the first level.

Algorithm $\mathcal{B}$ generates a one-time signature key pair $(ssk^\star, svk^\star) \leftarrow \mathcal{G}(\lambda)$ and the same public parameters as in theorem 1. Namely, it sets $u = A_1^{\alpha_1}$ and $v = A_1^{-\alpha_1 svk^\star} \cdot A_2^{\alpha_2}$ with $\alpha_1, \alpha_2 \xleftarrow{R} \mathbb{Z}_p^*$ so that $F(svk) = u^{svk} \cdot v = A_1^{\alpha_1 (svk - svk^\star)} \cdot A_2^{\alpha_2}$. As in the proof of theorem 1, $i^\star$ identifies the target receiver. The attack environment is simulated as follows.

- *Key generation*: for corrupt users $i \in CU$ and almost all honest ones $i \in HU \backslash \{i^\star\}$, $\mathcal{B}$ sets $X_i = g^{x_i}$ for a random $x_i \xleftarrow{R} \mathbb{Z}_p^*$. The target user's public key is defined as $X_{i^\star} = A_1$. For corrupt users $i \in CU$, X_i and x_i are both revealed. All re-encryption keys are computable and given to $\mathcal{A}$. Namely, $R_{ij} = g^{x_j / x_i}$ if $i, j \neq i^\star$; $R_{i^\star j} = A_{-1}^{x_j}$ and $R_{j i^\star} = A_1^{1/x_j}$ for $j \neq i^\star$.
- *First level decryption* queries: when the decryption of a ciphertext $C_j = (C_1, C_2', C_2'', C_2''', C_3, C_4, \sigma)$ is queried for a public key X_j, $\mathcal{B}$ returns '`invalid`' if relations (3)-(5) do not hold. We assume that $j = i^\star$ since $\mathcal{B}$ can decrypt using the known private key x_j otherwise. We have $C_2' = A_1^t$,

$C_2'' = g^{1/t}$, $C_2''' = A_1^{rt}$ for unknown exponents $r, t \in \mathbb{Z}_p^*$. Since $e(C_2'', C_2''') = e(g,g)^{ar}$ and

$$e(C_4, A_{-1}) = e(g,g)^{\alpha_1 r(svk - svk^\star)} \cdot e(g,g)^{ar\alpha_2},$$

$\mathcal{B}$ can obtain

$$e(g,g)^r = \left(\frac{e(C_4, A_{-1})}{e(C_2'', C_2''')^{\alpha_2}} \right)^{\frac{1}{\alpha_1(svk - svk^\star)}}$$

which reveals the plaintext $m = C_3/e(g,g)^r$ as long as $svk \neq svk^\star$. In the event that $C_1 = svk^\star$ in a post-challenge query,

- If $e(C_2'', C_2''') = e(C_2''^\star, C_2'''^\star)$, $\mathcal{B}$ returns $\perp$, meaning that C_j is simply a re-randomization (and thus a Derivative) of the challenge ciphertext.
- Otherwise, we necessarily have $(C_3^\star, C_4^\star, \sigma^\star) \neq (C_3, C_4, \sigma)$, which is an occurrence of F_{OTS} and implies $\mathcal{B}$'s termination.

In the "guess" stage, $\mathcal{B}$ must ensure that m differs from messages m_0, m_1 of the challenge phase before answering the query.

- *Challenge*: when the first phase is over, $\mathcal{A}$ outputs messages (m_0, m_1) and $\mathcal{B}$ flips a bit $d^\star \stackrel{R}{\leftarrow} \{0,1\}$. Then, it chooses $\mu \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and sets

$$\begin{array}{llll} C_2'^\star = A_2^\mu & C_2''^\star = A_{-1}^{1/\mu} & C_2'''^\star = B^\mu \\ C_1^\star = svk^\star & C_3^\star = m_{d^\star} \cdot T & C_4^\star = B^{\alpha_2} \end{array}$$

and $\sigma = \mathcal{S}(ssk^\star, (C_3, C_4))$.

Since $X_{i^\star} = A_1$ and $B = g^b$, $C^\star$ is a valid encryption of $m_{d^\star}$ with the random exponents $r = b/a^2$ and $t = a\mu$ whenever $T = e(g,g)^{b/a^2}$. When T is random, $C^\star$ perfectly hides $m_{d^\star}$ and $\mathcal{A}$ cannot guess $d^\star$ with better probability than $1/2$. Eventually, $\mathcal{B}$ bets that $T = e(g,g)^{b/a^2}$ if $\mathcal{A}$ correctly guesses $d^\star$ and that T is random otherwise. □

3.3 Efficiency

The first level decryption algorithm can be optimized using ideas from [23,25]. Namely, verification tests (3)-(4) can be simultaneously achieved with high confidence by the receiver who can choose a random $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and test whether

$$\frac{e(C_2', C_2'' \cdot C_4^\alpha)}{e(C_2''', u^{svk} \cdot v)^\alpha} = e(g,g)^{x_j}.$$

Hence, computing a quotient of two pairings (which is faster than evaluating two independent pairings [18]) and two extra exponentiations suffice to check the validity of the ciphertext.

It could also be desirable to shorten ciphertexts that are significantly lengthened by one-time signatures and their public keys. To this end, ideas from Boneh and Katz [9] can be used as well as those of Boyen, Mei and Waters [10]. In the latter case, ciphertexts can be made fairly compact as components C_1 and σ

become unnecessary if the checksum value C_4 is computed using the Waters "hashing" technique [30] applied to a collision-resistant hash of C_3. This improvement in the ciphertext size unfortunately comes at the expense of a long public key (made of about 160 elements of $\mathbb{G}$ as in [30]) and a loose reduction.

4 A Scheme with Temporary Delegation

This section describes a variant of our scheme supporting temporary delegation. Like the temporary unidirectional PRE suggested in [2,3], it only allows the proxy to re-encrypt messages from A to B during a limited time period. If the scheme must be set up for T periods, we assume that a trusted server publishes randomly chosen elements $(h_1, \ldots, h_T) \in \mathbb{G}^T$ as global parameters. Alternatively, the server could publish a new value h_i that erases h_{i-1} at period i so as to keep short public parameters.

Global-setup(λ, T)**:** is as in section 3 with the difference that additional random group elements $h_1, \ldots, h_T$ (where T is the number of time intervals that the scheme must be prepared for) are chosen. Global parameters are

$$\mathsf{par} := \{\mathbb{G}, \mathbb{G}_T, g, u, v, h_1, \ldots, h_T, \mathsf{Sig}\}.$$

Keygen(λ)**:** user i's public key is set as $X_i = g^{x_i}$ for a random $x_i \xleftarrow{R} \mathbb{Z}_p^*$.

ReKeygen$(x_i, D_{(\ell,j)})$**:** when user j is willing to accept delegations during period $\ell \in \{1, \ldots, T\}$, he publishes a delegation acceptance value $D_{(\ell,j)} = h_\ell^{x_j}$. Given his private key x_i, user i then generates the temporary re-encryption key is $R_{ij\ell} = D_{(\ell,j)}^{1/x_i} = h_\ell^{x_j/x_i}$.

Enc$_1(m, X_i, \ell, \mathsf{par})$**:** to encrypt $m \in \mathbb{G}_T$ under the public key X_i at the first level during period $\ell \in \{1, \ldots, T\}$, the sender conducts the following steps.

1. Choose a one-time signature key pair $(ssk, svk) \xleftarrow{R} \mathcal{G}(\lambda)$; set $C_1 = svk$.
2. Pick $r, t \xleftarrow{R} \mathbb{Z}_p^*$ and compute

$$C_2' = X_i^t \quad C_2'' = h_\ell^{1/t} \quad C_2''' = X_i^{rt} \quad C_3 = e(g, h_\ell)^r \cdot m \qquad C_4 = (u^{svk} \cdot v)^r$$

3. Generate a one-time signature $\sigma = \mathcal{S}(ssk, (\ell, C_3, C_4))$ on (ℓ, C_3, C_4).

The ciphertext is $C_i = \big(\ell, C_1, C_2', C_2'', C_2''', C_3, C_4, \sigma\big)$.

Enc$_2(m, X_i, \ell, \mathsf{par})$**:** to encrypt $m \in \mathbb{G}_T$ under the public key X_i at level 2 during period ℓ, the sender does the following.

1. Pick a one-time signature key pair $(ssk, svk) \xleftarrow{R} \mathcal{G}(\lambda)$ and set $C_1 = svk$.
2. Choose $r \xleftarrow{R} \mathbb{Z}_p^*$ and compute

$$C_2 = X_i^r \qquad C_3 = e(g, h_\ell)^r \cdot m \qquad C_4 = (u^{svk} \cdot v)^r$$

3. Generate a one-time signature $\sigma = \mathcal{S}(ssk, (\ell, C_3, C_4))$ on (ℓ, C_3, C_4).

The ciphertext is $C_i = \big(\ell, C_1, C_2, C_3, C_4, \sigma\big)$.

ReEnc$(R_{ij\ell}, \ell, C_i)$**:** on input of the re-encryption key $R_{ij\ell} = h_\ell^{x_j/x_i}$ and a ciphertext $C_i = (C_1, C_2, C_3, C_4, \sigma)$, the validity of the latter can be checked exactly as in section 3 (i.e. conditions (1)-(2) must be satisfied). If ill-formed, C_i is declared '`invalid`'. Otherwise, it can be re-encrypted by choosing $t \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and computing

$$C_2' = X_i^t \qquad C_2'' = R_{ij\ell}^{1/t} = h_\ell^{(x_j/x_i)t^{-1}} \qquad C_2''' = C_2^t = X_i^{rt}$$

The re-encrypted ciphertext is $C_j = \big(\ell, C_1, C_2', C_2'', C_2''', C_3, C_4, \sigma\big)$.

Dec$_1(C_j, sk_j)$**:** a first level ciphertext C_j is deemed valid if it satisfies similar conditions to (3)-(5) in the scheme of section 3. Namely, we must have

$$e(C_2', C_2'') = e(X_j, h_\ell) \tag{9}$$
$$e(C_2''', u^{C_1} \cdot v) = e(C_2', C_4) \tag{10}$$
$$\mathcal{V}(svk, \sigma, (\ell, C_3, C_4)) = 1 \tag{11}$$

If C_j is valid, the plaintext $m = C_3/e(C_2'', C_2''')^{1/x_j}$ is returned. Otherwise, the message '`invalid`' is returned.

Dec$_2(C_i, sk_i)$**:** receiver i outputs '`invalid`' if the second level ciphertext $C_i = (\ell, C_1, C_2, C_3, C_4, \sigma)$ is ill-formed. Otherwise, it outputs $m = C_3/e(C_2, h_\ell)^{1/x_i}$.

For such a scheme with temporary delegation, replayable chosen-ciphertext security can be defined by naturally extending definition 2. At the beginning of each time period, the attacker obtains all honest-to-honest, corrupt-to-corrupt and corrupt-to-honest re-encryption keys. At the end of a time interval, she also receives all honest-to-corrupt re-encryption keys if she did not choose to be challenged during that period. When she decides to enter the challenge phase at some period $\ell^\star$, she obtains a challenge ciphertext as well as honest-to-corrupt keys $R_{ij\ell^\star}$ for $i \neq i^\star$.

Throughout all periods, she can access a first level decryption oracle and a re-encryption oracle that uses the current re-encryption keys. As she obtains re-encryption keys in chronological order, it is reasonable to expect that queries are made in chronological order as well. Here, a second level decryption oracle is again useless since second level ciphertexts can be publicly "sent" to the first level while keeping the plaintext and the receiver unchanged.

With this security definition, we can prove the security of this scheme under a slightly stronger (but still reasonable) assumption than in section 3. This assumption, that we call 4-QDBDH, states that it dwells hard to recognize $e(g,g)^{b/a}$ given $(g^a, g^{(a^2)}, g^{(a^3)}, g^{(a^4)}, g^b)$. Again, this assumption is not stronger than the q-DBDHI assumption [6] for $q \geq 4$.

Theorem 3. *Assuming the strong unforgeability of the one-time signature, the scheme is RCCA-secure at both levels under the 4-QDBDH assumption.*

Proof. Detailed in the full version of the paper. □

5 Conclusions and Open Problems

We presented the first unidirectional proxy re-encryption scheme with chosen-ciphertext security in the standard model (i.e. without using the random oracle heuristic). Our construction is efficient and demands a reasonable intractability assumption in bilinear groups. In addition, we applied the same ideas to construct a chosen-ciphertext secure PRE scheme with temporary delegation.

Many open problems still remain. For instance, Canetti and Hohenberger suggested [12] to investigate the construction of a multi-hop unidirectional PRE system. They also mentioned the problem of securely obfuscating CCA-secure re-encryption or other key translation schemes. It would also be interesting to efficiently implement such primitives outside bilinear groups (the recent technique from [8] may be useful regarding this issue). Finally, as mentioned in the end of section 2.1, the design a scheme withstanding transfer of delegation attacks is another challenging task.

Acknowledgements

We are grateful to Jorge Villar for many useful comments and suggestions. We also thank anonymous PKC referees for their comments and Susan Hohenberger for helpful discussions on security models. The first author was supported by the Belgian National Fund for Scientific Research (F.R.S.-F.N.R.S.).

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. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage. In: NDSS (2005)
3. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. ACM TISSEC 9(1), 1–30 (2006)
4. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: ACM CCS 1993, pp. 62–73. ACM Press, New York (1993)
5. Blaze, M., Bleumer, G., Strauss, M.: Divertible Protocols and Atomic Proxy Cryptography. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 127–144. Springer, Heidelberg (1998)
6. Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
7. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
8. Boneh, D., Gentry, C., Hamburg, M.: Space-Efficient Identity Based Encryption Without Pairings. In: FOCS 2007 (to appear, 2007)

9. Boneh, D., Katz, J.: Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
10. Boyen, X., Mei, Q., Waters, B.: Direct Chosen Ciphertext Security from Identity-Based Techniques. In: ACM CCS 2005, pp. 320–329. ACM Press, New York (2005)
11. Boyen, X., Waters, B.: Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
12. Canetti, R., Hohenberger, S.: Chosen-Ciphertext Secure Proxy Re-Encryption. In: ACM CCS 2007, pp. 185–194. ACM Press, New York (2007)
13. 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)
14. Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
15. Dodis, Y., Ivan, A.-A.: Proxy Cryptography Revisited. In: NDSS 2003 (2003)
16. Dodis, Y., Yampolskiy, A.: A Verifiable Random Function with Short Proofs and Keys. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 416–431. Springer, Heidelberg (2005)
17. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985)
18. Granger, R., Smart, N.P.: On Computing Products of Pairings. Cryptology ePrint Archive, Report 2006/172 (2006)
19. Green, M., Ateniese, G.: Identity-Based Proxy Re-encryption. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 288–306. Springer, Heidelberg (2007)
20. Hohenberger, S.: Advances in Signatures, Encryption, and E-Cash from Bilinear Groups. Ph.D. Thesis, MIT (May 2006)
21. Hohenberger, S., Rothblum, G.N., Shelat, A., Vaikuntanathan, V.: Securely Obfuscating Re-encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 233–252. Springer, Heidelberg (2007)
22. Jakobsson, M.: On Quorum Controlled Asymmetric Proxy Re-encryption. In: Imai, H., Zheng, Y. (eds.) PKC 1999. LNCS, vol. 1560, pp. 112–121. Springer, Heidelberg (1999)
23. Kiltz, E.: Chosen-Ciphertext Security from Tag-Based Encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
24. Kiltz, E.: On the Limitations of the Spread of an IBE-to-PKE Transformation. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T.G. (eds.) PKC 2006. LNCS, vol. 3958, pp. 274–289. Springer, Heidelberg (2006)
25. Kiltz, E., Galindo, D.: Direct Chosen-Ciphertext Secure Identity-Based Key Encapsulation without Random Oracles. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 336–347. Springer, Heidelberg (2006)
26. Mambo, M., Okamoto, E.: Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts. IEICE Trans. Fund. Elect. Communications and CS, E80-A/1, 54–63 (1997)
27. Rackoff, C., Simon, D.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)

28. Shamir, A.: Identity based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
29. Shoup, V.: A proposal for the ISO standard for public-key encryption (version 2.1). manuscript (2001), http://shoup.net/
30. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)