Improved Security Notions for Proxy Re-Encryption to Enforce Access Control

Ela Lee [†]
Ela.Lee.2010@live.rhul.ac.uk

Royal Holloway, University of London

Abstract Proxy Re-Encryption (PRE) allows a ciphertext encrypted under Alice's public key to be transformed to an encryption under Bob's public key without revealing either the plaintext or the decryption keys. PRE schemes have clear applications to cryptographic access control by allowing outsourced data to be selectively shared to users via re-encryption to appropriate keys. One concern for this application is that the server should not be able to perform unauthorised re-encryptions. We argue that current security notions do not adequately address this concern. We revisit existing definitions for PRE, starting by challenging the concept of unidirectionality, which states that re-encryption tokens from A to B cannot be used to re-encrypt from B to A. We strengthen this definition to reflect realistic scenarios in which adversaries may try to reverse a re-encryption by retaining information about prior ciphertexts and update tokens. We then strengthen the adversarial model to consider malicious adversaries that may collude with corrupt users and attempt to perform unauthorised re-encryptions; this models a malicious cloud service provider aiming to subvert the re-encryption process to leak sensitive data. Finally, we revisit the notion of authenticated encryption for PRE. This currently assumes the same party who created the message also encrypted it, which is not necessarily the case in re-encryption. We thus introduce the notion of *ciphertext origin authentication* to determine which party encrypted the message (or initiated the most recent re-encryption) and show how to fufil this requirement in practice.

Keywords: Proxy re-encryption, applied cryptography, unidirectional, multi-hop, malicious model, access control

1 Introduction

There are many practical situations in which a ciphertext encrypted under one key must be re-encrypted such that it represents an encryption of the same message under a different key. This is trivial when data is stored locally, but is less straightforward when data is stored remotely by an untrusted server such as a cloud service provider. Proxy Re-Encryption (PRE) [4] enables a third party to re-encrypt a ciphertext using an *update token* generated by the client, in such a way that neither the decryption keys nor plaintext are revealed.

A common motivation cited for PRE in the literature is *email forwarding* [2–4, 6, 7, 21], where Alice wants to forward her emails to Bob and have him read them on her behalf whilst she is away, without revealing her secret key. With PRE, she can generate an update token which the email server uses to re-encrypt emails encrypted under Alice's key to become ciphertexts under Bob's key, without the server reading her emails.

[†]Née Berners-Lee. The author was supported by the EPSRC and the UK government as part of the Centre for Doctoral Training in Cyber Security at Royal Holloway, University of London (EP/K035584/1)

Another motivation, which we focus on in this paper, is enforcing cryptographic access control over remotely stored files [3, 20]. If data is given a classification level, and keys are shared with users according to an access control policy, then re-encryption is used to enforce changes to the policy. In particular, re-encryption can signify a change in user access rights, particularly revocation or key expiry. In this work, we revisit the security notions for PRE with a particular focus on enforcing access control as an application. We show that, in many cases, existing notions are insufficient to ensure the necessary security in this setting.

The main issue not addressed by existing literature is that a malicious server should not be able to perform an unauthorised re-encryption. We break this down into two main security notions: the inability to create a valid update token even having seen a number of valid tokens, and a stronger notion of unidirectionality which considers reversal attacks. The summary of our contributions is as follows:

Malicious adversaries. Most previous work considers *honest-but-curious* adversaries that follow the protocol honestly but try to learn the underlying plaintexts. For stronger security in the access control setting, we must also consider malicious adversaries who can deviate from the protocol to try to perform unauthorised re-encryptions by colluding with corrupted users (thereby leaking confidential data). We break this down into two main security notions: the inability to create a valid update token even having seen a number of valid tokens, and a stronger notion of unidirectionality which considers reversal attacks.

Token robustness. Existing work [10] tackles the issue of controlling which ciphertexts are re-encrypted by defining *ciphertext dependence*, where tokens are created to only be valid for specific ciphertexts. We strengthen this definition to create a security notion which states that an adversary cannot generate a valid update token which re-encrypts to a key the ciphertext has never previously been encrypted under, even having seen a number of legitimate tokens. We then give an example which shows that this notion is stronger than ciphertext dependence. We call this *token robustness*.

Unidirectionality. To tackle re-encryptions to keys which a ciphertext has previously been encrypted under, we revisit the existing notion of unidirectionality, which states that a re-encryption token can only be used to transform a ciphertext under pk_i to pk_j and not from pk_j to pk_i (otherwise the scheme is *bidirectional*). The ability to re-encrypt back to the old key can grant access back to an unauthorised user or re-encrypt to an expired key. Current notions do not consider reversal attacks where a server may retain some limited information about an old ciphertext and update token to reverse the re-encryption. This consideration is particularly important for token robust and ciphertext dependent schemes where the token used to perform the update is crucial in reverting a ciphertext back to an expired key. We formally define reversal attacks with respect to the size of the state the server must retain in order to reverse a re-encryption. We use this to form an upper bound on security definitions for directionality. This is stronger than existing notions for unidirectionality as it gives the adversary more control over the information they have access to than traditional notions, which only consider tokens given to the adversary. We then use this together with token robustness to define *best-achievable unidirectionality*. Overall, our security model covers a wider range of attacks than prior definitions. We show that these definitions can be met by introducing a simple adaptation of ElGamal-based PRE in Section 6.1.

Ciphertext Origin Authentication. Finally, we revisit the notion of data origin authentication for PRE. Typically, data origin authentication assumes that the same party who created the message also encrypted it, hence tying the data owner's identity to the message within the ciphertext is suffient. Whilst this is a valid assumption for many encryption scenarios, for access control where more than one party shares a key, the same assumption cannot be made. We create a new notion of *Ciphertext Origin Authentication (COA)* where the identity of the initiator of the most recent re-encryption (or the encryptor for fresh ciphertexts) is tied to the ciphertext as opposed to the message, and re-encryption updates this accordingly. We offer an extension to our unidirectional token robust scheme, and show how to develop similar extensions for other schemes.

The structure of this paper is as follows: In Section 2 and Section 3 we formally discuss existing work and current notions of security for PRE. We define *token robustness* in Section 4. In Section 5 we give a formal definition for the existing intuition of unidirecionality, and present *maximal irreversibility* – the first security definition that considers reversibility. In Section 6 we use unidirectionality together with token robustness to create a stronger definition for *best-achievable unidirectionality*, and give a scheme which meets this definition. Finally, in Section 7 we define COA to provide authenticated PRE and discuss how to achieve this, giving an extension to our scheme.

2 Preliminaries

We begin by defining PRE in the public-key setting. Note that we will always assume that encryption is a probabilistic algorithm (so the encryption scheme is Indistinguishable against Chosen Plaintext Attacks (IND-CPA)).

Definition 1 (PRE scheme). *A* (public-key, multi-hop) Proxy Re-Encryption (PRE) scheme *consists of the following algorithms:*

- $\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{params}$*: Takes the security parameter* 1^λ *and outputs the set of parameters* params*, including a description of the message space* $\mathcal{M}$ *and token space* $\mathcal{T}$*. We note that* params *is an input for all subsequent algorithms but we omit it for compactness.*
- $\mathsf{KeyGen}(1^\lambda) \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$*: Generates a public-private key pair.*
- $\mathsf{Enc}(\mathsf{pk}, m) \xrightarrow{\$} C$*: Given a public key* pk *and message* $m \in \mathcal{M}$*, returns the ciphertext* C*.*
- $\mathsf{Dec}(\mathsf{sk}, C) \rightarrow m'$ *or* $\perp$*: Given a secret key* sk *and a ciphertext* C*, returns either a message* m' *or the error symbol* $\perp$*.*
- $\mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j) \rightarrow \Delta_{i,j}$*: Outputs an update token* $\Delta_{i,j}$ *from* pk_i *to* pk_j*.*
- $\mathsf{ReEnc}(\Delta_{i,j}, C) \rightarrow C'$*: Takes a ciphertext* C *and a token* $\Delta_{i,j}$ *and translates the ciphertext to output a new ciphertext* C'*.*

A PRE scheme is correct *if for all* $m \in \mathcal{M}$*,*

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m,$$

where $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda)$*, and if*

$$\Pr[\mathsf{Dec}(\mathsf{sk}_j, \mathsf{ReEnc}(\Delta_{i,j}, C_i)) = \mathsf{Dec}(\mathsf{sk}_i, C_i)] \geq 1 - \mathsf{negl}(\lambda),$$

for some negligible function $\mathsf{negl}(\lambda)$*, where* $(\mathsf{pk}_i, \mathsf{sk}_i), (\mathsf{pk}_j, \mathsf{sk}_j) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda)$ *and* $\Delta_{i,j} \leftarrow \mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j)$*.*

In other words, in a correct PRE scheme all ciphertexts decrypt correctly including ciphertexts which are re-encryptions of other correct ciphertexts. Definition 1 can be intuitively adapted to the symmetric key setting, where token generation requires knowledge of both secret keys.

For PRE to be practically useful, it must be cheaper either computationally or in terms of the bandwidth used. Amazon Web Services charges users for downloads but not uploads, giving some justification for schemes which upload data to cloud storage but require no download [10]. Therefore, the expense of the scheme could be measured either with respect to size of messages exchanged or with respect to price.

2.1 Mathematical preliminaries and notation

We now give some mathematical preliminaries and notation. For a group $\mathbb{G}$, sampling an element x uniformly at random from that group is denoted $x \xleftarrow{\$} \mathbb{G}$. We sometimes abuse this notation by writing $y \xleftarrow{\$} \mathsf{Alg}(x)$ to convey that the algorithm $\mathsf{Alg}(x)$ generates some randomness in creating output y, such as in Definition 1. We denote by $\mathbb{Z}_n$ the set of integers modulo n, and by $\mathbb{Z}_n^*$ the set $\{1, 2, \dots, n-1\}$.

The Decisional Diffie-Hellman (DDH) problem: Let p be a prime and let g be a generator of the group $\mathbb{Z}_p$. Given a tuple g, g^a, g^b, g^c where $a, b \xleftarrow{\$} \mathbb{Z}_p^*$ and determine whether $c \xleftarrow{\$} \mathbb{Z}_p^*$ or $c = ab$. The *DDH assumption* states that there is no Probabilistic Polynomial-Time (PPT) algorithm that can solve the DDH problem.

Diffie-Hellman assumptions underpin the security of ElGamal-based PRE schemes, which make up the majority of the literature, particularly the original PRE scheme [4] as well as many subsequent schemes such as [2, 7, 18].

2.2 Additional PRE properties

There are some additional properties which a PRE scheme can have. Whether or not these properties are required depends on the security model and the application. We define some of these properties below.

Directionality. Informally, a PRE scheme is *bidirectional* if a re-encryption token from pk_i to pk_j can be used to re-encrypt from pk_j to pk_i. Otherwise, it is *unidirectional.* We take this description from existing work [3, 6, 10, 16] (see Section 2.3). To date unidirectionality is not given as a separate definition, instead when the PRE scheme is defined, it is implicitly defined as either unidirectional or bidirectional. We observe that it lacks the necessary formalism to be considered a security definition as needed for access control and key expiry. This motivates our extracting the current notion of unidirectionality from the literature and formulating it as a formal definition Section 5.1, before giving a stronger definition in Section 5.2. We note that there are some applications where bidirectionality may be considered a desirable feature.

Single/Multi-hop. Some PRE schemes are *single-hop* meaning ciphertexts can only be re-encrypted once. Typically in these cases, ReEnc changes the format of the ciphertext and there are two decryption algorithms $\mathsf{Dec}_1, \mathsf{Dec}_2$, one for ciphertexts which can be re-encrypted and the other for ciphertexts which cannot. In contrast, a *multi-hop* scheme allows ciphertexts to be re-encrypted multiple times. Single-hop schemes only have limited use and are mainly considered for unidirectionality purposes. Since we focus on the practical application of access control as a motivation, we will consider multi-hop a necessary requirement of a PRE scheme in the remainder of this work.

Ciphertext dependence. Informally, in a ciphertext-dependent PRE scheme $\mathsf{ReKeyGen}$ takes some additional information about the ciphertext to be re-encrypted as input. Tokens can only correctly re-encrypt the intended ciphertext. Let $\mathsf{ReKeyGen}$ in a PRE scheme be redefined to take additional information $\tilde{C}$ about ciphertext C as input: $\mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j, \tilde{C}) \rightarrow \Delta_{i,j}^{C}$. The PRE scheme is *ciphertext dependent* if for all $C_1 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_i, m_1)$ and $C_2 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_i, m_2)$ such that $C_1 \neq C_2$, and all re-encryption tokens $\Delta_{i,j,}^{C_1} \xleftarrow{\$} \mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j, \tilde{C}_1)$, then:

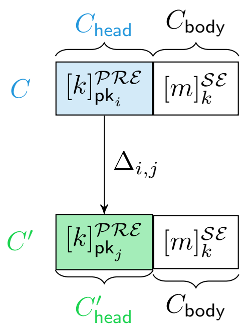
$$\Pr\left[\mathsf{Dec}(\mathsf{sk}_j, \mathsf{ReEnc}(\Delta_{i,j}^{C_1}, C_2)) = m_2\right] \leq \mathsf{negl}(\lambda) \qquad (1)$$

for some negligible function $\mathsf{negl}(\lambda)$.

In existing work [10] and in our scheme, $\tilde{C}$ denotes the header of the ciphertext. For simplicity we will assume this is the case for ciphertext-dependent schemes for the remainder of this paper. We note that not all applications require ciphertext dependence. For example in key expiry, all ciphertexts under the old key need to be re-encrypted. We give variants of security definitions for both ciphertext independent and ciphertext dependent schemes.

2.3 Existing Work

PRE was first introduced in [4]. The common approach to PRE is the key encapsulation approach, where the ciphertext header contains the data encryption key k, encrypted with a key encryption key pk. Typically, such schemes perform re-encryption by replacing pk — so the ciphertext header now contains the same k encrypted with the new key encryption key pk' and the body of the ciphertext remains unchanged. The effect of a re-encryption in the key encapsulation approach can be visualised as:



where $[a]^{\mathcal{PRE}}_{\mathsf{pk}}$ represents a PRE ciphertext which produces a when decrypted using the appropriate sk, and $[m]^{\mathcal{SE}}_{k}$ is a symmetric encryption of the message m using k. The full description is given in Figure 1.

```
$\mathsf{KEM.KeyGen}(1^\lambda)$ : $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathcal{PRE}.\mathsf{KeyGen}(1^\lambda)$
$\mathsf{KEM.Enc}(\mathsf{pk}, m)$ : $k \xleftarrow{\$} \mathcal{SE}.\mathsf{KeyGen}(1^\lambda)$,
        $C_{\mathcal{PRE}} \xleftarrow{\$} \mathcal{PRE}.\mathsf{Enc}(\mathsf{pk}, k)$,
        $C_{\mathcal{SE}} \xleftarrow{\$} \mathcal{SE}.\mathsf{Enc}(k, m)$.
        $C \leftarrow (C_{\mathcal{PRE}}, C_{\mathcal{SE}})$
$\mathsf{KEM.Dec}(\mathsf{sk}, C)$ : $(C_{\mathcal{PRE}}, C_{\mathcal{SE}}) \leftarrow C$
        $k' \leftarrow \mathcal{PRE}.\mathsf{Dec}(\mathsf{sk}, C_{\mathcal{PRE}})$,
        $m' \leftarrow \mathcal{SE}.\mathsf{Dec}(k', C_{\mathcal{SE}})$
$\mathsf{KEM.ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j)$ : $\Delta_{i,j} \leftarrow \mathcal{PRE}.\mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j)$
$\mathsf{KEM.ReEnc}(\Delta_{i,j}, C_i)$ : $(C_{\mathcal{PRE}}, C_{\mathcal{SE}}) \leftarrow C_i$,
        $C'_{\mathcal{PRE}} \leftarrow \mathcal{PRE}.\mathsf{ReEnc}(\Delta_{i,j}, C_{\mathcal{PRE}})$,
        $C_j \leftarrow (C'_{\mathcal{PRE}}, C_{\mathcal{SE}})$
```

Figure 1: A full description of the key encapsulation approach to PRE.

The appeal of this approach is that it is efficient and can use hybrid encryption. It is used widely, for example in Amazon's Key Management Service [1]. Whilst these schemes are simple and easy to implement, they do not completely re-randomise a ciphertext during re-encryption. A particular concern with the key encapsulation approach is that a malicious revoked user can simply retain k and be able to decrypt the message, regardless of how many times it has been re-encrypted.

Indistinguishability notions for PRE are described in [7] and [10]. A similar definition for CCA-security is used in [3,6,18]. Existing notions which imply complete re-randomisation for public-key PRE include *unlinkability* in [6], and for symmetric key PRE include *ciphertext independence*[1] in [5] and UP-REENC security in [10].

Ciphertext dependence was first introduced in [10] for symmetric PRE, but has not yet been picked up by subsequent work. However their work does not explicitly consider unauthorised re-encryptions or unidirectionality.

One attempt to formalise the definition of directionality is given by Ivan and Dodis in [14], but the authors do not view directionality as a security definition, rather a classification of PRE schemes. They therefore give one definition for a unidirectional PRE scheme another for a bidirectional PRE scheme as opposed to defining directionality separately from PRE. Furthermore the definition of unidirectionality in [14] assumes that a unidirectional scheme is single-hop, which is not necessarily the case. A more recent work which informally describes unidirectionality is based on the idea that no PPT algorithm can out output a token that can re-encrypt to the old key [10], whereas other works such as [20] are based on the idea that no PPT algorithm can output an equivalent encryption of the old ciphertext.

For a long time it was an open problem to create a scheme which is both unidirectional and multi-hop. Unidirectionality was achieved in the past by restricting to single-hop, and as such there exist a number of PRE schemes which are unidirectional and single-hop [7, 18, 21]. These schemes achieve unidirectionality by having two distinct levels of ciphertext which have different formats — level 2 for ciphertexts which can be re-encrypted, and level 1 for ciphertexts which cannot be re-encrypted. It is this format change which prevents a ciphertext from being re-encrypted more than once. This approach is undesirable, since it does not allow for multiple re-encryptions meaning it is unsuitable for all applications. It also does not convey how easy it is for a malicious server to reverse the re-encryption process. We discuss this in Section 5. Reversal attacks where the adversary is permitted to retain information on the original ciphertext prior to re-encryption are considered in [19], for a variant of PRE where both keypairs are needed to create an update token.

In many multi-hop schemes the number of re-encryptions is fixed, and the size of the ciphertext grows linearly with each re-encryption [6, 16]. The related problem of multi-hop unidirectional proxy re-signatures is addressed in [17], however the message must be provided with the signature and thus this technique cannot be easily adapted to re-encryption. The first PRE scheme which is both unidirectional and multi-hop was given in [20], but does not address ciphertext dependence and current methods for achieving this cannot be applied to their scheme.

There does not appear to be existing work in either symmetric or public-key PRE that considers a malicious server which may perform unauthorised re-encryptions.

3 Indistinguishability

The most common notion of indistinguishability for PRE is that a re-encryption of a ciphertext should *preserve* the indistinguishability given by the underlying encryption scheme, which we

[1] We reserve this terminology for PRE schemes for which token generation is not specific to a given ciphertext as in Section 2.2.

```
$\mathsf{preIND\text{-}CPA}^{\mathcal{A}}_{\mathcal{PRE}}(\lambda, \kappa)$
    $\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda)$
    $b \xleftarrow{\$} \{0,1\}$
    $(\mathsf{pk}_1, \mathsf{sk}_1), \ldots (\mathsf{pk}_\kappa, \mathsf{sk}_\kappa) \leftarrow \mathsf{KeyGen}(1^\lambda)$
    $\mathcal{K}_{\mathsf{corrupt}} = \{\mathsf{sk}_1, \ldots, \mathsf{sk}_t\}$
    $b' \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{ReKeyGen}}, \mathsf{O}_{\mathsf{ReEnc}}, \mathsf{O}_{\mathsf{Challenge}}}(1^\lambda, \mathcal{K}_{\mathsf{corrupt}}, \mathsf{pk}_1, \ldots, \mathsf{pk}_\kappa)$
    return $b' = b$
```

```
$\mathsf{O}_{\mathsf{ReKeyGen}}(i, j \boxed{, C})$
    if $(\mathsf{sk}_i \notin \mathcal{K}_{\mathsf{corrupt}}) \wedge (\mathsf{sk}_j \in \mathcal{K}_{\mathsf{corrupt}})$ :
        return $\perp$
    if ($\mathcal{PRE}$ unidirectional) :
        if $(\mathsf{sk}_j \notin \mathcal{K}_{\mathsf{corrupt}}) \wedge (\mathsf{sk}_i \in \mathcal{K}_{\mathsf{corrupt}})$ :
            return $\perp$
    $\Delta^{\boxed{C}}_{i,j} \xleftarrow{\$} \mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j \boxed{, \tilde{C}})$
    return $\Delta^{\boxed{C}}_{i,j}$
```

```
$\mathsf{O}_{\mathsf{ReEnc}}(i, j, C)$
    if $(\mathsf{sk}_i \notin \mathcal{K}_{\mathsf{corrupt}}) \wedge (\mathsf{sk}_j \in \mathcal{K}_{\mathsf{corrupt}})$ :
        return $\perp$
    if ($\mathcal{PRE}$ unidirectional) :
        if $(\mathsf{sk}_j \notin \mathcal{K}_{\mathsf{corrupt}}) \wedge (\mathsf{sk}_i \in \mathcal{K}_{\mathsf{corrupt}})$ :
            return $\perp$
    $\Delta^{\boxed{C}}_{i,j} \xleftarrow{\$} \mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j \boxed{, \tilde{C}})$
    $C' \leftarrow \mathsf{ReEnc}(\Delta^{\boxed{C}}_{i,j}, C)$
    return $C'$
```

```
$\mathsf{O}_{\mathsf{Challenge}}(i, m_0, m_1)$
    if $(|m_0| \neq |m_1|) \vee (\mathsf{sk}_i \in \mathcal{K}_{\mathsf{corrupt}})$ :
        return $\perp$
    $C_0 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_i, m_0)$
    $C_1 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_i, m_1)$
    return $C_b$
```

Figure 2: The preIND-CPA game which reflects the most common notion of indistinguishability for PRE. $\boxed{\text{Boxed}}$ values show the variant for a ciphertext dependent scheme.

call preIND-CPA. We note that since [6], a main notion for security in PRE schemes is that the scheme should be Indistinguishable against Chosen Ciphertext Attacks (IND-CCA) as opposed to IND-CPA. However since the focus of this paper is to revisit definitions with respect to unauthorised re-encryptions, for simplicity we restrict security to IND-CPA and leave IND-CCA as an open problem. We do not consider this a significant weakening of security in comparison with existing practical schemes, as recent schemes which are both unidirectional and multi-hop such as [11, 20] also do not consider IND-CCA security.

The following definition is a formalism of the preservation of indistinguishability introduced in [6] adapted to IND-CPA security.

Definition 2. *A PRE scheme $\mathcal{PRE}$ is* Indistinguishable against Chosen Plaintext Attacks (preIND-CPA) *if for all* PPT *algorithms $\mathcal{A}$ there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that:*

$$\Pr\left[\mathsf{preIND\text{-}CPA}^{\mathcal{A}}_{\mathcal{PRE}}(\lambda, \kappa) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda),$$

where κ is polynomial with respect to λ and preIND-CPA *is given in Figure 2.*

Informally, the PRE scheme is still IND-CPA even when the adversary is given access to a re-encryption and token generation oracle. Clearly the underlying PKE scheme must be IND-CPA in order for the PRE scheme to be preIND-CPA.

```
$\mathsf{ReEnc\text{-}IND\text{-}CPA}^{\mathcal{A}}_{\mathcal{PRE}}(\lambda, \kappa)$
$\mathsf{params} \leftarrow \mathsf{Setup}(\lambda)$
$b \xleftarrow{\$} \{0,1\}$
$(\mathsf{pk}_1, \mathsf{sk}_1), \dots (\mathsf{pk}_\kappa, \mathsf{sk}_\kappa) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$\mathcal{K}_{\mathsf{corrupt}} = \{\mathsf{sk}_1, \dots, \mathsf{sk}_t\}$
$b' \leftarrow \mathcal{A}^{\mathsf{LR\text{-}ReEnc}, \mathsf{O}_{\mathsf{ReKeyGen}}}(1^\lambda, \mathcal{K}_{\mathsf{corrupt}}, \mathsf{pk}_1, \dots, \mathsf{pk}_\kappa)$
return $b' = b$

$\mathsf{LR\text{-}ReEnc}(i, j, C_0, C_1)$
if $|C_0| \neq |C_1|$ or $\mathsf{sk}_j \in \mathcal{K}_{\mathsf{corrupt}}$
  return $\perp$
$\Delta_{i,j}^{\boxed{C_b}} \xleftarrow{\$} \mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j \boxed{, \tilde{C_b}})$
$C' \leftarrow \mathsf{ReEnc}(\Delta_{i,j}^{\boxed{C_b}}, C_b)$
return $(C')$

$\mathsf{O}_{\mathsf{ReKeyGen}}(i, j \boxed{, C})$
if $\mathsf{sk}_j \in \mathcal{K}_{\mathsf{corrupt}}$ :
  return $\perp$
$\Delta_{i,j}^{\boxed{C}} \xleftarrow{\$} \mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j \boxed{, \tilde{C}})$
return $\Delta_{i,j}^{C}$
```

Figure 3: ReEnc-IND-CPA game. Schemes that meet this definition must fully re-randomise a ciphertext upon re-encryption. $\boxed{\text{Boxed}}$ values show the variant for a ciphertext dependent scheme.

Observe that the above definition applies whether or not the PRE scheme is ciphertext-dependent or unidirectional, where bidirectional schemes have an additional constraint that $\mathsf{O}_{\mathsf{ReKeyGen}}$ will not return tokens from corrupted to uncorrupted keys. We can easily extend Definition 2 to symmetric PRE by providing the adversary with encryption oracles for both keys, see [10].

The concept of preIND-CPA is the most common notion of indistinguishability in the re-encryption literature. This means it is often not considered a requirement that re-encryption fully re-randomises a ciphertext. However, full re-randomisation must be considered a necessary security property for applications such as access control (revocation) and key expiry. Definition 3 addresses this, based on UP-REENC in [10] adapted to the public-key setting. Figure 3 models a revoked user trying to distinguish a re-encrypted ciphertext from two potential original ciphertexts. In this game, the adversary needs to distinguish which of two possible ciphertexts is re-encrypted by the LR-ReEnc oracle. For stronger security, the adversary has access to a token generation oracle, $\mathsf{O}_{\mathsf{ReKeyGen}}$. The adversary is given t secret keys to model revocation scenarios where a user knows the old key and oracles have the restriction that they will not return tokens to a corrupted key.

Definition 3. *A PRE scheme $\mathcal{PRE}$ is* Re-Encryption Indistinguishable Against Chosen Plaintext Attacks (ReEnc-IND-CPA) *if for all* PPT *adversaries $\mathcal{A}$ there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that:*

$$\Pr\left[\mathsf{ReEnc\text{-}IND\text{-}CPA}^{\mathcal{A}}_{\mathcal{PRE}}(\lambda, \kappa) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda)\,, \tag{2}$$

where κ is polynomial with respect to λ and ReEnc-IND-CPA *is given in Figure 3.*

This definition can be easily extended to symmetric PRE by providing the adversary with encryption oracles for both keys, see [10].

Lemma 1. *Let $\mathcal{PRE}$ be a PRE scheme where both* ReKeyGen *and* ReEnc *are deterministic. Then $\mathcal{PRE}$ is not ReEnc-IND-CPA.*

```
$\mathsf{Tok\text{-}Rob}_{\mathcal{A}}^{\mathcal{PRE}}(\lambda, \kappa)$
    $\mathsf{params} \leftarrow \mathsf{Setup}(1^\lambda)$
    $(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_\kappa, \mathsf{sk}_\kappa) \leftarrow \mathsf{KeyGen}(1^\lambda)$
    $\mathcal{C}_{\mathsf{honest}}, \mathcal{T}_{\mathsf{honest}} := \emptyset$
    for all $m \in \mathcal{M}$ : $\mathsf{chain}[m] := \emptyset$
    $(C, i, j, \Delta^{\mathcal{A}}) \leftarrow \mathcal{A}^{\mathsf{O_{Enc}}, \mathsf{O_{ReKeyGen}}}(1^\lambda, \mathsf{pk}_1, \ldots, \mathsf{pk}_\kappa)$
    if $(i, C) \notin \mathcal{C}_{\mathsf{honest}}$ : return $\perp$
    $C' \leftarrow \mathsf{ReEnc}(\Delta^{\mathcal{A}}, C)$
    if $(\mathsf{Dec}(\mathsf{sk}_i, C) = \mathsf{Dec}(\mathsf{sk}_j, C')) \wedge (j \notin \mathsf{chain}[m])$ :
      return 1
    else : return 0

$\mathsf{O_{Enc}}(i, m)$
    $C \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_i, m)$
    $\mathsf{chain}[m].\mathsf{add}(i)$
    $\mathcal{C}_{\mathsf{honest}}.\mathsf{add}(i, C)$
    return $C$

$\mathsf{O_{ReKeyGen}}(i, j \boxed{, C})$
    $\Delta_{i,j}^{\boxed{C}} \leftarrow \mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j \boxed{, \tilde{C}})$
    $\mathsf{chain}[\mathsf{Dec}(\mathsf{sk}_i, C)].\mathsf{add}(j)$
    $\mathcal{T}_{\mathsf{honest}}.\mathsf{add}(i, j \boxed{, C}, \Delta_{i,j}^{\boxed{C}})$
    return $\Delta_{i,j}^{\boxed{C}}$

$\mathsf{O_{ReEnc}}(i, j, \Delta_{i,j}^{C}, C)$
    if $\Delta_{i,j}^{\boxed{C}} = \perp$: $\Delta_{i,j}^{\boxed{C}} \leftarrow \mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j \boxed{, \tilde{C}})$
    else if $(i, j, \Delta_{i,j}^{\boxed{C}}) \notin \mathcal{T}_{\mathsf{honest}}$ : return $\perp$
    $C' \leftarrow \mathsf{ReEnc}(\Delta_{i,j}^{\boxed{C}}, C)$
    if $(i, C) \in \mathcal{C}_{\mathsf{honest}}$
      $\mathcal{C}_{\mathsf{honest}}.\mathsf{add}(j, C')$
    return $C'$
```

Figure 4: The token robustness game Tok-Rob for assessing whether a ciphertext dependent PRE scheme allows an adversary to create unauthorised update tokens.

Proof. If ReKeyGen and ReEnc are deterministic, the adversary can win by calling $\mathsf{O_{ReKeyGen}}(i, j, C_0)$ to obtain Δ, compute $\mathsf{ReEnc}(\Delta, C_1)$ and compare this to $\mathsf{LR\text{-}ReEnc}(i, j, C_0, C_1)$.

4 Token Robustness

Recall that in a ciphertext dependent PRE scheme, ReKeyGen takes an additional input $\tilde{C}$ from the ciphertext to be re-encrypted, with the intention that the resulting update token can only re-encrypt that ciphertext. This section defines *token robustness* - a stronger notion than ciphertext dependence.

Informally, token robustness states that even with access to a token generation oracle, an adversary cannot create a new valid token which re-encrypts a ciphertext to a key it was never previously encrypted under. We cover re-encryption to keys a ciphertext was previously encrypted under when we discuss directionality in Section 5. This extends ciphertext dependence by explicitly modelling an adversary who sees a number of legitimate update tokens and attempts to perform an unauthorised re-encryption. This is designed to compliment unidirectionality, which examines re-encrypting to a key which the ciphertext has previously been under.

Definition 4. *We say that a PRE scheme $\mathcal{PRE}$ has* κ-token robustness *if for all* PPT *adversaries* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr\left[\mathsf{Tok\text{-}Rob}^{\mathcal{A}}_{\mathcal{PRE}}(\lambda, \kappa) = 1\right] \leq \mathsf{negl}(\lambda)\,, \tag{3}$$

where $\mathsf{Tok\text{-}Rob}$ *is given in Figure 4. If* κ *is polynomial with respect to* λ *then we say that* $\mathcal{PRE}$ *has* token robustness.

In the $\mathsf{Tok\text{-}Rob}$ game, the adversary attempts to output a token $\Delta^{\mathcal{A}}$ which re-encrypts a target ciphertext C from being under pk_i to being under pk_j, where the underlying message has never been encrypted under pk_j. It has access to an encryption oracle $\mathsf{O}_{\mathsf{Enc}}$, a token generation oracle $\mathsf{O}_{\mathsf{ReKeyGen}}$ and a re-encryption oracle $\mathsf{O}_{\mathsf{ReEnc}}$. The list $\mathsf{chain}[m]$ records keys m has been encrypted (or can be re-encrypted) under by appending the appropriate keys whenever $\mathsf{O}_{\mathsf{Enc}}$ or $\mathsf{O}_{\mathsf{ReKeyGen}}$ are called. The adversary cannot win by submitting a token that re-encrypts the key to another key in the chain. This game aims to model a malicious server trying to re-encrypt ciphertexts provided by the client, and therefore another condition is that the adversary's target ciphertext C must be an output of the $\mathsf{O}_{\mathsf{Enc}}$ oracle (or a re-encryption of such a ciphertext), and the challenger maintains a list $\mathcal{C}_{\mathsf{honest}}$ to keep track of these ciphertexts. This ensures the adversary gains no additional advantage from storing information created when encrypting the ciphertext. For example, in an ElGamal-based scheme (such as our scheme in Section 6.1), encryption selects a random y and sets $\tilde{C} = g^y$. If the adversary encrypts the message for themselves then they learn y, which the server would not know in the cloud storage application. The list $\mathcal{C}_{\mathsf{honest}}$ is used to keep track of oracle-generated ciphertexts.

Theorem 1. *No ciphertext independent PRE scheme has token robustness.*

Proof. If a PRE scheme is not ciphertext dependent, then the same update token can be used to re-encrypt more than one ciphertext. In the $\mathsf{Tok\text{-}Rob}$ game, let $C_1 \leftarrow \mathsf{O}_{\mathsf{Enc}}(i, m_1), C_2 \leftarrow \mathsf{O}_{\mathsf{Enc}}(i, m_2)$. Then the adversary can submit (i, j, C_1) to $\mathsf{O}_{\mathsf{ReKeyGen}}$ to obtain $\Delta^{\tilde{C_1}}_{i,j}$ and then submit $(C_2, i, j, \Delta_{i,j})$ (in other words, set $C^{\mathcal{A}} = C_2, \Delta^{\mathcal{A}} = \Delta_{i,j}$). Since $j \notin \mathsf{chain}[m_2]$, the adversary wins the game with probability 1. $\square$

Ciphertext dependence alone does not imply token robustness. For example, suppose that $\mathsf{ReKeyGen}(\mathsf{pk}_i, \mathsf{sk}_j, \tilde{C_1})$ outputs $\Delta^{\tilde{C_1}}_{i,j} = (\Delta_0 = \tilde{C_1}, \Delta_1)$, and that $\mathsf{ReEnc}(\Delta, C)$ incorporates ciphertext dependence by verifying that $\Delta_0 = \tilde{C}$, with re-encryption only proceeding if this is true. Then an adversary can trivially craft a valid token for a different ciphertext by setting $\Delta^{\mathcal{A}} = (\tilde{C_2}, \Delta_1)$. We therefore see that token robustness is a stronger notion than ciphertext dependence. Token robustness requires more than a check that the given token is meant for a specific ciphertext by comparing values – it requires some mathematical computation where a correct re-encryption is only possible using the appropriate update token.

5 Directionality revisited

Whilst token robustness covers the risk of ciphertexts being re-encrypted to keys they have never previously been under, the problem of re-encrypting to a key the ciphertext was previously encrypted under relates to *directionality.* Recall that the existing idea of unidirectionality states that an update token $\Delta_{i,j}$ cannot be used to re-encrypt a ciphertext under pk_j to pk_i. We argue that this notion is not sufficient for access control as it is lacks formality and is unclear in whether it covers the possibility that re-encryptions from pk_i to pk_j can be reversed. This informal notion also does not couple well with ciphertext dependence or token robustness. In this section we will elaborate on this claim before offering a more fine-grained security definition, which we call $\bar{\lambda}$*-irreversible.*

5.1 Problems with traditional directionality

Unidirectionality is required in a number of applications for security reasons. However, it has not yet been defined as a security property. This is problematic, as it is not immediately obvious whether different authors' informal descriptions of unidirectionality amount to the same thing. Formal security allows us a precise understanding of what kinds of re-encryptions are prevented. A particular issue is that, if a single-hop PRE scheme prevents $\Delta_{j,i}$ being learned from $\Delta_{i,j}$ (which is one possible interpretation of unidirectionality), this does not convey whether it is possible for $\Delta_{i,j}$ to reverse a re-encrypted ciphertext $C' \leftarrow \mathsf{ReEnc}(\Delta_{i,j}, C)$ back to C, which may be possible if ReEnc is deterministic.

We therefore formulate a security game UNI in Figure 6, which we believe formalises the informal understanding of unidirectionality in the literature. In UNI, the adversary receives two public keys $\mathsf{pk}_i, \mathsf{pk}_j$. Their aim is to take a challenge ciphertext C_j which is an encryption of an unknown message, and produce a ciphertext C_i such that $\mathsf{Dec}(\mathsf{sk}_i, C_i) = \mathsf{Dec}(\mathsf{sk}_j, C_j)$. The adversary receives access to a token oracle $\mathsf{O}_{\mathsf{ReKeyGen}}$ that returns tokens from pk_i to pk_j[2]. The adversary can choose the number of times N that a challenge ciphertext can have been re-encrypted (its level), in which case the challenger generates the intermediate keys, and returns the public keys to the adversary together with the ciphertext. This accommodates both single and multi-hop PRE schemes, where $N \in \{0, 1\}$ for single-hop schemes.

Definition 5. *A PRE scheme $\mathcal{PRE}$ is* unidirectional *if for all* PPT *algorithms $\mathcal{A}$ there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\Pr\left[\mathsf{UNI}^{\mathcal{A}}_{\mathcal{PRE}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda)\,. \tag{4}$$

A PRE scheme $\mathcal{PRE}$ is bidirectional *if there exists a* PPT *algorithm $\mathcal{A}$ that wins the unidirectionality game* UNI *with overwhelming probability.*

This can be easily extended to symmetric PRE schemes by making the obvious adjustments. We believe that this definition formally captures the intuition behind unidirectionality as it is currently understood in the literature. We hope that for applications where unidirectionality is a security requirement, this definition can be used to assess the suitability of potential PRE schemes.

5.2 Directionality reconsidered

Whilst reversal attacks are not of concern for email forwarding, the original motivation for PRE, it is an important consideration for access control and key expiry as reversing a re-encryption can mean regranting access to a revoked user. UNI does not cover reversal attacks, as challenge ciphertexts have never been encrypted under pk_i. A server or revoked user aiming to perform a reversal attack would at some point know the old ciphertext and potentially the information used to perform the re-encryption. As a malicious server aiming to perform an unauthorised re-encryption could attempt such an attack, reversal attacks warrant further consideration.

Before we proceed, we must address the obvious point. An assumption often made, particularly in the access control literature, is that once a message (or ciphertext) is known, it is known forever. However, this assumption does not always make sense in cloud storage applications when considering revoked users. One of the most common reasons for outsourcing storage is limited

[2]In a ciphertext dependent scheme, this oracle takes a ciphertext C as input, otherwise it takes no input. Tokens generated by the challenge oracle also take the appropriate ciphertexts as input.

```
$\mathsf{UNI}^{\mathcal{PRE}}_{\mathcal{A},\bar{\lambda}}(\lambda)$
--------------------------------
$\mathsf{params} \leftarrow \mathsf{Setup}(\lambda)$
$(\mathsf{pk}_i, \mathsf{sk}_i), (\mathsf{pk}_j, \mathsf{sk}_j) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda)$
$\mathcal{C}_{\mathsf{chal}} = \emptyset$
$(C_j, C_i) \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{ReKeyGen}}, \mathsf{O}_{\mathsf{Challenge}}}(1^\lambda, \mathsf{pk}_i, \mathsf{pk}_j)$
if $(j, C_j) \notin \mathcal{C}_{\mathsf{chal}}$ : return $\perp$
if $(\mathsf{Dec}(\mathsf{sk}_i, C_i) = \mathsf{Dec}(\mathsf{sk}_j, C_j))$ :
   return 1
else : return 0

$\mathsf{O}_{\mathsf{ReKeyGen}}(\boxed{C})$
--------------------------------
$\Delta_{i,j}^{\boxed{C}} \leftarrow \mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j \boxed{, \tilde{C}})$
return $\Delta_{i,j}^{\boxed{C}}$

$\mathsf{O}_{\mathsf{Challenge}}(N)$
--------------------------------
$m \xleftarrow{\$} \mathcal{M}$
if $(N = 0)$ :
   $C_N \leftarrow \mathsf{Enc}(\mathsf{pk}_j, m)$
else :
   $(\mathsf{pk}_0, \mathsf{sk}_0), \ldots, (\mathsf{pk}_{N-1}, \mathsf{sk}_{N-1}) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda)$
   $\mathsf{pk}_N = \mathsf{pk}_j$
   $C_0 \leftarrow \mathsf{Enc}(\mathsf{pk}_0, m)$
   for $\ell \in \{1, \ldots, N\}$ :
      $\Delta_{\ell-1,\ell}^{\boxed{C_{\ell-1}}} \leftarrow \mathsf{ReKeyGen}(\mathsf{sk}_{\ell-1}, \mathsf{pk}_\ell \boxed{, \tilde{C}_{\ell-1}})$
      $C_\ell \leftarrow \mathsf{ReEnc}(\Delta_{\ell-1,\ell}^{\boxed{C_{\ell-1}}}, C_{\ell-1})$
$\mathcal{C}_{\mathsf{chal}}.\mathsf{add}(j, C_N)$
return $(C_N, \mathsf{pk}_0, \ldots, \mathsf{pk}_{N-1})$
```

Figure 5: The unidirectionality game UNI – a formal security game capturing the traditional notion of unidirectionality. $\boxed{\text{Boxed}}$ values show the variant for a ciphertext dependent scheme.

local space. Therefore, a user's ability to retrieve a file at one point does not necessarily mean they have the capacity to retain all files they have ever had access to. Revoked users may try to decrypt a file that was previously encrypted under a key known to them. In many existing schemes, if a revoked user is able to retain limited information about the old ciphertext, they have an advantage in creating a re-encryption under the old key. For example, in the key encapsulation method, as the data encryption key is static, any entity who knows this key can create a re-encryption of the message under any public key. Since the data encryption key is typically much shorter than the message, it is unnecessary for such adversaries to retain full messages if they can instead retain keys, This could make a significant difference in how realistic such an attack is for constrained devices. Given increasing trends in use external storage as the default storage option, we believe definitions addressing revocation that consider bounded storage are worthy of consideration.

We also note that existing notions of unidirectionality do not couple well with ciphertext dependence. For ciphertext dependent schemes, reversal attacks become more significant than traditional unidirectionality since tokens are specific to each ciphertext and thus the act of maliciously retaining the token implies a willingness to store something for each ciphertext. If it is assumed that the server retains the update token, we should also consider other information which they might retain.

Now that we have argued that current notions of unidirectionality are not suitable for access control, we present a security defintion for reversal attacks. We explain the key principles behind the motivation behind this definition.

- **Principle 1:** *Malicious storage cannot be prevented.* It is impossible for one party to prevent another from storing extra information without additional assumptions. In particular we cannot prevent the server from retaining the old ciphertext.
- **Principle 2:** *The amount of storage required to reverse a process has a lower bound.* Whilst we cannot prevent a malicious server from retaining an old ciphertext, we can ensure there is no 'easier' way for them to obtain the old ciphertext. By 'easier', we mean that the server needs significantly less storage than keeping the components of the original ciphertext that were updated. This is similar to the motivation behind an *economically rational server* considered in [9].

There are schemes such as [2] where storage of the update token alone cannot reverse the re-encryption, but retaining elements of the update token and elements of the original ciphertext can reverse a re-encryption. For example, if the update token contains a replacement ciphertext header, then reversal can be trivial by retaining the original ciphertext header. If storing some component of the original ciphertext makes a reversal attack successful, then we should assume that the adversary will do so, especially if the amount of storage needed is at most the size of the update token. For practical reasons, header values and update tokens are often designed to be small [10], and thus can easily be retained. Current models of unidirectionality permit an adversary to retain the update token in order to attempt to re-encrypt. We argue that there is no reason to restrict the information an adversary may store to just update tokens, especially if the adversary can retain other information thand have a greater chance of success.

We now define a reversal attack game which takes into account the amount of information an adversarial server may have retained during the re-encryption process using a storage parameter $\bar{\lambda}$. The following game has an adversary in three stages $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$. All three adversaries receive the security parameter λ, storage parameter $\bar{\lambda}$, public keys and system parameters as input.

Stage 1: $\mathcal{A}_0$ receives a randomly chosen message m and decides which keys $\mathsf{pk}_i, \mathsf{pk}_j$ should be used for encryption and re-encryption.

Stage 2: $\mathcal{A}_1$ receives the ciphertext C and update token $\Delta_{i,j}$ and determines what should be retained in the state $st_{\mathcal{A}}$, which is bounded by a storage parameter $\bar{\lambda}$. Note this adversary never receives the message. Since this adversary knows the storage bound, it can compute many potential states before selecting which one will be passed on to $\mathcal{A}_2$.

Stage 3: $\mathcal{A}_2$ receives the state $st_{\mathcal{A}}$ forwarded by $\mathcal{A}_1$ and the re-encrypted ciphertext C', and uses this to try to output a ciphertext $C_{\mathcal{A}}$ which is an encryption of the same message under the original key. This adversary never receives the message, original ciphertext or the update token — they only receive the information retained by $\mathcal{A}_1$.

Note that the ciphertext $C_{\mathcal{A}}$ output by $\mathcal{A}_2$ does not need to be the original ciphertext — it can be any encryption of m under pk_i. This emulates the scenario where the server must decide how much information to retain about the old ciphertext and update token before later attempting to reverse the re-encryption (or revert to an equivalent ciphertext).

```
$\mathsf{Rev\text{-}ReEnc}_{\mathcal{A}}^{\mathcal{PRE}}(\lambda, \kappa, \bar{\lambda})$
$\mathsf{params} \leftarrow \mathsf{Setup}(\lambda)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_\kappa, \mathsf{sk}_\kappa) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda)$
$m \xleftarrow{\$} \mathcal{M}$
$(i, j) \leftarrow \mathcal{A}_0(1^\lambda, 1^{\bar{\lambda}}, \mathsf{pk}_1, \ldots, \mathsf{pk}_\kappa, m)$
$C \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_i, m)$
$\Delta_{i,j}^{\boxed{C}} \xleftarrow{\$} \mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j \boxed{, \tilde{C}})$
$st_{\mathcal{A}} \leftarrow \mathcal{A}_1(1^\lambda, 1^{\bar{\lambda}}, \mathsf{pk}_1, \ldots \mathsf{pk}_\kappa, \Delta_{i,j}^{\boxed{C}}, C)$
$\textbf{if } |st_{\mathcal{A}}| > \bar{\lambda} \textbf{ return } \perp$
$C' \leftarrow \mathsf{ReEnc}(\Delta_{i,j}^{\boxed{C}}, C)$
$C_{\mathcal{A}} \leftarrow \mathcal{A}_2(1^\lambda, 1^{\bar{\lambda}}, \mathsf{pk}_1, \ldots, \mathsf{pk}_\kappa, st_{\mathcal{A}}, C')$
$\textbf{if } \mathsf{Dec}(\mathsf{sk}_i, C_{\mathcal{A}}) = \mathsf{Dec}(\mathsf{sk}_j, C') :$
    $\textbf{return } 1$
$\textbf{else } : \textbf{return } 0$
```

Figure 6: The reversal game Rev-ReEnc. The $\mathcal{A}_0$ maintains a state bounded by $\bar{\lambda}$ which is passed onto $\mathcal{A}_1$ who attempts to reverse the re-encryption. $\boxed{\text{Boxed}}$ values show the variant for a ciphertext dependent scheme.

Definition 6. *Given a PRE scheme $\mathcal{PRE}$, let $|\Delta|$ denote the size of tokens given by $\mathcal{PRE}$ with security parameter λ, let $|\tilde{C}|$ denote the size of the ciphertext headers, let $|\bar{C}|$ denote the size of the ciphertext body, and let s be the total size of ciphertext components updated by* ReEnc. *Then for $\bar{\lambda} \in \{0, |\Delta|, |\tilde{C}|, |\bar{C}|, s\}$, we define the* advantage *of an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$ in winning the* Rev-ReEnc *game given in Figure 6 as:*

$$\mathbf{adv}_{\mathcal{A},\bar{\lambda}}^{\mathsf{Rev\text{-}ReEnc}}(1^\lambda, \kappa) = \left| \Pr\left[\mathsf{Rev\text{-}ReEnc}_{\mathcal{A}}^{\mathcal{PRE}}(\lambda, \kappa, \bar{\lambda}) = 1\right] - \frac{1}{2^{s-\bar{\lambda}}} \right|, \tag{5}$$

where κ is polynomial in terms of λ and $\frac{1}{2^{s-\bar{\lambda}}}$ is the probability that an adversary who has retained $\bar{\lambda}$ bits of C can correctly guess the remaining bits.

We say that a proxy re-encryption scheme $\mathcal{PRE}$ is $\bar{\lambda}$-irreversible *if for all* PPT *adversaries $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2)$, the advantage of winning the game is negligible:*

$$\mathbf{adv}_{\mathcal{A},\bar{\lambda}}^{\mathsf{Rev\text{-}ReEnc}}(\lambda) \leq \mathsf{negl}(\lambda)\,. \tag{6}$$

Conversely, a PRE scheme is $\bar{\lambda}$-reversible *if there exists a* PPT *algorithm $\mathcal{A}$ that can win the* $\mathsf{Rev\text{-}ReEnc}_{\mathcal{A},\bar{\lambda}}^{\mathcal{PRE}}(\lambda)$ *game with state $st_{\mathcal{A}}$ of size at most $\bar{\lambda}$, with non-negligible probability.*

We restrict ourselves to logical choices of $\bar{\lambda}$. We note that typically $|\Delta| = |\tilde{C}|$, and that in PRE schemes which do not rely on a hybrid key encapsulation mechanism, components of the ciphertext header and body are also usually the same size.

We briefly note that constructing this notion as an indistinguishability game is difficult since the state which the adversary outputs is not fixed. For example, if the adversary stores a truncated hash of the original ciphertext then the game cannot compute another ciphertext, which makes indistinguishability difficult. Since this definition aims to convey that an adversary should not be able to re-encrypt back to the old key, we do not consider the lack of indistinguishability as a significant drawback.

We now formulate a definition for *maximum irreversibility.* Informally, the amount of storage needed to reverse a re-encryption is at least the size of the old ciphertext components that were updated.

Definition 7. *A PRE scheme is* maximally irreversible *if it is s-reversible, where s is the total size of the ciphertext components updated by* ReEnc.

Clearly, *maximum irreversibility* is stronger than traditional notions of unidirectionality as the adversary sees an original ciphertext under pk_i.

Observations

1. The storage bound $\bar{\lambda}$ can be considered similarly to the security parameter λ in that the larger $\bar{\lambda}$ is, the more secure the scheme is. However, even small values for $\bar{\lambda}$ are still meaningful since they convey how easy it is to reverse a re-encryption and can therefore be used to compare different schemes.
2. The most useful values which $\bar{\lambda}$ can take are $\bar{\lambda} = |\Delta|$, as this is comparable to traditional bidirectionality, or $\bar{\lambda} = s$, as this makes a scheme maximally irreversable. In general, useful values are in the range $|\Delta| \leq \bar{\lambda} \leq |C|$.
3. If a scheme is both ReEnc-IND-CPA and maximally irreversible then it is $|C|$-irreversible.
4. All traditionally bidirectional schemes can be shown to be $|\Delta|$-reversible, but there also exist $|\Delta|$-reversible schemes which are *not* traditionally bidirectional, as we shall see in Section 5.3. Since more attacks are covered, saying that a scheme is $|\Delta|$-reversible is stronger than saying it is bidirectional in the traditional sense.
5. Our definition also applies to schemes where the ciphertext is not fully re-randomised upon re-encryption (not ReEnc-IND-CPA), since best-achievable unidirectionality only requires that the state is the size of the number of components which are updated as opposed to the entire ciphertext.
6. We assume that the ciphertext is as compact as it could be, so for all ciphertexts C there is no compression function which allows C to be stored as C^* with $|C^*| < |C|$. For example, in an IND\$-CPA scheme all ciphertexts are indistinguishable from random strings, so only storing a subset of the bits means the adversary can do no better than guessing to obtain the remainder of the ciphertext. If the ciphertexts can be compressed, we can apply the definition to a compressed version of the PRE scheme by having encryption compress the ciphertext, decryption decompress it and have the re-encryption function perform decompression and compression.

Note that if a scheme is token robust and we can prove that the only way of reversing a re-encryption is by storing a state the size of the original ciphertext, then this is the best notion of unidirectionality that can be achieved without assuming that the adversary honestly deletes old ciphertext.In Section 6 we use this definition to define *best-achievable unidirectionality.* By considering unidirectionality in this way, the problem of creating a unidirectional multi-hop PRE scheme may be solved more easily using token robustness and could therefore lead to more unidirectional multi-hop schemes that are practically implementable.

5.3 Existing schemes under the new definition

Some traditionally unidirectional schemes are $|\Delta|-$reversible. For ciphertext-dependent schemes, this means they are bidirectional. In both [10] and [5], the update token consists of the new header and another value used to change the body of the ciphertext using an arithmetic operation. We can generalise this by saying $\Delta = (\Delta_0, \Delta_1)$, where $\Delta_0 = \tilde{C}'$ and $(\Delta_1)^{-1}$ is easily computable. To reverse the re-encryption, the adversary $\mathcal{A}_1$ retains the old header $\tilde{C}$, and computes the inverse of Δ_1, setting $st_{\mathcal{A}} = (\tilde{C}, \Delta_1^{-1})$. Then $\mathcal{A}_2$ can recover $C \leftarrow \mathsf{ReEnc}(st_{\mathcal{A}}, C')$ to win the game. Note that $\mathcal{A}$ does not need to retain Δ_0 as this is contained in the new ciphertext. The state $st_{\mathcal{A}}$ is clearly the same size as $\Delta = (\tilde{C}', \Delta_1)$, and we can therefore consider such schemes to be $|\Delta|$-reversible. Since any adversary willing to store information of size up to $|\Delta|$ will not restrict themselves to retaining Δ alone, our definition better reflects the intuition behind bidirectionality. In particular, because [10] is ciphertext-dependent, it should be considered bidirectional under these realistic assumptions.

Some existing bidirectional schemes are maximally irreversible. In the multi-hop PRE scheme of [6], $\mathsf{ReKeyGen}$ takes two secret keys as input and the ciphertext includes a number of components including $B = (g^a)^r$, where $\mathsf{pk} = g^a, \mathsf{sk} = a$ and $r \xleftarrow{\$} \mathbb{Z}_q$. The re-encryption token takes as input two secret keys a, b and outputs $\Delta_{a,b} = b/a$, which is then used to update B and no other part of the ciphertext. Since both the ciphertext element B and the re-keying token $\Delta_{a,b}$ are integers modulo q, an adversary hoping to reverse the re-encryption by storing $\Delta_{a,b}$ could have simply retained B. Particularly for applications where there is one message per keypair, the server would need to store one token per re-encrypted ciphertext, in which case they could have retained every original ciphertext [3]. Similarly, the original symmetric PRE scheme [4] may also be considered maximally irreversible.

6 Proxy Re-Encryption in the Malicious model

We now describe the requirements for a PRE scheme to be secure in the a model that explores malicious behaviours other than breaking confidentiality. More specifically, we define a model where the server cannot perform encryptions not initiated by a client. We discuss some conditions which apply to re-encryption generally, before explaining the stronger conditions specific to our setting. We consider ReEnc-IND-CPA as another necessary condition for revocation and key expiry, despite the fact that this is not the case in much existing work [3, 4, 6, 7, 17].

In the malicious model, we must ensure that giving the server the ability to perform some re-encryptions does not mean they can perform unauthorised re-eencryptions. In particular, we want our setting to consider revoked users who are honest-but-curious in that they may try to decrypt re-encrypted ciphertexts, but do not collude with the server directly. We thus require a means of ensuring that only authorised re-encryptions are possible. The inability to perform unauthorised re-encryptions breaks down to two necessary properties:

Token Robustness. *No matter how many re-encryption tokens the server sees, the server cannot use these to form a token which encrypts a ciphertext to a new key.* This means the adversary is unable to share messages with users who have not had access to them before. Ciphertext dependence is reasonably trivial to achieve for ElGamal-based schemes such as [4] by having the randomness used to encrypt the message input to $\mathsf{ReKeyGen}$. We build on this existing technique [10] to create a token robust scheme.

[3]As the value B is unique for each ciphertext, retaining B for one ciphertext does not allow a different ciphertext to be re-encrypted whereas the update token can re-encrypt any ciphertext in either direction. This further demonstrates why token robustness is a necessary requirement.

Maximal irreversibility. *The token used to perform a re-encryption cannot be used to reverse that re-encryption.* This is particularly necessary when considering revocation and key expiry. If re-encryption has been performed to revoke access, then reversing that re-encryption regrants access to the revoked user. Our definition of maximal irreversability conveys this under realistic assumptions.

We combine these definitions to form the following definition which is a requirement for a PRE scheme used to enforce changes to access control policy on a malicious server.

Definition 8. *A PRE scheme is* best-achievable unidirectional *if it is both token robust and maximally irreversable.*

Token robustness implies that a token $\Delta_{i,j}^{C_i}$ cannot be used to re-encrypt a ciphertext $C_j \xleftarrow{\$} \mathsf{Enc}(pk_j, m)$ where $C_j \neq C_i$ (except with negligible probability), which covers the traditional notion of unidirectionality. Coupled with maximal irreversability, this means that given a re-encrypted ciphertext C_j under pk_j, the only way that the adversary can produce a ciphertext C_i such that $\mathsf{Dec}(\mathsf{sk}_i, C_i) = \mathsf{Dec}(\mathsf{sk}_j, C_j)$, where pk_i is the original key, is by retaining a state the size of the original ciphertext during the re-encryption process.

We show in Section 6.1 that it is possible to have a scheme which is best-achievable unidirectional using a simple adaptation of ElGamal-based PRE, and prove its security under the DDH assumption.

6.1 A Secure PRE scheme in the malicious model

Recall that for PRE suitable for access control, we require a multi-hop scheme. For PRE in the malicious model, we require a scheme which is unidirectional, ciphertext-dependent and token robust. We give our scheme in Figure 7, based on ElGamal encryption.

Lemma 2. *The scheme given in Figure 7 is correct.*

Proof. Encryptions have the form $C = (g^y, m \cdot g^{xy})$ for message m and public key g^x. Decrypting using secret key x results in $m' = \tilde{C}^{-x} \cdot \bar{C} = (g^y)^{-x} \cdot m \cdot g^{xy} = m$, as required.

The update token resulting from $\mathsf{ReEnc}(\mathsf{sk}_i = x_i, \mathsf{pk}_j = g^{x_j}, \tilde{C} = g^y)$ has the form $\Delta_{i,j}^{C} = (g^{y'}, g^{x_j y' - x_i y})$, meaning that re-encrypted ciphertexts have the form $C_j = (g^{y'}, m \cdot g^{x_i y} \cdot g^{x_j y' - x_i y}) = (g^{y'}, m \cdot g^{x_j y'})$, which is the same as a fresh ciphertext under g^{x_j}, so correctness follows.

6.2 Security analysis

First we show that this scheme is ReEnc-IND-CPA, then best-achievable unidirectional, and finally token robust.

Theorem 2. *The scheme described in Figure 7 is ReEnc-IND-CPA under the decisional Diffie-Hellman assumption.*

Proof. Recall that in the ReEnc-IND-CPA game, the adversary does not learn the precise update token used to perform the re-encryption (similarly to [10]). Therefore it suffices to show that a re-encrypted ciphertext has the same form as a fresh encryption of the message under the new key. In other words, re-encrypted ciphertexts under x_j are identically distributed to ciphertexts encrypted for the first time under x_j. We have already demonstrated this in the proof of Lemma 2.

Proving ReEnc-IND-CPA therefore reduces to ElGamal being IND-CPA. □

```
$\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{params}$
    $p$ large prime
    $g$ a generator of $\mathbb{Z}_p$
    $\mathcal{M} = \mathbb{Z}_p$
    $\mathcal{T} = \mathcal{K} = \mathbb{Z}_p^*$
    return $\mathsf{params} = (p, g, \mathcal{M}, \mathcal{K}, \mathcal{T})$

$\mathsf{KeyGen}(1^\lambda) \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$
    $x \xleftarrow{\$} \mathbb{Z}_p^*$
    $\mathsf{pk} = g^x, \mathsf{sk} = x$
    return $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{Enc}(\mathsf{pk}, m) \xrightarrow{\$} C$
    $y \xleftarrow{\$} \mathbb{Z}_p^*$
    $\tilde{C} = g^y$
    $\bar{C} = m \cdot \mathsf{pk}^y = m \cdot g^{xy}$
    return $C = (\tilde{C}, \bar{C})$

$\mathsf{Dec}(\mathsf{sk}, C) \rightarrow m$
    $m' = \tilde{C}^{-\mathsf{sk}} \cdot \bar{C} = (g^y)^{-x} \cdot m \cdot g^{xy}$
    return $m'$

$\mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j, \tilde{C}) \xrightarrow{\$} \Delta_{i,j}^{C}$
    $y' \xleftarrow{\$} \mathbb{Z}_p^*$
    $a = g^{y'}$
    $b = (\mathsf{pk}_j)^{y'} \cdot \tilde{C}^{-\mathsf{sk}_i} = g^{x_j y'} \cdot g^{-x_i y}$
    return $\Delta_{i,j}^{C} = (a, b)$

$\mathsf{ReEnc}(\Delta_{i,j,}^{C}, C) \rightarrow C'$
    $(\tilde{C}, \bar{C}) \leftarrow C$
    $(a, b) \leftarrow \Delta_{i,j}^{C}$
    $\tilde{C}' = a = g^{y'}$
    $\bar{C}' = \bar{C} \cdot b = m \cdot g^{x_i y} \cdot g^{x_j y' - x_i y}$
    return $C' = (\tilde{C}', \bar{C}')$
```

Figure 7: An ElGamal-based scheme similar to the one given by Blaze et al [4] which is best-achievable unidirectional and token robust.

Theorem 3. *The scheme in Figure 7 is best-achievable unidirectional.*

We prove this through two lemmas, first showing that the scheme is maximally irreversible then that it has token robustness.

Lemma 3. *The scheme described in Figure 7 is maximally irreversible under the DDH assumption.*

In this proof we need to show that the adversary must retain that $\bar{\lambda} = s = 2|x|$, where $x \in \mathbb{Z}_p$, as the token and ciphertext components are all the same size ($|\Delta| = |\tilde{C}| = |\bar{C}|$) and all elements of $\mathbb{Z}_p$, and both ciphertext components are updated during re-encryption. We do not consider the advantage of the adversary storing $g^{x_i y - x_j y'}$ and only part of g^y (or vice versa) here, but in Appendix B we give a general discussion unidirectionality when $\bar{\lambda} \in \{0, 1, \ldots, s-1, s\}$.

Proof. As the old header is replaced independently of the update token, the update token alone cannot be used to reverse a ciphertext. If an adversary has both the re-encryption token used as well as the first component of the old ciphertext, then reversing the re-encryption is trivial. We demonstrate that retaining both values would require the same amount of storage as retaining the original ciphertext, and therefore is maximally irreversibile.

First we observe that $\bar{C}$ and $\Delta_{i,j}^{C(1)}$ can be considered interchangeably as components to be retained in $st_{\mathcal{A}}$. For simplicity, we only consider $\Delta_{i,j}^{C(1)}$ as a potential input to $st_{\mathcal{A}}$. There are only two values which can revert re-encryption of the ciphertext – the old ciphertext header $\tilde{C}$ and $\Delta_{i,j}^{C(1)}$. For the scheme to be best-achievable unidirectional, we need to show that both values must be retained for a successful reversal attack.

We begin by showing that an adversary who only retains $\Delta_{i,j}^{C(1)} = g^{x_i y - x_j y'}$ cannot derive $\tilde{C} = g^y$. We first recall the DDH assumption which states, given (g^a, g^b, g^e) where either $e = ab$ or $e \xleftarrow{\$} \mathbb{Z}_p^*$, no PPT adversary can distinguish which was given.

In addition to $g^{x_i y - x_j y'}$, we also assume that the server knows the public keys g^{x_i}, g^{x_j} and the header of the new file, $g^{y'}$. For simpler notation, let $a = x_i, b = x_j, c = y, d = y'$. The adversary's task is therefore: given (a, g^b, g^d, g^{bd-ac}), find g^c. We show that if there exists an adversary $\mathcal{B}$ such that $\mathcal{B}(a, g^b, g^d, g^{bd-ac}) \to g^c$ for $a, b, c, d \xleftarrow{\$} \mathbb{Z}_p^*$, then this would break DDH. Given a DDH challenge (g^a, g^b, g^e), sample $a, c \xleftarrow{\$} \mathbb{Z}_p^*$ and call $\mathcal{B}(a, g^b, g^d, g^{e-ac}) \to g^{c'}$ If $g^{c'} = g^c$ then $e = bd$, breaking DDH. We therefore conclude that g^y cannot be recovered from $(x_i, g^{x_j}, g^{y'}, g^{x_j y' - x_i y})$.

For the other direction, we note that the adversary being able to calculate g^{bd-ac} from (a, g^b, g^d, g^c) would clearly also break DDH. We therefore conclude that $g^{x_j y' - x_i y}$ cannot be recovered from $(x_i, g^{x_j}, g^{y'}, g^y)$.

It remains to show that the adversary cannot output an equivalent ciphertext $(g^{\hat{y}}, m \cdot g^{x_i \hat{y}})$ for some $\hat{y} \in \mathbb{Z}_p^*$ given $(g^{y'}, m \cdot g^{x_j y'}), x_i, g^{x_j}$. To do this the adversary would need to be able to derive m by calculating $g^{x_j y'}$, which would again imply the DDH assumption can be broken. □

This proof shows that the attacker needs to store just as much as if they were storing the original ciphertext, so they could have avoided needing to perform the attack by simply retaining the original ciphertext.

Lemma 4. *The scheme in Figure 7 has token robustness under the Computational Diffie-Hellman (CDH) assumption.*

Proof. To win the token robustness game, the adversary must output a token which re-encrypts an honestly-generated ciphertext so that it is under a key that it has not been encrypted under before.

Recall that the encryption oracle $\mathsf{OEnc}(i, m)$ outputs a ciphertext of the form $(g^y, m \cdot g^{x_i y})$ and the token generation oracle $\mathsf{OReKeyGen}(i, j, C)$ outputs a re-encryption token of the form $(g^{y'}, g^{x_j y' - x_i y})$. Let the challenge input ciphertext be denoted $C_i^{\mathcal{A}} = (g^y, m \cdot g^{x_i y})$. Then the adversary must output a token of the form $\Delta_{\mathcal{A}} = (g^{y'}, g^{x_j y' - x_i y})$, where $j \notin \mathsf{chain}(m)$. It may be possible that querying two completely different keys and a different ciphertext $C_l = (g^{y_l}, m \cdot g^{x_l y_l})$ results in a token $\Delta_{l,k}^{C_l} = (g^{y_k}, g^{x_k y_k - x_l y_l})$, where $g^{x_k y_k - x_l y_l} = g^{x_i y - x_j y'}$, but this only occurs with negligible probability.

The token generation oracle means that the adversary receives tokens of the form $\Delta_{i,k}^{C_i^{\mathcal{A}}} = (g^{y''}, g^{x_k y'' - x_i y})$. The adversary's goal is to output an update token $\Delta^{\mathcal{A}} = (g^{y'}, g^{x_j y' - x_i y})$, where $x_i = sk_i$ and $C = (g^y, m \cdot g^{xy})$ is the ciphertext output by $\mathcal{A}$ to be re-encrypted to a new key. We note that it is trivial for the adversary to calculate $g^{x_j y'}$ for some $y' \xleftarrow{\$} \mathbb{Z}_p^*$ from pk_j by setting $\mathsf{pk}_j^{y'} = g^{x_j y'}$. To create a winning $\Delta_{i,j}^{\mathcal{A}}$, it remains for $\mathcal{A}$ to calculate $g^{x_i y}$. Since the ciphertext is honestly generated ($C_i^{\mathcal{A}} \in \mathcal{C}_{\mathsf{honest}}$), the adversary does not know y.

Since factoring is a difficult problem modulo p (otherwise, ElGamal would not be IND-CPA secure), g^{xy} cannot be easily extracted from tokens of the form $\Delta_{i,k,}^{C_i^{\mathcal{A}}} = g^{x_k y'' - x_i y}$ for some $x_k \leftarrow \mathsf{KeyGen}(1^\lambda), y'' \in \mathbb{Z}_p^*$. We conclude that the adversary can only gain ciphertexts of the correct form for keys pk_i where $i \in \mathsf{chain}(\mathsf{Dec}(\mathsf{sk}_i, C_i^{\mathcal{A}}))$. Since the adversary must output a token for a new key to win the game, we conclude the scheme is token robust. □

We have shown that our scheme is suitable for the malicious model according to the goals we outlined in Section 1. This means a malicious server will be unable to perform unauthorised re-encryptions on stored files, as much as can be guaranteed given realistic storage asumptions.

7 Ciphertext Origin Authentication (COA)

We now present an alternative approach to ensuring that re-encryptions have been initiated by honest parties by revisiting the traditional notion of data origin authentication.Traditionally, data origin authentication is intended for settings where the party who created the message also encrypts it. This means that the appropriate tag that ties the identity of the sender to the ciphertext can be associated either with the message prior to encryption, or with the ciphertext itself. However, for PRE this is not always the case. In situations where there is one key per party, then the act of having created the re-encryption key would imply knowledge of sk_i, meaning that the receiver can be sure that re-encryption was initiated by the party whose public key is pk_i. Therefore a correct re-encryption implies that the re-encryption was initiated by a party with access to the message. However, in applications where more than one party shares an encryption key, such as some access control systems, proof of having used this key is not sufficient to authenticate the encryptor / re-encryption initiator.

It may be beneficial in some applications to use individual signature keys to verify specific identities, in addition to the encryption keys for confidentiality. Both signatures and PRE could be combined to create an authenticated PRE scheme. This is useful in auditing changes to access control policy, or enabling users to verify which user has revoked their access.

Now that we have outlined this distinction, we formulate the notion of *Ciphertext Origin Authentication (COA)*.

7.1 Correctness upon verification

Most Authenticated Encryption (AE) and Signcryption schemes [22] use a mechanism where during the decryption process an authentication check is made, and the receiver is supposed to terminate the decryption process if the check fails. Such a check is also made in [10], which to our knowledge is the only PRE scheme to provide authentication. However, if corrupted users are being considered then honest termination of a process is not guaranteed – a malicious user could ignore the outcome of the authentication check and derive the message regardless. We therefore aim for compulsory COA that provides stronger security against users who want to change policy without being caught, as the message can *only* be derived if identity is verified correctly. We call this *correctness upon verification* and consider it a secondary goal.

The most intuitive way of proving which entity encrypted a message is to show proof of knowledge of the secret information used to form the new ciphertext. In our scheme outlined in Figure 7 this is the value y. However, the COA check must not leak y as this will enable decryption using the public key. Therefore, we need the initiator to prove that they know y without revealing it.

Recall the basic ElGamal signature scheme [12]:

1. $\mathsf{Setup}(1^\lambda) \to p, g$, where p is a prime and g is a generator of $\mathbb{Z}_p$.
2. $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{svk}, \mathsf{ssk}) = (g^{\mathsf{x}}, \mathbf{x})$, where $\mathbf{x} \xleftarrow{\$} \mathbb{Z}_p$.
3. $\mathsf{Sign}(\mathsf{ssk}, m) \to \sigma = (r, s)$: $r = g^k$ for $k \xleftarrow{\$} \mathbb{Z}_p$ and $s = (h(m) - \mathsf{ssk} \cdot r)k^{-1} \mod p-1 = (h(m) - \mathbf{x}r)k^{-1} \mod p-1$, for hash function $h()$.
4. $\mathsf{Verify}(\mathsf{svk}, \sigma, m) \to \{0,1\}$: if $g^{h(m)} = \mathsf{svk}^{\sigma_0} \cdot {\sigma_0}^{\sigma_1} = (g^{\mathsf{x}})^r \cdot r^s$ return 1, else return 0.

The hash of the message m is signed so as to enable the signing of arbitrary length messages. We can see correctness as $(g^{\mathbf{x}})^r \cdot r^s = g^{\mathbf{x}r} \cdot (g^k)^{(h(m)-\mathbf{x}r)k^{-1}} = g^{\mathbf{x}r} \cdot g^{h(m)-\mathbf{x}r} = g^{h(m)}$.

We can obtain a mandatory COA check by replacing g^y in our original scheme (Figure 7) with an ElGamal signature on y signed using the initiator's public key. Note that we do not need to sign the hash of y, as y is already a random value in the correct range. We also adapt the Verify algorithm so that instead of it returning 0 or 1 to indicate a pass or fail, it derives a specific value a. We call the resulting algorithm $\mathsf{VerRetrieve}$.

```
$\mathsf{sigKeyGen}(1^\lambda)$
    $\mathbf{x} \xleftarrow{\$} \mathbb{Z}_p^*$
    $\mathsf{ssk} = \mathbf{x}, \mathsf{svk} = (g^{\mathbf{x}})$
    return $(\mathsf{ssk}, \mathsf{svk})$

$\mathsf{Sign}(\mathsf{ssk}, y)$
    $\mathbf{x} \leftarrow \mathsf{ssk}$
    $k \xleftarrow{\$} \mathbb{Z}_p, r = g^k$
    $s = (y - \mathbf{x}r)k^{-1} \mod p-1$
    $\sigma = (r, s)$
    return $\sigma$

$\mathsf{VerRetrieve}(\mathsf{svk}, \sigma)$
    $(r, s) = \sigma$
    $a = \mathsf{svk}^r \cdot r^s = (g^{\mathbf{x}})^r \cdot r^s$
    return $a$
```

Ciphertexts now have the form $C = (\sigma, m \cdot g^{xy})$. By the correctness of ElGamal signatures, $\mathsf{VerRetrieve}(\mathsf{svk} = g^{\mathbf{x}}, \sigma)$ should return $a = g^y$ if σ was signed using $\mathsf{ssk} = \mathbf{x}$, since $(g^{\mathbf{x}})^r \cdot r^s = g^{\mathbf{x}r} \cdot g^{k(y-\mathbf{x}r)k^{-1}} = g^y$. Since y is not revealed during this process, this adaptation should still meet the security properties proven in Section 6.2. Since g^y is needed for decryption, if follows that correctly deriving g^y is necessary to derive the correct message. Correctly deriving g^y relies on correct verification during decryption, meaning that verification is not optional, and so the decryptor cannot deny knowledge of who encrypted the message.

However, if the scheme is only adapted with the change outlined above then there is no confirmation that the obtained message m is the correct one, so by extension the receiver cannot verify that g^y was correct. Suppose that Alice encrypts a message m to obtain a ciphertext (σ_A, C_A), and that she now shares this with Mallory (so Mallory knows x). Mallory can forge Alice's signature on a message m_M of her choice encrypted for public keys for which Mallory knows the corresponding secret $\mathsf{sk}' = x'$ by computing the following:

```
$\mathsf{Forge}(m_M, g_A^{\mathbf{x}}, x', \sigma_A)$
    $\sigma_A = (r_A, s_A)$
    $g^y = (g_A^{\mathbf{x}})^r \cdot r_A^{s_A}$
    $C_M = (g^y, m_M \cdot (g^y)^{x'})$
    return $(\sigma_A, C_M)$
```

Then receivers of this ciphertext will conclude that Alice encrypted the message m_M, not Mallory. We note that the encryption used here can either be done using the same secret key $(x' = x)$ or a different one $(x' \neq x)$. This is a realistic scenario in access control schemes where decryption keys are shared, and the attack means that anyone who knows the secret key can forge the encryptor's signature. Therefore in order to have COA, we also need a message integrity check.

We propose adapting the encryption mechanism to replace $\bar{C} = m \cdot g^{xy}$ with $\bar{C} = (g^{x\hat{y}h(m)}, m \cdot g^{xy})$ where $\hat{y} \xleftarrow{\$} \mathbb{Z}_p^*$ and adding the matching signature to the header. So $\tilde{C}$ now contains two signatures – one used to derive g^y and the other used to verify the message m.

We redefine $\mathsf{ReKeyGen}$ as $\mathsf{AuthReKeyGen}$, which additionally takes the currect signature verification key svk_A and re-encryption initator's signing key ssk_B as input. To achieve best-achievable unidirectionality, $\mathsf{AuthReKeyGen}$ selects a new $\hat{y}$ uniformly at random so that new

```
$\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{params}$
    $p$ large prime
    $g$ a generator of $\mathbb{Z}_p$
    return $(p, g)$

$\mathsf{EncKeyGen}(1^\lambda) \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$
    $x \xleftarrow{\$} \mathbb{Z}_p^*$
    $\mathsf{pk} = g^x, \mathsf{sk} = x$
    return $(\mathsf{pk}, \mathsf{sk})$

$\mathsf{sigKeyGen}(1^\lambda) \xrightarrow{\$} (\mathsf{svk}, \mathsf{ssk})$
    $\mathbf{x} \xleftarrow{\$} \mathbb{Z}_p^*$
    $\mathsf{ssk} = \mathbf{x}, \mathsf{svk} = g^{\mathbf{x}}$
    return $(\mathsf{svk}, \mathsf{ssk})$

$\mathsf{Sign}(\mathsf{ssk}, y) \xrightarrow{\$} \sigma$
    $k \xleftarrow{\$} \mathbb{Z}_p, r = g^k$
    $s = (y - \mathsf{ssk} \cdot r)k^{-1} \mod p - 1$
    $\sigma = (r, s)$
    return $\sigma$

$\mathsf{VerRetrieve}(\mathsf{svk}, \sigma) \rightarrow a$
    $(r, s) \leftarrow \sigma$
    $a = \mathsf{svk}^r \cdot r^s$
    return $a$

$\mathsf{AuthEnc}(\mathsf{ssk}, \mathsf{pk}, m) \xrightarrow{\$} C$
    $y \xleftarrow{\$} \mathbb{Z}_p^*, \sigma \leftarrow \mathsf{Sign}(\mathsf{ssk}, y)$
    $\hat{y} \xleftarrow{\$} \mathbb{Z}_p^*, \hat{\sigma} \leftarrow \mathsf{Sign}(\mathsf{ssk}, \hat{y})$
    $\tilde{C} = (\sigma, \hat{\sigma})$
    $\bar{C} = (\mathsf{pk}^{\hat{y} \cdot h(m)}, m \cdot \mathsf{pk}^y)$
    return $C = (\tilde{C}, \bar{C})$

$\mathsf{AuthDec}(\mathsf{svk}, \mathsf{sk}, C) \rightarrow m$
    $(\tilde{C}, \bar{C}) \leftarrow C$
    $(\sigma, \hat{\sigma}) \leftarrow \tilde{C}$
    $a \leftarrow \mathsf{VerRetrieve}(\mathsf{svk}, \sigma)$
    $\hat{a} \leftarrow \mathsf{VerRetrieve}(\mathsf{svk}, \hat{\sigma})$
    $m' = a^{-\mathsf{sk}} \cdot \bar{C}_1$
    if $\hat{a}^{\mathsf{sk} \cdot h(m')} \neq \bar{C}_0$ :
        return $\perp$
    else :
        return $m'$

$\mathsf{AuthReKeyGen}(\mathsf{svk}_A, \mathsf{ssk}_B, \mathsf{sk}_i, \mathsf{pk}_j, \tilde{C}) \xrightarrow{\$} \Delta_{i,j}^C$
    $(\sigma, \hat{\sigma}) = \tilde{C}$
    $a \leftarrow \mathsf{VerRetrieve}(\mathsf{svk}_A, \sigma)$
    $\hat{a} \leftarrow \mathsf{VerRetrieve}(\mathsf{svk}_A, \hat{\sigma})$
    $y' \xleftarrow{\$} \mathbb{Z}_p^*, \sigma' \leftarrow \mathsf{Sign}(\mathsf{ssk}_B, y')$
    $\hat{y}' \xleftarrow{\$} \mathbb{Z}_p^*, \hat{\sigma}' \leftarrow \mathsf{Sign}(\mathsf{ssk}_B, \hat{y}')$
    $\tilde{C}' = (\sigma', \hat{\sigma}')$
    $\Delta_{i,j}^C = (\tilde{C}', (\mathsf{pk}_j)^{\hat{y}'} \cdot \hat{a}^{-\mathsf{sk}_i}, (\mathsf{pk}_j)^{y'} \cdot a^{-\mathsf{sk}_i})$
    return $\Delta_{i,j}^C$

$\mathsf{ReEnc}(\Delta_{i,j}^C, C) \rightarrow C$
    $(\tilde{C}', \Delta_1, \Delta_2) = \Delta_{i,j}^C$
    $\bar{C}' = (\bar{C}_0 \cdot \Delta_1, \bar{C}_1 \cdot \Delta_2)$
    return $(\tilde{C}', \bar{C}')$
```

Figure 8: Adapted scheme with ciphertext origin authentication and message integrity. Receivers must verify the party that encrypted (initiated re-encryption for) the ciphertext to decrypt the message.

entropy is added throughout the ciphertext. Then to verify the derived message m', the receiver derives $\hat{a}$ from σ and confirms that $a^{xh(m')} = \bar{C}_0$. If they match then we have both message integrity and COA. Therefore this mechanism provides a means of pairing g^y with m and associating this with the identity of the encryptor (or re-encryption initiator). The full adapted scheme is given in Figure 8.

7.2 COA in other schemes

COA can be added to other schemes. It is sufficient to create a signature using the encryptor's signing key and the randomness used to form the ciphertext and changing re-encryption and decryption accordingly. We now demonstrate how to extend existing schemes to offer COA. For other ElGamal-based schemes including [4, 15, 18], a similar adaptation to the one we made in Section 7.1 can be used.

We now propose an extension to ReCrypt [10] (a description of which can be found in Appendix A). Whilst ReCrypt uses symmetric PRE for re-encryption, we believe that COA is still valid for symmetric encryption. Recall that a COA check can either be optional or compulsory,

```
Setup($1^\lambda$)
    $p$ large prime
    $g$ a generator of $\mathbb{Z}_p$
    return $p, g$

encKeyGen($1^\lambda$)
    $k \xleftarrow{\$} \mathcal{KG}(1^\lambda)$

sigKeyGen($1^\lambda$)
    $\mathbf{x} \xleftarrow{\$} \mathbb{Z}_p^*$
    $\mathsf{ssk} = \mathbf{x}, \mathsf{svk} = g^{\mathbf{x}}$ return $(\mathsf{ssk}, \mathsf{svk})$

AuthEnc($k, \mathsf{ssk}, m$)
    $r \xleftarrow{\$} \mathbb{Z}_p^*$
    $x, y \xleftarrow{\$} \{i, j, : i + j = g^r\}$
    $\sigma \leftarrow \mathsf{Sign}(\mathsf{ssk}, r)$
    $\tau = h(m) + F(x, 0)$
    $\tilde{C} = (\mathcal{E}_k(\sigma, \tau), y)$
    for $0 \leq i \leq n$ :
        $\bar{C}_i = m_i + F(x, i+1)$
    return $(\tilde{C}, \bar{C})$

AuthDec($k, \mathsf{svk}, (\tilde{C}, \bar{C})$)
    $(\sigma, \tau) \leftarrow \mathcal{D}_k(\tilde{C}_0)$
    if $(\sigma, \tau) = \perp$ return $\perp$
    $\chi \leftarrow \mathsf{VerRetrieve}(\mathsf{svk}, \sigma)$
    for $0 \leq i \leq n$ :
        $m_i = \bar{C}_i - F(\chi - y, i+1)$
    if $h(m) + F(\chi - y, 0) \neq \tau$
        return $\perp$
    else return $m$

AuthReKeyGen($k_i, \mathsf{svk}_A, k_j, \mathbf{x}_B, \tilde{C}$)
    $(\sigma, \tau) \leftarrow \mathcal{D}_{k_i}(\tilde{C}_0)$
    if $(\sigma, \tau) = \perp$ return $\perp$
    $\chi \leftarrow \mathsf{VerRetrieve}(\mathsf{svk}_A, \sigma)$
    $y = \tilde{C}_1, x = \chi - y$
    $r' \xleftarrow{\$} \mathbb{Z}_p^*$
    $x', y' \xleftarrow{\$} \{i, j, : i + j = g^{r'}\}$
    $\sigma' \leftarrow \mathsf{Sign}(\mathsf{ssk}_B, r')$
    $a = x' - x$
    $\tau' = \tau + F(a, 0)$
    $\tilde{C}' = (\mathcal{E}_{k_j}(\sigma', \tau'), y')$
    return $\Delta_{i,j}^C = (\tilde{C}', a)$

ReEnc($\Delta_{i,j}^C, C$)
    $(\tilde{C}', a) \leftarrow \Delta_{i,j}^C$
    for $0 \leq i \leq n$ :
        $\bar{C}'_i = \bar{C}_i + F(a, i+1)$
    return $(\tilde{C}', \bar{C}')$
```

Figure 9: ReCrypt [10] with compulsory COA

depending on whether or not corrupted users are considered. In ReCrypt, the message integrity check is optional and therefore it would be arguable that an optional COA check suffices in their model. We first describe a simple extension which creates an optional check here.

For a basic extension, the ciphertext should replace χ with $(\chi, \sigma = \mathsf{Sign}(\mathbf{x}, \chi))$ and decryption should include the step $\mathsf{Verify}((g^{\mathbf{x}}), \sigma, \chi)$ and only return m if this outputs 1. We note that this may be preferable to our scheme from Figure 7 depending on the application, as it permits the re-encryption of longer messages, with the caveat that it is not best-achievable unidirectional.

Since ReCrypt is not an ElGamal-based scheme, the adaptation for a mandatory COA check is more complicated. Essentially, we replace $x, y \xleftarrow{\$} \mathcal{K}$ and $\chi = x + y$ in the ciphertext header by having $x, y \xleftarrow{\$} \{i, j, : i+j = g^r\}$ for $r \xleftarrow{\$} \mathbb{Z}_p^*$ and setting $\chi = g^r$. Then by signing r, VerRetrieve will return $g^r = \chi$ and decryption is adjusted accordingly. Note that we move y from the ciphertext body into the ciphertext header, as it is now needed to generate update tokens. The full extension is given in Figure 9.

8 Conclusions and Open Problems

We revisited the notion of unidirectionality in PRE schemes and provided a formal security definition that covers reversal attacks. We have shown how, under this new definition, existing PRE schemes which are considered traditionally bidirectional may be considered unidirectional, and vice versa. We also outlined properties that a PRE scheme needs in order to be considered secure in the malicious model, in particular defining token robustness — the inability of the server to forge update tokens. Finally, we introduced a new notion of COA for authenticated PRE and discussed how to implement this. Schemes meeting these definitions are given in the appendices.

A useful extension of this work is the design of a best-achievable unidirectional token robust scheme which can be used for longer messages. This could be achieved trivially by having the update token be as long as the ciphertext, but this is an inefficient solution, going against the motivations for outsourcing re-encryption. Developing a best-achievable unidirectional PRE scheme with small update tokens has similar challenges to white-box cryptography and obfuscation [13]. We leave these as open problems.

We also leave to future work creating an IND-CCA secure PRE scheme which is best-achievable unidirectional and token robust, as well as defining what it means for a PRE scheme to be post-compromise secure [8] and the creation of a post-compromise secure PRE scheme.

9 Acknowledgements

Many thanks to my supervisor Keith Martin for guiding my ideas into a worthwhile piece of work. Also many thanks to those who gave up their time to proofread my work, especially Christian Janson and James Alderman.

References

1. Amazon Web Services: Protecting data using client-side encryption (2017), http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html
2. Ateniese, G., Benson, K., Hohenberger, S.: Key-private proxy re-encryption. In: Fischlin, M. (ed.) Topics in Cryptology - CT-RSA 2009, The Cryptographers' Track at the RSA Conference 2009, San Francisco, CA, USA, April 20-24, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5473, pp. 279–294. Springer (2009), https://doi.org/10.1007/978-3-642-00862-7_19
3. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Trans. Inf. Syst. Secur. 9(1), 1–30 (2006), http://doi.acm.org/10.1145/1127345.1127346
4. Blaze, M., Bleumer, G., Strauss, M.: Divertible protocols and atomic proxy cryptography. In: Nyberg, K. (ed.) Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding. Lecture Notes in Computer Science, vol. 1403, pp. 127–144. Springer (1998), https://doi.org/10.1007/BFb0054122
5. Boneh, D., Lewi, K., Montgomery, H.W., Raghunathan, A.: Key homomorphic prfs and their applications. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8042, pp. 410–428. Springer (2013), https://doi.org/10.1007/978-3-642-40041-4_23
6. Canetti, R., Hohenberger, S.: Chosen-ciphertext secure proxy re-encryption. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (eds.) Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007. pp. 185–194. ACM (2007), http://doi.acm.org/10.1145/1315245.1315269

7. Chow, S.S.M., Weng, J., Yang, Y., Deng, R.H.: Efficient unidirectional proxy re-encryption. In: Bernstein, D.J., Lange, T. (eds.) Progress in Cryptology - AFRICACRYPT 2010, Third International Conference on Cryptology in Africa, Stellenbosch, South Africa, May 3-6, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6055, pp. 316–332. Springer (2010), `https://doi.org/10.1007/978-3-642-12678-9_19`
8. Cohn-Gordon, K., Cremers, C.J.F., Garratt, L.: On post-compromise security. In: IEEE 29th Computer Security Foundations Symposium, CSF 2016, Lisbon, Portugal, June 27 - July 1, 2016. pp. 164–178. IEEE Computer Society (2016), `https://doi.org/10.1109/CSF.2016.19`
9. van Dijk, M., Juels, A., Oprea, A., Rivest, R.L., Stefanov, E., Triandopoulos, N.: Hourglass schemes: how to prove that cloud files are encrypted. In: Yu, T., Danezis, G., Gligor, V.D. (eds.) the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012. pp. 265–280. ACM (2012), `http://doi.acm.org/10.1145/2382196.2382227`
10. Everspaugh, A., Paterson, K.G., Ristenpart, T., Scott, S.: Key rotation for authenticated encryption. IACR Cryptology ePrint Archive 2017, 527 (2017), `http://eprint.iacr.org/2017/527`
11. Fuchsbauer, G., Kamath, C., Klein, K., Pietrzak, K.: Adaptively secure proxy re-encryption. Cryptology ePrint Archive, Report 2018/426 (2018), `https://eprint.iacr.org/2018/426`
12. Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Information Theory 31(4), 469–472 (1985), `https://doi.org/10.1109/TIT.1985.1057074`
13. Hohenberger, S., Rothblum, G.N., Shelat, A., Vaikuntanathan, V.: Securely obfuscating re-encryption. J. Cryptology 24(4), 694–719 (2011), `https://doi.org/10.1007/s00145-010-9077-7`
14. Ivan, A., Dodis, Y.: Proxy cryptography revisited. In: Proceedings of the Network and Distributed System Security Symposium, NDSS 2003, San Diego, California, USA. The Internet Society (2003), `http://www.isoc.org/isoc/conferences/ndss/03/proceedings/papers/14.pdf`
15. Lehmann, A., Tackmann, B.: Updatable encryption with post-compromise security. In: Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III. pp. 685–716 (2018), `https://doi.org/10.1007/978-3-319-78372-7_22`
16. Liang, X., Cao, Z., Lin, H., Shao, J.: Attribute based proxy re-encryption with delegating capabilities. In: Li, W., Susilo, W., Tupakula, U.K., Safavi-Naini, R., Varadharajan, V. (eds.) Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2009, Sydney, Australia, March 10-12, 2009. pp. 276–286. ACM (2009), `http://doi.acm.org/10.1145/1533057.1533094`
17. Libert, B., Vergnaud, D.: Multi-use unidirectional proxy re-signatures. In: Ning, P., Syverson, P.F., Jha, S. (eds.) Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, October 27-31, 2008. pp. 511–520. ACM (2008), `http://doi.acm.org/10.1145/1455770.1455835`
18. Libert, B., Vergnaud, D.: Unidirectional chosen-ciphertext secure proxy re-encryption. In: Cramer, R. (ed.) Public Key Cryptography - PKC 2008, 11th International Workshop on Practice and Theory in Public-Key Cryptography, Barcelona, Spain, March 9-12, 2008. Proceedings. Lecture Notes in Computer Science, vol. 4939, pp. 360–379. Springer (2008), `https://doi.org/10.1007/978-3-540-78440-1_21`
19. Myers, S., Shull, A.: Efficient hybrid proxy re-encryption for practical revocation and key rotation. IACR Cryptology ePrint Archive 2017, 833 (2017), `http://eprint.iacr.org/2017/833`
20. Polyakov, Y., Rohloff, K., Sahu, G., Vaikuntanthan, V.: Fast proxy re-encryption for publish/subscribe systems. IACR Cryptology ePrint Archive 2017, 410 (2017), `http://eprint.iacr.org/2017/410`
21. Shao, J., Cao, Z.: Cca-secure proxy re-encryption without pairings. In: Jarecki, S., Tsudik, G. (eds.) Public Key Cryptography - PKC 2009, 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5443, pp. 357–376. Springer (2009), `https://doi.org/10.1007/978-3-642-00468-1_20`
22. Zheng, Y.: Digital signcryption or how to achieve cost(signature & encryption) $<<$ cost(signature) + cost(encryption). In: Jr., B.S.K. (ed.) Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings. Lecture Notes in Computer Science, vol. 1294, pp. 165–179. Springer (1997), `https://doi.org/10.1007/BFb0052234`

```
$\mathsf{Setup}(1^\lambda)$
    $p$ large prime
    $g$ a generator of $\mathbb{Z}_p$
    return $p, g$

$\mathsf{KeyGen}(1^\lambda)$
    $k \xleftarrow{\$} \mathcal{KG}(1^\lambda)$

$\mathsf{Enc}(k, m)$
    $x, y \xleftarrow{\$} \mathcal{K}$
    $\chi = x + y$
    $\tau = h(m) + F(x, 0)$
    $\tilde{C} = \mathcal{E}_k(\chi, \tau)$
    $\bar{C}_0 = y$
    for $0 \le i \le n$ :
        $\bar{C}_i = m_i + F(x, i)$
    return $(\tilde{C}, \bar{C})$

$\mathsf{Dec}(k, (\tilde{C}, \bar{C}))$
    $(\chi, \tau) \leftarrow \mathcal{D}_k(\tilde{C})$
    if $(\chi, \tau) = \perp$: return $\perp$
    $y = \bar{C}_0$
    for $0 \le i \le n$ :
        $m_i = \bar{C}_i - F(\chi - y, i)$
    if $h(m) + F(\chi - y, 0) \neq \tau$ :
        return $\perp$
    else return $m$

$\mathsf{ReKeyGen}(k_i, k_j, \tilde{C})$
    $(\chi, \tau) \leftarrow \mathcal{D}_k(\tilde{C}_0)$
    if $(\chi, \tau) = \perp$: return $\perp$
    $x', y' \xleftarrow{\$} \mathcal{K}$
    $\chi' = \chi + x' + y'$
    $\tau' = \tau + F(x', 0)$
    $\tilde{C}' = \mathcal{E}_{k_j}(\chi', \tau')$
    return $\Delta^C_{i,j} = (\tilde{C}', x', y')$

$\mathsf{ReEnc}(\Delta^C_{i,j}, C)$
    $(\tilde{C}', x', y') \leftarrow \Delta^C_{i,j}$
    $y = \bar{C}_0$
    $\bar{C}'_0 = y' + y$
    for $0 \le i \le n$ :
        $\bar{C}'_i = \bar{C}_i + F(x', i)$
    return $(\tilde{C}', \bar{C}')$
```

Figure 10: ReCrypt [10], a general construction relying on a symmetric encryption scheme $(\mathcal{KG}, \mathcal{E}, \mathcal{D})$ and key-homomorphic PRF F.

A ReCrypt [10]

We give an extension to the ReCrypt scheme given in [10] which includes a compulsory COA check. A description of ReCrypt is given in Figure 10.

B More fine-grained choices for $\bar{\lambda}$

Recall that in our definition of unidirectionality Definition 6, the security parameter $\bar{\lambda}$ is determined in terms of components $\bar{\lambda} \in \{0, |\Delta|, |\tilde{C}|, |\bar{C}|, s\}$ where s is the total size of ciphertext components updated during re-encryption. A more fine-grained definition would be to allow $\bar{\lambda}$ to be any number of bits $\bar{\lambda} \in \{0, 1, \ldots, s\}$. We decided against this definition since, although it is more fine-grained in terms of security, it makes proofs much harder for what we consider to not be a significant distinction in practice. The main difference in terms of proofs is that the more fine-grained options for $\bar{\lambda}$ require additional proofs that an adversary who only retains part of a component or component has no real advantage in calculating the rest. Ultimately, whilst retaining certain values such as the old header or the update token make sense, it's harder to envisage or justify an adversary who is additionally retaining some number of bits of other values.

To demonstrate this in how proofs become more complex for more fine-grained choices of $\bar{\lambda}$, as well as for general interest, we include a proof here that shows that our scheme in Figure 7 is also best-achievable unidirectional in this stricter sense, on the condition that the prime p chosen by the Setup algorithm is a Mersenne prime, which is to say that $p = 2^n - 1$ for some n. To prove that our scheme is best-achievable unidirectional, we need the following lemma, after which best-achievable unidirectionality follows from a trivial adaptation of Lemma 3.

Lemma 5. *Let $p = 2^n - 1$ be a (Mersenne) prime and let $a, b \xleftarrow{\$} \mathbb{Z}_p^*$. Then, for all* PPT *algorithms $\mathcal{B}$ there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that:*

$$\Pr\left[\mathcal{B}(g^a, g^b, [g^{ab}]_{\bar{\lambda}}) \to g^{ab}\right] \leq \frac{1}{2^{n-\bar{\lambda}}} + \mathsf{negl}(\lambda) \qquad (7)$$

where $[g^{ab}]_{\bar{\lambda}}$ denotes the $\bar{\lambda}$ known bits of g^{ab}.

Proof. Since p is a Mersenne prime it has the form $2^n - 1$, and so is n bits long and there is a bijection between integers in $\mathbb{Z}_p$ and bitstrings of length n. In other words every integer in $\mathbb{Z}_p$ can be represented as a bitstring. Therefore random elements of $\mathbb{Z}_p$ of the form can be modelled as random n-bit strings. More specifically, elements g^c where c is is chosen uniformly at random can be considered as random n-bit strings.

We proceed by induction. We note that an element of this $\mathbb{Z}_p$ has size n.
Base case: $\bar{\lambda} = 0$. By the CDH assumption, the adversary cannot compute g^{ab} with probability significantly greater than $\frac{1}{2^n}$.
Assume for $\bar{\lambda} = i$. An adversary who knows the first i bits of g^{ab} cannot calculate the remaining $n - i$ bits with probability higher than $\frac{1}{2^{n-i}}$.
We now consider a variant of the DDH game which factors in the adversary's knowledge of i bits of g^{ab}. We denote by $[g^{ab}]_{\bar{\lambda}}$ the $\bar{\lambda}$ known bits of g^{ab}. Let a, b be selected uniformly at random and let c be selected at random with the condition that $[g^c]_i = [g^{ab}]_i$: $a, b \xleftarrow{\$} \mathbb{Z}_p^*, c \xleftarrow{\$} \{x \in \mathbb{Z}_p^* : [g^x]_i = [g^{ab}]_i\}$. A consequence of assuming our hypothesis is correct for i is that an adversary cannot distinguish between (g^a, g^b, g^{ab}) and (g^a, g^b, g^c) with non-negligible probability. If our assumption were false, the adversary would have an advantage in this game with probability $> \frac{1}{2^{n-i}}$. We will use this result moving forward.
Show for $\bar{\lambda} = i + 1$. For $a, b \xleftarrow{\$} \mathbb{Z}_p^*, c \xleftarrow{\$} \{x \in \mathbb{Z}_p^* : [g^x]_i = [g^{ab}]_i\}$, we have that $[g^c]_{i+1} \neq [g^{ab}]_{i+1}$ with probability $\frac{1}{2}$. In other words, the bit which the adversary is trying to predict is going to be different from g^{ab} in g^c with probability $\frac{1}{2}$. Suppose for a contradiction that the adversary has an non-negligible advantage in distinguishing (g^a, g^b, g^{ab}) from (g^a, g^b, g^c). This would mean an adversary has an advantage in winning the variant of DDH in case i half the time, whenever $[g^c]_{i+1} \neq [g^{ab}]_{i+1}$. This makes the assumption in case i false. Equivalently, the adversary would have an advantage in winning DDH with probability $\frac{1}{2^i}$. Therefore by contradiction, an adversary has no significant advantage calculating another bit of g^{ab} over guessing. Overall, the adversary cannot compute the remaining bits g^{ab} with probability significantly greater than $\frac{1}{2^{n-i-1}}$

We conclude that knowledge of the first $\bar{\lambda}$ bits of g^{ab} in addition to (g^a, g^b) does not help the adversary compute the remaining bits of g^{ab} with probability higher than $\frac{1}{2^{n-\bar{\lambda}}}$ when p is a Mersenne prime.