Unidirectional Chosen-Ciphertext Secure Proxy Re-Encryption*

Benoît Libert[1] and Damien Vergnaud[2] **

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

Abstract. In 1998, Blaze, Bleumer and Strauss introduced a cryptographic primitive called *proxy re-encryption* (PRE) in which a proxy can transform – without seeing the plaintext – a ciphertext encrypted under one key into an encryption of the same plaintext under another key. The concept has recently drawn renewed interest. Notably, Canetti and Hohenberger showed how to properly define (and realize) chosen-ciphertext security for the primitive. Their system is *bidirectional* as the translation key allows converting ciphertexts in both directions. This paper presents the first *unidirectional* proxy re-encryption schemes with chosen-ciphertext security in the standard model (*i.e.* without the random oracle idealization). The first system provably fits a unidirectional extension of the Canetti-Hohenberger security model. As a second contribution, the paper considers a more realistic adversarial model where attackers may choose dishonest users' keys on their own. It is shown how to modify the first scheme to achieve security in the latter scenario. At a moderate expense, the resulting system provides additional useful properties such as non-interactive temporary delegations. Both constructions are efficient and rely on mild complexity assumptions in bilinear groups. Like the Canetti-Hohenberger scheme, they meet a relaxed flavor of chosen-ciphertext security introduced by Canetti, Krawczyk and Nielsen.

1 Introduction

The concept of proxy re-encryption (PRE) dates back to the work of Blaze, Bleumer, and Strauss in 1998 [11]. 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 [19] 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* in that any information to translate ciphertexts from Alice to Bob can also be used to translate from Bob to Alice. This paper deals with the case of unidirectional PRE schemes and answers the question of how to secure them against chosen-ciphertext attacks while keeping them efficient. We first achieve this goal in the sense of a natural extension of the Canetti-Hohenberger security definition to the unidirectional setting. Then, we re-consider chosen-ciphertext security in a model where weaker assumptions are made on how malicious parties' public keys are generated.

1.1 Background

In a PRE scheme, a proxy is given a piece of information that allows turning a ciphertext encrypted under a given public key into an encryption of the same message under a different key. A naive way for Alice to implement such a mechanism is to simply store her private key at the proxy when she is unavailable: when a ciphertext is heading for her, the proxy decrypts it using its copy of her secret key and re-encrypts the plaintext using Bob's public key. The obvious

* This is the full version of a paper with the same title presented in Public Key Cryptography 2008 [37].

** The first author acknowledges the financial support of the Belgian National Fund for Scientific Research (F.R.S.-F.N.R.S.) and the BCRYPT Interuniversity Attraction Pole. The second author is supported by the European Commission through the IST Program under Contract IST-2002-507932 ECRYPT and by the French *Agence Nationale de la Recherche* through the PACE project.

problem with this strategy is that the proxy learns the plaintext and Alice's private key.

In 1998, Blaze, Bleumer and Strauss [11] (whose work is sometimes dubbed BBS) proposed the first proxy re-encryption scheme where the proxy is kept from knowing plaintexts and secret keys. It is based on a simple modification of the ElGamal encryption scheme [26]: 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 sender 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 $y/x \bmod p$ 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 aforementioned problem since the proxy is unable to learn the plaintext or secret keys x and 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 situations where trust relationships are not symmetric. They left open the problem of designing PRE methods 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 [3, 4] 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 [3, 4] are also efficient, semantically secure assuming the intractability of decisional variants of the Bilinear Diffie-Hellman problem [15].

These PRE schemes only ensure chosen-plaintext security, which seems definitely insufficient for many practical applications. Very recently, Canetti and Hohenberger [19] 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 [9]). Like the BBS scheme [11], their construction is *bidirectional* and they left as an open problem to come up with a chosen-ciphertext secure unidirectional scheme.

1.2 Related Work

Many papers in the literature – the first one of which being [38] – 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 [33, 24], a receiver Alice allows a delegatee Bob to decrypt ciphertexts intended for 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 [24] present efficient proxy encryption schemes based on RSA, the Decision Diffie-Hellman problem as well as in an identity-based setting [42, 15] under bilinear-map-related assumptions.

Proxy re-encryption schemes are a special kind of proxy cryptosystems where delegatees only need to store their own decryption key. They find applications in secure e-mail forwarding, digital rights management (DRM) or distributed storage systems (e.g. [3, 4]). The signature analogue, also suggested by Blaze, Bleumer and Strauss [11] in 1998, of PRE systems was formalized by Ateniese and Hohenberger [5] in 2005. The two techniques were notably combined [44] to design interoperable DRM systems where digital content can be translated between devices from different DRM domains.

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 [32]: they proved the existence of an efficient program obfuscator for a family of circuits implementing re-encryption.

In [29], Green and Ateniese studied the problem of identity-based PRE and proposed a unidirectional scheme that can be made chosen-ciphertext secure. Their security results are presented only in the random oracle model. Also, the recipient of a re-encrypted ciphertext needs to know who the original receiver was in order to decrypt a re-encryption. In the standard model, Chu and Tzeng [23] described another identity-based PRE scheme that extends to provide chosen-ciphertext security. Their scheme is both multi-hop and unidirectional but fails to provide collusion-resistance (also called *master secret security* in [3, 4]) as the delegator's private key is trivially exposed when a dishonest delegatee and a proxy pool their information.

More recently, Ateniese, Benson and Hohenberger [6] analyzed the notion of ciphertext anonymity (a.k.a. key privacy) in proxy re-encryption. This notion demands that even the proxy performing translations be unable to infer useful information on the identities of the participants between which it re-encrypts ciphertexts.

1.3 Our contributions

In the unidirectional case, this paper aims at achieving chosen-ciphertext security in the standard model without sacrificing other security properties such as collusion-resistance. While the scheme of [23] can be modified to satisfy some model of chosen-ciphertext security, it fails to protect the delegator against colluding delegatees and proxies. In particular, this scheme fails to satisfy our security modeling for first level ciphertexts. We believe that, as stressed in [3, 4], the security of delegators against malicious delegatees and proxies should be one of the pursued goals in the design of unidirectional schemes.

To achieve this goal, we first generalize the work of Canetti and Hohenberger [19] and present the first construction of chosen-ciphertext secure and collusion-resistant[3] *unidirectional* proxy re-encryption scheme in the standard model. Although only single-hop (like all known unidirectional schemes that resist collusions), our system is efficient and its security proof requires a non-interactive (and thus falsifiable [39]) complexity assumption in bilinear groups. It builds on the first unidirectional scheme from [3, 4], which we briefly recall 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 [19] to [3], 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.

As a second contribution (and a novelty w.r.t. the proceedings version of the paper [37]), we further strengthen our security model by allowing adversaries to inject their own keys in the system. A limitation of all known proxy re-encryption systems – even including passively secure or bidirectional ones [3, 19] – is that their security is analyzed in a model that implicitly makes the knowledge of secret key (KOSK) assumption [12] and does not capture a scenario where the generation malicious users' public keys is left to adversaries themselves. The KOSK model is frequently used to hedge against certain harmful adversarial behaviors. It typically requires that, before being introduced in a multi-user system, any adversarially-controlled public key should be properly registered and knowledge of the matching secret key should be proven to the certification authority (CA). For the sake of simplicity, security proofs (e.g. [12, 8]) frequently assume that adversaries merely reveal their private key to a so-called 'key registration authority' whenever they create a public key for themselves. As will be discussed in the second part of

[3] The earlier version [37] of this paper appeared before we became aware of the independent work [23] that focuses on the identity-based setting. The present one assumes traditional (*i.e.*, non-identity-based) public keys and however considers stronger adversarial models: dishonest delegation partners are notably allowed to collude with proxies.

section 2.1, this requirement, that amounts to assume some ideal trusted key generation phase, may be worrisome to rely on in practice. We therefore show how to modify our first scheme to prove it secure in a more powerful model, called *chosen key model*, where the adversary can freely choose her own public keys without honestly following the specification of the key generation algorithm. In particular, she may even come up with public keys that are calculated as functions of honest users' keys and for which she does not know the corresponding secret keys. To handle queries involving such wicked public keys, we use techniques that were first introduced for identity-based encryption [13].

Whenever users delegate some of their rights to another party, there is always the chance that they will either need or want to revoke those rights later on. In [3, 4], Ateniese *et al.* designed a unidirectional PRE scheme that allows for temporary delegations: that is, re-encryption keys can only be used during a restricted time interval outside which translations are not possible any longer. The latter temporary PRE assumes a trusted server periodically updating public parameters and also entails the participation of delegatees in each temporary delegation. As a third contribution, we devise[4] a chosen-ciphertext secure scheme with temporary delegation in the chosen key model. Beyond its security against strong adversaries, one of the advantages of this new scheme is that temporary delegations remain non-interactive (*i.e.*, no action from the delegatee is required at each temporary delegation) and do not require to rely on a trusted server publishing modified parameters at discrete time intervals. We additionally outline how to optimize the storage at the proxy when re-encryption rights are granted for several consecutive time periods. Our new scheme also lends itself to extensions such as keyword-controlled delegations, where proxy keys can only re-encrypt ciphertexts that are tagged with specific keywords.

1.4 Roadmap

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 main new scheme, gives the intuition behind its construction and a security proof. We give in section 4 the description of the scheme secure in the *chosen-key* model that additionnaly provides 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.* [3, 4]. 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 [19] in the bidirectional case. Like [19], we consider security in the *replayable* CCA sense [20] where a harmless mauling of the challenge ciphertext is tolerated.

Definition 1. *A (single hop) unidirectional PRE scheme consists of a tuple of algorithms* $\mathsf{Global\text{-}setup}$, Keygen, $\mathsf{ReKeygen}$, Enc_1, Enc_2, ReEnc, Dec_1, 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 public parameters to be used by all parties in the scheme.*

[4] This novel construction improves the first example of chosen-ciphertext secure PRE with temporary delegations given in the first version [37] of the paper. Like [3, 4], the latter system required interactive delegations and dynamically changing public parameters.

- $\mathsf{Keygen}(\lambda, \mathsf{par}) \rightarrow (sk, pk)$*: on input of 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 translating second level ciphertexts intended for* 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 public key* pk *and a plaintext* m*, this randomized algorithm outputs a second level ciphertext that can be re-encrypted into a first level one (intended for a possibly different receiver) using the suitable 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 plaintext* $m \in \{0,1\}^*$ *or a 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:*

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

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 first consider is naturally inspired from the bidirectional case [19] which in turn extends ideas from Canetti, Krawczyk and Nielsen [20] to the proxy re-encryption setting. For traditional public key cryptosystems, in this relaxation of the Rackoff-Simon definition [40], 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 [20] to suffice for most practical applications.

Security of second level ciphertexts. This first definition considers a challenger that produces a number of public keys. As in [19], 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 [19] is that the adversary $\mathcal{A}$ 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 [19], and rather follow [3, 4], in that we let the target public key be determined by the challenger at the beginning of the game. Unlike [3, 4], 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 (as $\mathcal{A}$ 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*

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

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 the 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 such other honest parties, keys are subscripted by h or h' and we subscript corrupt keys by x. The adversary is given access to all non-trivial[5] *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 $\mathcal{O}_{1\text{-}dec}$: 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 returned to $\mathcal{A}$.* Derivative*s of $(pk^\star, C^\star)$ are defined as follows.*

  If C is a first level ciphertext and $pk = pk^\star$ or pk belongs to another honest user, we say that (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 $\{R_{\star h}\}$ and the resulting ciphertext can be queried for decryption at the first level.

Remark. 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)

[5] A *non-trivial* re-encryption key is one that the adversary cannot compute on her own. For instance, corrupt-to-honest proxy keys $\{R_{xh}\}$ are trivial since the adversary can compute them using sk_x.

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 $O(1/n)$.

Security of first level ciphertexts. The above definition provides adversaries with a second level ciphertext in the challenge phase. A complementary definition of security captures their inability to distinguish first level ciphertexts as well. For *single-hop* schemes, $\mathcal{A}$ 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 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, $\mathsf{Derivatives}$ of the challenge ciphertext are simply defined as encryptions of either m_0 or m_1 for the same public key $pk^\star$. A single-hop scheme is said RCCA-secure at level 1 if it satisfies this notion.

Master secret security. In [3], 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.

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

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 [3] however, *all* known unidirectional schemes fail to satisfy such a stronger security level. It remains an open problem to construct systems withstanding these *transfer of delegation* attacks.

In single-hop schemes, 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. In the general multi-hop setting, the notion of master secret security appears to be the most appropriate one. However, no viable construction of multi-hop unidirectional system is known to date. As mentioned earlier, the scheme of [23] indeed fails to be master secret secure.

Chosen-ciphertext security in the chosen-key model. In the previous definitions, we assume a static corruption model as in [19]. In definition 2 as well as in the model of [19], the challenger generates public keys for all parties and allows the adversary to obtain private keys for some of them. These models do not capture a scenario where adversaries may generate public keys on behalf of corrupt parties (possibly non-uniformly or as a function of honest parties' public keys). All previous PRE systems as well as our first scheme are analyzed in models that implicitly make the knowledge of secret key (KOSK) assumption according to which users only publish public keys when they know the underlying private keys. In other settings (such as [12, 8]), similar restrictions are frequently imposed on adversarial behaviors: attackers are allowed to come up with their own public key but are required to also reveal the matching secret key. This mirrors the fact that, upon certification of their public key, users should provide certification authorities (CAs) with a proof of knowledge of their private key.

In the known secret key model, security proofs take advantage of the fact that the simulator itself knows dishonest users' secrets. It is tempting to justify this knowledge by arguing that, upon key registration, the simulator can rewind the adversary to extract her private key using the knowledge extractor [7] of the proof. However, attention must be paid to the fact that rewinding is very problematic in inherently concurrent environments like the Internet. Then, CAs should mandate users to provide more involved and computationally more expensive proofs of knowledge (such as [27] in the random oracle model) that guarantee online extractability. As discussed in [10], current public key infrastructures (PKIs) do not bother to apply such thorough verifications that would suffice to realize the abstract KOSK model.

In this paragraph, we consider a more realistic model where the adversary can arbitrarily choose public keys without demonstrating knowledge of the private keys. The only limitation is that all public keys should fall into some public key space (which is pre-determined by system-wide parameters shared by all parties in the system). This provides the adversary with much more flexibility and power in attacking other honest parties in the system. Schemes that are secure in the known secret key model may not necessarily be secure in the chosen-key model (although we did not find a strict separation in the context of proxy re-encryption).

Security of second level ciphertexts. The definition of second level security in the chosen-key model considers a challenger that produces a set HU of honest users' public keys. As in definition 2, the adversary is allowed to adaptively query a decryption oracle and a re-encryption oracle. This time however, the latter can be queried on input of adversarially-chosen delegatees' public keys. The attacker is again directly provided with all re-encryption keys for which the delegator and the delegatee are both honest. As another enhancement w.r.t. definition 2, she is granted access to a delegation oracle that returns re-encryption keys on behalf of honest delegators for arbitrary delegatees' public keys. By "arbitrary", we mean that the adversary can choose any element of the pre-determined (and publicly recognizable) public key space without necessarily knowing the corresponding secret key. Such a key may even be invalid if the scheme supports invalid public keys (for which no private keys exists) that look like well-formed ones. In the next definition, we also let the target public key be chosen by the adversary (among all public keys in HU) in the challenge phase.

Definition 3. *A (single-hop) unidirectional PRE scheme is replayable chosen-ciphertext secure in the chosen-key model (RCCA-CK) at level* 2 *if the probability*

$$\begin{aligned}\Pr[\{(pk_i, sk_i) \leftarrow \mathsf{Keygen}(\lambda)\}_{i\in HU}, \{R_{ii'} \leftarrow \mathsf{ReKeygen}(sk_i, pk_{i'})\}_{i,i'\in HU}\\ (m_0, m_1, i^\star, St) \leftarrow \mathcal{A}^{\mathcal{O}_{1\text{-}dec}, \mathcal{O}_{renc}, \mathcal{O}_{deleg}}\big(\{pk_i\}_{i\in HU}, \{R_{ii'}\}_{i,i'\in HU}\big),\\ d^\star \xleftarrow{R} \{0,1\}\\ C^\star = \mathsf{Enc}_2(m_{d^\star}, pk_{i^\star})\\ d' \leftarrow \mathcal{A}^{\mathcal{O}_{1\text{-}dec}, \mathcal{O}_{renc}, \mathcal{O}_{deleg}}(C^\star, St):\\ d' = d^\star]\end{aligned}$$

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 the state maintained by $\mathcal{A}$ while $pk_{i^\star}$ denotes the public key of the target user that is chosen by the adversary in the set HU. The adversary $\mathcal{A}$ is given access to all re-encryption keys between honest users. She is also allowed to query any re-encryption key but those that would allow re-encrypting from the target user $i^\star$ to some user under her control. 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:*

- *Delegation $\mathcal{O}_{deleg}$: on input (pk_i, pk) with, where pk_i is a public key in HU (and $i \neq i^\star$ in any stage) and pk is a public key of $\mathcal{A}$'s choosing (for which she is not required to reveal the private key), this oracle responds with the re-encryption key* $\mathsf{ReKeygen}(sk_i, pk)$. *We insist* *that no such query can involve $pk_{i^\star}$ as a delegator at any time.*

– *Re-encryption $\mathcal{O}_{renc}$: on input (pk_i, pk_j, C), where C is a second level ciphertext, $pk_i \in HU$ was part of $\mathcal{A}$'s input and pk_j may be an arbitrary public key supplied by $\mathcal{A}$ (again without handing over the matching private key), this oracle responds with '`invalid`' if C is not properly shaped w.r.t. pk_i. It returns a special symbol $\perp$ if $pk_j \notin HU$ and $(pk_i, C) = (pk_{i^\star}, C^\star)$. Otherwise, $\mathcal{A}$ obtains the re-encrypted ciphertext $C' = \mathsf{ReEnc}(\mathsf{ReKeygen}(sk_i, pk_j), C)$.*
– *First level decryption $\mathcal{O}_{1\text{-}dec}$: given a pair (pk, C), where C is a first level ciphertext and $pk \in HU$, this oracle returns '`invalid`' if C is ill-formed w.r.t. pk. If the query occurs in the post-challenge phase, it outputs a special symbol $\perp$ if (pk, C) is a* Derivative *of the challenge pair $(pk_{i^\star}, C^\star)$. Otherwise, the plaintext $m = \mathsf{Dec}_1(sk, C)$ is revealed to $\mathcal{A}$.* Derivative*s of $(pk_{i^\star}, C^\star)$ are defined as previously.*

  If C is a first level ciphertext and $pk \in HU$, we say that (pk, C) is a Derivative *of $(pk_{i^\star}, C^\star)$ if $\mathsf{Dec}_1(sk, C) \in \{m_0, m_1\}$.*

Although more power is granted to the adversary, the above model is still non-adaptive. In a truly adaptive model, $\mathcal{A}$ would be allowed to dynamically corrupt users that are initially honest. In fact, the scenario of definition 3 is easily seen to be equivalent to a completely similar game (in particular, delegation and re-encryption queries are treated in the same way) where $\mathcal{A}$ first statically chooses which players she wants to corrupt within a set of n users before being given all public keys and corrupt users' private keys. We leave open the problem of handling fully adaptive adversaries here.

Security of first level ciphertexts. As in the known secret key model, a second definition of security captures the 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 (*i.e.*, this time, even $pk_{i^\star}$ can be the delegator's public key when $\mathcal{A}$ invokes oracle $\mathcal{O}_{deleg}$ with delegatees' public keys of her choosing). As above, the re-encryption oracle becomes useless since all possible re-encryption keys are made available to $\mathcal{A}$. Again, Derivatives of the challenge ciphertext are simply defined as encryptions of either m_0 or m_1 for the same target public key $pk_{i^\star}$.

In fact the security of first level encryptions can be captured by a simpler definition where the adversary is challenged on a single honest user's public key pk_0 and is allowed to generate herself any other public key for which she makes delegation queries or re-encryption queries (the honest user being the delegator in either case).

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 $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ 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, introduced for the first time in [14], of the Decision Bilinear Diffie-Hellman (DBDH) problem which is, given (g^a, g^b, g^c) with $a, b, c \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, to distinguish $e(g, g)^{abc}$ from random elements of $\mathbb{G}_T$.

Definition 4 ([14]). *The q-***weak Decision Bilinear Diffie-Hellman Inversion** *assumption (q-wDBDHI) posits the computational infeasibility of distinguishing $e(g, g)^{b/a}$ from random* *given* $(g, g^a, g^{(a^2)}, \ldots, g^{(a^q)}, g^b)$. *A distinguisher* $\mathcal{B}$ (t, ε)*-breaks the assumption if it runs in time* t *and*

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

The q-wDBDHI problem is obviously not easier than the q-DBDHI problem [13], where the challenge is to recognize $e(g,g)^{1/a}$ given $(g, g^a, \ldots, g^{(a^q)}) \in \mathbb{G}^{q+1}$. Dodis and Yampolskiy [25] showed the generic hardness of q-DBDHI and their result implies the generic computational infeasibility of q-wDBDHI. Boneh, Boyen and Goh [14] also gave generic intractability results for a wide class of assumptions that encompasses q-wDBDHI and many others.

To prove our results, we only use the above assumption for *constant* values of q whereas this parameter depends on the number of adversarial queries in several earlier applications (e.g. [25]). In our setting, the intractability of q-wDBDHI can be classified among *mild* decisional assumptions (according to the terminology of [18]) as its strength does not depend on the number of queries allowed to adversaries whatsoever. In some of our schemes, $q = 1$ even suffices and we obtain a slightly relaxed variant of the DBDH problem. The 1-wDBDHI assumption is indeed equivalent to the *Squared Decision Bilinear Diffie-Hellman* assumption which is the infeasibility of deciding whether $T = e(g,g)^{a^2 b}$ on input of (g^a, g^b).

2.3 One-time signatures

As an underlying tool for applying the Canetti-Halevi-Katz methodology [22], 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 [22], 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 [2]).

Definition 5. $\mathsf{Sig} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ *is a strong one-time signature if the probability*

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

where St *denotes* $\mathcal{F}$*'s state information across stages, is negligible for any PPT forger* $\mathcal{F}$.

3 A Unidirectional Scheme in the Known Secret Key Model

Our construction is inspired from the first unidirectional scheme suggested in [3, 4] where second level ciphertexts $(A, B) = (X^r, m \cdot e(g,g)^r)$, that are encrypted under the public key $X = g^x$, can be re-encrypted into first level ciphertexts $(e(A, R_{xy}), B) = (e(g,g)^{ry}, m \cdot e(g,g)^r)$ 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 [22, 17, 34] 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 [3] 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 A with a randomized pair $(A', A'') = (R_{xy}^{1/t}, C_1^t) = (g^{y/(tx)}, X^{rt})$, for a random "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 $m = C_2/e(A', A'')^{1/y}$. 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 [2, 20, 43]) 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 [22] to the second selective-ID secure identity-based encryption scheme described in [13][6].

Like the Canetti-Hohenberger construction [19], 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 [22]. 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 $\mathbb{Z}_p^*$.

- $\mathsf{Global\text{-}setup}(\lambda)$: 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

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

- $\mathsf{Keygen}(\lambda)$: user i sets his public key as $X_i = g^{x_i}$ for a random $x_i \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$.
- $\mathsf{ReKeygen}(x_i, X_j)$: given user i's private key x_i and user j's public key X_j, generate the unidirectional re-encryption key $R_{ij} = X_j^{1/x_i} = g^{x_j/x_i}$.
- $\mathsf{Enc}_1(m, X_i, \mathsf{par})$: 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 $(ssk, svk) \stackrel{R}{\leftarrow} \mathcal{G}(\lambda)$ and set $C_1 = svk$.
  2. Pick $r, t \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and compute

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

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

  The ciphertext is $\mathbf{C}_i = \big(C_1, C_2', C_2'', C_2''', C_3, C_4, \sigma\big)$.

[6] It was actually shown in [35] that, although the security of the underlying IBE scheme relies on a rather strong assumption, a weaker assumption such as the one considered here suffices to prove the security of the resulting public key encryption scheme.

- $\mathsf{Enc}_2(m, X_i, \mathsf{par})$: 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 $\mathbf{C}_i = \big(C_1, C_2, C_3, C_4, \sigma\big)$.
- $\mathsf{ReEnc}(R_{ij}, \mathbf{C}_i)$: on input of the re-encryption key $R_{ij} = g^{x_j/x_i}$ and a ciphertext $\mathbf{C}_i = (C_1, C_2, C_3, C_4, \sigma)$, 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, $\mathbf{C}_i$ is re-encrypted by choosing $t \xleftarrow{R} \mathbb{Z}_p^*$ and computing

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

  The re-encrypted ciphertext is

  $$\mathbf{C}_j = \big(C_1, C_2', C_2'', C_2''', C_3, C_4, \sigma\big).$$

  If ill-formed, $\mathbf{C}_i$ is declared '`invalid`'.
- $\mathsf{Dec}_1(\mathbf{C}_j, sk_j)$: the validity of a level 1 ciphertext $\mathbf{C}_j$ 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`'.
- $\mathsf{Dec}_2(C_i, sk_i)$: if the level 2 ciphertext $\mathbf{C}_i = (C_1, C_2, C_3, C_4, \sigma)$ 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 [3], 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-wDBDHI assumption.

Lemma 1. *The 3-wDBDHI 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 elements $(g, g^{1/a}, g^a, g^{(a^2)}, g^b, T)$, we can construct a 3-wDBDHI 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-wDBDHI assumption.*

Proof. Let $(g, A_{-1} = g^{1/a}, A_1 = g^a, A_2 = g^{(a^2)}, B = g^b, T)$ be a modified 3-wDBDHI instance. We build an algorithm $\mathcal{B}$ deciding if $T = e(g,g)^{b/a^2}$ 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 $\mathbf{C}^\star = (svk^\star, C_2^\star, C_3^\star, C_4^\star, \sigma^\star)$ denote the challenge ciphertext given to $\mathcal{A}$ in the game. Let F_{OTS} be the event that, at some point, $\mathcal{A}$ issues a decryption query for a first level ciphertext $\mathbf{C} = (svk^\star, C_2', C_2'', C_2''', C_3, C_4, \sigma)$ or a re-encryption query $\mathbf{C} = (svk^\star, C_2, C_3, C_4, \sigma)$ 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 5, 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 $\alpha_1, \alpha_2 \xleftarrow{R} \mathbb{Z}_p^*$. 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 $X_i = A_1^{x_i} = g^{ax_i}$ for a randomly chosen $x_i \xleftarrow{R} \mathbb{Z}_p^*$. The target user's public key is set as $X_{i^\star} = A_2^{x_{i^\star}} = g^{(x_{i^\star} a^2)}$ with $x_{i^\star} \xleftarrow{R} \mathbb{Z}_p^*$. The key pair of a corrupt user $i \in CU$ is set as $(X_i = g^{x_i}, x_i)$, for a random $x_i \xleftarrow{R} \mathbb{Z}_p^*$, 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 $i \in HU \backslash \{i^\star\}$ and $j = i^\star$, $\mathcal{B}$ returns $R_{ii^\star} = A_1^{x_{i^\star}/x_i} = g^{x_{i^\star} a^2/(ax_i)}$ which is a valid re-encryption key.
  - If $i = i^\star$ and $j \in HU \backslash \{i^\star\}$, $\mathcal{B}$ responds with $R_{i^\star j} = A_{-1}^{x_i/x_{i^\star}} = g^{(ax_i/(x_{i^\star} a^2))}$ that has also the correct distribution.
  - If $i, j \in HU \backslash \{i^\star\}$, $\mathcal{B}$ returns $R_{ij} = g^{x_j/x_i} = g^{(ax_j)/(ax_i)}$.
  - If $i \in HU \backslash \{i^\star\}$ and $j \in CU$, $\mathcal{B}$ outputs $R_{ij} = A_{-1}^{x_j/x_i} = g^{x_j/(ax_i)}$ which is also computable.
- *Re-encryption* queries: when facing a re-encryption query from user i to user j for a second level ciphertext $\mathbf{C}_i = (C_1, C_2, C_3, C_4, \sigma)$, $\mathcal{B}$ 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

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

  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 $\mathbf{C}_j = (C_1, C_2', C_2'', C_3''', C_3, C_4, \sigma)$ which has the proper distribution. Indeed, if we set $\tilde{t} = at/x_j$, we have

$$C_2' = X_j^{\tilde{t}} \qquad C_2'' = g^{1/\tilde{t}} \qquad C_2''' = X_j^{r\tilde{t}}.$$

- *First level decryption* queries: at any time, $\mathcal{A}$ may ask for the decryption of a first level ciphertext $\mathbf{C}_j = (C_1, C_2', C_2'', C_2''', C_3, C_4, \sigma)$ under a public key X_j. For such a request, $\mathcal{B}$ 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,

$$e(C_4, A_{-1}) = e(C_4, g^{1/a}) = e(g, g)^{\alpha_1 r(svk - svk^\star)} \cdot e(g, g)^{ar\alpha_2} \tag{7}$$

  and

$$e(g, g)^r = \left(\frac{e(C_4, A_{-1})}{e(C_2'', C_2''')^{\alpha_2/x_j}} \right)^{\frac{1}{\alpha_1(svk - svk^\star)}} \tag{8}$$

  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 we know that

$$e(C_2'', C_2''') = e(X_{i^\star}, g)^r = e(g, g)^{a^2 r x_{i^\star}}$$
$$e(C_4, g) = e(g, g)^{\alpha_1 ar(svk - svk^\star)} \cdot e(g, g)^{a^2 r\alpha_2},$$

  $\mathcal{B}$ 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 RCCA-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 generates the challenge ciphertext $\mathbf{C}^\star$ 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 $\sigma = \mathcal{S}(ssk^\star, (C_3^\star, C_4^\star))$.

Since $X_{i^\star} = A_2^{x_{i^\star}} = g^{x_{i^\star} a^2}$ and $B = g^b$, $\mathbf{C}^\star$ 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 $\mathbb{G}_T$, $\mathbf{C}^\star$ 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-*wDBDHI assumption.*

Proof. The proof is very similar to the one of theorem 1. We construct an algorithm $\mathcal{B}$ that is given a 3-wDBDHI instance $(g, A_{-1} = g^{1/a}, A_1 = g^a, A_2 = g^{(a^2)}, B = g^b, T)$ and uses the adversary $\mathcal{A}$ to decide 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 $\mathbf{C}^\star = (C_1^\star, C_2'^\star, C_2''^\star, C_2'''^\star, C_3^\star, C_4^\star, \sigma^\star)$ denote the challenge ciphertext at the first level.

Algorithm $\mathcal{B}$ generates a one-time 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 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 $R_{ji^\star} = A_1^{1/x_j}$ for $j \neq i^\star$.
- *First level decryption* queries: when a ciphertext $\mathbf{C}_j = (C_1, C_2', C_2'', C_2''', C_3, C_4, \sigma)$ is queried for decryption w.r.t. 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 $\mathbf{C}_j$ 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

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

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

Since $X_{i^\star} = A_1$ and $B = g^b$, $\mathbf{C}^\star$ 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, $\mathbf{C}^\star$ 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. $\square$

3.3 Efficiency

The first level decryption algorithm can be optimized using ideas from [34, 36]. 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 [28]) 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 Boyen, Mei and Waters [17] allow for fairly compact ciphertexts as components C_1 and σ become unnecessary if the checksum value C_4 is computed using the Waters "hashing" technique [45] 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 [45]) and a loose reduction.

In the random oracle model, we can simultaneously keep short public keys and ciphertexts if we compute $C_4 = H(C_3)^r$ using a random oracle $H : \{0,1\}^* \to \mathbb{G}$. By programming the latter using standard techniques in the security proof, we additionally get a tight security reduction.

It is also worth mentioning that the random oracle model allows dispensing with trusted setup assumptions for the generation of $u, v \in \mathbb{G}$, the discrete logarithms of which must be safely erased by the trusted party performing the setup in the above description of the scheme.

4 Schemes in the Chosen-Key Model

In this section, we suggest modifications of our first scheme that can be proven secure in the sense of definition 3, where dishonest users' public keys can be arbitrarily chosen *on-the-fly* by the adversary invoking the delegation oracle and the re-encryption oracle.

The main construction that we describe allows for temporary delegations: re-encryption keys are associated with definite time periods during which they can be used to translate ciphertexts. The simpler case where delegations are permanent is tackled with by merely instantiating the scheme with a single time period as explained in section 4.2.

4.1 A Non-Interactive Scheme with Temporary Delegation

We describe a scheme supporting temporary delegation. Like temporary unidirectional PRE suggested in [3, 4, 37][7], it only allows the proxy to re-encrypt messages from A to B during a limited time period, but takes a different approach. Prior proposals involve a trusted server that changes system-wide parameters at discrete time intervals: at the beginning of period i, the server publicizes a group element $h_i \in \mathbb{G}$ that erases the old one h_{i-1}. If the scheme must be prepared for L time periods, elements $h_1, \ldots, h_L$ can alternatively be generated all at once at setup time. This removes the need for a trusted server but incurs linear public storage in the number of time periods. In the random oracle model, the sequence $\{h_i\}_{i=1,\ldots,L}$ can be derived from a random oracle but, even in this case, schemes of [3, 4, 37] retain an interactive (albeit simple) delegation protocol where delegatees publish a delegation acceptance value at the beginning of each period during which they must be able to receive delegations.

We depart from [3, 4, 37] in that we do not assume changing public parameters produced by a trusted server. Public parameters are fixed for the lifetime of the system and we do not require the random oracle model either. Also, our delegation mechanism is kept entirely non-interactive and does not require any action from the delegatee who remains entirely passive: delegation is achieved via a single message sent by the delegator to the proxy as in section 3.

We assume that the scheme is prepared for a polynomial (in λ) number L of time periods. In the description hereafter, both encryption algorithms and the re-encryption algorithm all take the period number ℓ as additional input.

The scheme mixes the ideas of our first construction with the first identity-based encryption scheme suggested by Boneh and Boyen [13]. More precisely, the generation of re-encryption keys is randomized and actually reminiscent of the algorithm deriving decryption keys from identities in the IBE system (when period numbers are seen as identities). In a nutshell, the translation key from i to j during period $\ell \in \{1, \ldots, L\}$ consists of a pair $(A_{ij\ell}, B_{ij\ell}) = (X_j^{1/x_i} \cdot F_i(\ell)^r, X_i^r)$ for some $r \xleftarrow{R} \mathbb{Z}_p^*$ and where $F_i : \{1, \ldots, L\} \to \mathbb{G}$ is an identity-hashing function such as the one used in [13]. The pair $(A_{ij\ell}, B_{ij\ell})$ satisfies $e(X_i, A_{ij\ell}) = e(X_j, g) \cdot e(F_i(\ell), B_{ij\ell})$. Then, when ciphertexts are computed as $(X_i^s, F_i(\ell)^s, m \cdot e(g, g)^s)$ at level 2, the underlying idea of the re-encryption algorithm is to translate them into

$$(e(X_j, g)^s,\ m \cdot e(g, g)^s) = \big(e(X_i^s, A_{ij\ell})/e(F_i(\ell)^s, B_{ij\ell}),\ m \cdot e(g, g)^s\big).$$

In order to preserve the publicly verifiable validity of first level ciphertexts however, the technique of section 3 must be applied twice to postpone the (implicit) calculation of both $e(X_i^s, A_{ij\ell})$ and $e(F_i(\ell)^s, B_{ij\ell})$ until the decryption at level 1.

– $\mathsf{Global\text{-}setup}(\lambda)$: is exactly as in section 4. Common parameters consist of

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

[7] The original version of this paper [37] described a temporary scheme in the same vein as the one suggested by section 3.2 in [3, 4]. This section re-considers the problem of temporary unidirectional delegation by removing interaction with the delegatee in the generation of temporary re-encryption keys and avoiding the reliance on a time server.

– $\mathsf{Keygen}(\lambda)$: user i sets his public key as a pair $pk_i = (X_i = g^{x_i}, Y_i = g^{y_i})$ with $(x_i, y_i) \xleftarrow{R} (\mathbb{Z}_p^*)^2$. Those values implicitly define a function $F_i : \{1, \ldots, L\} \to \mathbb{G}$ such that $F_i(\ell) = g^\ell \cdot Y_i$.

– $\mathsf{ReKeygen}(sk_i, pk_j, \ell)$: given user i's private key $sk_i = (x_i, y_i)$, the public key $pk_j = (X_j, Y_j)$ of user j and a period number $\ell \in \{1, \ldots, L\}$, the delegator i generates a unidirectional re-encryption key for period ℓ as

$$R_{ij\ell} = (A_{ij\ell},\ B_{ij\ell}) = \big(X_j^{1/x_i} \cdot F_i(\ell)^r,\ X_i^r\big)$$

where X_i, Y_i are part of i's key pk_i, $r \in \mathbb{Z}_p^*$ is a randomly chosen exponent.

– $\mathsf{Enc}_1(m, pk_i, \ell, \mathsf{par})$: to encrypt $m \in \mathbb{G}_T$ under the public key $pk_i = (X_i, Y_i)$ at the first level during period ℓ, choose $s, t_1, t_2 \xleftarrow{R} \mathbb{Z}_p^*$ and output

$$\mathbf{C}'_j = (\ell, C_0, C'_1, C''_1, C'''_1, C'_2, C''_2, C'''_2, C_3, C_4, \sigma)$$

where

$$C_0 = svk \qquad C_3 = e(g,g)^s \cdot m \qquad C_4 = (u^{svk} \cdot v)^s \qquad \sigma = \mathcal{S}(ssk, (\ell, C_3, C_4)),$$

for a freshly generated one-time key pair $(ssk, svk) \xleftarrow{R} \mathcal{G}(\lambda)$, and

$$\begin{array}{lll} C'_1 = X_i^{t_1} & C''_1 = (F_i(\ell) \cdot g)^{1/t_1} & C'''_1 = X_i^{st_1} \\ C'_2 = F_i(\ell)^{t_2} & C''_2 = X_i^{1/t_2} & C'''_2 = F_i(\ell)^{st_2}. \end{array}$$

– $\mathsf{Enc}_2(m, pk_i, \ell, \mathsf{par})$: to encrypt $m \in \mathbb{G}_T$ under the public key $pk_i = (X_i, Y_i)$ at level 2, the sender does the following.
  1. Generate a one-time key pair $(ssk, svk) \xleftarrow{R} \mathcal{G}(\lambda)$ and set $C_0 = svk$.
  2. Pick a random exponent $s \xleftarrow{R} \mathbb{Z}_p^*$ and compute $\mathbf{C}$ as

$$\mathbf{C} = (\ell, C_0, C_1, C_2, C_3, C_4, \sigma) = \Big(\ell,\ svk,\ X_i^s,\ F_i(\ell)^s,\ e(g,g)^s \cdot m,\ (u^{svk} \cdot v)^s,\ \sigma\Big)$$

  where $\sigma = \mathcal{S}(ssk, (\ell, C_3, C_4))$.

– $\mathsf{ReEnc}(R_{ij\ell}, \mathbf{C}_i, \ell)$: given the re-encryption key $R_{ij\ell} = (A_{ij\ell}, B_{ij\ell})$ and a second level ciphertext $\mathbf{C}_i = (\ell, C_0, C_1, C_2, C_3, C_4, \sigma)$, reject $\mathbf{C}_i$ if its first component ℓ does not match $R_{ij\ell}$, if $\mathcal{V}(C_0, (\ell, C_3, C_4), \sigma) \neq 1$ or if one of the next equalities fails to hold:

$$e(X_i, C_2) = e(C_1, F_i(\ell)) \qquad e(X_i, C_4) = e(C_1, u^{C_0} \cdot v). \tag{9}$$

Otherwise, choose $t_1, t_2 \xleftarrow{R} \mathbb{Z}_p^*$ and output

$$\mathbf{C}'_j = (\ell, C_0, C'_1, C''_1, C'''_1, C'_2, C''_2, C'''_2, C_3, C_4, \sigma)$$

where

$$\begin{array}{lll} C'_1 = X_i^{t_1} & C''_1 = A_{ij\ell}^{1/t_1} & C'''_1 = C_1^{t_1} = X_i^{st_1} \\ C'_2 = F_i(\ell)^{t_2} & C''_2 = B_{ij\ell}^{1/t_2} & C'''_2 = C_2^{t_2} = F_i(\ell)^{st_2}. \end{array}$$

– $\mathsf{Dec}_1(\mathbf{C}_j, sk_j)$: given $sk_j = (x_j, y_j)$, parse $\mathbf{C}_j$ as $(\ell, C_0, C'_1, C''_1, C'''_1, C'_2, C''_2, C'''_2, C_3, C_4, \sigma)$. Return '`invalid`' if $\mathcal{V}(C_0, (\ell, C_3, C_4), \sigma) \neq 1$ or if these relations are not satisfied.

$$e(C'_1, C_4) = e(C'''_1, u^{C_0} \cdot v) \tag{10}$$

$$e(C'_2, C_4) = e(C'''_2, u^{C_0} \cdot v) \tag{11}$$

$$e(C'_1, C''_1) = e(X_j, g) \cdot e(C'_2, C''_2). \tag{12}$$

Otherwise, return $m = C_3 \cdot \big(e(C''_2, C'''_2)/e(C''_1, C'''_1)\big)^{1/x_j}$.

– $\mathsf{Dec}_2(\mathbf{C}_i, sk_i)$: parse $\mathbf{C}_i$ as $\mathbf{C}_i = (C_0, C_1, C_2, C_3, C_4, \sigma)$ and sk_i as (x_i, y_i). Return '`invalid`' if $\mathcal{V}(C_0, (\ell, C_3, C_4), \sigma) \neq 1$ or if relation (9) does not hold. Otherwise, return $m = C_1/e(C_3, g)^{1/x_i}$.

As in section 4, the correctness of the re-encryption procedure follows from the fact that re-encryption keys $R_{ij\ell} = (A_{ij\ell},\ B_{ij\ell})$ satisfy

$$\begin{aligned} e(A_{ij\ell}, X_i) &= e(X_j^{1/x_i}, X_i) \cdot e\big(F_i(\ell)^r, X_i\big) \\ &= e(X_j, g) \cdot e(F_i(\ell), B_{ij\ell}). \end{aligned}$$

When raising both members to the power $s \in \mathbb{Z}_p^*$, we find

$$e(X_i^s, A_{ij\ell}) = e(X_j, g)^s \cdot e\big(F_i(\ell)^s, B_{ij\ell}\big).$$

Since $C_4 = (u^{C_0} \cdot v)^s$, where $s \in \mathbb{Z}_p^*$ is the encryption exponent, relations (10)-(11) imply that $C_1''' = {C_1'}^s$ and $C_2''' = {C_2'}^s$. From (12), it comes that

$$\frac{e(C_1'', C_1''')}{e(C_2'', C_2''')} = e(X_j, g)^s.$$

The scheme is slightly less efficient and features longer ciphertexts than in section 3. On the other hand, it offers security guarantees in a stronger model. As established by the next two theorems, its security additionally rests on a weaker intractability assumption which is the q-wDBDHI assumption with $q = 1$.

At level 2, the considered security model is a straightforward extension (where the adversary chooses both a target user $i^\star$ and a target period $\ell^\star$ that the end of the "find" stage) of the one expressed by definition 3 with simple restrictions: the adversary is not allowed to query delegations from user $i^\star$ for the attacked period $\ell^\star$. In addition, she is disallowed to query the re-encryption of the challenge pair $(pk_{i^\star}, C^\star)$ during the target period $\ell^\star$ or a first level decryption of its derivatives (the notion of derivative being generalized by imposing that a second level encryption and its derivatives pertain to the same period number).

Theorem 3. *If* Sig *is a strongly secure one-time signature, the scheme with temporary delegation is RCCA-CK-secure at level* 2 *under the* 1*-wDBDHI assumption.*

Proof. We show how to solve a 1-wDBDHI instance $(g, A = g^a, B = g^b, T \stackrel{?}{=} e(g,g)^{b/a})$ using an RCCA adversary $\mathcal{A}$ in the chosen key model.

As in previous proofs, we first call F_{OTS} the event the $\mathcal{A}$ comes up with a query on a valid ciphertext including components $(svk, \ell, C_3, C_4, \sigma)$ such that $svk = svk^\star$ is the same as in the challenge phase but $(\ell, C_3, C_4, \sigma) \neq (\ell^\star, C_3^\star, C_4^\star, \sigma^\star)$. In the "find" stage, $\mathcal{A}$ has simply no information on $svk^\star$ so that F_{OTS} occurs with probability at most $q_O \cdot \delta$ if q_O is the overall number of oracle queries and δ denotes the maximal probability (which does not exceed $1/p$) that any one-time verification key svk is produced 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 Adv^{OTS} denotes the maximal probability of defeating the one-time signature security, must be negligible by assumption.

We now describe a 1-wDBDHI solver $\mathcal{B}$. In a preparation phase, the latter generates a one-time key pair $(ssk^\star, svk^\star) \leftarrow \mathcal{G}(\lambda)$ and provides $\mathcal{A}$ with public parameters including $u = g^{\alpha_1}$ and $v = g^{-\alpha_1 svk^\star} \cdot A^{\alpha_2}$ for random $\alpha_1, \alpha_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$. Observe that u and v define a function $F(svk) = u^{svk} \cdot v = g^{\alpha_1(svk - svk^\star)} \cdot A^{\alpha_2}$. In the model that extends definition 3, $\mathcal{B}$ has to guess upfront the honest user that will be $\mathcal{A}$'s prey. In addition, it must foresee the time period $\ell^\star$ for which the challenge ciphertext will have to be generated. Hence, $\mathcal{B}$ draws two integers

$i^\star \stackrel{R}{\leftarrow} HU = \{1,\ldots,N\}$ as $\ell^\star \stackrel{R}{\leftarrow} \{1,\ldots,L\}$, hoping that the attack will pertain to user $i^\star$ at period $\ell^\star$ (the probability $1/LN$ of this event is non-negligible as long as L and N are both polynomial). The whole attack environment is then emulated as follows:

- *Key generation*:
  - The expected target user's public key $pk_{i^\star} = (X_{i^\star}, Y_{i^\star})$ is chosen as $X_{i^\star} = A = g^a$, $Y_{i^\star} = g^{-\ell^\star} \cdot A^{y_{i^\star}}$ for a randomly chosen $y_{i^\star} \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$. The corresponding private key sk_i includes the unknown exponent $\tilde{x}_{i^\star} = a$.
  - For other users $i \in HU\backslash\{i^\star\}$, public keys are chosen as $X_i = A^{x_i} = g^{ax_i}$, $Y_i = g^{y_i}$, with $x_i, y_i \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, so that the first element of the underlying secret key is $\tilde{x}_i = ax_i$.
  - To generate re-encryption keys $R_{ij\ell}$ for players $i,j \in HU$ and period ℓ, $\mathcal{B}$ can first compute $X_j^{1/\tilde{x}_i}$ which in turn allows for the generation of $R_{ij\ell}$ for known random exponents $r \in \mathbb{Z}_p^*$ (alternatively, it can be directly given to the adversary as a meta-key enabling the generation of keys for all periods). Three situations must be distinguished:
    - If $i \in HU\backslash\{i^\star\}$ and $j = i^\star$, $\mathcal{B}$ can compute $R_{ii^\star}$ using $X_{i^\star}^{1/\tilde{x}_i} = g^{1/x_i}$ which yields a correct key since $g^{1/x_i} = (g^a)^{1/(ax_i)} = X_{i^\star}^{1/\tilde{x}_i}$.
    - If $i = i^\star$ and $j \in HU\backslash\{i^\star\}$, $\mathcal{B}$ computes $R_{i^\star j\ell}$ using $X_j^{1/\tilde{x}_{i^\star}} = g^{x_j}$ which has the correct shape since $g^{x_j} = (g^{ax_j})^{1/a} = X_j^{1/\tilde{x}_{i^\star}}$.
    - If $i,j \in HU\backslash\{i^\star\}$, $X_j^{1/\tilde{x}_i} = g^{x_j/x_i}$ is also computable since $\tilde{x}_i = ax_i$, $\tilde{x}_j = ax_j$.

- *Queries*:
  - *Delegation queries:* the simulator halts and declares failure if $\mathcal{A}$ ever queries a re-encryption key $R_{i^\star j\ell^\star}$ for a delegatee $j \notin HU$ (which means that $\mathcal{B}$ was unfortunate in its choice of $i^\star$ and $\ell^\star$ at the beginning of the game). Otherwise,
    - For a query involving a honest delegator $i \in HU\backslash\{i^\star\}$ and a delegatee's public key $pk_j = (X_j, Y_j)$ supplied by $\mathcal{A}$ (recall that we may have $pk_j \neq pk_i$ for all $i \in HU$, in which case $\mathcal{A}$ does *not* have to reveal the matching sk_j), we have $X_i = g^{ax_i}, Y_i = g^{y_i}$, for known values $x_i, y_i \in \mathbb{Z}_p^*$, and $\mathcal{B}$ uses techniques from [13] to generate re-encryption keys $R_{ij\ell}$. Namely, it chooses $r \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and outputs

$$R_{ij\ell} = \Big(g^{r(\ell+y_i)}, X_i^r \cdot X_j^{-1/(\ell+y_i)}\Big). \tag{13}$$

    If we define $\tilde{r} = r - \frac{x_j}{ax_i(\ell+y_i)}$, we observe that $R_{ij\ell}$ has the required distribution since

$$\begin{aligned} X_j^{1/\tilde{x}_i} \cdot (g^\ell \cdot Y_i)^{\tilde{r}} &= X_j^{1/ax_i} \cdot (g^{\ell+y_i})^{\tilde{r}} \\ &= X_j^{1/ax_i} \cdot (g^{\ell+y_i})^r \cdot (g^{\ell+y_i})^{-\frac{x_j}{ax_i(\ell+y_i)}} \\ &= g^{r(\ell+y_i)} \end{aligned}$$

    and $X_i^{\tilde{r}} = X_i^r \cdot X_j^{-1/(\ell+y_i)}$. We observe that, provided $\ell + y_i \neq 0$, $\mathcal{B}$ can compute both components of (13) without knowing the delegatee's private key $x_j = \log_g(X_j)$. Since y_i is chosen at random for $i \in HU\backslash\{i^\star\}$, the probability to have $\ell + y_i \neq 0$ for all $\ell \in \{1,\ldots,L\}$ and all i is at least $1 - LN/p$, which is overwhelming when L and N are both polynomial in λ.

- in the case $i = i^\star$ and $\ell \neq \ell^\star$, let $pk_j = (X_j, Y_j)$ be the delegatee's public key supplied by $\mathcal{A}$. Since $X_{i^\star} = A = g^a$ and $Y_{i^\star} = g^{-\ell^\star} \cdot A^{y_{i^\star}}$ for known values $y_{i^\star}, \ell^\star \in \mathbb{Z}_p^*$, $\mathcal{B}$ can generate re-encryption keys as pairs

$$R_{i^\star j\ell} = \Big((g^\ell \cdot Y_{i^\star})^r \cdot X_j^{-\frac{y_{i^\star}}{\ell-\ell^\star}}, X_{i^\star}^r \cdot X_j^{-\frac{1}{\ell-\ell^\star}} \Big). \tag{14}$$

If we set $\tilde{r} = r - \frac{x_j}{a(\ell-\ell^\star)}$, we see that $R_{i^\star j\ell}$ has the proper shape since

$$\begin{aligned} & X_j^{1/\tilde{x}_{i^\star}} \cdot (g^\ell \cdot Y_{i^\star})^{\tilde{r}} \\ &= X_j^{1/a} \cdot (g^{\ell-\ell^\star} \cdot A^{y_{i^\star}})^{\tilde{r}} \\ &= X_j^{1/a} \cdot (g^{\ell-\ell^\star} \cdot A^{y_{i^\star}})^r \cdot (g^{\ell-\ell^\star})^{-\frac{x_j}{a(\ell-\ell^\star)}} \cdot A^{-\frac{y_{i^\star} x_j}{a(\ell-\ell^\star)}} \\ &= (g^{\ell-\ell^\star} \cdot A^{y_{i^\star}})^r \cdot X_j^{-\frac{y_{i^\star}}{(\ell-\ell^\star)}} \end{aligned}$$

and $X_{i^\star}^{\tilde{r}} = X_{i^\star}^r \cdot X_j^{-1/(\ell-\ell^\star)}$. Again, the above computation can be carried out without knowing $x_j = \log_g(X_j)$.

- *Re-encryption* queries: for any adversarially-chosen public key $pk_j = (X_j, Y_j)$, $\mathcal{B}$ can compute re-encryption keys for any delegator $i \in HU \backslash \{i^\star\}$. It can also compute $R_{i^\star j\ell}$ on behalf of the target user $i^\star$ whenever $\ell \neq \ell^\star$. We thus assume that $i = i^\star$ and $\ell = \ell^\star$. If relations (9) do not hold, $\mathcal{B}$ returns '`invalid`'. Otherwise,
  - · If $C_0 = svk^\star$, we necessarily have an occurrence of F_{OTS} since, after the challenge phase re-encryptions of $C^\star$ to users outside HU are not permitted for period $\ell^\star$.
  - · If $C_0 \neq svk^\star$, $i = i^\star$ and $j \notin HU$. Given $C_1 = X_{i^\star}^s = A^s$ and $C_4 = F(svk)^s$, $\mathcal{B}$ can compute $g^s = (C_4/C_1^{\alpha_2})^{1/\alpha_1(svk-svk^\star)}$. Also, we have $C_2 = (g^{\ell^\star} \cdot Y_{i^\star})^s = A^{sy_{i^\star}}$. Then, $\mathcal{B}$ picks $t_1, t_2 \xleftarrow{R} \mathbb{Z}_p^*$ to compute

$$\begin{aligned} C_1' &= g^{t_1} \cdot (A^{y_{i^\star}})^{t_1} \\ C_1'' &= X_j^{1/t_1} \\ C_1''' &= (g^s)^{t_1} \cdot C_2^{t_1} = (g^s)^{t_1} \cdot (A^{sy_{i^\star}})^{t_1} \end{aligned}$$

    and

$$C_2' = (A^{y_{i^\star}})^{t_2} \qquad C_2'' = X_j^{1/t_2} \qquad C_2''' = C_2^{t_2} = (A^{sy_{i^\star}})^{t_2}$$

    and return the re-encrypted ciphertext $\mathbf{C}_j = (C_0, C_1', C_1'', C_1''', C_2', C_2'', C_2''', C_3, C_4, \sigma)$ which satisfies the validity test (10)-(12).

- *First level decryption* queries: when a ciphertext

$$\mathbf{C}_i = (\ell, C_0, C_1', C_1'', C_1''', C_2', C_2'', C_2''', C_3, C_4, \sigma)$$

is queried w.r.t. a public key X_i with $i \in HU$ at the first level, we necessarily have $e(C_1'', C_1''')/e(C_2'', C_2''') = e(X_i, g)^s$, for an unknown $s \in \mathbb{Z}_p^*$, if $\mathbf{C}_i$ is valid. Since $X_i = A^{x_i}$, for a known value x_i (which equals 1 if $i = i^\star$), and

$$e(C_4, g) = e(g, g)^{\alpha_1 s(svk-svk^\star)} \cdot e(A, g)^{as\alpha_2},$$

$\mathcal{B}$ obtains

$$e(g, g)^s = \left(\frac{e(C_4, g) \cdot e(C_2'', C_2''')^{\alpha_2 x_i}}{e(C_1'', C_1''')^{\alpha_2 x_i}} \right)^{\frac{1}{\alpha_1(C_0 - svk^\star)}} \tag{15}$$

which reveals the plaintext $m = C_3/e(g, g)^s$ as long as $svk \neq svk^\star$. In the event that $C_0 = svk^\star$ after the challenge phase,

- If $\ell \neq \ell^\star$, we have an occurrence of F_{OTS} since ℓ is signed along with C_3 and C_4 in both encryption algorithms.
- If $\ell = \ell^\star$ and $C_4 = C_4^\star$, $\mathcal{B}$ returns $\perp$ to indicate that $\mathbf{C}_i$ is a re-randomization (and thus a Derivative) of the challenge ciphertext.
- Otherwise, we have $(C_3^\star, C_4^\star, \sigma^\star) \neq (C_3, C_4, \sigma)$ and thus another occurrence of F_{OTS}.

Again, when handling post-challenge queries, $\mathcal{B}$ only returns m if $m \notin \{m_0, m_1\}$.

- *Challenge*: when $\mathcal{A}$ comes up with messages $m_0, m_1 \in \mathbb{G}_T$ and indices $i \in HU$, $\ell \in \{1, \ldots, L\}$, $\mathcal{B}$ aborts if $i \neq i^\star$ or $\ell \neq \ell^\star$. With probability $1/LN$, such an undesirable event is avoided and, we have $g^\ell \cdot Y_i = A^{y_{i^\star}}$. Then, $\mathcal{B}$ draws $d^\star \stackrel{R}{\leftarrow} \{0,1\}$ and constructs the challenge $\mathbf{C}^\star$ as

$$C_0^\star = svk^\star \qquad C_1^\star = B \qquad C_2^\star = B^{y_{i^\star}} \qquad C_3^\star = m_{d^\star} \cdot T \qquad C_4^\star = B^{\alpha_2}$$

and $\sigma^\star = \mathcal{S}(ssk^\star, (C_3^\star, C_4^\star))$. As one can see, $\mathbf{C}^\star$ encrypts $m_{d^\star}$ under $pk_{i^\star}$ with the random exponent $s = b/a$ if $T = e(g,g)^{b/a}$ whereas $\mathcal{A}$'s view is independent of $d^\star$ if T is random. As usual, $\mathcal{B}$ outputs 1 (meaning that $T = e(g,g)^{b/a}$) if $\mathcal{A}$ successfully guesses $d^\star$ and returns 0 otherwise. $\square$

At level 1, the model does not change and the adversary can still query all re-encryption keys without restrictions.

Theorem 4. *Assuming the strong unforgeability of the one-time signature, the scheme is RCCA-CK-secure at level* 1 *under the* 1*-wDBDHI assumption.*

Proof. Let $\mathcal{A}$ be a RCCA adversary at level 1. We show an algorithm $\mathcal{B}$ that decides if $T = e(g,g)^{b/a}$ given $(g, A = g^a, B = g^b)$. Let $\mathbf{C}^\star = (\ell^\star, C_0^\star, C_1'^\star, C_1''^\star, C_1'''^\star, C_2'^\star, C_2''^\star, C_2'''^\star, C_3^\star, C_4^\star, \sigma^\star)$ be the challenge ciphertext. As above, we start by defining an event F_{OTS} which is the same as in the proof of theorem 3. Assuming the strong security of the one-time signature, this event comes about with negligible probability as detailed in the proof of prior theorems. We now describe our simulator $\mathcal{B}$ that simply halts and outputs a random bit if F_{OTS} ever happens.

As in the previous proof, the simulator $\mathcal{B}$ picks a one-time key pair $(ssk^\star, svk^\star) \leftarrow \mathcal{G}(\lambda)$ and sets up public parameters as $u = g^{\alpha_1}$ and $v = g^{-\alpha_1 svk^\star} \cdot A^{\alpha_2}$, with $\alpha_1, \alpha_2 \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, so that we have $F(svk) = u^{svk} \cdot v = g^{\alpha_1(svk - svk^\star)} \cdot A^{\alpha_2}$. The adversary's view is then simulated as follows.

- *Key generation*: $\mathcal{B}$ generates a public key $pk_0 = (X_0, Y_0) = (A, g^y)$, for a random $y \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$, so that $sk_0 = (\tilde{x}_0, \tilde{y}_0) = (a, y)$ is the implicitly defined secret.
- *Delegation queries:* at any time, $\mathcal{A}$ can output a public key (X_j, Y_j) of her choosing and a time period ℓ and request $\mathcal{B}$ to generate a re-encryption key $R_{0j\ell}$ on behalf of user 0 acting as a delegator. Since $X_0 = A$ and $Y_0 = g^y$ for a known value $y \in \mathbb{Z}_p^*$, $\mathcal{B}$ can proceed as in the proof of theorem 3 by drawing $r \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and returning

$$R_{0j\ell} = \left(g^{r(\ell+y)}, X_0^r \cdot X_j^{-1/(\ell+y)}\right). \qquad (16)$$

which has the correct distribution since, if we define $\tilde{r} = r - \frac{x_j}{a(\ell+y)}$, we have

$$\begin{aligned} X_j^{1/\tilde{x}_0} \cdot (g^\ell \cdot Y_0)^{\tilde{r}} &= X_j^{1/a} \cdot (g^{\ell+y})^{\tilde{r}} \\ &= X_j^{1/a} \cdot (g^{\ell+y})^r \cdot (g^{\ell+y})^{-\frac{x_j}{a(\ell+y)}} \\ &= g^{r(\ell+y)} \end{aligned}$$

and $X_0^{\tilde{r}} = X_0^r \cdot X_j^{-1/(\ell+y)}$. Both parts of (16) are computable (without knowing the discrete logarithm $x_j = \log_g(X_j)$) whenever $\ell + y \neq 0$. Since y is drawn at random, this is the case for any $\ell \in \{1, \ldots, L\}$ with probability at least $1 - L/p$.

- *First level decryption* queries: when faced with a decryption query for a first level ciphertext $\mathbf{C} = (\ell, C_0, C_1', C_1'', C_1''', C_2', C_2'', C_2''', C_3, C_4, \sigma)$, $\mathcal{B}$ returns '`invalid`' if relations (10)-(12) do not hold. If they do, we must have $e(C_1'', C_1''')/e(C_2'', C_2''') = e(X_0, g)^s$ where s is the unknown exponent such that $C_4 = (u^{C_0} \cdot v)^s$. Therefore, as in the proof of theorem 3, $\mathcal{B}$ can compute

$$e(g,g)^s = \left(\frac{e(C_4, g) \cdot e(C_2'', C_2''')^{\alpha_2}}{e(C_1'', C_1''')^{\alpha_2}} \right)^{\frac{1}{\alpha_1 (C_0 - svk^\star)}}$$

  (and the plaintext) as long as $C_0 \neq svk^\star$. If $C_0 = svk^\star$ in a post-challenge query,

  - If $C_4 = C_4^\star$, $\mathcal{B}$ returns $\perp$ to indicate that $\mathbf{C}$ is a re-randomization of the challenge ciphertext.
  - Otherwise, we necessarily have an occurrence of F_{OTS} and $\mathcal{B}$ terminates.

  To comply with replayable CCA security rules after the challenge phase, $\mathcal{B}$ must always check that $m \notin \{m_0, m_1\}$ before answering the query.

- *Challenge*: at the challenge step, $\mathcal{A}$ outputs messages (m_0, m_1) and a time period $\ell^\star$. $\mathcal{B}$ tosses a coin $d^\star \stackrel{R}{\leftarrow} \{0,1\}$. It chooses $t_1, \mu \stackrel{R}{\leftarrow} \mathbb{Z}_p^*$ and prepares the challenge $\mathbf{C}^\star$ as

$$\begin{array}{lll} C_1'^\star = A^{t_1} & C_1''^\star = (g^{\ell^\star+1} \cdot Y_0)^{1/t_1} & C_1'''^\star = B^{t_1} \\ C_2'^\star = A^{\mu(\ell^\star+y)} & C_2''^\star = g^{1/\mu} & C_2'''^\star = B^{\mu(\ell^\star+y)} \\ C_0^\star \ = svk^\star & C_3^\star \ = m_{d^\star} \cdot T & C_4^\star \ = B^{\alpha_2} \end{array}$$

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

Recall that $X_0 = A$, $Y_0 = g^y$ and $B = g^b$. Whenever $T = e(g,g)^{b/a}$, $\mathbf{C}^\star$ is a valid encryption of $m_{d^\star}$ with the encryption exponent $s = b/a$ and the blinding exponents t_1, $t_2 = a\mu$. When T is random, $\mathbf{C}^\star$ leaks no information on $m_{d^\star}$ or the bit $d^\star \in \{0,1\}$. Finally, $\mathcal{B}$ bets that $T = e(g,g)^{b/a}$ if $\mathcal{A}$ correctly guesses $d^\star$ and that T is random otherwise. □

4.2 A Non-Temporary PRE in the Chosen-Key Model

In settings where delegations should be permanent rather than temporary, one can simply instantiate the above scheme with a single time period. In this case, the scheme can be further simplified by defining the functions F_i as constants $F_i = Y_i$ for any user i.

4.3 PRE with Windowed Delegation

It may happen that temporary delegations should take place during several consecutive time periods whereas these periods should be short enough to give fine-time granularity. For instance, the delegator may want to set up his public key for one-day periods and grant specific decryption rights during several months. In such situations, the temporary PRE suggested in section 4.1 requires the generation of a new re-encryption key at each time period, and thus incurs re-encryption keys that have linear length in the duration of the delegation.

By appropriately modifying the scheme using ideas borrowed from forward-secure public key encryption [21, 14], re-encryption key sizes can be decreased from $O(\Delta L)$ to $O(\log^2(\Delta L))$, where ΔL denotes the length of the windowed delegation (*i.e.*, the number periods during which translation keys should be effective). Each user's public key now comprises $O(\log L)$ additional group elements $h_{i,0}, h_{i,1}, \dots h_{i,\theta} \in \mathbb{G}$, where $L = 2^\theta - 1$ is the total number of periods that the key is prepared for. The scheme described in section 4.1 bears salient resemblance with the selective-ID secure IBE scheme of Boneh-Boyen [13]: indeed, the function $F_i(\ell) = g^\ell \cdot Y_i$ applies the Boneh-Boyen identity hashing by seeing period numbers ℓ as identities. For such windowed delegations, the forementioned number theoretic hash function is more convenient to instantiate as $F_i(\ell) = h_{i,0} \cdot \prod_{k=1}^{\theta} h_{i,k}^{\ell_k}$, using the binary expansion $\ell = \ell_1 \dots \ell_\theta \in \{0,1\}^\theta$ of ℓ.

We imagine binary tree of height $\theta + 1$ where the root (at depth 0) has label ε. When a node at depth $\leq \theta$ has label w, its children are labeled with $w0$ and $w1$. The leaves of the tree correspond to time periods in the obvious way, periods being indexed from 0 to $L-1$ and stage ℓ being associated with the leaf labeled by $\ell_1 \dots \ell_\theta$. Let use assume that a delegator i holds a public key $pk_i = (X_i = g^{x_i}, h_{i,0}, h_{i,1}, \dots, h_{i,\theta})$ and wishes to delegate to user j, whose public key pk_j includes $X_j = g^{x_j}$, during periods $\{L_0 + 1, \dots, L_1\}$. First, to each tree node with label $w = w_1 \dots w_d$ at depth $d \leq \theta$, user i assigns the node key

$$R_{ij,w} = \big(X_j^{1/x_i} \cdot (h_{i,0} \cdot \prod_{k=1}^{d} h_{i,k}^{w_k})^r,\ X_i^r,\ h_{i,d+1}^r, \dots, h_{i,\theta}^r\big)$$

according the Boneh-Boyen-Goh HIBE system [14] (by seeing $(w_1, \dots, w_d) \in \{0,1\}^d$ as a vector of binary identities). As in [14], such a node key allows iteratively deriving similar keys for all w's descendants until reaching the leaves for which keys only consist of

$$R_{ij,\ell} = \big(X_j^{1/x_i} \cdot (h_{i,0} \cdot \prod_{k=1}^{\theta} h_{i,k}^{\ell_k})^r,\ X_i^r\big)$$

and actually suffice to re-encrypt ciphertexts during period $\ell = \ell_1 \dots \ell_\theta \in \{0,1\}^\theta$. To allow re-encryptions for a window $\{L_0 + 1, \dots, L_1\}$ of $\Delta L = L_1 - L_0$ time periods, the delegator only provides the proxy with the smallest set of node keys that contains an ancestor of each leaf falling in $\{L_0 + 1, \dots, L_1\}$ (and no ancestor of leaves outside this interval). Then, the proxy only has to store $O(\log^2(\Delta L))$ group elements instead of $O(\Delta L)$ using the method of section 4.1.

The price to pay is that a stronger assumption (*i.e.*, the θ-wDBDHI assumption where $\theta = O(\log L) > 1$) is needed to prove security results in a security model that naturally extends the one used in 4.1. Namely, at level 2 (the model obviously does not change at level 1), security is captured by a game where the attacked period $\ell^\star$ must be outside the union of all time-windows for which the adversary has requested delegations from the target user. We omit to give detailed security proofs here but it is not hard to convince oneself that security in this game more or less trivially follows from the selective-ID security of the underlying HIBE [14].

4.4 Introducing Warrants and Keywords in Proxy Re-Encryption

It may be desirable for delegators to only permit the re-encryption of ciphertexts that are tagged with specific keywords. For example, a traveling businessman may want the proxy to only redirect incoming encrypted emails to his secretary when the tagged keyword is "urgent". Rather than keywords, second level ciphertexts can be tagged with a warrant that specifies conditions under which re-encryption should be permitted. A natural way to impose such restrictions is to introduce these warrants or keywords in re-encryption keys in such a way that proxies will be limited to only translate a particular class of ciphertexts.

The above scheme is actually amenable to provide warrant-based and keyword-based delegations. It suffices to replace the Boneh-Boyen [13] identity hashing $F_i(\ell) = g^\ell \cdot Y_i$ with Waters' adaptive-ID secure identity hashing [45] $F_i : \{0,1\}^n \rightarrow \mathbb{G}$ that, on input of n-bit strings

$W = w_1 \ldots, w_n \in \{0,1\}^n$, calculates $F_i(W) = U_{i,0} \cdot \prod_{k=1}^{n} U_{i,k}^{w_k}$ using a random $(n+1)$-vector $\overline{U}_i = (U_{i,0}, U_{i,1}, \ldots, U_{i,n}) \in \mathbb{G}^{n+1}$ that supersedes Y_i in user i's public key. To generate a re-encryption key using a delegatee's public key $pk_j = (X_j, \overline{U}_j)$ and the warrant W, the delegator i computes

$$R_{ij,W} = (A_{ij,W}, B_{ij,W}) = (X_j^{1/x_i} \cdot F_i(W)^r, X_i^r).$$

Such a key only allows translating second level ciphertexts that are calculated as per

$$\mathbf{C} = (W, C_0, C_1, C_2, C_3, C_4, \sigma) = \Big(W, svk,\ X_i^s,\ F_i(W)^s,\ e(g,g)^s \cdot m,\ (u^{svk} \cdot v)^s,\ \sigma\Big), \quad (17)$$

where $\sigma = \mathcal{S}(ssk, (W, C_3, C_4))$. The security proofs (in a model that naturally generalizes the one of the scheme with temporary delegation) rely on the same assumption but with a looser reduction due to the use of Waters' technique.

Identity-based techniques and proxy re-encryption can be mixed in several settings. Other extensions are indeed possible in a natural analogue of the selective-ID security model [13] for IBE schemes (*i.e.*, a model defined by selective-keyword games where the adversary should choose the target keyword upfront and before seeing any public key). By borrowing ideas from the identity-based broadcast encryption with constant-size ciphertexts (derived from the Boneh-Boyen-Goh [14] hierarchical IBE) suggested in [1], we can construct a keyword-based PRE where ciphertexts are tagged with multiple keywords. Re-encryption is then permitted as long as the proxy has a translation key corresponding to at least one of them. In this case, ciphertexts retain constant (*i.e.*, independent of the number of tagging keywords) size at the expense of private keys that have quadratic size in the maximal number of keywords that a ciphertext can be associated with. Using ideas from Sahai-Waters [41], one can also imagine to design keyword-based PRE systems with error-tolerance: the proxy is allowed to re-encrypt ciphertexts if it holds a translation key for a keyword being sufficiently close (according to some metric) to that of the ciphertext. More generally, if ciphertexts are tagged with a set of descriptive attributes, attribute-based encryption techniques [30] can even be used to enable re-encryption when ciphertext attributes fit the access structure of the re-encryption key.

5 Conclusions and Open Problems

We presented the first unidirectional PRE realizations with chosen-ciphertext security in the standard model (*i.e.*, without using the random oracle heuristic). We also refined our security definitions by allowing adversaries to introduce arbitrary delegatees' public keys in the system. To the best of our knowledge, these are the first security results in the so-called chosen key model for the proxy re-encryption primitive. One of the new schemes additionally allows for temporary delegations and other extensions.

Many open problems still remain. One of them would be to devise secure schemes in a fully adaptive corruption model. The very existence of collusion-resistant multi-hop unidirectional systems dwells a (perhaps even more) challenging open question. Canetti and Hohenberger [19] also mentioned the problem of securely obfuscating CCA-secure re-encryption. Ateniese, Benson and Hohenberger [6] raised the one of key-private PRE in the chosen-ciphertext setting. It would also be interesting to efficiently implement such primitives outside bilinear groups (recent results [16] in the context of identity-based encryption may be encouraging in these regards).

References

1. M. Abdalla, E. Kiltz, G. Neven. Generalized Key Delegation for Hierarchical Identity-Based Encryption. In *ESORICS'07*, *LNCS* 4734, pp. 139–154. Springer, 2007.
2. J.-H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Eurocrypt'02*, *LNCS* 2332, pages 83–107. Springer, 2002.
3. G. Ateniese, K. Fu, M. Green, S. Hohenberger. Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage. In *NDSS*, 2005.
4. G. Ateniese, K. Fu, M. Green, S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In *ACM TISSEC* , 9(1): pp. 1–30, 2006.
5. G. Ateniese, S. Hohenberger. Proxy re-signatures: new definitions, algorithms, and applications. In *ACM Conference on Computer and Communications Security*, pages 310–319, ACM Press, 2005
6. G. Ateniese, K. Benson, S. Hohenberger. Key-Private Proxy Re-Encryption. Cryptology ePrint Archive: Report 2008/463, 2008.
7. M. Bellare, O. Goldreich. On defining proofs of knowledge. In *Crypto'92*, pp. 390–420, 1993.
8. M. Bellare, T. Kohno, V. Shoup. Stateful public-key cryptosystems: how to encrypt with one 160-bit exponentiation. In *ACM CCS'06* pp. 380–389, ACM Press, 2006.
9. M. Bellare, P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS'93*, pp. 62–73, ACM Press, 1993.
10. M. Bellare, G. Neven. Multi-signatures in the plain public-Key model and a general forking lemma. In *ACM CCS*, pp. 390–399, 2006.
11. M. Blaze, G. Bleumer, M. Strauss. Divertible Protocols and Atomic Proxy Cryptography. In *Eurocrypt'98*, *LNCS* 1403, pages 127–144, 1998.
12. A. Boldyreva. Efficient Threshold Signature, Multisignature and Blind Signature Schemes Based on the Gap-Diffie-Hellman-group Signature Scheme. In *PKC*, pp. 31–46, 2003.
13. D. Boneh, X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. In *Eurocrypt'04*, *LNCS* 3027, pp. 223–238. Springer, 2004.
14. D. Boneh, X. Boyen, E.-J. Goh. Hierarchical Identity Based Encryption with Constant Size Ciphertext. In *Eurocrypt'05*, *LNCS* 3494, pp. 440–456, 2005. Extended version available from Cryptology ePrint Archive: Report 2005/015.
15. D. Boneh, M. Franklin. Identity-based encryption from the Weil pairing. In *Crypto'01*, *LNCS* 2139, pp. 213–229. Springer, 2001.
16. D. Boneh, C. Gentry and M. Hamburg Space-Efficient Identity Based Encryption Without Pairings. In *FOCS'07*, to appear.
17. X. Boyen, Q. Mei, B. Waters. Direct Chosen Ciphertext Security from Identity-Based Techniques. In *ACM CCS'05*, ACM Press, pages 320–329, 2005.
18. X. Boyen, B. Waters. Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In *Crypto'06*, *LNCS* 4117, pages 290–307. Springer, 2006.
19. R. Canetti, S. Hohenberger. Chosen-Ciphertext Secure Proxy Re-Encryption. In *ACM CCS'07*. pages 185–194. ACM Press, 2007.
20. R. Canetti, H. Krawczyk, J. B. Nielsen. Relaxing Chosen-Ciphertext Security. In *Crypto'03*, *LNCS* 2729, pages 565–582. Springer, 2003.
21. R. Canetti, S. Halevi, J. Katz. A forward secure public key encryption scheme. In *Eurocrypt'03*, volume 2656 of *LNCS*, pages 254–271. Springer, 2003.
22. R. Canetti, S. Halevi, J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *Eurocrypt'04*, *LNCS* 3027, pp. 207–222, Springer, 2004.
23. C.-K. Chu, W. G. Tzeng. Identity-Based Proxy Re-encryption Without Random Oracles. In *ISC'07*, *LNCS* 4779, pp. 189–202, Springer, 2007.
24. Y. Dodis, A.-A. Ivan. Proxy Cryptography Revisited. In *NDSS'03*, 2003.
25. Y. Dodis, A. Yampolskiy. A Verifiable Random Function with Short Proofs and Keys. In *PKC'05*, *LNCS* 3386, pages 416–431, Springer, 2005.
26. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Crypto'84*, *LNCS* 196, pages 10–18. Springer, 1985.
27. M. Fischlin. Communication-Efficient Non-interactive Proofs of Knowledge with Online Extractors. In *Crypto'05*, *LNCS* 3621, pp. 152–168, 2005.
28. R. Granger, N. P. Smart. On Computing Products of Pairings. Cryptology ePrint Archive: Report 2006/172, 2006.
29. M. Green, G. Ateniese. Identity-Based Proxy Re-encryption. In *ACNS'07*, *LNCS* 4521, pages 288–306. Springer, 2007.
30. V. Goyal, O. Pandey, A. Sahai, B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS'06*, pp. 89–98, 2006.

31. S. Hohenberger. Advances in Signatures, Encryption, and E-Cash from Bilinear Groups. Ph.D. Thesis, MIT, May 2006.
32. S. Hohenberger, G. N. Rothblum, a. shelat, V. Vaikuntanathan. Securely Obfuscating Re-encryption. In *TCC'07*, *LNCS* 4392, pages 233–252. Springer, 2007.
33. M. Jakobsson. On Quorum Controlled Asymmetric Proxy Re-encryption. In *PKC'99*, *LNCS* 1560, pages 112–121. Springer, 1999.
34. E. Kiltz. Chosen-Ciphertext Security from Tag-Based Encryption. In *TCC'06*, *LNCS* 3876, pp. 581–600. Springer, 2006.
35. E. Kiltz. On the Limitations of the Spread of an IBE-to-PKE Transformation. In *PKC'06*, *LNCS* 3958, pp. 274–289, Springer, 2006.
36. E. Kiltz, D. Galindo. Direct Chosen-Ciphertext Secure Identity-Based Key Encapsulation without Random Oracles. In *ACISP'06*, *LNCS* 4058, pp. 336–347 Springer, 2006.
37. B. Libert, D. Vergnaud. Unidirectional Chosen-Ciphertext Secure Proxy Re-Encryption. In *PKC'08*, *LNCS* 4939. Springer, 2008.
38. M. Mambo, E. Okamoto. Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts. In *IEICE* Trans. Fund. Elect. Communications and CS, E80-A/1, pages 54–63, 1997.
39. M. Naor. On Cryptographic Assumptions and Challenges. In *Crypto'03*, *LNCS* 2729, pp. 96–109. Springer, 2003.
40. C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Crypto'91*, *LNCS* 576, pages 433–444. Springer, 1991.
41. A. Sahai, B. Waters. Fuzzy Identity-Based Encryption In *Eurocrypt'05*, *LNCS* 3494, pp. 457–473. Springer, 2005.
42. A. Shamir. Identity based cryptosystems and signature schemes. In *Crypto'84*, *LNCS* 196, pages 47–53. Springer, 1984.
43. V. Shoup. A proposal for the ISO standard for public-key encryption (version 2.1). manuscript, 2001. `http://shoup.net/`.
44. G. Taban, A. Cárdenas, V. D. Gligor. Towards a secure and interoperable DRM architecture. In *DRM'06*, ACM workshop on Digital rights management, pp. 69–78. ACM, 2006.
45. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *Eurocrypt'05*, *LNCS* 3494, pages 114–127. Springer 2005.