Revisiting Proxy Re-Encryption: Forward Secrecy, Improved Security, and Applications

David Derler[1], Stephan Krenn[2], Thomas Lorünser[2], Sebastian Ramacher[1], Daniel Slamanig[2], and Christoph Striecks[2]

[1] IAIK, Graz University of Technology, Austria
[2] AIT Austrian Institute of Technology, Vienna, Austria
{firstname.lastname}@tugraz.at, {firstname.lastname}@ait.ac.at

Abstract. We revisit the notion of proxy re-encryption (PRE), an enhanced public-key encryption primitive envisioned by Blaze et al. (Eurocrypt'98) and formalized by Ateniese et al. (NDSS'05) for delegating decryption rights from a delegator to a delegatee using a semi-trusted proxy. PRE notably allows to craft re-encryption keys in order to equip the proxy with the power of transforming ciphertexts under a delegator's public key to ciphertexts under a delegatee's public key, while not learning anything about the underlying plaintexts.

We study an attractive cryptographic property for PRE, namely that of forward secrecy. In our forward-secret PRE (fs-PRE) definition, the proxy periodically evolves the re-encryption keys and permanently erases old versions while the delegator's public key is kept constant. As a consequence, ciphertexts for old periods are no longer re-encryptable and, in particular, cannot be decrypted anymore at the delegatee's end. Moreover, delegators evolve their secret keys too, and, thus, not even they can decrypt old ciphertexts once their key material from past periods has been deleted. This, as we will discuss, directly has application in short-term data/message-sharing scenarios.

Technically, we formalize fs-PRE. Thereby, we identify a subtle but significant gap in the well-established security model for conventional PRE and close it with our formalization (which we dub fs-PRE^+). We present the first provably secure and efficient constructions of fs-PRE as well as PRE (implied by the former) satisfying the strong fs-PRE^+ and PRE^+ notions, respectively. All our constructions are instantiable in the standard model under standard assumptions and our central building block are hierarchical identity-based encryption (HIBE) schemes that only need to be selectively secure.

Keywords: Forward secrecy, proxy re-encryption, improved security model

1 Introduction

The security of cryptosystems essentially relies on the secrecy of the respective secret key. For example, if for an encryption scheme a secret key is (accidentally) leaked, the confidentiality of all the data encrypted with respect to this key so far is immediately destroyed. One simple mitigation strategy for such a secret-key leakage is to frequently change secret keys such that leaking a secret key only affects a small amount of data. Implementing this in a naïve way, for instance in context of public-key encryption, means that one either has to securely and interactively distribute copies of new public keys frequently or to have huge public keys[3], which is rather inconvenient in practice. Consequently, cryptographic research focused on the design of cryptosystems that inherently provide such a property, being denoted as *forward secrecy* (or, *forward security*) [28]. The goal hereby is that key leakage at some point in time does not affect the data which was encrypted before the key leakage, while mitigating the drawbacks of the naïve solution discussed before. That is, one aims at efficient non-interactive solutions that have fixed sublinear-size public keys in the number of key switches/time periods. Those (strong) properties are the minimal requirements in the de-facto standard notion of forward secrecy in the cryptographic literature.

Within the last two decades, forward secrecy has been identified as an important property of various different cryptographic primitives such as digital signatures [6], identification schemes [1], public-key encryption [15], and private-key cryptography [7]. Only recently, another huge step forward has been made by Green and Miers [27] as well as Günther, Jager, Hale, and Lauer [29] to bring forward secrecy to important practical applications in the context of asynchronous messaging and zero round-trip time (0-RTT) key exchange. Given revelations and leaks about large-scale surveillance activities of security agencies within the last years, it is of utmost importance to further develop and deploy cryptosystems that inherently provide forward secrecy. We aim at advancing the research on forward secrecy with respect to other practically important public-key primitives, ideally, to ones with slightly more functionality.

Proxy re-encryption. Proxy re-encryption (PRE), envisoned by Blaze, Bleumer, and Strauss [9] and formalized by Ateniese, Fu, Green, and Hohenberger [4, 5], is a cryptographic primitive that can be seen as an extension of public-key encryption. A central feature of PRE is that senders can craft so-called re-encryption keys, which are usually created using only public information of the designated delegatee and the delegators' key material. Those re-encryption keys have the power to transform ciphertexts under a delegator's public key to ciphertexts under the delegatees' public keys. Within PRE, this transformation is done by a semi-trusted[4] proxy. The widely accepted model for PRE security (i.e., the conventional or plain PRE model) [4] requires that the proxy does not learn anything about the plaintexts which underlie the ciphertexts to be transformed.[5]

[3] With size $O(n)$ for n key switches/time periods.

[4] A semi-trusted proxy honestly follows the protocols, i.e., stores consistent re-encryption keys and re-encrypts correctly.

[5] The well-established security notions for PRE leave a potentially critical gap open. To look ahead, our proposed security model for *forward-secret* PRE closes this gap (implicitly also for plain PRE) and goes even beyond.

Proxy re-encryption is considered very useful in applications such as encrypted e-mail forwarding or access control in secure file systems, which was already discussed heavily in earlier work, e.g., in [4]. Furthermore, PRE has been object of significant research for almost two decades now, be it in a conventional setting [9, 4, 5], PRE with temporary delegation [4, 5, 34], identity-based PRE [26, 37], extensions to the chosen-ciphertext setting [16, 34], type-based/conditional PRE [39, 41], anonymous (or key-private) PRE [3], traceable PRE [32], or PRE from lattice-based assumptions [18, 36]. Generic constructions of PRE schemes from fully-homomorphic encryption [24] and from non-standard building blocks such as resplittable-threshold public key encryption as proposed in [30] are known, where different constructions of secure obfuscators for the re-encryption functionality have been given [31, 19, 18]. Despite PRE being an object of such significant research, forward-secret constructions remain unknown.[6]

On modeling forward-secret proxy re-encryption. Forward secrecy in the context of PRE is more complex than in standard public-key primitives, as PRE involves multiple different parties (i.e., delegator, proxy, and delegatees), where delegator and delegatees all have their own secret-key material and the proxy additionally holds all the re-encryption keys. One may observe that the proxy needs to be considered as a semi-trusted (central) party being always online, and, thus, it is reasonable to assume that this party is most valuable to attack. Consequently, we model forward secrecy in the sense that the re-encryption-key material can be evolved by the proxy to new periods while past-period re-encryption keys are securely erased. Hence, ciphertexts under the delegator's public key with respect to past-periods can no longer be re-encrypted. In addition, we model forward secrecy for the delegator's key material in a way that it is consistent with the evolution of the re-encryption material at the proxy.

For now, we do not consider forward secrecy at the delegatee, who can be seen as a passive party and does not need to take any further interaction with the delegator during the life-time of the system, except providing her public key once after set-up (e.g., via e-mail or public key server). It also does not have to be online when ciphertexts are re-encrypted for her by the proxy. Nevertheless, we leave it as a path for future research to cover the third dimension, i.e., model forward secrecy for the delegator and proxy as well as forward secrecy for the delegatee with efficient non-trivial constructions. However, it seems highly non-trivial to achieve efficient constructions that support forward secrecy for the delegatee additionally. In particular, we believe that the difficulty of achieving such strong type of forward secrecy is due to the circumstance that one has to carefully integrate three dimension of evolving key-material, one at the delegator, one at the proxy, and one at the delegatee. All dimensions seem to interfere

[6] We stress that we only aim at efficient non-trivial (non-interactive) forward-secret PRE constructions that have sublinear-size public and re-encryption keys in the number of time periods.

with each other.[7] As it will be confirmed by our application, covering the two dimensions already yields an interesting tool.

Moreover, to achieve forward secrecy for delegator and proxy key material, we face the following obstacles. First, it has to be guaranteed that the honest proxy must not be able to gain any information from the ciphertexts while at the same time being able to transform such ciphertexts *and* to update re-encryption key material consistently to newer time periods *without* any interaction with the delegator. Secondly, any delegatee *must not* be able to decrypt past-period ciphertexts. In this work, we give an affirmative answer to overcome those obstacles.

A practical application of forward-secret PRE. We believe that forward secrecy is an essential topic nowadays for any application. Also PRE is increasingly popular, be it in applied cryptographic literature [10, 14, 42, 36, 35], working groups such as the CFRG of the IRTF[8], large-scale EU-funded projects[9], and meanwhile also companies[10] that foster transition of such technologies into applications.

A practical application for forward-secret PRE is disappearing 1-to-n messaging. Here, a user encrypts a message under his public key and sends it to the proxy server that is responsible for distributing the encrypted messages to all pre-determined n receivers (note that receivers do not have to be online at the time the encrypted message is sent and an initial public-key exchange has to be done only in the beginning, but no more interactivity is needed). During setup time, the user has equipped the server with re-encryption keys (one for each receiver) while new keys can be added any time once a new receiver is present. Furthermore, the user does not need to manage a potentially huge list of public keys for each message to be sent. After a period, the data gets deleted by the proxy server, the re-encryption keys get evolved to a new period (without any interactions), and old-period re-encryption keys get deleted. The security of forward-secret PRE then guarantees that the proxy server does not learn the sensitive messages, neither can the two types of parties access disappeared messages later on. Once period-i re-encryption keys leak from the proxy server, only present and future encrypted messages (from period i onward) are compromised, while period-$(i-1)$ messages stay confidential. More generally, we believe that forward-secret PRE can be beneficially used in all kinds of settings that require access revocation, e.g., in outsourced encrypted data storage.

We also stress that within our forward-secret PRE instantiations, each user is only required to manage her own public and secret keys on her device and not a list of recipient public keys (or, identities). This deviates significantly from other primitives such as broadcast encryption (BE) [12, 22, 38], which could also be

[7] It is currently unknown to us how to solve the problem with efficient cryptographic tools, e.g., in the bilinear-maps setting. For efficiency reasons, multilinear maps and obfuscation are out of focus.

[8] https://www.ietf.org/id/draft-hallambaker-mesh-recrypt-00.txt

[9] https://credential.eu/

[10] e.g., http://www.nucypher.com, https://besafe.io/

suitable in such scenarios. However, practical BE schemes, e.g., [13], need large public keys and are computationally expensive.

1.1 Contribution

In this paper, we investigate forward secrecy in the field of proxy re-encryption (PRE) and term it fs-PRE. More precisely, our contributions are as follows:

- We first port the security model of PRE to the forward-secret setting (fs-PRE^-). Thereby, we observe a subtle but significant gap in existing (plain) security models for conventional PRE with regard to the granularity of delegations of decryption rights. In particular, existing models allow that a recipient, who has once decrypted a re-encrypted ciphertext, can potentially decrypt all re-encryptable ciphertexts of the same sender without further involvement of the proxy. In the forward-secret setting, it would essentially require to trust the delegatees to delete their re-encrypted ciphertexts whenever the period is switched, which is a problematic trust assumption.[11]
- We close this gap by introducing an additional security notion which inherently requires the involvement of a proxy in every re-encryption and in particular consider this notion in the forward-secret setting (fs-PRE^+). We also note that, when considering only a single time interval, this implicitly closes the aforementioned gap in the conventional PRE setting.[12] We also provide an explicit separation of the weaker fs-PRE^- notion (resembling existing PRE models) and our stronger notion fs-PRE^+.
- We then continue by constructing the first forward-secret PRE schemes (in the weaker as well as our stronger model) that are secure in the standard model under standard assumptions. On a technical side, only few approaches to forward secrecy are known. Exemplary, in the public-key-encryption (PKE) setting, we essentially have two ways to construct forward secrecy, i.e., the Canetti-Halevi-Katz (CHK) framework [15] from selectively secure hierarchical identity-based encryption (HIBE) [25] schemes and the more abstract puncturable-encryption (PE) approaches by [27, 29] (where both works either explicitly or implicitly use the CHK techniques). Particularly, we are not aware of any framework to achieve forward secrecy for PKE schemes based on "less-complex" primitives in comparison to selectively secure HIBE schemes. Consequently, we also base our constructions on selectively secure HIBE schemes [25], which we combine with linearly homomorphic encryption schemes, e.g., (linear) ElGamal.
- As a side result, we generalize the recent work of PE [27, 21, 17, 29] to what we call fully puncturable encryption (FuPE) in the full version of this paper and show how we can use FuPE to construct fs-PRE.

[11] Clearly, we still have to trust that the proxy deletes past-period re-encryption key material.

[12] In the conventional PRE setting, this gap was very recently independently addressed by Cohen [20].

1.2 Intuition and Construction Overview

To obtain more general results and potentially also more efficient instantiations, we use a relaxation of HIBEs denoted as binary-tree encryption (BTE) which was introduced by Canetti, Halevi, and Katz (CHK) in [15]. As an intermediate step, we introduce the notion of a forward-secret delegatable public-key encryption (fs-DPKE) scheme and present one instantiation which we obtain by combining the results of CHK with a suitable homomorphic public-key encryption (HPKE) scheme. Loosely speaking, a fs-DPKE scheme allows to delegate the decryption functionality of ciphertexts computed with respect to the public key of some user A to the public key of some other user B. Therefore, A provides a *public* delegation key to B. B then uses the delegation key *together* with the secret key corresponding to B's public key to decrypt any ciphertext that has been produced for A. A fs-DPKE scheme moreover incorporates forward secrecy in a sense that the originator A can evolve it's secret key and the scheme additionally allows to *publicly* evolve delegation keys accordingly. Interestingly, such a scheme is already sufficient to construct a fs-PRE$^-$-secure PRE scheme. Finally, we demonstrate how to strengthen this construction to a fs-PRE$^+$-secure PRE scheme, by solely relying on a certain type of key-homomorphism of the underlying fs-DPKE scheme. The intermediate step of introducing fs-DPKE is straightforward yet interesting, since we believe fs-DPKE is the "next natural step" to lift PKE to a setting which allows for controlled delegation of decryption rights.

Instantiation. In Table 1, we present an instantiation including the resulting key and ciphertext sizes. Thereby, we only look at fs-PRE instantiations that are fs-PRE$^+$-secure and note that the asymptotic sizes for fs-PRE$^-$-secure fs-PRE schemes are identical. For our instantiation, we use the BTE (or any selectively secure HIBE) from [15] and the linear encryption scheme from [11] as HPKE scheme under the Bilinear Decisional Diffie-Hellman (BDDH) and decision linear (DLIN) assumption respectively.

Building Blocks	$\lvert\mathsf{pk}\rvert$	$\lvert\mathsf{rk}^{(i)}\rvert$	$\lvert\mathsf{sk}^{(i)}\rvert$	$\lvert C\rvert$	Assumption
BTE [15], HPKE [11]	$\mathcal{O}(\log n)$	$\mathcal{O}((\log n)^2)$	$\mathcal{O}((\log n)^2)$	$\mathcal{O}(\log n)$	BDDH, DLIN

Table 1. Our fs-PRE$^+$-secure instantiation. All parameters additionally scale asymptotically in a security parameter k which is, hence, omitted. Legend: $n \ldots$ number of periods, $\lvert\mathsf{pk}\rvert \ldots$ public key size, $\lvert\mathsf{rk}^{(i)}\rvert \ldots$ size of re-encryption key for period i, $\lvert\mathsf{sk}^{(i)}\rvert \ldots$ size of secret key for period i, $\lvert C\rvert \ldots$ ciphertext size.

A note on a side result. Additionally, in the full version, we include the definition and a construction of a so called fully puncturable encryption (FuPE) scheme which is inspired by techniques known from HIBEs and the recent PE schemes in [27, 29]. We then show that FuPE schemes closely capture the essence which is required to construct fs-PRE$^+$-secure schemes by presenting a construction of a fs-PRE$^+$-secure PRE scheme from FuPE and HPKE.

1.3 Related Work and Outline

Work related to forward-secret PRE. Tang et al. [39, 41] introduced type-based/conditional PRE, which allows re-encryption of ciphertexts at the proxy only if a specific condition (e.g., a time period) is satisfied by the ciphertext. Furthermore, PRE with temporary delegations was proposed by Ateniese et al. [4, 5] and improved by Libert and Vernaud (LV) [34]. All those approaches yield a weak form of forward secrecy. Notably, the LV schemes provide fixed public parameters and non-interactivity with the delegatee as well. However, in contrast to our approach, LV and Tang et al. require at least to update the re-encryption keys for each time period with the help of the delegator (i.e., one message per time period from the delegator to the proxy) and also do not allow for exponentially many time periods, which do not suit our (stronger) forward-secret scenario.

Concurrent work on PRE. There is a considerable amount of very recent independent and concurrent work on different aspects of PRE and its applications [20, 8, 35, 23]. The works in [8, 35, 23] are only related in that they also deal with various aspects of PRE, but not fs-PRE. Those aspects are however unrelated to the work presented in this paper. In contrast, the work presented in [20] is related to one aspect of our work. It formalizes a security property for conventional PRE, which can be seen as a special case of our fs-PRE$^+$ notion which we introduce in context of fs-PRE. More precisely, our notion generalizes the notion of [20] and implies it if we fix the numbers of time periods to $n = 1$.

Outline. After discussing preliminaries in Section 2, we define fs-PRE in Section 3, discuss the gap in previous models and also briefly discuss its consequences to conventional PRE. We then give the first construction of a fs-PRE scheme from binary tree encryption in Section 4. We also show a separation result for the weaker fs-PRE$^-$ (resembling existing PRE models) and our stronger notion fs-PRE$^+$.

2 Preliminaries

For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$ and let $k \in \mathbb{N}$ be the security parameter. For an algorithm A, let $y \leftarrow A(1^k, x)$ be the process of running A, on input 1^k and x, with access to uniformly random coins and assigning the result to y. We assume that all algorithms take 1^k as input and we will sometimes not make this explicit in the following. To make the random coins r explicit, we write $A(1^k, x; r)$. An algorithm A is probabilistic polynomial time (PPT) if its running time is polynomially bounded in k. A function f is negligible if $\forall c \exists k_0 \forall k \geq k_0 : |f(k)| \leq 1/k^c$. For binary trees, we denote the root node with ε and all other nodes are encoded as binary strings, i.e., for a node w we denote child nodes as $w0$ and $w1$.

Homomorphic public-key encryption. A $\mathcal{F}$-homomorphic public key encryption (HPKE) scheme is a public-key encryption (PKE) scheme that is homomorphic with respect to a class of functions $\mathcal{F}$, i.e., given a sequence of ciphertexts to messages $(M_i)_{i \in [n]}$ one can evaluate a function $f : \mathcal{M}^n \rightarrow \mathcal{M} \in \mathcal{F}$ on the ciphertexts such that the resulting ciphertext decrypts to $f(M_1, \ldots, M_n)$.

Definition 1 **(($\mathcal{F}$-)HPKE).** *A $\mathcal{F}$-homomorphic public key encryption ($\mathcal{F}$-HPKE or HPKE for short) scheme with message space $\mathcal{M}$, ciphertext space $\mathcal{C}$ and a function family $\mathcal{F}$ consists of the PPT algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$:*

$\mathsf{Gen}(1^k)$: *On input security parameter k, outputs public and secret keys $(\mathsf{pk}, \mathsf{sk})$.*
$\mathsf{Enc}(\mathsf{pk}, M)$: *On input a public key pk, and a message $M \in \mathcal{M}$, outputs a ciphertext $C \in \mathcal{C}$.*
$\mathsf{Dec}(\mathsf{sk}, C)$: *On input a secret key sk, and ciphertext C, outputs $M \in \mathcal{M} \cup \{\perp\}$.*
$\mathsf{Eval}(f, (C_i)_{i\in[n]})$: *On input a function $f : \mathcal{M}^n \to \mathcal{M} \in \mathcal{F}$, a sequence of ciphertexts $(C_i)_{i\in[n]}$ encrypted under the same public key, outputs C.*

In addition to the standard and folklore correctness definition for public-key encryption (PKE), we further require for HPKE that for all security parameters $k \in \mathbb{N}$, all key pairs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^k)$, all functions $f : \mathcal{M}^n \to \mathcal{M} \in \mathcal{F}$, all message sequences $(M_i)_{i\in[n]}$ it holds that $\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(f, (\mathsf{Enc}(\mathsf{pk}, M_i))_{i\in[n]})) = f(M_1, \ldots, M_n)$. We are particularly interested in the case where $\mathcal{M}$ is a group and $\mathcal{F}$ is the set of all *linear functions* on products of $\mathcal{M}$. In that case, we call the HPKE scheme *linearly homomorphic*. For a HPKE, we require conventional IND-CPA security as with PKE schemes and recall an efficient instantiation of a linearly homomorphic scheme, i.e., linear ElGamal [11], in the full version.

Proxy re-encryption. Subsequently, we define proxy re-encryption and defer a formal treatment of security to Section 3.

Definition 2 **(PRE).** *A proxy re-encryption (PRE) scheme with message space $\mathcal{M}$ consists of the PPT algorithms $(\mathsf{Setup}, \mathsf{Gen}, \mathbf{Enc}, \mathbf{Dec}, \mathsf{ReGen}, \mathsf{ReEnc})$ where $\mathbf{Enc} = (\mathsf{Enc}^{(j)})_{j\in[2]}$ and $\mathbf{Dec} = (\mathsf{Dec}^{(j)})_{j\in[2]}$. For $j \in [2]$, they are defined as follows.*

$\mathsf{Setup}(1^k)$: *On input security parameter k, outputs public parameters pp.*
$\mathsf{Gen}(\mathsf{pp})$: *On input public parameters pp, outputs public and secret keys $(\mathsf{pk}, \mathsf{sk})$.*
$\mathsf{Enc}^{(j)}(\mathsf{pk}, M)$: *On input a public key pk, and a message $M \in \mathcal{M}$ outputs a level j ciphertext C.*
$\mathsf{Dec}^{(j)}(\mathsf{sk}, C)$: *On input a secret key sk, and level j ciphertext C, outputs $M \in \mathcal{M} \cup \{\perp\}$.*
$\mathsf{ReGen}(\mathsf{sk}_A, \mathsf{pk}_B)$: *On input a secret key sk_A and a public key pk_B for B, outputs a re-encryption $\mathsf{rk}_{A\to B}$.*
$\mathsf{ReEnc}(\mathsf{rk}_{A\to B}, C_A)$: *On input a re-encryption key $\mathsf{rk}_{A\to B}$, and a ciphertext C_A for user A, outputs a ciphertext C_B for user B.*

Binary tree encryption. Binary tree encryption (BTE) [15] is a relaxed version of hierarchical identity-based encryption (HIBE) [25]. Similar to a HIBE scheme, a BTE scheme has a (master) public key associated to a binary tree where each node in the tree has a corresponding secret key. To encrypt a message for some node, one uses both the public key and the name of the target node. Using the node's secret key, the resulting ciphertext can then be decrypted. Additionally, the secret key of a node can be used to derive the secret keys of its child nodes.

In contrast to BTE defined in [15], we make the part of the secret key used to perform the key derivation explicit, i.e., we will have secret keys for the decryption and derivation keys to derive secret keys. In case, an instantiation does not support a clear distinction, it is always possible to assume that the derivation key is empty and everything is contained in the secret key.

Definition 3. *A binary tree encryption (*BTE*) scheme with message space* $\mathcal{M}$ *consists of the PPT algorithms* $(\mathsf{Gen}, \mathsf{Evo}, \mathsf{Enc}, \mathsf{Dec})$ *as follows:*

$\mathsf{Gen}(1^k, \ell)$: *On input security parameter* k *and depth of the tree* ℓ*, outputs public, secret, and derivation keys* $(\mathsf{pk}, \mathsf{sk}^{(\varepsilon)}, \mathsf{dk}^{(\varepsilon)})$*.*

$\mathsf{Der}(\mathsf{sk}^{(w)}, \mathsf{dk}^{(w)})$: *On input secret key* $\mathsf{sk}^{(w)}$ *and derivation key* $\mathsf{dk}^{(w)}$*, for node* $w \in \{0,1\}^{<\ell}$*, outputs secret keys* $\mathsf{sk}^{(w0)}, \mathsf{sk}^{(w1)}$ *and derivation keys* $\mathsf{dk}^{(w0)}$*,* $\mathsf{dk}^{(w1)}$ *for the two children of* w*.*

$\mathsf{Enc}(\mathsf{pk}, M, w)$: *On input a public key* pk*, a message* $M \in \mathcal{M}$*, and node* $w \in \{0,1\}^{\leq \ell}$*, outputs a ciphertext* C*.*

$\mathsf{Dec}(\mathsf{sk}^{(w)}, C)$: *On input a secret key* $\mathsf{sk}^{(w)}$*, for node* $w \in \{0,1\}^{\leq \ell}$*, and ciphertext* C*, outputs* $M \in \mathcal{M} \cup \{\perp\}$*.*

For correctness, we require that for all security parameters $k \in \mathbb{N}$, all depths $\ell \in \mathbb{N}$, all key pairs $(\mathsf{pk}, (\mathsf{sk}^{(\varepsilon)}, \mathsf{ek}^{(\varepsilon)}))$ generated by $\mathsf{Gen}(1^k, \ell)$, any node $w \in \{0,1\}^{\leq \ell}$, any derived key $\mathsf{sk}^{(w)}$ derived using Der from $(\mathsf{sk}^{(\varepsilon)}, \mathsf{dk}^{(\varepsilon)})$, and all messages $M \in \mathcal{M}$, it holds that $\mathsf{Dec}(\mathsf{sk}^{(w)}, \mathsf{Enc}(\mathsf{pk}, M, w)) = M$.

The indistinguishability against selective node, chosen plaintext attacks (IND-SN-CPA) is a generalization of the standard IND-CPA security notion of PKE schemes. Essentially, the security notion requires the adversary to commit to the node to be attacked in advance. The adversary gets access to all secret keys except the secret keys for all nodes that are on the path from the root node to the targeted node.

```
Experiment $\mathsf{Exp}^{\mathsf{ind-sn-cpa}}_{\mathsf{BTE},A}(1^k, \ell)$
$(\mathsf{pk}, \mathsf{sk}^{(\varepsilon)}, \mathsf{dk}^{(\varepsilon)}) \leftarrow \mathsf{Gen}(1^k, \ell)$
$b \overset{R}{\leftarrow} \{0,1\}$
$(w^*, \mathsf{st}) \leftarrow A(1^k, \ell)$
Let $W$ be the set of all nodes that are siblings to the path from the root node to $w^*$
and (if possible) $w^*0$ and $w^*1$.
Compute $(\mathsf{sk}^{(w)}, \mathsf{dk}^{(w)})$ for all $w \in W$ from $(\mathsf{sk}^{(\varepsilon)}, \mathsf{dk}^{(\varepsilon)})$ using $\mathsf{Der}$.
$(M_0, M_1, \mathsf{st}) \leftarrow A(\mathsf{st}, \mathsf{pk}, (\mathsf{sk}^{(w)}, \mathsf{dk}^{(w)})_{w \in W})$
$b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}(\mathsf{pk}, M_b, w^*))$
if $b = b^*$ return 1, else return 0
```

Experiment 1. The IND-SN-CPA security experiment for a BTE scheme.

Definition 4 (IND-SN-CPA). *For a polynomially bounded function* ℓ*, a PPT adversary* A*, we define the advantage function in the sense of IND-SN-CPA as*

$$\mathsf{Adv}^{\mathsf{ind-sn-cpa}}_{\mathsf{BTE},A}(1^k, \ell(k)) = \left| \Pr\left[\mathsf{Exp}^{\mathsf{ind-sn-cpa}}_{\mathsf{BTE},A}(1^k, \ell(k)) = 1\right] - \frac{1}{2} \right| .$$

If for all ℓ*, and any* A *there exists a negligible function* ε *such that* $\mathsf{Adv}^{\mathsf{ind-sn-cpa}}_{\mathsf{BTE},A}($ $1^k, \ell(k)) < \varepsilon(k)$*, then a* BTE *scheme is IND-SN-CPA secure.*

The CHK Compiler. The technique of Canetti et al. [15] can be summarized as follows. To build a forward-secret PKE scheme with n periods, one uses a BTE of depth ℓ such that $n < 2^{\ell+1}$. Associate each period with a node of the tree and write w^i to denote the node for period i. The node for period 0 is the root node, i.e. $w^0 = \varepsilon$. If w^i is an internal node, then set $w^{i+1} = w^i0$. Otherwise, if w^i is a leaf node and $i < N-1$, then set $w^{i+1} = w'1$ where w' is the longest string such that $w'0$ is a prefix of w^i. The public key is simply the public key of the BTE scheme. The secret key for period i consists of the secret key for node w^i.

3 Security of (Forward-Secret) Proxy Re-Encryption

Proxy re-encryption (PRE) schemes can exhibit several important properties. In the following, we focus on the most common PRE properties in the cryptographic literature, i.e., uni-directionality (Alice is able to delegate decryption rights to Bob but not from Bob to Alice), non-interactivity (Alice can generate delegation key material without interacting with Bob), and collusion-safeness (even if Bob and other delegatees are colluding with the proxy, they cannot extract Alice' full secret key). Moreover, we consider PRE schemes that only allow a single hop, i.e., a ciphertext can be re-encrypted only a single time in contrast to multiple times in a row (multi-hop). Latter can be problematic due to unwanted transitivity.

In this work, we examine a further property of PRE schemes, namely the property of forward secrecy and propose the first uni-directional, non-interactive, collusion-safe, single hop, and forward-secret PRE scheme (dubbed fs-PRE) in the standard model from generic assumptions. Subsequently, in Section 3.1, we present the formal model for fs-PRE, while in Section 3.3 we discuss the relation and application of our stronger model to the conventional (i.e., plain) PRE security model.

3.1 Syntax of Forward-Secret Proxy Re-Encryption

To realize forward-secure PRE (fs-PRE), we lift the definitions and security models of uni-directional, single-hop, non-interactive, and collusion-safe PRE to a setting where we can have several periods. Thereby, we allow re-encryptions in every period such that re-encryption keys—in the same way as secret keys—are bound to a period. Furthermore, we align our PRE definitions with Ateniese et al. as well as Libert and Vergnaud [4, 5, 33] such that if we only have a single period, then they are equivalent to the definitions for plain PRE in [5, 33].[13]

Definition 5 (fs-PRE). *A forward-secure proxy re-encryption (fs-PRE) scheme with message space $\mathcal{M}$ consists of the PPT algorithms* $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Evo}, \mathbf{Enc}, \mathbf{Dec}, \mathsf{ReGen}, \mathsf{ReEvo}, \mathsf{ReEnc})$ *where* $\mathbf{Enc} = (\mathsf{Enc}^{(j)})_{j\in[2]}$ *and* $\mathbf{Dec} = (\mathsf{Dec}^{(j)})_{j\in[2]}$ *for levels $j \in [2]$. We denote level-2 ciphertext as re-encryptable ciphertexts, whereas level-1 ciphertexts are not re-encryptable.*

[13] Observe that for a single period, i.e., $n = 1$, Evo and ReEvo in Definition 5 are not defined. Dropping these algorithms and the corresponding evolution keys ek and rek yields a plain PRE scheme.

$\mathsf{Setup}(1^k)$: *On input security parameter k, outputs public parameters* pp.

$\mathsf{Gen}(\mathsf{pp}, n)$: *On input public parameters* pp, *and number of periods $n \in \mathbb{N}$, outputs public and secret keys* $(\mathsf{pk}, (\mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}))$.

$\mathsf{Evo}(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})$: *On input secret key* $\mathsf{sk}^{(i)}$ *and evolution key* $\mathsf{ek}^{(i)}$ *for period $i \in \{0, \ldots, n-2\}$, outputs a secret key* $\mathsf{sk}^{(i+1)}$ *and evolution key* $\mathsf{ek}^{(i+1)}$ *for period $i + 1$.*

$\mathsf{Enc}^{(j)}(\mathsf{pk}, M, i)$: *On input a public key* pk, *a message $M \in \mathcal{M}$, and period $i \in \{0, \ldots, n-1\}$, outputs a level-j ciphertext C.*

$\mathsf{Dec}^{(j)}(\mathsf{sk}^{(i)}, C)$: *On input a secret key* $\mathsf{sk}^{(i)}$, *for period $i \in \{0, \ldots, n-1\}$, and level-j ciphertext C, outputs $M \in \mathcal{M} \cup \{\perp\}$.*

$\mathsf{ReGen}(\mathsf{sk}_A^{(i)}, \mathsf{ek}_A^{(i)}, \mathsf{pk}_B)$: *On input a secret key* $\mathsf{sk}_A^{(i)}$ *and a evolution key* $\mathsf{ek}_A^{(i)}$ *(or $\perp$) for A and period $i \in \{0, \ldots, n-1\}$, and a public key* pk_B *for B, outputs a re-encryption* $\mathsf{rk}_{A \to B}^{(i)}$ *and re-encryption-evolution key* $\mathsf{rek}_{A \to B}^{(i)}$ *(or $\perp$).*

$\mathsf{ReEvo}(\mathsf{rk}_{A \to B}^{(i)}, \mathsf{rek}_{A \to B}^{(i)})$: *On input a re-encryption key* $\mathsf{rk}_{A \to B}^{(i)}$, *and a re-encryption-evolution key* $\mathsf{rek}_{A \to B}^{(i)}$ *for period $i \in \{0, \ldots, n-2\}$, outputs a re-encryption key* $\mathsf{rk}_{A \to B}^{(i+1)}$ *and re-encryption evolution key* $\mathsf{rek}_{A \to B}^{(i+1)}$ *for the period $i + 1$.*

$\mathsf{ReEnc}(\mathsf{rk}_{A \to B}^{(i)}, C_A)$: *On input a re-encryption key* $\mathsf{rk}_{A \to B}^{(i)}$, *and a (level-2) ciphertext C_A for user A, outputs a (level-1) ciphertext C_B for user B.*

Correctness. For correctness, we basically require on the one hand that every ciphertext encrypted for some period i can be decrypted with the respective secret key from period i. On the other hand—when also considering re-encryptable and re-encrypted ciphertexts—we require that level-2 ciphertexts encrypted for period i can be re-encrypted with a suitable re-encryption key for the same period and then decrypted using the (delegatee's) respective secret key for period i. More formally, for all security parameters $k \in \mathbb{N}$, all public parameters $\mathsf{pp} \leftarrow \mathsf{Setup}(1^k)$, any number of periods $n \in \mathbb{N}$ and users $U \in \mathbb{N}$, all key tuples $(\mathsf{pk}_u, \mathsf{sk}_u^{(0)}, \mathsf{ek}_u^{(0)})_{u \in [U]}$ generated by $\mathsf{Gen}(1^k, n)$, any period $i \in \{0, \ldots, n-1\}$, for any $u \in [U]$, any evolved key $\mathsf{sk}_u^{(i+1)}$ generated by $\mathsf{Evo}(\mathsf{sk}_u^{(i)})$, for all $u' \in [U], u \neq u'$, any (potentially evolved) re-encryption and re-encryption-evolution keys $\mathsf{rk}_{u \to u'}^{(i)}$ and $\mathsf{rek}_{u \to u'}^{(i)}$, respectively, for period i generated using ReGen from (potentially evolved) secret and evolution keys as well as the target public key, and all messages $M \in \mathcal{M}$, it holds that

$$\forall j \in [2]\ \exists j' \in [2] :\ \mathsf{Dec}^{(j')}(\mathsf{sk}_u^{(i)}, \mathsf{Enc}^{(j)}(\mathsf{pk}_u, M, i)) = M,$$
$$\mathsf{Dec}^{(1)}(\mathsf{sk}_{u'}^{(i)}, \mathsf{ReEnc}(\mathsf{rk}_{u \to u'}^{(i)}, \mathsf{Enc}^{(2)}(\mathsf{pk}_u, M, i))) = M.$$

3.2 Security of Forward-Secret Proxy Re-Encryption

The security notions for fs-PRE are heavily inspired by the security notions of (plain) PRE [4, 5, 33] and forward-secret PKE [15]. We will discuss multiple notions, combine them carefully, and introduce forward-secret indistinguishably under chosen-plaintext attacks for level-1 and level-2 ciphertexts (termed fs-IND-CPA-1 and fs-IND-CPA-2, respectively) which we argue to be reasonable notions in our setting. Additionally, we define a new (stronger) variant of indistinguishably-under-chosen-plaintext-attacks security for fs-PRE (dubbed fs-RIND-CPA) that focuses on malicious users in the face of honest proxies. In particular, the latter strengthen the folklore PRE security notion.

For all experiments defined in this section, the environment keeps initially empty lists of dishonest ($\mathtt{DU}$) and honest users ($\mathtt{HU}$). The oracles are defined as follows:

$\mathsf{Gen}^{(h)}(\mathsf{pp}, n)$: Run $(\mathsf{pk}, \mathsf{sk}, \mathsf{ek}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$, set $\mathtt{HU} \leftarrow \mathtt{HU} \cup \{(\mathsf{pk}, \mathsf{sk}, \mathsf{ek})\}$, and return pk.

$\mathsf{Gen}^{(d)}(\mathsf{pp}, n)$: Run $(\mathsf{pk}, \mathsf{sk}, \mathsf{ek}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$, set $\mathtt{DU} \leftarrow \mathtt{DU} \cup \{(\mathsf{pk}, \mathsf{sk}, \mathsf{ek})\}$, and return $(\mathsf{pk}, \mathsf{sk}, \mathsf{ek})$.

$\mathsf{ReGen}^{(h)}(j, \mathsf{pk}_u, \mathsf{pk})$: On input a period j, a public key pk_u and a public key pk, abort if $(\mathsf{pk}_u, \cdot, \cdot) \notin \mathtt{HU}$. Otherwise, look up $\mathsf{sk}_u^{(0)}$ and $\mathsf{ek}_u^{(0)}$ corresponding to pk_u from $\mathtt{HU}$. If $j > 0$ set $(\mathsf{sk}_u^{(i)}, \mathsf{ek}_u^{(i)}) \leftarrow \mathsf{Evo}(\mathsf{sk}_u^{(i-1)}, \mathsf{ek}_u^{(i-1)})$ for $i \in [j]$. Return $\mathsf{ReGen}(\mathsf{sk}_u^{(j)}, \mathsf{ek}_u^{(j)}, \mathsf{pk})$.

$\mathsf{ReGen}^{(h')}(j, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \mathsf{pk}_u)$: On input a period j, secret key $\mathsf{sk}^{(0)}$, evolution key $\mathsf{ek}^{(0)}$, and a public key pk_u, abort if $(\mathsf{pk}_u, \cdot, \cdot) \notin \mathtt{HU}$. Otherwise, if $j > 0$ set $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(i-1)}, \mathsf{ek}^{(i-1)})$ for $i \in [j]$. Return $\mathsf{ReGen}(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}, \mathsf{pk}_u)$.

$\mathsf{ReGen}^{(d)}(j, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \mathsf{pk}_d)$: On input a period j, secret key $\mathsf{sk}^{(0)}$, evolution key $\mathsf{ek}^{(0)}$, and a public key pk_d, abort if $(\mathsf{pk}_d, \cdot, \cdot) \notin \mathtt{DU}$. Otherwise, if $j > 0$ set $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(i-1)}, \mathsf{ek}^{(i-1)})$ for $i \in [j]$. Return $\mathsf{ReGen}(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}, \mathsf{pk}_d)$.

fs-IND-CPA-i security. We start with the definition of fs-IND-CPA-1 and fs-IND-CPA-2 security for fs-PRE. Inspired by the work on forward secrecy due to Canetti, Halevi, and Katz [15], our experiments lift standard PRE security notions as defined in Ateniese et al. [4] (AFGH) to the forward-secrecy setting. More concretely, after the selection of a target period j^* by the adversary A, A gets access to the secret and the evolution key $(\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$ of the target period j^*, while the challenge ciphertext for A-chosen message M_b is generated for period $j^* - 1$, for uniform $b \leftarrow \{0, 1\}$. Eventually, A outputs a guess on b. We say A is valid if A only outputs equal-length messages $|M_0| = |M_1|$ and $1 \leq j^* \leq n$.

Furthermore, we adapted the AFGH security experiment such that A has access to re-encryption and re-encryption-evolution keys for period $j^* - 1$. Analogously to previous work on PRE, we present two separate notions for level-1 and level-2 ciphertexts. The corresponding security experiments are given in Experiment 2 and Experiment 3. The only difference in Experiment 2 is that for level-1 ciphertexts, i.e., the ones which can no longer be re-encrypted, the adversary gets access to more re-encryption and re-encryption-evolution keys (obviously, the challenge ciphertext in that experiment is a level-1 ciphertext).

```
Experiment $\mathsf{Exp}^{\mathsf{fs-ind-cpa-1}}_{\mathsf{fs\text{-}PRE},A}(1^k, n)$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^k), (\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n), b \xleftarrow{R} \{0,1\}$
$(j^*, \mathsf{st}) \leftarrow A(\mathsf{pp}, n, \mathsf{pk})$
$(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(j-1)}, \mathsf{ek}^{(j-1)})$ for $j \in [j^*]$.
$\mathcal{O} \leftarrow \{\mathsf{Gen}^{(h)}, \mathsf{ReGen}^{(h)}(j^* - 1, \cdot, \mathsf{pk}), \mathsf{ReGen}^{(h')}(j^* - 1, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \cdot), \mathsf{Gen}^{(d)}, \mathsf{ReGen}^{(d)}(j^* - 1, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \cdot)\}$
$(M_0, M_1, \mathsf{st}) \leftarrow A^{\mathcal{O}}(\mathsf{st}, \mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$
$b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}^{(1)}(\mathsf{pk}, M_b, j^* - 1))$
if $b = b^*$ return 1, else return 0
```

Experiment 2. The fs-IND-CPA-1 security experiment for level-1 ciphertexts of fs-PRE schemes.

```
Experiment $\mathsf{Exp}^{\mathsf{fs-ind-cpa-2}}_{\mathsf{fs\text{-}PRE},A}(1^k, n)$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^k), (\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n), b \xleftarrow{R} \{0,1\}$
$(j^*, \mathsf{st}) \leftarrow A(\mathsf{pp}, n, \mathsf{pk})$
$(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(j-1)}, \mathsf{ek}^{(j-1)})$ for $j \in [j^*]$.
$\mathcal{O} \leftarrow \{\mathsf{Gen}^{(h)}, \mathsf{ReGen}^{(h)}(j^* - 1, \cdot, \mathsf{pk}), \mathsf{ReGen}^{(h')}(j^* - 1, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \cdot)\}$
$(M_0, M_1, \mathsf{st}) \leftarrow A^{\mathcal{O}}(\mathsf{st}, \mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$
$b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}^{(2)}(\mathsf{pk}, M_b, j^* - 1))$
if $b = b^*$ return 1, else return 0
```

Experiment 3. The fs-IND-CPA-2 security experiment for level-2 ciphertexts of fs-PRE schemes.

Definition 6 (fs-IND-CPA-i). *For a polynomially bounded function $n(\cdot) > 1$, a PPT adversary A, we define the advantage function for A in the sense of fs-IND-CPA-i for level-i ciphertexts as*

$$\mathsf{Adv}^{\mathsf{fs-ind-cpa-i}}_{fs-\mathsf{PRE},A}(1^k, n(k)) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{fs-ind-cpa-i}}_{fs-\mathsf{PRE},A}(1^k, n(k)) = 1 \right] - \frac{1}{2} \right| .$$

*A fs-*PRE *scheme is fs-IND-CPA-i secure if for all polynomially bounded $n(\cdot) > 1$ and any valid PPT A there exists a negligible function ε such that $\mathsf{Adv}^{\mathsf{fs-ind-cpa-i}}_{fs-\mathsf{PRE},A}($ $1^k, n(k)) < \varepsilon(k)$, where $\mathsf{Exp}^{\mathsf{fs-ind-cpa-i}}_{fs-\mathsf{PRE},A}$, for all $i \in [2]$, are defined in Experiment 2 and Experiment 3, respectively.*

Master-secret security. As discussed in [33], the security notion for level-1 (i.e., non re-encryptable) ciphertexts already implies classical master-secret security notion for PRE [4].[14] However, this must not be the case in the forward-secret setting. To formally close this gap, we give a trivial lemma (cf. Lemma 1) which states that fs-IND-CPA-1 implies master-secret security in the sense of Experiment 4 in the forward-secrecy setting. Essentially, master-secret security ensures collusion safeness such that re-encryption keys in period j do not leak the secret key corresponding to level-1 ciphertexts which can not be re-encrypted in period $j - 1$. In Experiment 4, we lift master-secret security in the classical PRE

[14] As we will discuss below, this notion seems to suggest false guarantees and leaves a critical gap in the security model open.

sense to the forward-secret setting. In the experiment, the adversary A selects an target period j^* and receives the secret and evolution keys $(\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$ for the target period in return. Within the experiment, A has access to several oracles, e.g., to obtain re-encryption and re-encryption-evolution keys for period j^*. Eventually, A outputs secret and evolutions keys $(\mathsf{sk}^*, \mathsf{ek}^*)$ and the experiment returns 1 (i.e., A wins) if $(\mathsf{sk}^*, \mathsf{ek}^*) = (\mathsf{sk}^{(j^*-1)}, \mathsf{ek}^{(j^*-1)})$. We say A is valid if A only outputs $1 \leq j^* \leq n$.

```
**Experiment** $\mathsf{Exp}^{\mathsf{fs-msk}}_{fs-\mathsf{PRE},A}(1^k, n)$
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^k), (\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$
$(j^*, \mathsf{st}) \leftarrow A(\mathsf{pp}, n, \mathsf{pk})$
$(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(j-1)}, \mathsf{ek}^{(j-1)})$ for $j \in [j^*]$.
$\mathcal{O} \leftarrow \{\mathsf{Gen}^{(h)}, \mathsf{ReGen}^{(h)}(j^*, \cdot, \mathsf{pk}), \mathsf{ReGen}^{(h')}(j^*, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \cdot), \mathsf{Gen}^{(d)}, \mathsf{ReGen}^{(d)}(j^*, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \cdot)\}$
$(\mathsf{sk}^*, \mathsf{ek}^*) \leftarrow A^{\mathcal{O}}(\mathsf{st}, \mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$
if $(\mathsf{sk}^*, \mathsf{ek}^*) = (\mathsf{sk}^{(j^*-1)}, \mathsf{ek}^{(j^*-1)})$ return 1, else return 0
```

Experiment 4. The forward secure master secret security experiment for fs-PRE schemes.

Definition 7 (fs-master-secret security). *For a polynomially bounded function $n(\cdot) > 1$ and a PPT adversary A, we define the advantage function for A in the sense of fs-master-secret security as*

$$\mathsf{Adv}^{\mathsf{fs-msk}}_{fs-\mathsf{PRE},A}(1^k, n(k)) := \Pr\left[\mathsf{Exp}^{\mathsf{fs-msk}}_{fs-\mathsf{PRE},A}(1^k, n(k)) = 1\right].$$

*A fs-*PRE *scheme is fs-master-secret secure if for all polynomially bounded $n(\cdot) > 1$ and any valid PPT A there exists a negligible function ε such that $\mathsf{Adv}^{\mathsf{fs-msk}}_{fs-\mathsf{PRE},A}(1^k, n(k)) < \varepsilon(k)$, where $\mathsf{Exp}^{\mathsf{fs-msk}}_{fs-\mathsf{PRE},A}$ is defined in Experiment 4.*

We now show that this notion in the sense of Definition 7 is trivially implied by fs-IND-CPA-1 security for fs-PRE in the sense of Definition 6.

Lemma 1. *If a fs-*PRE *scheme is fs-IND-CPA-1 secure in the sense of Definition 6, then the same fs-*PRE *scheme is fs-master-secret secure in the sense of Definition 7.*

Proof sketch. It is trivial to see that any successful PPT adversary on the fs-master-secret security of a fs-PRE scheme can be transformed into a PPT adversary on the fs-IND-CPA-1 security of that fs-PRE scheme. (Essentially, any PPT adversary that is able to gain access to the secret key of the prior period can trivially distinguish ciphertexts for the same period.)

The problem with (fs-)PRE security. A problem with the notion of standard (i.e., IND-CPA and master secret) security for (plain) PRE and also our fs-PRE notions so far is that the secret keys used for level-1 (i.e., non re-encryptable) and level-2 (i.e., re-encryptable) ciphertexts can be independent. Consequently, although ciphertexts on both levels can be shown to be indistinguishable, this does not rule out the possibility that ciphertexts on level-2 reveal the respective level-2 secret key of the sender to an legitimate receiver. This is exactly the reason for the gap in the plain PRE model which allows to leak a "level-2 secret key" once a re-encryption has been performed while all security properties are still satisfied (we provide an example for such a scheme in Section 4.4). In particular, this allows the receiver to potentially decrypt *any* level-2 ciphertext. We provide a solution in form of a stronger security notion which we term fs-RIND-CPA security in the following.

fs-RIND-CPA security. We observe that existing PRE notions only consider that (1) as long as the users are honest, the proxy learns nothing about any plaintext, and (2) if proxies and users collude they do not learn anything about the ciphertexts which are not intended to be re-encrypted. We go a step further and consider malicious users in the face of an honest proxy in the forward-secret and, hence, also in the plain PRE sense. That is, we want to enforce that a malicious user can only read the ciphertexts which were actually re-encrypted by the proxy and can not tell anything about the ciphertexts which can potentially be re-encrypted. We capture this via the notion of fs-RIND-CPA security. In this scenario, an adversary receives re-encrypted ciphertexts generated by an honest proxy, that it is able to decrypt. Nevertheless, for all other level-2 ciphertexts, the adversary should still be unable to recover the plaintext. In Experiment 5, we model this notion where the adversary gets access to a ReEnc-oracle which is in possession of the re-encryption key from the target user to the adversary. We say A is valid if A only outputs $1 \leq j^* \leq n$ and equal length messages $|M_0| = |M_1|$.

Experiment $\mathsf{Exp}^{\mathsf{fs-rind-cpa}}_{fs-\mathsf{PRE},A}(1^k, n)$

```
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^k), (\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n), b \xleftarrow{R} \{0,1\}$
$(j^*, \mathsf{pk}^*, \mathsf{st}) \leftarrow A(\mathsf{pp}, n, \mathsf{pk})$
$(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(j-1)}, \mathsf{ek}^{(j-1)})$ for $j \in [j^*]$
$\mathsf{rk} \leftarrow \mathsf{ReGen}(\mathsf{sk}^{(j^*)}, \perp, \mathsf{pk}^*)$
$(M_0, M_1, \mathsf{st}) \leftarrow A^{\{\mathsf{ReEnc}(\mathsf{rk},\cdot)\}}(\mathsf{st})$
$b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}^{(2)}(\mathsf{pk}, M_b, j^*))$
if $b = b^*$ return 1, else return 0
```

Experiment 5. The fs-RIND-CPA security experiment for fs-PRE schemes.

Definition 8 (fs-RIND-CPA). *For a polynomially bounded function $n(\cdot)$ and a PPT adversary A, we define the advantage function for A in the sense of fs-RIND-CPA as*

$$\mathsf{Adv}^{\mathsf{fs-rind-cpa}}_{fs-\mathsf{PRE},A}(1^k, n(k)) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{fs-rind-cpa}}_{fs-\mathsf{PRE},A}(1^k, n(k)) = 1\right] - \frac{1}{2} \right| .$$

A fs-PRE scheme is fs-RIND-CPA if for all polynomially bounded $n(\cdot)$ and any valid PPT A there exists a negligible function ε such that $\mathsf{Adv}^{\mathsf{fs-rind-cpa}}_{fs-\mathsf{PRE},A}($ $1^k, n(k)) < \varepsilon(k)$, where $\mathsf{Exp}^{\mathsf{fs-rind-cpa}}_{fs-\mathsf{PRE},A}$ is defined in Experiment 5.

We distinguish fs-PRE schemes based on this last notion:

Definition 9 (fs-PRE^--security). *If a fs-*PRE *scheme is fs-IND-CPA-1 and fs-IND-CPA-2 secure, then we say this fs-*PRE *scheme is fs-*PRE^-*-secure.*

Definition 10 (fs-PRE^+-security). *If a fs-*PRE *scheme is fs-IND-CPA-1, fs-IND-CPA-2, and fs-RIND-CPA secure, then we say this fs-*PRE *scheme is fs-*PRE^+*-secure.*

3.3 Stronger Security for Proxy Re-Encryption

To conclude the discussion of the security model of fs-PRE schemes, we first observe that it is interesting to consider the notion of fs-RIND-CPA security in the classical setting for PRE, i.e., Experiment 5 with fixed $n = 1$ and no call to the Evo algorithm. The notion again ensures involvement of the proxy for the re-encryption of every ciphertext, and can, thus, enforce that malicious users cannot learn anything beyond the explicitly re-encrypted ciphertexts. This immediately leads to a stronger security model for classical PRE (given in the full version), which we denote as PRE^+. In particular, it extends the classical model [4], dubbed PRE^-, which covers standard (IND-CPA) and master-secret security definitions, by our fs-RIND-CPA security notion ported to the PRE setting. As our fs-IND-CPA-i notions for fs-PRE are generalizations of the established standard security notions of PRE as defined in [4], we consequently obtain a PRE^+-secure PRE scheme from any fs-PRE^+-secure fs-PRE scheme. We formalize this observation via Lemma 2.

Lemma 2. *Any fs-*PRE^+*-secure fs-*PRE *scheme yields a* PRE^+*-secure* PRE *scheme.*

In the full version, we formally prove this lemma. This immediately gives us a construction for a PRE^+-secure PRE scheme.

Corollary 1. *Scheme 3 when limited to a single time period, i.e., setting $n = 1$, represents a* PRE^+*-secure* PRE *scheme.*

4 Constructing fs-PRE from Binary Tree Encryption

In this section we present our construction of fs-PRE which is based on BTEs. Along the way, we introduce the notion of forward-secret delegatable PKE (fs-DPKE) as intermediate step. Such a fs-DPKE scheme then directly gives us a first fs-PRE satisfying fs-PRE^- security. To extend our construction to satisfy the stronger fs-PRE^+ notion generically, we require a relatively mild homomorphic property of the fs-DPKE. This property is in particular satisfied by our fs-DPKE instantiation, which yields the first fs-PRE scheme with strong security.

4.1 Forward-Secret Delegatable Public-Key Encryption

We now formalize fs-DPKE. In such a scheme decryption rights within a public-key encryption scheme can be delegated from a delegator to a delegatee and secret keys of delegators can be evolved so that a secret key for some period e_i is no longer useful to decrypt ciphertexts of prior periods e_j with $j < i$.

Definition 11 (fs-DPKE). *A forward-secret delegatable* PKE *(fs-*DPKE*) scheme with message space $\mathcal{M}$ consists of the PPT algorithms* ($\mathsf{Setup}, \mathsf{Gen}, \mathsf{Evo}, \mathsf{Del}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{DelEvo}, \mathsf{DelDec}$) *as follows:*

$\mathsf{Setup}(1^k)$: *On input security parameter k, outputs public parameters* pp.

$\mathsf{Gen}(\mathsf{pp}, n)$: *On input public parameters* pp*, and maximum number of periods n, outputs public, secret and evolution keys* $(\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)})$.

$\mathsf{Evo}(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})$: *On input secret key* $\mathsf{sk}^{(i)}$*, and evolution key* $\mathsf{ek}^{(i)}$ *for period $i \in \{0, \ldots, n-2\}$, outputs secret key* $\mathsf{sk}^{(i+1)}$ *and evolution key* $\mathsf{ek}^{(i+1)}$ *for period $i+1$.*

$\mathsf{Del}(\mathsf{sk}_A^{(i)}, \mathsf{ek}_A^{(i)}, \mathsf{pk}_B)$: *On input secret key* $\mathsf{sk}_A^{(i)}$ *and evolution key* $\mathsf{ek}_A^{(i)}$ *(or $\perp$) for A and period $i \in \{0, \ldots, n-1\}$, and public key* pk_B *for B, outputs delegated key* $\mathsf{dk}^{(i)}$ *and delegated evolution key* $\mathsf{dek}^{(i)}$ *(or $\perp$) for period i.*

$\mathsf{Enc}(\mathsf{pk}, M, i)$: *On input a public key* pk*, a message $M \in \mathcal{M}$, and period $i \in \{0, \ldots, n-1\}$, outputs a ciphertext C.*

$\mathsf{Dec}(\mathsf{sk}^{(i)}, C)$: *On input a secret key* $\mathsf{sk}^{(i)}$*, for period $i \in \{0, \ldots, n-1\}$, and ciphertext C, outputs $M \in \mathcal{M} \cup \{\perp\}$.*

$\mathsf{DelEvo}(\mathsf{dk}^{(i)}, \mathsf{dek}^{(i)})$: *On input a delegation key* $\mathsf{dk}^{(i)}$ *and delegated evolution key* $\mathsf{dek}^{(i)}$ *for period $i \in \{0, \ldots, n-2\}$, output delegation key* $\mathsf{dk}^{(i+1)}$ *and delegated evolution key* $\mathsf{dek}^{(i+1)}$ *for period $i+1$.*

$\mathsf{DelDec}(\mathsf{sk}_B^{(i)}, \mathsf{dk}_{A \to B}^{(i)}, C_A)$: *On input secret key* $\mathsf{sk}_B^{(i)}$ *for B and period $i \in \{0, \ldots, n-1\}$, delegation key* $\mathsf{dk}_{A \to B}^{(i)}$ *from A for B and period i, and ciphertext C_A for A, outputs $M \in \mathcal{M} \cup \{\perp\}$.*

We note that the existence of the DelEvo algorithm is entirely optional. If provided, it allows the user in possession of a delegation key to evolve it for later periods without additional interaction with the delegator.

Correctness. For correctness we require that period i ciphertexts encrypted for user u can be decrypted if one is in possession of the secret key of u evolved to that period or one possess a delegation key of u to another user u' and the secret key for u' for that period. More formally, we require that for all security parameters $k \in \mathbb{N}$, all public parameters pp generated by $\mathsf{Setup}(1^k)$, all number of periods $n \in \mathbb{N}$, all users $U \in \mathbb{N}$, all key tuples $(\mathsf{pk}_u, \mathsf{sk}_u^{(0)}, \mathsf{ek}_u^{(0)})_{u \in [U]}$ generated by $\mathsf{Gen}(\mathsf{pp}, n)$, any period $i \in \{0, \ldots, n-1\}$, for any $u \in [U]$, any evolved keys $(\mathsf{sk}_u^{(i)}, \mathsf{ek}_u^{(i)})$ generated by Evo from $(\mathsf{sk}_u^{(0)}, \mathsf{ek}_u^{(0)})$, for all $u' \in [U], u \neq u'$, any (potentially evolved) delegation key $\mathsf{dk}_{u \to u'}^{(i)}$ for period i generated using Del from a (potentially evolved) secret key and the target public key, and all messages $M \in \mathcal{M}$ it holds that

$$\mathsf{Dec}(\mathsf{sk}_u^{(i)}, \mathsf{Enc}(\mathsf{pk}_u, M, i)) = \mathsf{DelDec}(\mathsf{sk}_{u'}^{(i)}, \mathsf{dk}_{u \to u'}^{(i)}, \mathsf{Enc}(\mathsf{pk}_u, M, i)) = M.$$

Security notions. The forward-secret IND-CPA notion is a straight-forward extension of the typical IND-CPA notion: the adversary selects a target period and gets access to secret and evolution keys of the targeted user for the selected period and is able to request delegation keys with honest and dishonest users for that period. The adversary then engages with an IND-CPA style challenge for the previous period. For the experiment, which is depicted in Experiment 6, the environment keeps a list of an initial empty list of honest users $\mathtt{HU}$.

$\mathsf{Gen}^{(h)}(\mathsf{pp}, n)$: Run $(\mathsf{pk}, \mathsf{sk}, \mathsf{ek}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$, set $\mathtt{HU} \leftarrow \mathtt{HU} \cup \{(\mathsf{pk}, \mathsf{sk}, \mathsf{ek})\}$, and return pk.

$\mathsf{Del}^{(h)}(j, \mathsf{pk}_u, \mathsf{pk})$: On input a period j, a public key pk_u and a public key pk, abort if $(\mathsf{pk}_u, \cdot) \notin \mathtt{HU}$. Otherwise, look up $\mathsf{sk}_u^{(0)}, \mathsf{ek}_u^{(0)}$ corresponding to pk_u from $\mathtt{HU}$, set $(\mathsf{sk}_u^{(i)}, \mathsf{ek}_u^{(i)}) \leftarrow \mathsf{Evo}(\mathsf{sk}_u^{(i-1)}, \mathsf{ek}_u^{(i-1)})$ for $i \in [j]$ if $j > 0$, and return $\mathsf{Del}(\mathsf{sk}_u^{(j)}, \mathsf{ek}_u^{(j)}, \mathsf{pk})$.

$\mathsf{Del}^{(h')}(j, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \mathsf{pk}_u)$: On input a period j, a secret key $\mathsf{sk}^{(0)}$, a evolution key $\mathsf{ek}^{(0)}$, and a public key pk_u, abort if $(\mathsf{pk}_u, \cdot) \notin \mathtt{HU}$. Otherwise, set $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)}) \leftarrow \mathsf{Evo}(\mathsf{sk}^{(i-1)}, \mathsf{ek}^{(i-1)})$ for $i \in [j]$ if $j > 0$, and return $\mathsf{Del}(\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)}, \mathsf{pk}_u)$.

Experiment $\mathsf{Exp}^{\mathsf{fs-ind-cpa}}_{fs-\mathsf{DPKE},A}(1^k, n)$

```
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^k), (\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n), b \xleftarrow{R} \{0,1\}$
$(j^*, \mathsf{st}) \leftarrow A(\mathsf{pp}, n, \mathsf{pk})$
$\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)} \leftarrow \mathsf{Evo}(\mathsf{sk}^{(j-1)}, \mathsf{ek}^{(j-1)})$ for $j \in [j^*]$.
$\mathcal{O} \leftarrow \{\mathsf{Gen}^{(h)}, \mathsf{Del}^{(h)}(j^* - 1, \cdot, \mathsf{pk}), \mathsf{Del}^{(h')}(j^* - 1, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \cdot)\}$
$(M_0, M_1, \mathsf{st}) \leftarrow A^{\mathcal{O}}(\mathsf{st}, \mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)}))$
$b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}(\mathsf{pk}, M_b, j^* - 1))$
if $b = b^*$ return 1, else return 0
```

Experiment 6. The fs-IND-CPA security experiment for a fs-DPKE scheme.

Definition 12 (fs-IND-CPA). *For a polynomially bounded function $n(\cdot) > 1$, a PPT adversary A, we define the advantage function in the sense of fs-IND-CPA as*

$$\mathsf{Adv}^{\mathsf{fs-ind-cpa}}_{fs-\mathsf{DPKE},A}(1^k, n(k)) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{fs-ind-cpa}}_{fs-\mathsf{DPKE},A}(1^k, n(k)) = 1\right] - \frac{1}{2} \right| .$$

If for all $n(\cdot) > 1$, and any A there exists a negligible function ε such that $fs - \mathsf{DPKE}, A(1^k, n(k)) < \varepsilon(k)$, then a fs-$\mathsf{DPKE}$ scheme is fs-IND-CPA secure.

4.2 Constructing fs-DPKE from BTE

Now we construct a fs-DPKE scheme from a BTE scheme by applying the CHK compiler to a BTE and combining it with an $\mathcal{F}$-HPKE scheme for handling the delegation keys, i.e., the fs-DPKE key contains a BTE and an $\mathcal{F}$-HPKE key. The evolution key contains the secret and derivation keys for all right siblings on the path from the root node to w^i as well as the evolution key for w^i. The evolution algorithms traverse the tree in a depth-first manner, hence the evolution keys are viewed as stack and when visiting a node, the derived secret and derivation keys are pushed onto the stack. To simplify the presentation of the scheme, we define an algorithm DFEval that performs the stack manipulation on a stack of pairs:

$\mathsf{DFEval}(s_1^{(w^i)}, s, \mathsf{Eval})$: On input the stack s and first element $s_1^{(w^i)}$ of the pair for node w^i, an algorithm Eval, perform the following steps:

- Pop the topmost element, $(\bot, s_2^{(w^i)})$, from the stack s.
- If w^i is an internal node, set $s^{(w^i 0)}, s^{(w^i 1)} \leftarrow \mathsf{Eval}(s_1^{(w^i)}, s_2^{(w^i)})$ and push $s^{(w^i 1)}, s^{(w^i 0)}$ onto s.
- Replace the topmost element, $(s_1^{(w^{i+1})}, s_2^{(w^{i+1})})$, with $(\bot, s_2^{(w^{i+1})})$.
- Return $s_1^{(w^{i+1})}$ and the new stack s.

The overall idea is now to encrypt the BTE secret key of the current period using the $\mathcal{F}$-HPKE scheme's public key of the target user. Using the homomorphic property of the encryption scheme, we are able to evolve the delegation keys in the same way as the secret keys of the nodes. In particular, we will require that the key derivation algorithm of the BTE can be represented by functions in $\mathcal{F}$, i.e., $\mathsf{Der}_{\mathsf{BTE}} = (f_i)_{i \in [m]}$. For notional simplicity, we will write $\mathsf{Eval}_{\mathsf{HPKE}}(\mathsf{Der}_{\mathsf{BTE}}, \cdot)$ instead of repeating it for each f_i that represents $\mathsf{Der}_{\mathsf{BTE}}$.

For our fs-DPKE scheme we need keys of different users to live in compatible key spaces. To that end, we introduce Setup algorithms for both schemes that fix the key spaces and we change the key generation algorithms to take the public parameters instead of the security parameter as argument. Note that when using the BTE from [15], linear ElGamal [11] as $\mathcal{F}$-HPKE to encrypt the BTE keys suffices for our needs.

Our construction. The fs-DPKE scheme is detailed in Scheme 1. We note that only the definition of DelEvo relies on the homomorphic properties of the HPKE scheme. So to obtain a fs-DPKE scheme without DelEvo algorithm, a compatible PKE scheme is sufficient. Yet, we will require the homomorphic properties later to achieve a suitable notion of adaptability regardless of the availability of DelEvo.

Similar to Canetti et al.'s construction, our fs-DPKE scheme inherits the fs-IND-CPA security from the BTE's IND-SN-CPA security.

Theorem 1. *If instantiated with an IND-SN-CPA secure* BTE *scheme and a IND-CPA secure* HPKE *scheme, then Scheme 1 is a fs-IND-CPA secure fs-*DPKE*.*

Proof. We prove the theorem using a sequence of games. We denote by W all the relevant nodes in the binary tree for period j. We note that the size of W is bounded by $\log_2(n)$. We index W as w_i for $i \in [|W|]$.

Game 0: The original game.

Game $1_{i,j}$ $(1 \leq i \leq q_{\mathsf{Del}^h}, 1 \leq j \leq 2|W|)$: As the previous game, but we replace all HPKE ciphertexts up to the j-th one in the i-th query with ciphertexts encrypting random plaintexts. That is, we modify the $\mathsf{Del}^{h'}$ in the i-th query as follows:

$\mathsf{Del}^{h'}(j, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \mathsf{pk}_i)$: Up to the j-th call to $\mathsf{Enc}_{\mathsf{HPKE}}$, encrypt a uniformly random value.

Let $(\mathsf{Setup}_{\mathsf{BTE}}, \mathsf{Gen}_{\mathsf{BTE}}, \mathsf{Der}_{\mathsf{BTE}}, \mathsf{Enc}_{\mathsf{BTE}}, \mathsf{Dec}_{\mathsf{BTE}})$ be a BTE scheme and $(\mathsf{Setup}_{\mathsf{HPKE}}, \mathsf{Gen}_{\mathsf{HPKE}}, \mathsf{Enc}_{\mathsf{HPKE}}, \mathsf{Dec}_{\mathsf{HPKE}}, \mathsf{Eval}_{\mathsf{HPKE}})$ a compatible $\mathcal{F}$-HPKE scheme with $\mathsf{Der}_{\mathsf{BTE}} \in \mathcal{F}$.

$\underline{\mathsf{Setup}(1^k)}$: Set $\mathsf{pp}_{\mathsf{BTE}} \leftarrow \mathsf{Setup}_{\mathsf{BTE}}(1^k)$, $\mathsf{pp}_{\mathsf{HPKE}} \leftarrow \mathsf{Setup}_{\mathsf{HPKE}}(1^k)$, and return $(\mathsf{pp}_{\mathsf{BTE}}, \mathsf{pp}_{\mathsf{HPKE}})$.

$\underline{\mathsf{Gen}(\mathsf{pp}, n)}$: Parse pp as $(\mathsf{pp}_{\mathsf{BTE}}, \mathsf{pp}_{\mathsf{HPKE}})$. Choose ℓ such that $n < 2^{\ell+1}$, set $(\mathsf{pk}_{\mathsf{BTE}}, \mathsf{sk}_{\mathsf{BTE}}^{(\varepsilon)}, \mathsf{dk}_{\mathsf{BTE}}^{(\varepsilon)}) \leftarrow \mathsf{Gen}_{\mathsf{BTE}}(\mathsf{pp}_{\mathsf{BTE}}, \ell)$ and $(\mathsf{pk}_{\mathsf{HPKE}}, \mathsf{sk}_{\mathsf{HPKE}}) \leftarrow \mathsf{Gen}_{\mathsf{HPKE}}(\mathsf{pp}_{\mathsf{HPKE}})$, and return $((\mathsf{pk}_{\mathsf{BTE}}, \mathsf{pk}_{\mathsf{HPKE}}), (\mathsf{sk}_{\mathsf{BTE}}^{(\varepsilon)}, \mathsf{sk}_{\mathsf{HPKE}}), (\perp, \mathsf{dk}_{\mathsf{BTE}}^{(\varepsilon)}))$.

$\underline{\mathsf{Evo}(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})}$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}_{\mathsf{BTE}}^{(w^i)}, \mathsf{sk}_{\mathsf{HPKE}})$ and view $\mathsf{ek}^{(i)}$ organized as a stack of secret key and evolution keys pairs. Set $\mathsf{sk}_{\mathsf{BTE}}^{(w^{i+1})}, \mathsf{ek}^{(i+1)} \leftarrow \mathsf{DFEval}(\mathsf{sk}_{\mathsf{BTE}}^{(w^i)}, \mathsf{ek}^{(i)}, \mathsf{Der}_{\mathsf{BTE}})$, and $\mathsf{sk}^{(i+1)} \leftarrow (\mathsf{sk}_{\mathsf{BTE}}^{(w^{i+1})}, \mathsf{sk}_{\mathsf{HPKE}})$. Return $\mathsf{sk}^{(i+1)}, \mathsf{ek}^{(i+1)}$.

$\underline{\mathsf{Enc}(\mathsf{pk}, M, i)}$: Parse pk as $(\mathsf{pk}_{\mathsf{BTE}}, \cdot)$, and return $\mathsf{Enc}_{\mathsf{BTE}}(\mathsf{pk}_{\mathsf{BTE}}, M, w^i)$.

$\underline{\mathsf{Dec}(\mathsf{sk}^{(i)}, C)}$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}_{\mathsf{BTE}}^{(w^i)}, \cdot)$, and return $\mathsf{Dec}_{\mathsf{BTE}}(\mathsf{sk}_{\mathsf{BTE}}^{(w^i)}, C)$.

$\underline{\mathsf{Del}(\mathsf{sk}_A^{(i)}, \mathsf{ek}_A^{(i)}, \mathsf{pk}_B)}$: Parse $\mathsf{sk}_A^{(i)}$ as $(\mathsf{sk}_{\mathsf{BTE}}^{(w^i)}, \cdot)$ and pk_B as $(\cdot, \mathsf{pk}_{\mathsf{HPKE}})$. If $\mathsf{ek}_A^{(i)} = \perp$, return $\mathsf{Enc}_{\mathsf{HPKE}}(\mathsf{pk}_{\mathsf{HPKE}}, \mathsf{sk}_{\mathsf{BTE}}^{(w^i)})$. Otherwise parse $\mathsf{ek}_A^{(i)}$ as $(\mathsf{sk}_{\mathsf{BTE}}^{(w)}, \mathsf{dk}_{\mathsf{BTE}}^{(w)})_{w \in W}, (\cdot, \mathsf{dk}_{\mathsf{BTE}}^{(w^i)})$, and set $\mathsf{dk}^{(w)} \leftarrow \mathsf{Enc}_{\mathsf{HPKE}}(\mathsf{pk}_{\mathsf{HPKE}}, \mathsf{sk}_{\mathsf{BTE}}^{(w)})$ and $\mathsf{dek}^{(w)} \leftarrow \mathsf{Enc}_{\mathsf{HPKE}}(\mathsf{pk}_{\mathsf{HPKE}}, \mathsf{dk}_{\mathsf{BTE}}^{(w)})$ for $w \in W \cup \{w^i\}$. Set $\mathsf{dk}^{(i)} \leftarrow \mathsf{dk}^{(w^i)}$ and $\mathsf{dek}^{(i)} \leftarrow (\mathsf{dk}^{(w)}, \mathsf{dek}^{(w)})_{w \in W}, (\perp, (\mathsf{dek}^{(w^i)}))$ and return $\mathsf{dk}^{(i)}, \mathsf{dek}^{(i)}$.

$\underline{\mathsf{DelEvo}(\mathsf{dk}_{A \to B}^{(i)}, \mathsf{dek}_{A \to B}^{(i)})}$: Parse $\mathsf{dk}_{A \to B}^{(i)}$ as $\mathsf{dk}_{A \to B}^{(w^i)}$ and view $\mathsf{dek}_{A \to B}^{(i)}$ organized as a stack of encrypted evolution keys. Set $\mathsf{dk}_{A \to B}^{(w^{i+1})}, \mathsf{dek}_{A \to B}^{(i+1)} \leftarrow \mathsf{DFEval}(\mathsf{dk}_{A \to B}^{(w^i)}, \mathsf{dek}_{A \to B}^{(i)}, \mathsf{Eval}_{\mathsf{HPKE}}(\mathsf{Der}_{\mathsf{BTE}}, \cdot))$, and $\mathsf{dk}^{(i+1)} \leftarrow \mathsf{dk}_{\mathsf{BTE}}^{(w^{i+1})}$. Return $\mathsf{dk}^{(i+1)}, \mathsf{dek}^{(i+1)}$.

$\underline{\mathsf{DelDec}(\mathsf{sk}_B^{(i)}, \mathsf{dk}_{A \to B}^{(i)}, C_A)}$: Parse $\mathsf{sk}_B^{(i)}$ as $(\cdot, \mathsf{sk}_{\mathsf{HPKE}})$, set $\mathsf{sk}_{\mathsf{BTE}}^{(w^i)} \leftarrow \mathsf{Dec}_{\mathsf{HPKE}}(\mathsf{sk}_{\mathsf{HPKE}}, \mathsf{dk}_{A \to B}^{(i)})$, and return $\mathsf{Dec}_{\mathsf{BTE}}(\mathsf{sk}_{\mathsf{BTE}}^{(w^i)}, C_A)$.

Scheme 1. fs-DPKE scheme from BTE scheme and a compatible HPKE scheme.

Transition$^{0 \to 1_{1,1}}$, **Transition**$^{1_{i,j} \to 1_{i,j+1}}$, **Transition**$^{1_{i,2|W|} \to 1_{i+1,1}}$**:** A distinguisher $\mathcal{D}^{0 \to 1_{1,1}}$ (respectively $\mathcal{D}^{1_{i,j} \to 1_{i,j+1}}$ or $\mathcal{D}^{1_{i,2|W|} \to 1_{i+1,1}}$) is an IND-CPA adversary against the HPKE scheme. We construct a reduction where we let $\mathcal{C}$ be a IND-CPA challenger. We modify $\mathsf{Del}^{h'}$ in the i-th query in the following way:

$\mathsf{Del}^{h'}(j, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}, \mathsf{pk}_{i'})$: Simulate everything honestly, but on the j-th query choose $\boxed{r}$ uniformly at random and run

$$\boxed{c \leftarrow \mathcal{C}(\mathsf{sk}_{BTE}^{(w_{(j/2)-1})}, r)} \text{ if } j \text{ is odd and } \boxed{c \leftarrow \mathcal{C}(\mathsf{ek}_{BTE}^{(w_{j/2})}, r)} \text{ if } j \text{ is even,}$$

where $c \leftarrow \mathcal{C}(m_0, m_b)$ denotes a challenge ciphertext with respect to m_0 and m_1.

Now, the bit b chosen by $\mathcal{C}$ switches between the distributions of the Games.

In Game $1_{q_{\mathsf{Del}^h}, 2|W|}$ all ciphertexts obtainable from $\mathsf{Del}^{h'}$ are with respect to random values. Now, an adversary B winning Game $1_{q_{\mathsf{Del}^h}, 2|W|}$ can be transformed into a IND-SN-CPA adversary A against the underlying BTE scheme:

1. When A is first started on $1^k, \ell$, choose $i^* \xleftarrow{R} [n]$ and output $w^{(i^*-1)}$.

2. When A is started on $\mathsf{pk}_{\mathsf{BTE}}, (\mathsf{sk}^{(w)}, \mathsf{dk}^{(w)})_{w \in W}$, compute $(\mathsf{pk}_{\mathsf{HPKE}}, \mathsf{sk}_{\mathsf{HPKE}}) \leftarrow \mathsf{Gen}_{\mathsf{HPKE}}(1^k)$. The secret key $\mathsf{sk}_{\mathsf{HPKE}}$ is stored in the state st and we extend the public key to $\mathsf{pk} \leftarrow (\mathsf{pk}_{\mathsf{BTE}}, \mathsf{pk}_{\mathsf{HPKE}})$. Now start B on the extended public key, i.e. $(j^*, \mathsf{st}) \leftarrow B(1^k, n, \mathsf{pk})$. If $i^* \neq j^*$, output a random bit and halt. Otherwise we have the secret-derivation key pairs of all nodes that are right siblings on the path from the root node to $w^{(j^*-1)}$ and (if they exist) all child nodes of $w^{(j^*-1)}$, hence we are able to simulate all oracle queries from B honestly. Similarly, we can compute $(\mathsf{sk}^{(j^*)}, \mathsf{dk}^{(j^*)})$ from the given keys. Thus we run $B^{\mathcal{O}}(\mathsf{st}, \mathsf{sk}^{(j^*)}, \mathsf{dk}^{(j^*)})$ and forward its result.
3. When A is finally started on the challenge ciphertext, the ciphertext is simply forwarded to B and when B outputs the bit b, A returns b and halts.

When B is running within A and $j^* = i^*$, B has exactly the same view as in Game $1_{q_{\mathsf{Gen}^h}, 2|W|}$. In this case the probability of A to win is exactly the same as the winning probability of B, and Game $1_{q_{\mathsf{Gen}^h}, 2|W|}$ is computationally indistinguishable from the initial game. The random guess of i^* so that $i^* = j^*$ induces a loss of $\frac{1}{n}$, which is however bounded by a polynomial in the security parameter. $\square$

4.3 Constructing fs-PRE from fs-DPKE

Now we present a construction of a fs-PRE^+-secure fs-PRE scheme from a fs-DPKE scheme. Therefore, we define additional properties of fs-DPKE and show that a fs-PRE can be directly obtained from a fs-DPKE. For our transformation to work, we need to define an additional algorithm that allows us to homomorphically shift ciphertexts and delegation keys. That is, ciphertexts and delegation keys are modified in such a way that the delegation keys look like randomly distributed fresh keys, which are only useful to decrypt ciphertexts adapted to this key. Formally, we introduce an algorithm Adapt that enables this adaption:

$\mathsf{Adapt}(\mathsf{dk}, C)$: On input a delegation key dk, a ciphertext C, outputs an adapted delegation key dk' and ciphertext C'.

Since the delegation keys in our construction are encrypted BTE secret keys, we essentially adapt secret keys and ciphertexts from a BTE. We will see that this adaption is possible as long as the HPKE scheme used to encrypt the BTE keys provides a suitable homomorphism on the message space.

To adapt ciphertexts and delegation keys we extend correctness to additionally require that for any message M encrypted under the public key of A, any delegation key $\mathsf{dk}^{(i)}_{A \to B}$, and any adapted delegation key-ciphertext pairs $(\mathsf{dk}', C') \leftarrow \mathsf{Adapt}(\mathsf{dk}^{(i)}_{A \to B}, C_A)$, it holds that $M = \mathsf{DelDec}_{\mathsf{DPKE}}(sk^{(i)}_B, \mathsf{dk}', C')$.

As security notion we introduce the fs-ADAP-IND-CPA notion, where the adversary may see multiple adapted delegation keys and ciphertexts, but the adversary should be unable to win an IND-CPA game for non-adapted ciphertexts. We give the formal definition of the security experiment in Experiment 7. This notion gives the delegator more control over the ciphertexts that should be readable for the delegatee. If given the delegation key, the delegatee can always decrypt all ciphertexts, but if just given an adapted delegation key, only a selected subset of ciphertexts is decryptable.

Experiment $\mathsf{Exp}^{\mathsf{fs-adap-ind-cpa}}_{fs-\mathsf{DPKE},A}(1^k, n)$

```
$\mathsf{pp} \leftarrow \mathsf{Setup}(1^k), (\mathsf{pk}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n), b \xleftarrow{R} \{0,1\}$
$(j^*, \mathsf{pk}^*, \mathsf{st}) \leftarrow A(\mathsf{pp}, n, \mathsf{pk})$
$\mathsf{sk}^{(j)}, \mathsf{ek}^{(j)} \leftarrow \mathsf{Evo}(\mathsf{sk}^{(j-1)}, \mathsf{ek}^{(j-1)})$ for $j \in [j^*]$, $\mathsf{dk} \leftarrow \mathsf{Del}(\mathsf{sk}^{(j^*)}, \perp, \mathsf{pk}^*)$
$(M_0, M_1, \mathsf{st}) \leftarrow A^{\{\mathsf{Adapt}(\mathsf{dk}, \cdot)\}}(\mathsf{st})$
$b^* \leftarrow A(\mathsf{st}, \mathsf{Enc}(\mathsf{pk}, M_b, j^*))$
if $b = b^*$ return 1, else return 0
```

Experiment 7. The fs-ADAP-IND-CPA security experiment for a fs-DPKE scheme.

Definition 13 (fs-ADAP-IND-CPA). *For a polynomially bounded function $n(\cdot) > 1$, a PPT adversary A, we define the advantage function in the sense of fs-IND-CPA as*

$$\mathsf{Adv}^{\mathsf{fs-adap-ind-cpa}}_{fs-\mathsf{DPKE},A}(1^k, n(k)) := \left| \Pr\left[\mathsf{Exp}^{\mathsf{fs-adap-ind-cpa}}_{\mathsf{DPKE},A}(1^k, n(k)) = 1\right] - \frac{1}{2} \right|.$$

If for all $n(\cdot) > 1$, and any A there exists a negligible function ε such that $\mathsf{Adv}^{\mathsf{fs-adap-ind-cpa}}_{fs-\mathsf{DPKE},A}(1^k, n(k)) < \varepsilon(k)$, then a fs-$\mathsf{DPKE}$ *scheme is fs-ADAP-IND-CPA secure.*

For Scheme 1, this adaption can be achieved solely from key-homomorphic properties of the BTE and homomorphic properties of the HPKE, respectively. Subsequently, we define the required homomorphisms. Our definitions are inspired by [2, 40]. We focus on schemes where the secret/derived key pairs, and public keys live in groups $(\mathbb{G}, +)$, and $(\mathbb{H}, \cdot)$, respectively. We will require two different properties: first, the public key is the image of the secret key under a group homomorphism, and second, given two secret keys with a known difference, we can map the binary tree of derived keys from one key to the other key. In other words, the difference in the keys propagates to the derived keys.

Definition 14. *Let Ω be a* BTE *scheme with secret/derived key space $(\mathbb{G}, +)$ and public key space $(\mathbb{H}, \cdot)$.*

1. *The scheme Ω provides a secret-key-to-public-key homomorphism, if there exists an efficiently computable group homomorphism $\mu : \mathbb{G} \to \mathbb{H}$ such that for all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$, it holds that $\mathsf{pk} = \mu(\mathsf{sk})$.*
2. *The scheme Ω provides a derived-key homomorphism, if there exists a family of efficiently computable group homomorphisms $\nu^{(w)} : \mathbb{G} \to \mathbb{G}^2$ such that for all $(\mathsf{pk}, \mathsf{sk}^{(\varepsilon)}) \leftarrow \mathsf{Gen}$, all nodes w it holds that $(\mathsf{sk}^{(w0)}, \mathsf{sk}^{(w1)}) = \nu^{(w)}(\mathsf{sk}^{(w)})$ and for all messages M it holds that $\mathsf{Dec}(\mathsf{sk}^{(w)}, \mathsf{Enc}(\mathsf{pk}, M, w)) = M$.*

We denote by Φ^+ the set of all possible secret key differences in $\mathbb{G}$. Alternatively, it is possible to view Φ^+ as set of functions representing all linear shifts in $\mathbb{G}$ and we simply identify each shift by an element $\Delta \in \mathbb{G}$.

Definition 15. *A* BTE *scheme* Ω *is called* Φ^+*-key-homomorphic, if it provides both a secret-key-to-public-key homomorphism and a derived key homomorphism and an additional PPT algorithm* Adapt*, defined as:*

$\mathsf{Adapt}(\mathsf{pk}, C, \Delta)$: *On input a delegation key* dk*, a ciphertext* C *and a secret key difference* Δ*, outputs a public key* pk' *and a ciphertext* C'*.*

such that for all $\Delta \in \Phi^+$*, and all* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\ldots)$*, all message* M*, and all* $C \leftarrow \mathsf{Enc}(\mathsf{pk}, M)$*, and* $(\mathsf{pk}', C') \leftarrow \mathsf{Adapt}(\mathsf{pk}, C, \Delta)$ *it holds that* $\mathsf{pk}' = \mathsf{pk} \cdot \mu(\Delta)$ *and* $\mathsf{Dec}(\mathsf{sk}^{(w)} + \nu^{(w)}(\Delta), C') = M$*.*

Definition 16 (Adaptability of ciphertexts). *A* Φ^+*-key-homomorphic* BTE *scheme provides adaptability of ciphertexts, if for every security parameter* $k \in \mathbb{N}$*, any public parameters* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^k)$*, every message* M *and every period* j*, it holds that* $\mathsf{Adapt}(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, M, j), \Delta)$ *and* $(\mathsf{pk} \cdot \mu(\Delta), \mathsf{Enc}(\mathsf{pk} \cdot \mu(\Delta), M, j))$ *as well as* $(\mathsf{sk}, \mathsf{pk})$ *and* $(\mathsf{sk}', \mu(\mathsf{sk}'))$ *are identically distributed, where* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$*,* $\mathsf{sk}' \overset{R}{\leftarrow} \mathbb{G}$ *and* $\Delta \leftarrow \Phi^+$*.*

Next, we discuss the BTE from [15] with respect to our notion of ciphertext adaptability. We first recall the BTE scheme in Scheme 2 where BGGen is a bilinear group generator. By [15, Proposition 1] this scheme is IND-SN-CPA secure if the decisional BDH assumption holds relative to BGGen.

$\mathsf{Setup}(1^k)$: Run to $\mathsf{BGGen}_p(1^k)$ to generate groups $\mathbb{G}_1, \mathbb{G}_2$ of prime order q and a bilinear map e and select a random generator $P \in \mathbb{G}_1$. Set $\mathsf{pp} \leftarrow (\mathbb{G}_1, \mathbb{G}_2, e, q, P)$ and return pp.

$\mathsf{Gen}(\mathsf{pp}, \ell)$: Choose $\alpha \leftarrow \mathbb{Z}_q$ and set $Q \leftarrow \alpha \cdot P$. Set $\mathsf{sk}^{(\varepsilon)} \leftarrow \alpha H(\varepsilon)$ and $\mathsf{pk} \leftarrow (Q, H)$. Return $(\mathsf{pk}, \mathsf{sk}^{(\varepsilon)})$.

$\mathsf{Der}(\mathsf{sk}^{(i)})$: Parse $\mathsf{sk}^{(w)}$ as $(R_{w|1}, \ldots, R_w, S_w)$. Choose $r_0, r_1 \overset{R}{\leftarrow} \mathbb{Z}_q$ and set $R_{wi} \leftarrow r_i P$ and $S_{wi} \leftarrow S_w + r_i \cdot H(wi)$ for $i \in [2]$ and return $((R_{w|1}, \ldots, R_w, R_{w0}, S_{w0}), (R_{w|1}, \ldots, R_w, R_{w1}, S_{w1}))$.

$\mathsf{Enc}(\mathsf{pk}, M, i)$: Choose $\gamma \leftarrow \mathbb{Z}_q$ and set $C \leftarrow (\gamma \cdot P, \gamma \cdot H(w|1), \ldots, \gamma \cdot H(w), M \cdot e(Q, \gamma \cdot H(\varepsilon)))$. Return C.

$\mathsf{Dec}(\mathsf{sk}^{(w)}, C)$: Parse $\mathsf{sk}^{(w)}$ as $(R_{w|1}, \ldots, R_w, S_w)$ and $\mathcal{C}$ as $(U_0, \ldots, U_t, V)$. Return $M = V/d$ where

$$d = \frac{e(U_0, S_w)}{\prod_{i=1}^{t} e(R_{w|i}, U_i)}.$$

Scheme 2. BTE scheme from [15]

Now we show that Scheme 2 also provides adaptability of ciphertexts:

Lemma 3. *Scheme 2 provides adaptability of ciphertexts under shared* H*.*

Proof. We show the existence of the homomorphisms and give the Adapt algorithm. Note that the master secret key can easily be viewed as containing α, hence, the secret-to-public-key homomorphism is simply $\mu : \alpha \mapsto \alpha P$. As the Der algorithm simply computes sums, the existence of the homomorphism is clear.

We now show the existence of Adapt:

$\mathsf{Adapt}(\mathsf{pk}, C, \varDelta)$: Parse pk as (Q, ℓ, H) and C as $(U_0, \ldots, U_t, V)$. Let $Q' \leftarrow Q + \varDelta \cdot P$ and set $\mathsf{pk}' \leftarrow (Q', \ell, H)$. Let $V' \leftarrow Ve(U_0, \varDelta \cdot H(\varepsilon))$ and set $C' \leftarrow (U_0, \ldots, U_t, V')$ and return (pk', C').

The adapted C' ciphertext is an encryption of the original message under the public key $Q' = Q + \varDelta \cdot P$. □

Now, given any $\varPhi^+$-key-homomorphic BTE scheme, it can be turned into an adaptable fs-DPKE by defining Adapt in a publicly computable way as follows:

$\mathsf{Adapt}(\mathsf{dk}^{(i)}_{A \to B}, C)$: Sample $\varDelta \xleftarrow{R} \varPhi^+$ and compute $\mathsf{dk}_\varDelta \leftarrow \mathsf{Enc}_{\mathsf{HPKE}}(\mathsf{pk}_B, \nu^{(w^i)}(\varDelta))$, and then $\mathsf{dk}' \leftarrow \mathsf{Eval}_{\mathsf{HPKE}}(+, \mathsf{dk}^{(i)}_{A \to B}, \mathsf{dk}_\varDelta)$. Set $(\cdot, C') \leftarrow \mathsf{Adapt}_{\mathsf{BTE}}(\mathsf{pk}_A, C, \varDelta)$. Return (dk', C').

Theorem 2. *If in addition to the premise in Theorem 1 the* BTE *scheme also provides adaptability of ciphertexts, then Scheme 1 is a fs-ADAP-IND-CPA secure fs-*DPKE *scheme.*

Proof. We prove this theorem with a sequence of games.

Game 0: The original game.
Game 1: We modify the simulation of the Adapt oracle as follows, where we denote the modified oracle by Adapt':

$\mathsf{Adapt}'(\boxed{\mathsf{sk}^{(i)}, \mathsf{pk}, \mathsf{pk}^*}, C)$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}^{(w^i)}_{\mathsf{BTE}}, \cdot)$, pk as $(\mathsf{pk}_{\mathsf{BTE}}, \cdot)$, and pk^* as $(\cdot, \mathsf{pk}^*_{\mathsf{HPKE}})$. Choose $\varDelta \leftarrow \varPhi^+$, run

$$\boxed{\mathsf{dk}' \leftarrow \mathsf{Enc}_{\mathsf{HPKE}}(\mathsf{pk}^*_{\mathsf{HPKE}}, \mathsf{sk}^{(w^i)}_{\mathsf{BTE}} + \nu^{(w^i)}(\varDelta))} \text{ and}$$
$$\boxed{C' \leftarrow \mathsf{Enc}_{\mathsf{BTE}}(\mathsf{pk} \cdot \mu(\varDelta), \mathsf{Dec}_{\mathsf{BTE}}(\mathsf{sk}^{(i)}, C), i)}. \text{ Return } (\mathsf{dk}', C').$$

Transition$^{0 \to 1}$: The distributions of Game 0 and Game 1 are indistinguishable under the BTE's adaptability of ciphertexts.
Game 2: We further modify the simulation of Adapt' as follows:

$\mathsf{Adapt}'(\boxed{\mathsf{sk}^{(i)}, \mathsf{pk}^*}, C)$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}^{(w^i)}_{\mathsf{BTE}}, \cdot)$, pk as $(\mathsf{pk}_{\mathsf{BTE}}, \cdot)$, and pk^* as $(\cdot, \mathsf{pk}^*_{\mathsf{HPKE}})$. Choose $\boxed{\mathsf{pk}'_{\mathsf{BTE}}, \mathsf{sk}'^{,(\varepsilon)}_{\mathsf{BTE}}, \mathsf{ek}'^{,(\varepsilon)}_{\mathsf{BTE}} \leftarrow \mathsf{Gen}_{\mathsf{BTE}}}$ and evolve the secret key to period i, run

$$\boxed{\mathsf{dk}' \leftarrow \mathsf{Enc}_{\mathsf{HPKE}}(\mathsf{pk}^*_{\mathsf{HPKE}}, \mathsf{sk}'^{,(w^i)}_{\mathsf{BTE}})} \text{ and}$$
$$\boxed{C' \leftarrow \mathsf{Enc}_{\mathsf{BTE}}(\mathsf{pk}'_{\mathsf{BTE}}, \mathsf{Dec}_{\mathsf{BTE}}(\mathsf{sk}^{(i)}, C), i)}. \text{ Return } (\mathsf{dk}', C').$$

Transition$^{1 \to 2}$: The change is conceptual.

In Game 2 all the secret BTE keys the adversary gets are chosen independently from the challenge key. Hence, Game 2 is a standard IND-CPA game and thus the success probability of Game 2 is negligible by Theorem 1. □

Now, given an adaptable fs-DPKE scheme, we use the Adapt algorithm to obtain a fs-PRE^+ secure fs-PRE scheme. While the algorithms Setup, Gen, Evo, $\mathsf{Enc}^{(i)}$, and $\mathsf{Dec}^{(i)}$ can simply be lifted from the fs-DPKE scheme, we note that for each period j in the fs-PRE scheme, we use two periods, i.e., $2j-1$ and $2j$, of the fs-DPKE scheme. The period $2j-1$ is used for level 1 ciphertexts whereas the period $2j$ is used for level 2 ciphertexts[15]. We use $\mathsf{Del}_{\mathsf{DPKE}}$ and $\mathsf{DelEvo}_{\mathsf{DPKE}}$ for ReGen and ReEvo, respectively. For the re-encryption algorithm ReEnc, we apply Adapt. $\mathsf{Dec}^{(1)}$ for re-encrypted ciphertexts then decrypts the ciphertext by running $\mathsf{DelDec}_{\mathsf{DPKE}}$ on the adapted delegation key and ciphertext. The full scheme is presented in Scheme 3.

Let $(\mathsf{Setup}_{\mathsf{DPKE}}, \mathsf{Gen}_{\mathsf{DPKE}}, \mathsf{Evo}_{\mathsf{DPKE}}, \mathsf{Del}_{\mathsf{DPKE}}, \mathsf{Enc}_{\mathsf{DPKE}}, \mathsf{Dec}_{\mathsf{DPKE}}, \mathsf{Adapt}_{\mathsf{DPKE}})$ be fs-DPKE scheme with adaption of ciphertexts and delegation keys.

$\underline{\mathsf{Setup}(1^k)}$: Return $\mathsf{Setup}_{\mathsf{DPKE}}(1^k)$.

$\underline{\mathsf{Gen}(\mathsf{pp}, n)}$: Set $(\mathsf{pk}_{\mathsf{DPKE}}, \mathsf{sk}^{(0)}_{\mathsf{DPKE}}, \mathsf{ek}^{(0)}_{\mathsf{DPKE}}) \leftarrow \mathsf{Gen}_{\mathsf{DPKE}}(\mathsf{pp}, 2n+1)$, obtain $(\mathsf{sk}^{(1)}_{\mathsf{DPKE}}, \mathsf{ek}^{(1)}_{\mathsf{DPKE}}) \leftarrow \mathsf{Evo}_{\mathsf{DPKE}}(\mathsf{sk}^{(0)}_{\mathsf{DPKE}}, \mathsf{ek}^{(0)}_{\mathsf{DPKE}})$, and return $(\mathsf{pk}_{\mathsf{DPKE}}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)})$, where

$$\mathsf{sk}^{(0)} \leftarrow (\mathsf{sk}^{(0)}_{\mathsf{DPKE}}, \mathsf{sk}^{(1)}_{\mathsf{DPKE}}),\ \mathsf{ek}^{(0)} \leftarrow (\mathsf{ek}^{(0)}_{\mathsf{DPKE}}, \mathsf{ek}^{(1)}_{\mathsf{DPKE}}).$$

$\underline{\mathsf{Evo}(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})}$: Parse $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})$ as $((\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(2i)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+1)}_{\mathsf{DPKE}}))$ and return $(\mathsf{sk}^{(i+1)}, \mathsf{ek}^{(i+1)}) = (\mathsf{sk}^{(2i+2)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+3)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(2i+2)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+3)}_{\mathsf{DPKE}}))$, where

$$(\mathsf{sk}^{(2i+1+j)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+1+j)}_{\mathsf{DPKE}}) \leftarrow \mathsf{Evo}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+j)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+j)}_{\mathsf{DPKE}}) \text{ for } j \in [2].$$

$\underline{\mathsf{Enc}^{(1)}(\mathsf{pk}, M, i)}$: Return $\mathsf{Enc}_{\mathsf{DPKE}}(\mathsf{pk}, M, 2i)$.

$\underline{\mathsf{Enc}^{(2)}(\mathsf{pk}, M, i)}$: Return $\mathsf{Enc}_{\mathsf{DPKE}}(\mathsf{pk}, M, 2i+1)$.

$\underline{\mathsf{Dec}^{(1)}(\mathsf{sk}^{(i)}, C)}$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}})$ and return $\mathsf{Dec}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i)}, C)$ if C was not re-encrypted. Otherwise parse C as (C_1, rk) and return $\mathsf{DelDec}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+1)}, \mathsf{rk}, C_1)$.

$\underline{\mathsf{Dec}^{(2)}(\mathsf{sk}^{(i)}, C)}$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}})$ and return $\mathsf{Dec}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+1)}, C)$.

$\underline{\mathsf{ReGen}(\mathsf{sk}^{(i)}_A, \mathsf{ek}^{(i)}_A, \mathsf{pk}_B)}$: Parse $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})$ as $((\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(2i)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+1)}_{\mathsf{DPKE}}))$, and $\mathsf{Del}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+1)}_A, \mathsf{ek}^{(2i+1)}_A, \mathsf{pk}_B)$.

$\underline{\mathsf{ReEvo}(\mathsf{rk}^{(i)}_{A\to B}, \mathsf{rek}^{(i)}_{A\to B})}$: Return $\mathsf{DelEvo}_{\mathsf{DPKE}}(\mathsf{DelEvo}_{\mathsf{DPKE}}(\mathsf{rk}^{(i)}_{A\to B}, \mathsf{rek}^{(i)}_{A\to B}))$.

$\underline{\mathsf{ReEnc}(\mathsf{rk}^{(i)}_{A\to B}, C_A)}$: Choose $\tau \xleftarrow{R} \mathbb{G}$ and return $\mathsf{Adapt}_{\mathsf{DPKE}}(\mathsf{rk}^{(i)}_{A\to B}, C_A, \tau)$.

Scheme 3. fs-PRE scheme from an adaptable fs-DPKE scheme.

We prove that our scheme is both fs-IND-CPA-1 and fs-IND-CPA-2 secure. Both security notions follow from the fs-IND-CPA security of the underlying fs-DPKE scheme. In contrast, to achieve fs-RIND-CPA, we require an fs-ADAP-IND-CPA fs-DPKE scheme.

Theorem 3. *If instantiated with a fs-IND-CPA and fs-ADAP-IND-CPA secure fs-*DPKE *scheme, Scheme 3 is a fs-*PRE^+*-secure fs-*PRE *scheme.*

[15] One can see the keys for period $2j$ as weak keys in the sense of [4, Third Attempt] whereas the keys for period $2j-1$ constitute the master secret keys.

Proof. Informally speaking, the security experiment for fs-IND-CPA-2 with a fixed period j^* corresponds to the fs-IND-CPA experiment for fs-DPKE for period $2j^*$. We can build a straightforward reduction from an adversary against fs-IND-CPA-2, A_2 to fs-IND-CPA for fs-DPKE:

- When started on pp, n and pk, run $(j^*, \mathsf{st}) \leftarrow A_2(\mathsf{pp}, \lceil \frac{n}{2}+1 \rceil, \mathsf{pk})$. Set $j' \leftarrow 2j^*$ and return (j', st).
- When started on $\mathsf{st}, \mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$, we simulate the $\mathsf{ReGen}^{(h)}$ and $\mathsf{ReGen}^{(h')}$ oracles using $\mathsf{Del}^{(h)}$ and $\mathsf{Del}^{(h')}$. Indeed, $\mathsf{Del}^{(h)}$ and $\mathsf{Del}^{(h')}$ return delegation keys for period $j'-1 = 2j^*-1$, which are re-encryption keys for period j^*-1. Using Evo we evolve $\mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$ to period $j'+1$. Set $(\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)}) \leftarrow ((\mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{sk}_{\mathsf{DPKE}}^{(j'+1)}), (\mathsf{ek}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j'+1)}))$ and start A_2 on $\mathsf{st}, (\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$ and simply forward the result.
- Finally, when started on st and $C_{j'-1}$, $C_{j'-1}$ is a level 2 ciphertext for j^*-1. Hence we start A_2 on the ciphertext and return its' result.

To show fs-IND-CPA-1 security, we perform a similar reduction:

- When started on pp, n and pk, run $(j^*, \mathsf{st}) \leftarrow A_1(\mathsf{pp}, \lceil \frac{n}{2}+1 \rceil, \mathsf{pk})$. Set $j' \leftarrow 2j^*-1$ and return (j', st).
- When started on $\mathsf{st}, \mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$, we simulate the $\mathsf{ReGen}^{(h)}$ and $\mathsf{ReGen}^{(h')}$ oracles using $\mathsf{Del}^{(h)}$ and $\mathsf{Del}^{(h')}$ and by running DelEvo on the result. Indeed, $\mathsf{Del}^{(h)}$ and $\mathsf{Del}^{(h')}$ return delegation keys for period $j'-1 = 2j^*-2$, hence after applying DelEvo we obtain re-encryption keys for period j^*-1. $\mathsf{ReGen}^{(d)}$ is simulated honestly by delegating $\mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$ to a dishonest user. Using Evo we evolve $\mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$ to period $j'+2$. Set $(\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)}) \leftarrow ((\mathsf{sk}_{\mathsf{DPKE}}^{(j'+1)}, \mathsf{sk}_{\mathsf{DPKE}}^{(j'+2)}), (\mathsf{ek}_{\mathsf{DPKE}}^{(j'+1)}, \mathsf{ek}_{\mathsf{DPKE}}^{(j'+2)}))$ and start A_1 on $\mathsf{st}, (\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$ and simply forward the result.
- Finally, when started on st and $C_{j'-1}$, $C_{j'-1}$ is a level 1 ciphertext for j^*-1. Hence we start A_1 on the ciphertext and return its' result.

To show receiver-IND-CPA security we build an fs-ADAP-IND-CPA adversary against the fs-DPKE scheme. The fs-RIND-CPA adversary is denoted as A_r.

- When started on pp, n and pk, run $(j^*, \mathsf{st}) \leftarrow A_r(\mathsf{pp}, \lceil \frac{n}{2}+1 \rceil, \mathsf{pk})$. Set $j' \leftarrow 2j^*+1$ and return (j', st).
- When started on st, we can simulate ReEnc honestly using Adapt.
- Wen started on st and C, the ciphertext is a level 2 ciphertext for period $j*$, hence we return $A_r(\mathsf{st}, C)$.

Note that all values are consistently distributed in all three reductions. □

4.4 Separating fs-PRE$^-$ from fs-PRE$^+$

To expand on the gap between fs-PRE^+ and fs-PRE^- schemes and to provide an explicit separation, we construct a counterexample. In particular, it is clear that every scheme that satisfies fs-PRE^+ also satisfies fs-PRE^-. For our separation we now present a scheme that is fs-PRE^- but trivially violates fs-PRE^+. The scheme is also built from a fs-DPKE scheme and presented in Scheme 4. In this scheme however, ReEnc simply embeds the delegation key in the re-encrypted ciphertext. The shortcomings of this construction compared to Scheme 3 are obvious: once the receiver is presented with one valid re-encrypted ciphertext, it can recover the delegation key from that ciphertext and can decrypt all level 2 ciphertexts for this period.

Let $(\mathsf{Setup}_{\mathsf{DPKE}}, \mathsf{Gen}_{\mathsf{DPKE}}, \mathsf{Evo}_{\mathsf{DPKE}}, \mathsf{Del}_{\mathsf{DPKE}}, \mathsf{Enc}_{\mathsf{DPKE}}, \mathsf{Dec}_{\mathsf{DPKE}})$ be fs-DPKE scheme.

$\underline{\mathsf{Setup}(1^k)}$: Return $\mathsf{Setup}_{\mathsf{DPKE}}(1^k)$.

$\underline{\mathsf{Gen}(\mathsf{pp}, n)}$: Set $(\mathsf{pk}_{\mathsf{DPKE}}, \mathsf{sk}^{(0)}_{\mathsf{DPKE}}, \mathsf{ek}^{(0)}_{\mathsf{DPKE}}) \leftarrow \mathsf{Gen}_{\mathsf{DPKE}}(\mathsf{pp}, 2n{+}1)$, obtain $(\mathsf{sk}^{(1)}_{\mathsf{DPKE}}, \mathsf{ek}^{(1)}_{\mathsf{DPKE}}) \leftarrow \mathsf{Evo}_{\mathsf{DPKE}}(\mathsf{sk}^{(0)}_{\mathsf{DPKE}}, \mathsf{ek}^{(0)}_{\mathsf{DPKE}})$, and return $(\mathsf{pk}_{\mathsf{DPKE}}, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)})$, where

$$\mathsf{sk}^{(0)} \leftarrow (\mathsf{sk}^{(0)}_{\mathsf{DPKE}}, \mathsf{sk}^{(1)}_{\mathsf{DPKE}}),\ \mathsf{ek}^{(0)} \leftarrow (\mathsf{ek}^{(0)}_{\mathsf{DPKE}}, \mathsf{ek}^{(1)}_{\mathsf{DPKE}}).$$

$\underline{\mathsf{Evo}(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})}$: Parse $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})$ as $((\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(2i)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+1)}_{\mathsf{DPKE}}))$ and return $(\mathsf{sk}^{(i+1)}, \mathsf{ek}^{(i+1)}) = (\mathsf{sk}^{(2i+2)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+3)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(2i+2)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+3)}_{\mathsf{DPKE}}))$, where

$$(\mathsf{sk}^{(2i+1+j)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+1+j)}_{\mathsf{DPKE}}) \leftarrow \mathsf{Evo}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+j)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+j)}_{\mathsf{DPKE}}) \text{ for } j \in [2].$$

$\underline{\mathsf{Enc}^{(1)}(\mathsf{pk}, M, i)}$: Return $\mathsf{Enc}_{\mathsf{DPKE}}(\mathsf{pk}, M, 2i)$.

$\underline{\mathsf{Enc}^{(2)}(\mathsf{pk}, M, i)}$: Return $\mathsf{Enc}_{\mathsf{DPKE}}(\mathsf{pk}, M, 2i+1)$.

$\underline{\mathsf{Dec}^{(1)}(\mathsf{sk}^{(i)}, C)}$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}})$ and return $\mathsf{Dec}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i)}, C)$ if C was not re-encrypted. Otherwise parse C as (C_1, rk) and return $\mathsf{DelDec}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+1)}, \mathsf{rk}, C_1)$.

$\underline{\mathsf{Dec}^{(2)}(\mathsf{sk}^{(i)}, C)}$: Parse $\mathsf{sk}^{(i)}$ as $(\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}})$ and return $\mathsf{Dec}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+1)}, C)$.

$\underline{\mathsf{ReGen}(\mathsf{sk}^{(i)}_A, \mathsf{ek}^{(i)}_A, \mathsf{pk}_B)}$: Parse $(\mathsf{sk}^{(i)}, \mathsf{ek}^{(i)})$ as $((\mathsf{sk}^{(2i)}_{\mathsf{DPKE}}, \mathsf{sk}^{(2i+1)}_{\mathsf{DPKE}}), (\mathsf{ek}^{(2i)}_{\mathsf{DPKE}}, \mathsf{ek}^{(2i+1)}_{\mathsf{DPKE}}))$, and return $(\mathsf{rk}^{(i)}_{A\to B}, \mathsf{rek}^{(i)}_{A\to B})$, where

$$(\mathsf{rk}^{(i)}_{A\to B}, \mathsf{rek}^{(i)}_{A\to B}) \leftarrow \mathsf{Del}_{\mathsf{DPKE}}(\mathsf{sk}^{(2i+1)}_A, \mathsf{ek}^{(2i+1)}_A, \mathsf{pk}_B).$$

$\underline{\mathsf{ReEvo}(\mathsf{rk}^{(i)}_{A\to B}, \mathsf{rek}^{(i)}_{A\to B})}$: Return $\mathsf{DelEvo}_{\mathsf{DPKE}}(\mathsf{DelEvo}_{\mathsf{DPKE}}(\mathsf{rk}^{(i)}_{A\to B}, \mathsf{rek}^{(i)}_{A\to B}))$.

$\underline{\mathsf{ReEnc}(\mathsf{rk}^{(i)}_{A\to B}, C_A)}$: Return $(C_A, \mathsf{rk}^{(i)}_{A\to B})$.

Scheme 4. fs-PRE scheme from a fs-DPKE scheme without adaption.

In the following Theorem, we first show that Scheme 4 is indeed fs-PRE^- secure, i.e., satisfies fs-IND-CPA-1 and fs-IND-CPA-2 security, but trivially does not satisfy fs-RIND-CPA security and thus is not fs-PRE^+ secure.

Theorem 4. *Scheme 4 when instantiated with a fs-IND-CPA secure fs-*DPKE *scheme satisfies fs-IND-CPA-1 and fs-IND-CPA-2 security, but not fs-RIND-CPA security.*

Proof. We follow the same strategy as for Theorem 3 to show fs-IND-CPA-2.

- When started on pp, n and pk, run $(j^*, \mathsf{st}) \leftarrow A_2(\mathsf{pp}, \lceil \frac{n}{2}{+}1\rceil, \mathsf{pk})$. Set $j' \leftarrow 2j^*$ and return (j', st).

- When started on $\mathsf{st}, \mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$, we simulate the ReGen^h and $\mathsf{ReGen}^{h'}$ oracles using Del^h and $\mathsf{Del}^{h'}$. Indeed, Del^h and $\mathsf{Del}^{h'}$ return delegation keys for period $j' - 1 = 2j^* - 1$, which are re-encryption keys for period $j^* - 1$. Using Evo we evolve $\mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$ to period $j' + 1$. Set $(\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)}) \leftarrow ((\mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{sk}_{\mathsf{DPKE}}^{(j'+1)}), (\mathsf{ek}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j'+1)}))$ and start A_2 on $\mathsf{st}, (\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$ and simply forward the result.
- Finally, when started on st and $C_{j'-1}$, $C_{j'-1}$ is a level 2 ciphertext for $j^* - 1$. Hence we start A_2 on the ciphertext and return its' result.

To show fs-IND-CPA-1 security, we perform a similar reduction:

- When started on pp, n and pk, run $(j^*, \mathsf{st}) \leftarrow A_1(\mathsf{pp}, \lceil \frac{n}{2} + 1 \rceil, \mathsf{pk})$. Set $j' \leftarrow 2j^* - 1$ and return (j', st).
- When started on $\mathsf{st}, \mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$, we simulate the ReGen^h and $\mathsf{ReGen}^{h'}$ oracles using Del^h and $\mathsf{Del}^{h'}$ and by running DelEvo on the result. Indeed, Del^h and $\mathsf{Del}^{h'}$ return delegation keys for period $j' - 1 = 2j^* - 2$, hence after applying DelEvo we obtain re-encryption keys for period $j^* - 1$. ReGen^d is simulated honestly by delegating $\mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$ to a dishonest user. Using Evo we evolve $\mathsf{sk}_{\mathsf{DPKE}}^{(j')}, \mathsf{ek}_{\mathsf{DPKE}}^{(j')}$ to period $j' + 2$. Set $(\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)}) \leftarrow ((\mathsf{sk}_{\mathsf{DPKE}}^{(j'+1)}, \mathsf{sk}_{\mathsf{DPKE}}^{(j'+2)}), (\mathsf{ek}_{\mathsf{DPKE}}^{(j'+1)}, \mathsf{ek}_{\mathsf{DPKE}}^{(j'+2)}))$ and start A_1 on $\mathsf{st}, (\mathsf{sk}^{(j^*)}, \mathsf{ek}^{(j^*)})$ and simply forward the result.
- Finally, when started on st and $C_{j'-1}$, $C_{j'-1}$ is a level 1 ciphertext for $j^* - 1$. Hence we start A_1 on the ciphertext and return its' result.

Following the initial observation on the recoverability of delegation keys, an receiver-IND-CPA adversary is straightforward to define:

- When started on pp, n and pk, honestly generate a key $(\mathsf{pk}^*, \mathsf{sk}^{(0)}, \mathsf{ek}^{(0)}) \leftarrow \mathsf{Gen}(\mathsf{pp}, n)$ and store it in st. Choose $j^* \xleftarrow{R} [n]$ and store it together with pk in st, and return $(j^*, \mathsf{pk}^*, \mathsf{st})$.
- When started on st to output the challenge messages, choose $M_0, M_1, M_2 \xleftarrow{R} \mathcal{M}$. Invoke the ReEnc oracle as $(\cdot, \mathsf{dk}) \leftarrow \mathsf{ReEnc}(\mathsf{rk}, \mathsf{Enc}^{(2)}(\mathsf{pk}, M_2, j^*))$ and store M_0, M_1, dk in st. Return M_0, M_1, st.
- Now when started on st and the challenge ciphertext C, use dk stored in st and obtain $M \leftarrow \mathsf{DelDec}_{\mathsf{DPKE}}(\mathsf{sk}^{(2j^*+1)}, \mathsf{dk}, C)$. Check for which $i \in \{0, 1\}$ $M = M_i$ and return i.

Regardless of the chosen period the adversary always wins, rendering the scheme insecure with respect to the fs-RIND-CPA notion. □

From this theorem we obtain the following corollary:

Corollary 2. *fs-PRE$^+$ is a strictly stronger notion than fs-PRE$^-$.*

Note that this also shows that for conventional PRE scheme there is a separation between the classical security notion of PRE (PRE^-) as defined by Ateniese et al. and the PRE^+ notion.

Acknowledgments. Supported by H2020 project PRISMACLOUD, grant agreement n°644962 and by H2020 project CREDENTIAL, grant agreement n°653454. We thank all anonymous reviewers for their valuable comments.

References

1. Abdalla, M., An, J.H., Bellare, M., Namprempre, C.: From identification to signatures via the fiat-shamir transform: Minimizing assumptions for security and forward-security. In: EUROCRYPT (2002)
2. Applebaum, B., Harnik, D., Ishai, Y.: Semantic Security under Related-Key Attacks and Applications. In: ICS (2011)
3. Ateniese, G., Benson, K., Hohenberger, S.: Key-private proxy re-encryption. In: CT-RSA (2009)
4. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. In: NDSS (2005)
5. 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) (2006)
6. Bellare, M., Miner, S.K.: A forward-secure digital signature scheme. In: CRYPTO (1999)
7. Bellare, M., Yee, B.S.: Forward-security in private-key cryptography. In: CT-RSA (2003)
8. Berners-Lee, E.: Improved security notions for proxy re-encryption to enforce access control. In: LATINCRYPT (2017)
9. Blaze, M., Bleumer, G., Strauss, M.: Divertible protocols and atomic proxy cryptography. In: EUROCRYPT (1998)
10. Blazy, O., Bultel, X., Lafourcade, P.: Two secure anonymous proxy-based data storages. In: SECRYPT (2016)
11. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: CRYPTO (2004)
12. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: CRYPTO (2005)
13. Boneh, D., Waters, B.: A fully collusion resistant broadcast, trace, and revoke system. In: CCS (2006)
14. Borceaa, C., Guptaa, A.B.D., Polyakova, Y., Rohloffa, K., Ryana, G.: Picador: End-to-end encrypted publish-subscribe information distribution with proxy re-encryption. Future Generation Comp. Syst. (2016)
15. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: EUROCRYPT (2003)
16. Canetti, R., Hohenberger, S.: Chosen-ciphertext secure proxy re-encryption. In: CCS (2007)
17. Canetti, R., Raghuraman, S., Richelson, S., Vaikuntanathan, V.: Chosen-ciphertext secure fully homomorphic encryption. In: PKC 2017 (2017)
18. Chandran, N., Chase, M., Liu, F., Nishimaki, R., Xagawa, K.: Re-encryption, functional re-encryption, and multi-hop re-encryption: A framework for achieving obfuscation-based security and instantiations from lattices. In: PKC (2014)
19. Chandran, N., Chase, M., Vaikuntanathan, V.: Functional re-encryption and collusion-resistant obfuscation. In: TCC (2012)
20. Cohen, A.: What about bob? the inadequacy of cpa security for proxy reencryption. Cryptology ePrint Archive, Report 2017/785 (2017)

21. Cohen, A., Holmgren, J., Nishimaki, R., Vaikuntanathan, V., Wichs, D.: Watermarking cryptographic capabilities. In: STOC (2016)
22. Delerablée, C.: Identity-based broadcast encryption with constant size ciphertexts and private keys. In: ASIACRYPT 2007 (2007)
23. Fan, X., Liu, F.H.: Proxy re-encryption and re-signatures from lattices. Cryptology ePrint Archive, Report 2017/456 (2017)
24. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC (2009)
25. Gentry, C., Silverberg, A.: Hierarchical id-based cryptography. In: ASIACRYPT (2002)
26. Green, M., Ateniese, G.: Identity-based proxy re-encryption. In: ACNS (2007)
27. Green, M.D., Miers, I.: Forward secure asynchronous messaging from puncturable encryption. In: IEEE S&P (2015)
28. Günther, C.G.: An identity-based key-exchange protocol. In: EUROCRYPT (1989)
29. Günther, F., Hale, B., Jager, T., Lauer, S.: 0-rtt key exchange with full forward secrecy. In: EUROCRYPT (2017)
30. Hanaoka, G., Kawai, Y., Kunihiro, N., Matsuda, T., Weng, J., Zhang, R., Zhao, Y.: Generic construction of chosen ciphertext secure proxy re-encryption. In: CT-RSA (2012)
31. Hohenberger, S., Rothblum, G.N., Shelat, A., Vaikuntanathan, V.: Securely obfuscating re-encryption. J. Cryptology (2011)
32. Libert, B., Vergnaud, D.: Tracing malicious proxies in proxy re-encryption. In: Pairing (2008)
33. Libert, B., Vergnaud, D.: Unidirectional chosen-ciphertext secure proxy re-encryption. In: PKC (2008)
34. Libert, B., Vergnaud, D.: Unidirectional chosen-ciphertext secure proxy re-encryption. IEEE Trans. Information Theory (2011)
35. Myers, S., Shull, A.: Efficient hybrid proxy re-encryption for practical revocation and key rotation. Cryptology ePrint Archive, Report 2017/833 (2017)
36. Polyakov, Y., Rohloff, K., Sahu, G., Vaikuntanathan, V.: Fast proxy re-encryption for publish/subscribe systems. ACM Trans. Priv. Secur. 20(4) (2017)
37. Ren, Y., Gu, D., Wang, S., Zhang, X.: Hierarchical identity-based proxy re-encryption without random oracles. Int. J. Found. Comput. Sci. 21(6), 1049–1063 (2010)
38. Sakai, R., Furukawa, J.: Identity-based broadcast encryption. IACR Cryptology ePrint Archive (2007)
39. Tang, Q.: Type-based proxy re-encryption and its construction. In: INDOCRYPT (2008)
40. Tessaro, S., Wilson, D.A.: Bounded-collusion identity-based encryption from semantically-secure public-key encryption: Generic constructions with short ciphertexts. In: PKC (2014)
41. Weng, J., Yang, Y., Tang, Q., Deng, R.H., Bao, F.: Efficient conditional proxy re-encryption with chosen-ciphertext security. In: ISC 2009 (2009)
42. Xu, P., Xu, J., Wang, W., Jin, H., Susilo, W., Zou, D.: Generally hybrid proxy re-encryption: A secure data sharing among cryptographic clouds. In: AsiaCCS (2016)