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

Ela Berners-Lee [†]

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 re-encryption 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 who encrypted the message (initiated a 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

[†]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)

a cloud service provider. Proxy Re-Encryption (PRE) [5] 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 is *email forwarding* [2, 3, 5, 7, 8, 18], where Alice wants to forward her emails to Bob and have him read them on her behalf, without revealing her secret key. With PRE, she can generate an update token which the email server uses to re-encrypt ciphertexts under Alice's key to ciphertexts under Bob's key, without the server reading her emails.

Another motivation, which we focus on in this paper, is enforcing cryptographic access control over remotely stored files [3, 17]. If data is given a classification level, and keys are shared with users according to 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, revocation or key expiry.

In this paper, 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 given having seen a number of valid tokens, and a stronger notion of unidirectionality which considers reversal attacks.

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).

Token robustness. Existing work [11] 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 previously unused key, 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 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 define *best-achievable unidirectionality*. Overall, our security model covers a wider range of attacks than prior definitions. We show in Appendix B that these definitions can be met by a simple adaptation of ElGamal-based PRE.

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* where the encryptor / re-encryption initiator's identity 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 Appendix C.

The structure of this paper is as follows: In Sections 2 and 3 we formally discuss existing work and current notions of security for PRE. We define *token robustness* in Section 4 and *maximal irreversibility* in Section 5. We then build on these to define *best-achievable unidirectionality*. In Section 6 we define the requirements for a PRE scheme to be secure for a malicious server and define *token robustness*. In Section 5 we critique existing notions of unidirectionality, and present the first security definition for that considers reversibility. A formal security definition for unidirectionality as it is currently considers we include in the full version [4]. We then use this to create a stronger definition for *best-achievable unidirectionality*. In Section 7 we define ciphertext origin authentication to provide authenticated PRE and discuss how to achieve this.

2 Preliminaries

In this work we only consider an IND-CPA schemes (see Appendix A) with a randomised encryption algorithm.

Definition 1 (multi-hop PRE scheme). *A* (multi-hop) proxy re-encryption (PRE) scheme *consists of the following algorithms:*

- $\mathsf{Setup}(1^\lambda) \rightarrow$ *param: Takes the security parameter* 1^λ *and outputs the set of parameters param, including a description of the message space* $\mathcal{M}$ *and token space* $\mathcal{D}$*. We note that param is an input for all subsequent algorithms but we leave it out for compactness of notation.*
- $\mathsf{KeyGen}(1^\lambda) \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$*: Generates a public-private key pair.*
- $\mathsf{Enc}(\mathsf{pk}, m) \xrightarrow{\$} C$*: Given a public key* pk *and message* $m \in \mathcal{M}$*, returning 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* $\perp$*.*
- $\mathsf{ReKeyGen}(\mathsf{sk}_1, \mathsf{pk}_2) \rightarrow \Delta_{1,2}$*: For two keypairs* $(\mathsf{pk}_1, \mathsf{sk}_1), (\mathsf{pk}_2, \mathsf{sk}_2)$*, outputs an update token* $\Delta_{1,2}$*.*
- $\mathsf{ReEnc}(\Delta_{1,2}, C_1) \rightarrow C_2$*: Takes a ciphertext* C_1 *and a token* $\Delta_{1,2}$ *and translates the ciphertext to output a new ciphertext* C_2*.*

A PRE scheme is correct *if for every message* $m \in \mathcal{M}$*, any sequence of* κ *keypairs* $(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_\kappa, \mathsf{sk}_\kappa) \leftarrow \mathsf{KeyGen}(1^\lambda)$*, all ciphertexts* $C_1 \leftarrow \mathsf{Enc}(\mathsf{pk}_1, m)$*, and all transformed ciphertexts* $\{C_i \leftarrow \mathsf{ReEnc}(\mathsf{ReKeyGen}(\mathsf{sk}_{i-1}, \mathsf{pk}_i), C_{i-1})\}_{i=1}^{\kappa}$*:*

$$\Pr[\mathsf{Dec}(\mathsf{sk}_i, C_i)) \neq m] \leq \mathsf{negl}(\lambda)\,,$$

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

In other words, 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.

2.1 Additional 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: 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 and we write $\Delta_{i \leftrightarrow j}$. Otherwise, it is *unidirectional* and we write $\Delta_{i \rightarrow j}$. We reserve the notation $\Delta_{i,j}$ for the general case where directionality is not specified.

Single/Multi-hop: Some PRE schemes are *single-hop* meaning ciphertexts can only be re-encrypted once. In contrast a *multi-hop* scheme can 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 assume multi-hop as a necessary requirement of a PRE scheme in the remainder of this work.

Ciphertext dependence: Informally $\mathsf{ReKeyGen}$ takes some additional information as input about the ciphertext that is to be re-encrypted under a new key. Let $\mathsf{ReKeyGen}$ in a PRE scheme be redefined to take additional information $\tilde{C}$ about ciphertext C as input: $\mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j, \tilde{C}) \rightarrow \Delta_{i,j,C}$. This PRE scheme is *ciphertext dependent* if for all $C_i^1 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_i, m_1)$ and $C_i^2 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_i, m_2)$ such that $C_i^1 \neq C_i^2$, and all re-encryption tokens $\Delta_{i,j,C_i^1} \xleftarrow{\$} \mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j, \tilde{C}_i^1)$, then:

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}_j, \mathsf{ReEnc}(\Delta_{i,j,C_i^1}, C_i^2)) \neq \mathsf{Dec}(\mathsf{sk}_i, C_i^2)\right] \leq 1 - \mathsf{negl}(\lambda)\,. \tag{1}$$

In existing work [11] and in our scheme, $\tilde{C}$ is the header of the ciphertext. For simplicity we will assume this is the case in 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. In subsequent definitions we will assume ciphertext dependence, but we note that these definitions can easily be extended to ciphertext independent schemes by setting $\tilde{C} = \emptyset$.

2.2 Existing Work

PRE was first introduced in [5]. The common approach to PRE in practice is the key encapsulation approach, where the ciphertext header contains the data encryption key k_D, encrypted with a key encryption key k_K. Typically, such schemes perform re-encryption by replacing k_K — so the ciphertext header now contains the same k_D encrypted with the new key encryption key k'_K and the body of the ciphertext remains unchanged:

$$C = ([k_D]_{k_K}, [m]_{k_D}) \xrightarrow{\mathsf{ReEnc}(\Delta, C)} C' = ([k_D]_{k'_K}, [m]_{k_D})$$

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 user can simply retain k_D and be able to decrypt the message, regardless of how many times it is re-encrypted.

Other indistinguishability notions for PRE are described in [8] and [11]. A similar definition for CCA-security is used in [3, 7, 16]. Existing notions which imply complete re-randomisation for public-key PRE include *unlinkability* in [7], and for symmetric key PRE include *ciphertext independence*[1] in [6] and UP-REENC security in [11].

Ciphertext dependence was first introduced in [11] 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 in [13], but they do not view directionality as a security definition, rather a classification of PRE schemes. They therefore define unidirectional PRE schemes and bidirectional PRE schemes as opposed to defining directionality separately. Furthermore the definition of unidirectionality in [13] assumes that a unidirectional scheme is single-hop, which is not necessarily the case. Other more recent work which informally describes unidirectionality say no PPT algorithm can out output a token that can re-encrypt to the old key [11], whereas other works [17] say 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 and, as such, there exist a number of PRE schemes which are unidirectional and single-hop [8, 16, 18]. They achieve unidirectionality by having two distinct levels of ciphertext which have different formats — level

[1] We reserve this terminology for PRE schemes for which token generation is not specific to a given ciphertext as in Section 2.1.

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 and therefore has limited practical application. Furthermore it does not convey how easy it is for a malicious server to reverse the re-encryption process. We discuss this in Section 5.

In many multi-hop schemes the number of re-encryptions is fixed, and the size of the ciphertext grows linearly with each re-encryption [7, 14]. The related problem of multi-hop unidirectional proxy re-signatures has a solution given in [15], however the message must be provided with the signature and thus such a scheme cannot be easily adapted to re-encryption. The first PRE scheme which is both unidirectional and multi-hop was given in [17], 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 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 call pres-IND-CPA (more details are given in Appendix A). This means that 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 2 addresses this, based on UP-REENC in [11] adapted to the public-key setting. Definition 2 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 $\mathsf{LR\text{-}ReEnc}$ oracle. For stronger security, the adversary has access to a token generation oracle. $\mathsf{OReKeyGen}$. The adversary is also 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 compromised key.

Definition 2. *A PRE scheme $\mathcal{PRE}$ is* re-encryption indistinguishable against chosen plaintext attack *(ReEnc-IND-CPA) if for all* PPT *adversaries $\mathcal{A}$ there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that:*

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

where $\mathsf{ReEnc\text{-}IND\text{-}CPA}$ *is given in Figure 1.*

This definition can be easily extended to symmetric PRE by providing the adversary with encryption oracles for both keys, see [11]. If $\mathsf{ReKeyGen}$ is deterministic, the adversary can win by calling $\mathsf{OReKeyGen}(i, j, C_0)$ to obtain Δ,

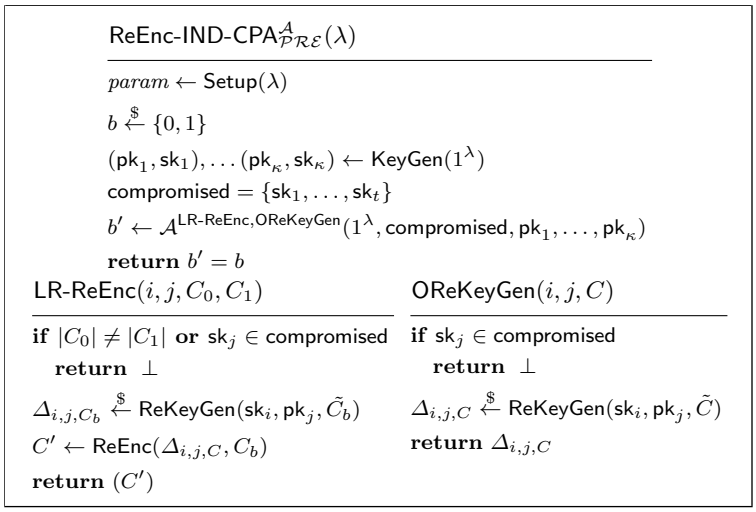

Figure 1: ReEnc-IND-CPA game. Schemes that meet this definition must fully re-randomise a ciphertext upon re-encryption.

compute $\mathsf{ReEnc}(\Delta, C_1)$ and compare this to $\mathsf{LR\text{-}ReEnc}(i, j, C_0, C_1)$. Since we consider re-randomisation a necessary property, from now on we will assume that ReKeyGen is randomised.

4 Token Robustness

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 in Section 5. Before we define token robustness, we need to define token validity.

Definition 3 (Token validity). *Let $\delta \in \mathcal{D}$. We say that δ is a* valid update token *if there exist keys $(\mathsf{pk}_1, \mathsf{sk}_1), (\mathsf{pk}_1, \mathsf{sk}_2) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and a ciphertext $C \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_1, m)$ for some $m \in \mathcal{M}$ such that:*

$$\Pr[\mathsf{Dec}(\mathsf{sk}_2, \mathsf{ReEnc}(\delta, C)) \neq m] \leq \mathsf{negl}(\lambda)\,. \tag{3}$$

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

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

where Tok-Rob *is given in Figure 2.*

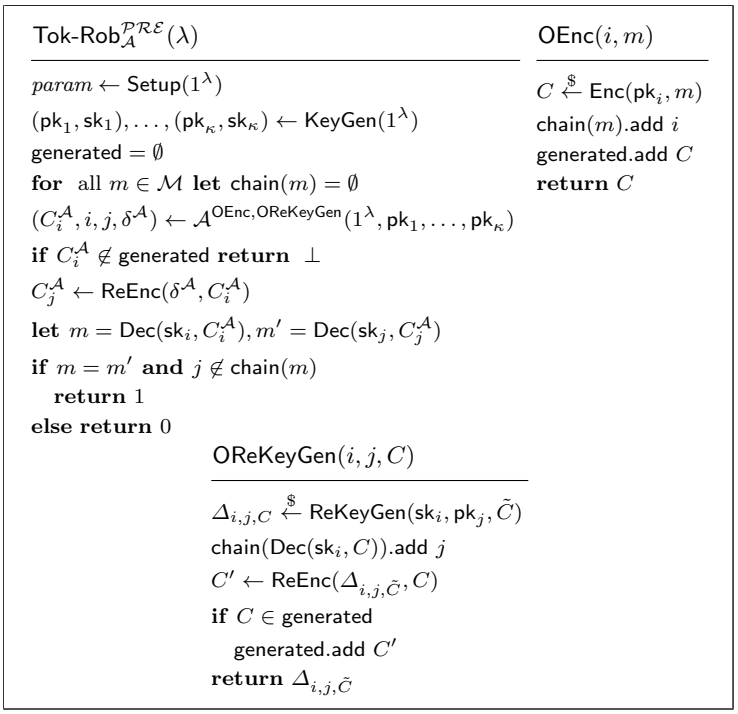

Figure 2: The token robustness game Tok-Rob

In the Tok-Rob game, the adversary has access to a token generation oracle OReKeyGen and an encryption oracle OEnc, and attempts to output a valid token $\delta^{\mathcal{A}}$ which re-encrypts a target ciphertext $C_i^{\mathcal{A}}$ from being under pk_i to being under pk_j. The list chain records which messages have been encrypted under which keys by adding the appropriate keys to $\mathsf{chain}(m)$ whenever OEnc or OReKeyGen are called. This means the adversary cannot trivially win by submitting a token which was the output of an OReKeyGen query. Another condition is the adversarys target ciphertext $C_i^{\mathcal{A}}$ must be an output of the OEnc oracle (or a re-encryption of such a ciphertext) to ensure that the adversary has no additional advantage from storing information created when encrypting the ciphertext. For example, in our scheme in Appendix B, 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 function generated is used to keep track of these 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 Tok-Rob game, let $C_1 \leftarrow \mathsf{OEnc}(i, m_1), C_2 \leftarrow \mathsf{OEnc}(i, m_2)$. Then the adversary can submit (i, j, C_1)

to $\mathsf{OReKeyGen}$ to obtain $\Delta_{i,j}$ 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 $j \neq \mathsf{chain}(m_2)$, the adversary wins the game with probability 1. □

However, ciphertext dependence alone does not imply token robustness. For example, suppose that $\mathsf{ReKeyGen}(\mathsf{pk}_i, \mathsf{sk}_j, \tilde{C}_1)$ outputs $\Delta_{i,j,C_1} = (\Delta_0 = \tilde{C}_1, \Delta_1)$, 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 see that token robustness is a stronger notion than ciphertext dependence.

5 Directionality revisited

Recall that the existing definition of unidirectionality states that an update token $\Delta_{i \to j}$ 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 not a security definition and does not consider reversal attacks where the adversary may have retained information on the old ciphertext. It also does not couple well with ciphertext dependence or token robustness. In this section we will elaborate on this claim before offering a security definition which covers reversal attacks, which we call $\bar{\lambda}$*-reversibility.*

5.1 Problems with traditional directionality

Unidirectionality is required in a number of applications for security reasons, despite the fact that it is not currently defined as a security property. In the full version [], we give a security definition for the current understanding of unidirectionality. However, this notion does not consider reversal attacks where the server has the new ciphertext and the information used to perform the re-encryption. However this is a possible means of a malicious server performing an unauthorised re-encryption and so warrants further consideration. Whilst this is not of any concern for email forwarding, it is an important consideration for access control as reversing a re-encryption can mean regranting access to a revoked user. Particularly for ciphertext dependent schemes where update tokens are randomised, reversal attacks become more significant as each re-encryption token should only be able to reverse one specific ciphertext. Therefore existing notions of unidirectionality do not couple well with ciphertext dependence.

There are schemes for which 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. We provide more explicit examples in Section 5.3. 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 (which they are presumed to retain in existing notions of unidirectionality). This shows that unidirectional schemes do not prevent an adversary from reversing a

re-encryption. For practical reasons, header values and update tokens are often designed to be small, and thus they 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 when adversarial sucess may be greater when considering other information that is already available to attackers, particularly in the case of a malicious server.

5.2 Directionality reconsidered

Now that we have argued that current notions of unidirectionality are not suitable for access control, we present a security defintion for reversal attacks (a security definition for unidirectionality is given in the full version [4]). Before we define this, we explain the key principles behind the motivation of the definition.
Principle 1: *Malicious storage cannot be prevented.* It is impossible for one part 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 needed 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 [10].

This definition is important in 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, which cannot be relied upon. 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.

We now define a reversal attack game which takes into account the amount of information the 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. All three adversaries receive the security parameter λ, storage parameter $\bar{\lambda}$, public keys and system parameters as input. The first stage adversary $\mathcal{A}_0$ receives a randomly chosen message and decides which keys should be used for encryption and re-encryption. The second stage adversary $\mathcal{A}_1$ receives the ciphertext and update token 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 $\mathcal{A}_2$. The final adversary $\mathcal{A}_2$ receives the re-encrypted ciphertext C' and state $st_{\mathcal{A}}$, and uses this to try to output a ciphertext which is an encryption of the same message under the original key.

$\mathsf{Rev\text{-}ReEnc}^{\mathcal{PRE}}_{\mathcal{A},\bar{\lambda}}(\lambda)$

$param \leftarrow \mathsf{Setup}(\lambda)$

$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots (\mathsf{pk}_\kappa, \mathsf{sk}_\kappa) \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda)$

$m \xleftarrow{\$} \mathcal{M}$

$(i, j) \leftarrow \mathcal{A}_0(1^\lambda, 1^{\bar{\lambda}}, \mathsf{pk}_1, \ldots \mathsf{pk}_\kappa, m)$

$C \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_i, m)$

$\Delta_{i,j,C} \xleftarrow{\$} \mathsf{ReKeyGen}(\mathsf{sk}_i, \mathsf{pk}_j, \tilde{C})$

$st_{\mathcal{A}} \leftarrow \mathcal{A}_1(1^\lambda, 1^{\bar{\lambda}}, \mathsf{pk}_1, \ldots \mathsf{pk}_\kappa, \Delta, C)$

if $|st_{\mathcal{A}}| > \bar{\lambda}$ **return** $\bot$

$C' \leftarrow \mathsf{ReEnc}(\Delta_{i,j,C}, C)$

$C_{\mathcal{A}} \leftarrow \mathcal{A}_2(1^\lambda, 1^{\bar{\lambda}}, \mathsf{pk}_1, \ldots \mathsf{pk}_\kappa, st_{\mathcal{A}}, C')$

return $\mathsf{Dec}(\mathsf{sk}_i, C_{\mathcal{A}}) = \mathsf{Dec}(\mathsf{sk}_j, C')$

Figure 3: The reversal game Rev-ReEnc.

This adversary never receives the message, original ciphertext or the update token — they only receive the information retained by $\mathcal{A}_1$. Note that this does not need to be the original ciphertext, it can be any ciphertext equivalent to the original. 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).

Definition 5. *Given a PRE scheme $\mathcal{PRE}$, let s be the size of the components in ciphertexts as estabilished by the scheme and let c be the number of ciphertext components updated by* ReEnc*. Then for $\bar{\lambda} \in \{0, s, \ldots, cs\}$, 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 3 as:*

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

where $\frac{1}{2^{cs-\bar{\lambda}}}$ is the probability that an adversary who has retained $\bar{\lambda}$ bits of (the updatable components 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:*

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

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

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 hindering this.

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 6. *A PRE scheme is* maximally irreversible *if it is cs-reversible, where c is the number of the number of ciphertext components updated by* ReEnc *and s is the component size.*

Clearly, that *maximum irreversibility* is stronger than traditional notions of unidirectionality because it covers directionality attacks for ciphertext dependent schemes in addition to ciphertext independent schemes. In Section 6 we use this notion together with token robustness (which is stronger than ciphertext dependence) to form a definition for best-achievable unidirectionality.

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 $\bar{\lambda} = cs$ 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.

5.3 Existing schemes under the new definition

Some traditionally unidirectional schemes are $|\Delta|-$reversible. In both [11] and [6], 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 $st_{\mathcal{A}} = (\tilde{C}, \Delta_1^{-1})$. This is equivalent to $st_{\mathcal{A}} = \Delta^{-1}$. 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 reflects stronger security than traditional bidirectionality. In particular, because [11] 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 [7], $\mathsf{ReKeyGen}$ 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 [2]. Similarly, the original symmetric proxy re-encryption scheme [5] 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 malicious model. We discuss some conditions which apply to re-encryption generally, before explaining the stronger conditions specific to our setting. Clearly, *correctness* is a necessary property of all PRE schemes. 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, 5, 7, 8, 15].

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 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:

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.

[2] 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.

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 by having the randomness used to encrypt the message input to $\mathsf{ReKeyGen}$. We build on this existing technique [11] to create a token robust scheme. 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 7. *A PRE scheme is* best-achievable unidirectional *if it is both token robust and maximally irreversable.*

Token robustness implies that a token Δ_{i,j,C_1} cannot be used to re-encrypt a ciphertext $C_2 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}_j, m)$ where $C_2 \neq C_1$ (except with negligible probability), which covers the traditional notion of unidirectionality [4]. Coupled with maximal irreversability, this means that given a re-encrypted ciphertext C_2 under pk_j, the only way that the adversary can produce a ciphertext C_1 such that $\mathsf{Dec}(\mathsf{sk}_i, C_1) = \mathsf{Dec}(\mathsf{sk}_j, C_2)$ 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 Appendix B 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 CDH assumption.

7 Ciphertext Origin Authentication

We now revisit the traditional notion of data origin authentication and how this needs tweaking for PRE. Traditionally, data origin authentication is intended for settings where the party who created the message also encrypts it. However, for PRE this is not always the case. Particularly in applications where more than one party shares an encryption key, 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 Authentication with corrupted users

Many authenticated encryption schemes including Signcryption [19] implement data origin authentication by having a check during decryption which terminates the process if the check fails. Such a check is also made in [11] which to

our knowledge is the only ReEnc-IND-CPA scheme to provide authentication. However, if corrupted users are being considered then honest termination of a process is not guaranteed. Therefore compulsory COA 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.

7.2 Correctness upon verification

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 5 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:

1. $\mathsf{Sign}(\mathbf{x}, m) \rightarrow \sigma = (r, s)$: $r = g^k$ for $k \xleftarrow{\$} \mathbb{Z}_p$ and $s = (h(m) - \mathbf{x}r)k^{-1} \mod p - 1$, for hash function $h()$.
2. $\mathsf{Verify}(\mathbf{X}, \sigma, m) \rightarrow \{0, 1\}$: if $g^{h(m)} = \mathbf{X}^r \cdot r^s$ return 1, else return 0.

We can obtain a non-optional COA check by replacing g^y in our original scheme (Figure 5) with an ElGamal signature on y signed using the initiator's public key. We also adapt the Verify algorithm so that it derives a specific value a. We call the resulting algorithm VerRetrieve.

$\mathsf{sigKeyGen}(1^\lambda)$	$\mathsf{Sign}(\mathbf{x}, y)$	$\mathsf{VerRetrieve}(\mathbf{X}, \sigma)$
$\mathbf{x} \xleftarrow{\$} \mathbb{Z}_p^*$	$k \xleftarrow{\$} \mathbb{Z}_p, r = g^k$	$(r, s) = \sigma$
$\mathbf{X} = g^{\mathbf{x}}$	$s = (y - \mathbf{x}r)k^{-1} \mod p - 1$	$a = \mathbf{X}^r \cdot r^s$
$\mathsf{ssk} = \mathbf{x}, \mathsf{svk} = \mathbf{X}$	$\sigma = (r, s)$	**return** a
return $(\mathbf{x}, \mathbf{X})$	**return** σ	

Ciphertexts now have the form $C = (\sigma, m \cdot g^{xy})$. By the correctness of ElGamal signatures, VerRetrieve should return $a = g^y$, since $\mathbf{X}^r \cdot r^s = g^{\mathbf{x}r} \cdot g^{k(y - \mathbf{x}r)k^{-1}} = g^y$.

Since obtaining g^y is necessary for decryption, and this can only be learned by successfully computing the VerRetrieve operation described above using the encryptor's public key, so verification is not optional.

However, if the scheme is only adapted with the change outlined above then there is no confirmation that the obtained g^y was correct, as the decryptor has no means verifying the message. 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. See Appendix B for full details.

7.3 COA in other schemes

Our notion of COA is not restricted to our scheme. 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 [5], a similar adaptation can be made.

We now propose an extension to ReCrypt [11] (described in Appendix C). Whilst this scheme uses symmetric PRE for re-encryption, we believe that COA is still valid for symmetric encryption. In ReCrypt, the message integrity check is optional and therefore it would be arguable that an optional COA check suffices in their model. We describe a simple extension which creates an optional check here, and a non-optional check in Appendix C.

For a basic extension, the ciphertext should replace χ with $(\chi, \sigma = \mathsf{Sign}(\mathbf{x}, \chi))$ and decryption should include the step $\mathsf{Verify}(\mathbf{X}, \sigma, \chi)$ and only return m if this outputs 1. We note that this ReCrypt may be preferable to our scheme from Figure 5 depending on the application, as it permits the re-encryption of longer messages, with the caveat that it is not best-achievable unidirectional.

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 a PRE scheme needs 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 ciphertext origin authentication for authenticated PRE and discuss how to implement this. Schemes meeting these definitions are given in the appendices.

A useful extension of this work is to create 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 of outsourcing re-encryption. Developing a best-achievable unidirectional PRE scheme with small update tokens has similar challenges to white-box cryptography and obfuscation [12]. Another related challenge is to create a best-achievable unidirectional token robust symmetric PRE scheme. We leave these as open problem.

We also leave to future work creating a 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 [9] and creating 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 and helping me improve my writing. Also many thanks to those who gave up their time to proofread my work and taught me how to better

explain technical explanations, especially Christian Janson, Kenny Paterson, Martin Albrecht 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. Berners-Lee, E.: Improved security notions for proxy re-encryption to enforce access control. Cryptology ePrint Archive, Report 2017/824 (2017), http://eprint.iacr.org/2017/824
5. 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
6. 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
7. 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
8. 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
9. 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
10. 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

11. 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
12. 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
13. 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
14. 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
15. 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
16. 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
17. 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
18. 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
19. 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

A Common definitions for confidentiality in PRE

We note that since [7], the main notion for security in PRE schemes is against chosen-ciphertext attacks (CCA) as opposed to chosen-plaintext (CPA) attacks. However since the focus of this paper is to revisit definitions with respect to unauthorised re-encryptions, for simplicity we restrict security to CPA and leave IND-CCA to future work. We further note that since recent practical schemes which are both unidirectional and multi-hop also only focus on CPA security [17]

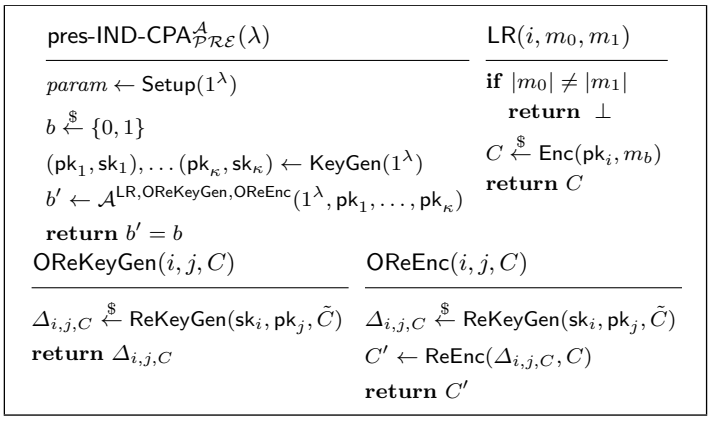

Figure 4: The pres-IND-CPA game which reflects the most common notion of indistinguishability for PRE.

we do not consider this to be a significant weakening of security in comparison with existing practical schemes.

The following definition is a formalism of the preservation of indistinguishability introduced in [7] adapted to CPA security. We note that this definition does not consider compromised keys.

Definition 8. *A PRE scheme* $\mathcal{PRE}$ preserves IND-CPA *if for all* PPT *algorithms* $\mathcal{A}$ *there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that:*

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

where pres-IND-CPA *is given in Figure 4.*

Informally, the PRE scheme is still IND-CPA secure 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 pres-IND-CPA.

Observe that the above definition applies whether or not the PRE scheme is ciphertext-dependent or unidirectional. It can be easily extended to symmetric PRE by providing the adversary with encryption oracles for both keys, see [11].

B 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. A multi-hop, ciphertext dependent scheme is given in Figure 5. We use our definitions in Definition 5 and Definition 4 to assess the unidirectionality and token robustness.

$\mathsf{Setup}(1^\lambda) \to param$	$\mathsf{KeyGen}(1^\lambda) \xrightarrow{\$} (X, x)$	$\mathsf{Enc}(X, m) \xrightarrow{\$} C$
p large prime	$x \xleftarrow{\$} \mathbb{Z}_p^*$	$y \xleftarrow{\$} \mathbb{Z}_p^*$
g a generator of $\mathbb{Z}_p$	$X = g^x$	$\tilde{C} = g^y$
$\mathcal{M} = \mathbb{Z}_p$	$\mathsf{pk} = X, \mathsf{sk} = x$	$\bar{C} = m \cdot X^y$
$\mathcal{D} = \mathcal{K} = \mathbb{Z}_p^*$	**return** $(\mathsf{pk}, \mathsf{sk})$	**return** $C = (\tilde{C}, \bar{C})$
return $param = (p, g, \mathcal{M}, \mathcal{K}, \mathcal{D})$		

$\mathsf{Dec}(x, C) \to m$	$\mathsf{ReKeyGen}(x_i, X_j, \tilde{C}) \xrightarrow{\$} \Delta_{i,j,C}$	$\mathsf{ReEnc}(\Delta_{i,j,C}, C) \to C'$
$m = \tilde{C}^{-x} \cdot \bar{C}$	$y' \xleftarrow{\$} \mathbb{Z}_p^*$	$C = (\tilde{C}, \bar{C})$
return m	$\Delta^0_{i,j,C} = g^{y'}$	$\tilde{C}' = \Delta^0_{i,j,C} = g^{y'}$
	$\Delta^1_{i,j,C} = X_j^{y'} \cdot \tilde{C}^{-x_i} = g^{x_j y' - x_i y}$	$\bar{C}' = \bar{C} \cdot \Delta^1_{i,j,C} = m \cdot X_j^{y'}$
	return $\Delta_{i,j,C} = (\Delta^0_{i,j,C}, \Delta^1_{i,j,C})$	**return** $C' = (\tilde{C}', \bar{C}')$

Figure 5: An ElGamal-based scheme similar to [5] which is best-achievable unidirectional and token robust.

Correctness: Let $C_i = (g^y, m \cdot g^{x_i y})$ be an encryption of m under g^{x_i}. The update token resulting from $\mathsf{ReEnc}(x_i, g^{x_j}, g^y)$ has the form $\Delta_{i,j,C} = (g^{y'}, X_j^{y'} \cdot (g^y)^{-x_i}) = (g^{y'}, g^{x_j y' - x_i y})$. Then re-encryption derives a ciphertext of the form $C_j = (g^{y'}, m \cdot g^{x_i y} \cdot g^{x_j y' - x_i y}) = (g^{y'}, m \cdot g^{x_j y'})$.

B.1 Security analysis

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

Theorem 2. *The scheme described in Figure 5 is ReEnc-IND-CPA under the decisional Diffie-Hellman assumption.*

Proof. In this scheme, a re-encrypted ciphertext has the same form as a ciphertext that was encrypted directly under the new key, so the domain of the re-encryption to the new key is equal to the domain of encryption 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. Therefore the problem reduces to ElGamal being IND-CPA, so we can assume the scheme is ReEnc-IND-CPA. □

To achieve best-achievable unidirectionality, we require that a new $\hat{y}$ is selected uniformly at random. Then to verify the received 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. The full adapted scheme is given in Figure 6.

Theorem 3. *The scheme in Figure 5 is best-achievable unidirectional.*

We prove this through two lemmas. For maximal irreversibility, observe that 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. However, we observe that retaining both values would require the same amount of storage as retaining the original ciphertext, and therefore is maximally irreversibile. Any reversal attacks are this obsolete. We prove this formally in Lemma 1.

Lemma 1. *The scheme described in Figure 5 is maximally irreversible under the Computational Diffie-Hellman (CDH) assumption.*

In this proof we assume that $\bar{\lambda} \in \{0, s, \dots cs\}$ where s is the size of components in the ciphertext and c is the number of components updated during re-encryption, as in Definition 5. 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). In Appendix D we provide a proof which shows the unidirectionality of the scheme when $\bar{\lambda} \in \{0, 1, \dots, cs\}$.

Proof. The CDH assumption states that, given (g^a, g^b), it is computationally infeasible to compute g^{ab}. The component size in this scheme is $s = p$, as we are working modulo p.

First we observe that $\bar{C}$ and $\Delta^1_{i,j,C}$ can be considered interchangeably as input to $st_{\mathcal{A}}$, as bits retained of $\Delta^1_{i,j,C}$ can be used with $\bar{C}'$ to calculate bits of $\bar{C}$. For simplicity, we only consider Δ_1 as a potential input to $st_{\mathcal{A}}$. There are only two values which can help reverse the re-encryption — the old ciphertext header $\tilde{C}$ and $\Delta^1_{i,j,C}$. 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 $\Delta^1_{i,j,C} = g^{x_i y - x_j y'}$ cannot derive $\tilde{C} = g^y$. 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 question becomes: given (a, g^b, g^d, g^{bd-ac}), find g^c. First we assume we have an oracle which solves the CDH problem: $\mathsf{OCDH}(g^a, g^b) \to g^{ab}$. In this case, given (a, g^b, g^d, g^{bd-ac}) we call $\mathsf{OCDH}(g^b, g^d) \to g^{db}$ and use this to retrieve $g^{ac} = (g^{bd-ac} \cdot g^{-bd})^{-1}$. We then calculate $g^c = (g^{ac})^{1/a}$.

For the other direction, assume we have an oracle $\mathsf{Og^c}(a, g^b, g^d, g^{bd-ac}) \to g^c$. We need to show that we can use this oracle to find g^{bd} given (g^b, g^d). First we observe that for all $x \in \mathbb{Z}_p^*$, there exists a c such that $g^x = g^{db-c} \mod p$. Therefore, calling $\mathsf{Og^c}(1, g^b, g^d, g^x)$ for some $x \in \mathbb{Z}_p^*$ returns g^c where $g^x = g^{db-c}$. We can then retrieve $g^{bd} = g^x \cdot g^c$. The problem statement is thus equivalent to the CDH problem. We conclude that an adversary who only retains $g^{x_i y - x_j y'}$ cannot derive g^y.

Analogously, an adversary who only retains g^y cannot calculate $g^{x_i y - x_j y'}$. The adversary must thus retain both g^y and $g^{x_i y - x_j y'}$. Hence, the number of components needed to reverse the re-encryption is equal to the number of

components transformed in the re-encryption process. We conclude the scheme is best-achievable unidirectional.

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 calculate $g^{x_j y'}$, which breaks the DDH assumption. □

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 2. *The scheme in Figure 5 has token robustness under the 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 which 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 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 $(g^{y'}, g^{x_j y' - x_i y})$, where $j \notin \mathsf{chain}(m)$. It may be possible that querying two completely different keys and a different ciphertext results in a token Δ, where $\Delta = g^{x_k y_k - x_l y_l} = g^{x_i y - x_j y'}$ for some y', but this only occurs with negligible probability.

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 $\mathsf{pk}_j^{y'} = g^{x_j y'}$. It remains for the adversary to calculate $g^{x_i y}$. Since the ciphertext is honestly generated ($\mathsf{generated}(C_i^{\mathcal{A}}) = \mathsf{true}$), the adversary does not know y and so cannot use this to compute $\delta^{\mathcal{A}} = X_j^{y'} \cdot (\mathsf{pk}_i^y)^{-1} = g^{x_j y' - x_i y}$. Since factoring is a difficult problem modulo p, g^{xy} cannot be easily separated from tokens of the form $\Delta_{i,k,C_i^{\mathcal{A}}} = g^{x_k y'' - x_i y}$ for some $x_k \leftarrow \mathsf{KeyGen}(1^\lambda), y'' \in \mathbb{Z}_p^*$. Finding the most common factor is also a difficult problem modulo p, so a string of tokens of this form also cannot be used to derive $g^{x_i y}$.

Alternatively, as each token output by $\mathsf{OReKeyGen}$ includes new randomisation y' which is not revealed to the adversary, these tokens blind the value $g^{x_i y}$ that the adversary must retrieve to win the game.

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.

An adapted version of this scheme which provides Ciphertext Origin Authentication (COA) is given in Figure 6.

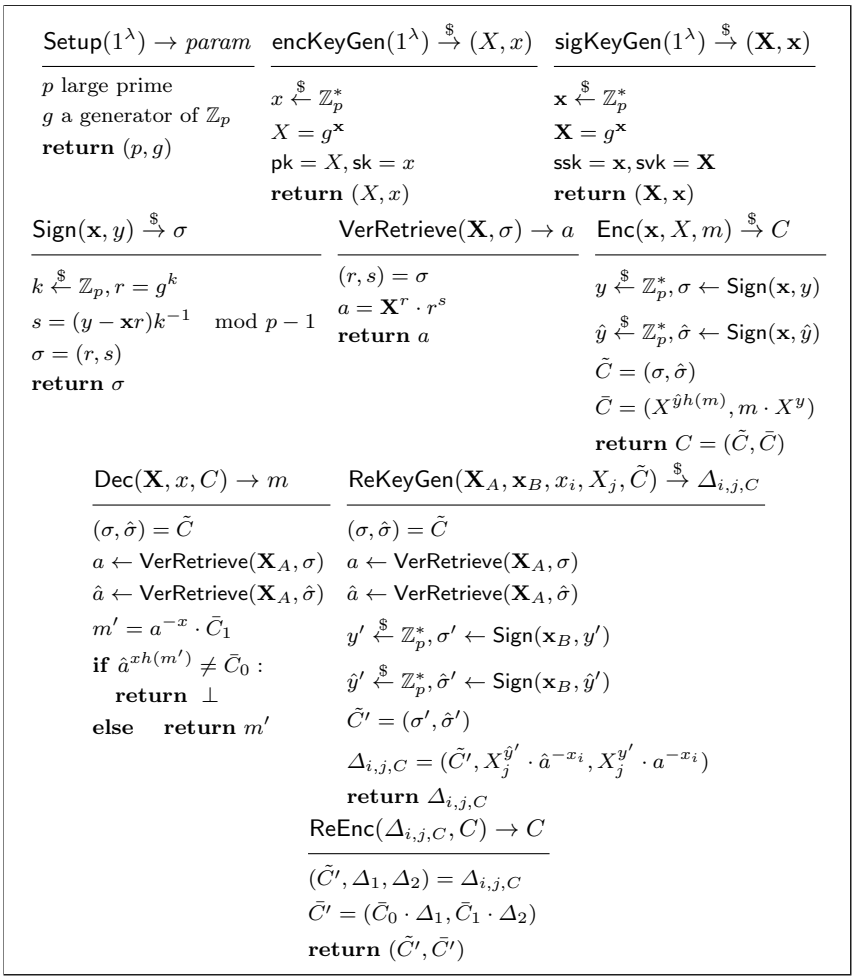

Figure 6: Adapted scheme with ciphertext origin authentication and message integrity.

C Non-optional COA extension for [11]

We give an extension to the ReCrypt scheme given in [11] which includes a compulsory COA check. A description of ReCrypt is given in Figure 7.

As with our scheme, we attach the identity of the encryptor / re-encryption initiator with the randomness used in the encryption. However, 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

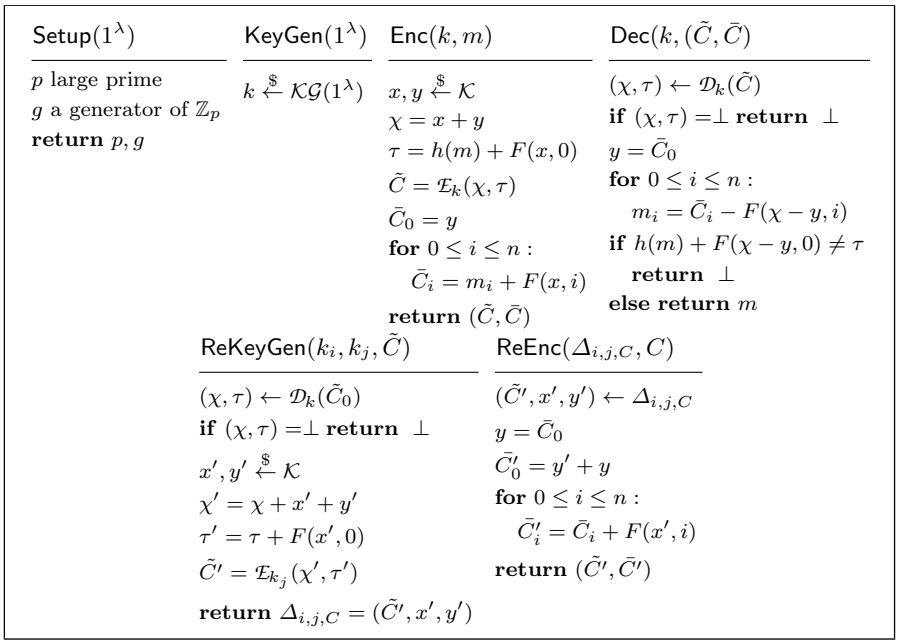

Figure 7: ReCrypt [11]

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 in generating update tokens. The full extension is given in Figure 8.

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

Recall that in our definition of unidirectionality Definition 5, the security parameter $\bar{\lambda}$ is determined in terms of components $\bar{\lambda} \in \{0, s, \ldots, cs\}$ where s is the component size and c is the number of ciphertext components updated during re-encryption. A more fine-grained definition would be to allow $\bar{\lambda}$ to be any number of bits $\bar{\lambda} \in \{0, 1, \ldots, cs\}$. We decided against this definition as whilst it be 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.

To demonstrate this as well as for general interest we include a proof here that shows that our scheme in Figure 5 is also best-achievable unidirectional in this stricter sense, as long as the prime p chosen by the Setup algorithm is a Mersenne prime.

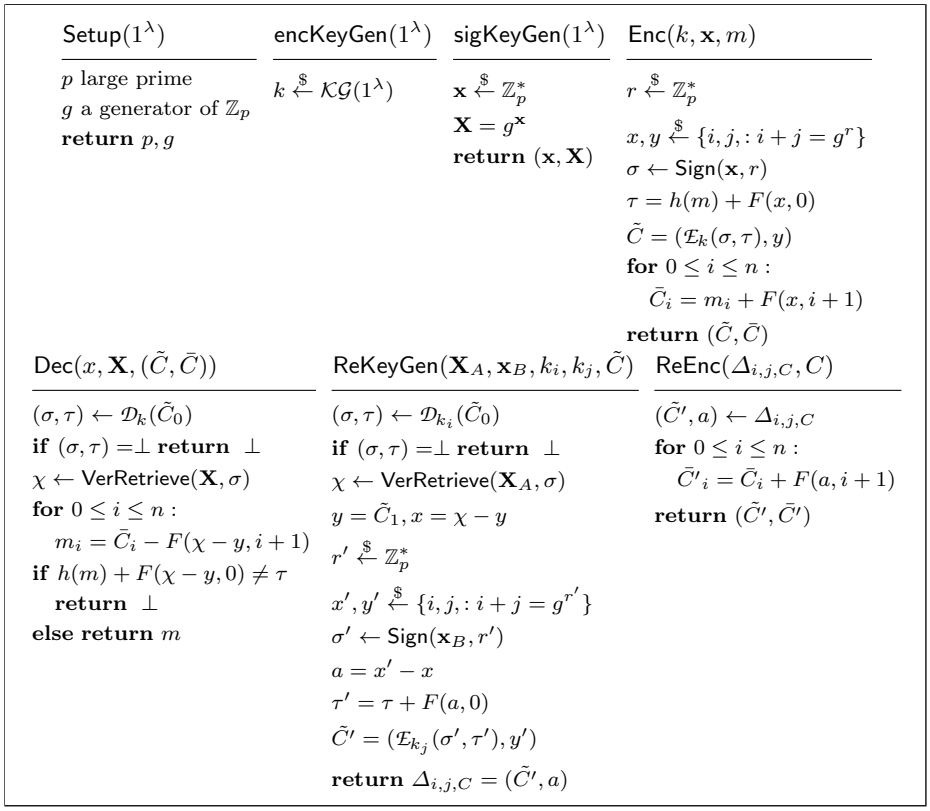

Figure 8: ReCrypt [11] with compulsory COA

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 1.

Lemma 3. *Let p be a Mersenne prime of length n, and let $a, b \xleftarrow{\$} \mathbb{Z}_p^*$. Then, for all* PPT *algorithms $\mathcal{B}$ there exists a negligible function* negl(λ) *such that:*

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

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 Computational Diffie-Hellman assumption (CDH), 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 $\frac{1}{2^i}$. 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.