Reusable Fuzzy Extractors for the Set Difference Metric and Adaptive Fuzzy Extractors

No Author Given

No Institute Given

Abstract. A fuzzy extractor (Dodis *et al.*, Eurocrypt 2004) is a pair of procedures that turns a noisy secret into a uniformly distributed key R. To eliminate noise, the generation procedure takes as input an enrollment value ω and outputs R and a helper string P that enables further reproduction of R from some close reading ω'.

Boyen highlighted the need for *reusable* fuzzy extractors (CCS 2004) that remain secure even when numerous calls to the generation procedure are made on a user's noisy secret. Boyen proved that any information-theoretically secure reusable fuzzy extractor is subject to strong limitations. In subsequent work, Simoens *et al.* (IEEE S&P, 2009) showed this is a practical vulnerability. Canetti *et al.* (Eurocrypt 2016) recently proposed moving to computational security and constructed a computationally secure reusable fuzzy extractor for the Hamming metric that corrects a *sublinear* fraction of errors.

In this work, we propose a different and generic approach: the idea is to separate the reusability property from key recovery. We define a new primitive called a *reusable pseudoentropic isometry* that projects an input metric space in a distance and entropy preserving manner even if applied multiple times. Generation of multiple randomized secrets Ωs via a reusable pseudoentropic isometry does not reveal information about the original fuzzy secret ω and can be used to "decorrelate" noisy versions of ω.

Given a reusable pseudoentropic isometric building a reusable fuzzy extractor is easy by 1) randomizing the noisy secret ω into Ω and 2) using a traditional fuzzy extractor to derive a secret key from Ω.

To show the promise of our framework, we construct a reusable pseudoentropic isometry for the set difference metric using composable digital lockers (Canetti and Dakdouk, Eurocrypt 2008). This construction allows us to build the first reusable fuzzy extractor that corrects a *linear* fraction of errors.

Lastly, we propose browser and device fingerprints as new authentication sources. These fingerprints are a list of features with entropy that undergo deeper variation over time than biometrics. However, they still enable user identification (Eckersley, PETS 2010). Extending reusable fuzzy extractors, we define *adaptive* fuzzy extractors to handle such sources. An adaptive fuzzy extractor enables recovery of R from ω' as long as ω' has *naturally* drifted from ω. We construct adaptive fuzzy extractors from reusable fuzzy extractors.

1 Introduction

Cryptography relies on uniformly distributed and reproducible long-term secrets to perform authentication or derive keys. Numerous high entropy randomness sources exist, such as biometrics and human-generated data [14,23], Physically Unclonable Functions (PUFs) [30] and quantum information [4]. Both PUFs and biometrics suffer the common issue of errors that prevent stable cryptographic key generation.

Randomness sources A PUF is a physical entity that is easy to evaluate but hard to predict. Unique by manufacturing process, PUFs are used to implement challenge-response authentication. Recently, researchers have attacked PUFs, creating software models for the PUF behavior [32,33]. These attacks can be avoided by first deriving stable cryptographic key from the PUF output and then creating a challenge response protocol using a function of this key.

Biometrics are unique characteristics of individuals based on either biological or behavioral characteristics. Biometrics are evaluated by their uniqueness, collectability and permanence [22]

where this last characteristic represents the period in which those traits are stable. Unlike passwords, biometrics suffer inevitable but minor variations. These variations are assumed to correspond to a bounded distance between repeated readings according to some metric. Dodis *et al.* [16] stated that Hamming distance looks like the "most natural metric to consider" [10,16,24]. However,with the exception of iris [14], set distance better suits concrete cases such as biometric matchers (e.g. digital fingerprints) or even the more exotic lover's problem [30,22,15,24,37]. Typical systems create a *template* reading from an initial reading; subsequent readings are directly compared to this initial template. These templates have privacy concerns [31,35], in the worst case a legitimate looking biometric can be reverse engineered from the template [19].

1.1 Fuzzy Extractors

Information reconciliation [4] enables retrieving identical values from noisy data. *Privacy amplification* [4] converts values with entropy into uniform random strings. Fuzzy Extractors (FEs) [16][1], are a pair of non-interactive algorithms (Gen, Rep) that simultaneously perform information reconciliation and privacy amplification. The algorithm Gen, used at enrollment, takes input ω from an entropy source and outputs a uniformly distributed key R and some public helper string P. The algorithm Rep takes the helper string P and ω' and reproduces the secret key R as long as ω' is close enough to ω relative to the distance metric, say $d(\omega, \omega') \leq t$. FEs exist with security against information-theoretic [15] or computational adversaries [18].

Metrics Dodis *et al.* proposed FE constructions for the Hamming, set difference and edit metrics drawing on prior work [24,23]. We focus on the set difference metric: inputs ω are subsets of size s of a universe $\mathcal{U}$ whose cardinality is n. For this metric, Dodis *et al.* distinguished two settings, referred as the *small* and *large* universe settings. Let s be some security parameter. In the former case, we have that $n = \texttt{poly}(s)$ while in the latter one n is superpolynomial in s. The large universe setting occurs in practice. For example, consider a list of book titles or a list of movies (movie lover's problem due to [23]). The small universe setting benefits from a reduction to the Hamming metric, referred as the bin-set equivalence (described in Section 2). We concentrate on the large universe setting where this transform is not applicable.

Reusability Boyen stated the need for *reusable* fuzzy extractors [8] for which numerous helper strings P^j from a user's fuzzy secret do not impact user's security. Boyen showed that information-theoretic FEs must leak substantial information about ω when numerous calls to Gen are made. On the positive side, Boyen demonstrated reusable security when the exclusive OR of the user's fuzzy secrets reveal no sensitive information. This is a restrictive class of correlations; we have no evidence that practical sources obey this condition. Subsequent works showed that existing FEs are not reusable in practice [6,34].

Recent work of Canetti *et al.* [10] constructs the first reusable fuzzy extractor (RFE) that makes no assumption about how repeating readings are correlated. It works for the Hamming distance and provides security against computationally bounded adversaries. It uses a strong form of symmetric encryption, called *digital lockers* [9] (Our construction will also use digital lockers but in a different way.) Their construction is secure for distributions with high entropy samples instead of global min-entropy. This is in contrast to traditional constructions that only assume the source has min-entropy.

[1] In the following, we will refer to the journal version [15].

Their main binary construction can be extended through bin-set equivalence to a FE in the set difference based metric but only in the small universe setting. Their scheme only allows an error rate (t/n) sublinear in n. Prior to this work, there were no known RFE correcting a linear error rate for any common metric. One should note that most biometrics demonstrate an error rate between 10-30%.

1.2 Authenticating with browser fingerprints

Industrial authentication solutions must make due with using software and available sensors (*e.g.* HCE payment [1]). While biometrics and PUFs have received attention in the authentication literature, these solutions can not be implemented using only software. The need for a pure software solution demands new authentication factors, we propose browser fingerprints as a new authentication factor.

Eckersley [17] showed how to create a fingerprint from characteristics of a web browser (user agent, list of fonts, list of plug-ins,...). Servers use this data to detect returning browsers even when features have changed over time. Subsequent studies [7,28,29,2] show such a system is deployable for personal computers. While early mobile solutions were insufficient [26,12,13,37], recent work of Kurtz *et al.* provides a comprehensive analysis in the mobile setting [25]. On Apple's iOS, they show how to compute a device fingerprint using 29 different configuration features. Using a dataset of 13,000 fingerprints, they show that fingerprints are unique and allow detection of returning devices with an accuracy of 97%. In their work, the list of installed applications and the top 50 songs are among the most identifying values present on a device. These fingerprints appear credible as a factor in designing a strong authentication protocol. Many of these device/browser fingerprints draw on features coming with large universes (*e.g.* songs, applications, plug-ins, ...).

Physiological biometrics undergo minor differences: if we consider an enrollment value ω, it is likely that any new reading ω' will stay within a certain distance. Browser fingerprints are different: even though they are identifying, they undergo deep variations over time. It is reasonable to expect each repeated reading to be close, that is, $d(\omega^j, \omega^{j+1}) \leq t$. However, ω^j will drift over time and $d(\omega, \omega^j) > t$.

To fairly identify users, current solutions have to handle variability of browser fingerprints. Some behavioral biometrics also suffer from a shorter permanence period. *Adaptive* biometric systems are designed to solve this problem [36]. In an adaptive biometric system, the template database is updated whenever a successful authentication occurs. Here, the goal is to decide if a given profile is a former one that has undergone variations or is a new one [17,2,25].

1.3 Our contributions

Secure sketches [15], which will be introduced later, are often used to build FEs. Both FEs and secure sketches are meant to recover a secret value from noisy inputs, the recovered value being either the noisy secret itself or an extracted cryptographic key. In both cases, noise elimination is performed using a helper data which leaks information.

FEs are mainly based on information-theoretic secure sketches and as such, prone to reusability issues [34,6]. Fuller *et al.* showed that computationally secure sketches are subject to many of the same limitations as information theoretic secure sketches [18, Theorem 3.6]. To avoid these negative results, we separate the task of reusability from the task of noise elimination. Our contributions are as follows.

1. We introduce a randomization stage captured by a new primitive we call a *pseudoentropic isometry* (PI). Informally, a PI pseudorandomly projects fuzzy secrets while maintaining distances between two noisy readings and entropy of the original secret. To be *reusable*, a PI must generate ρ uncorrelated values $\Omega^1, \ldots, \Omega^\rho$ from ρ enrollments values $\omega^1, \ldots, \omega^\rho$ drawn from the fuzzy secret ω. The reusability property is then defined as long as each Ω^{j_0} carries sufficient entropy even in presence of other Ω^js ($j_0 \neq j$). *Reusable pseudoentropic isometries* (RPIs), contrary to both FEs and secure sketches, do not perform any form of error correction and are not subject to many bounds from coding theory. These Ω^j can be used once by any nonreusable FE. If a user wants to authenticate herself toward provider j from a noisy reading ω', the RPI will generate $\Omega^{j\prime}$ where the distance between $\Omega^{j\prime}$ and Ω^j is the same as between ω^j and $\omega^{j\prime}$.
2. We show that combining a RPI and a traditional FE yields a RFE.
3. We instantiate a RPI for the set difference metric using digital lockers [9]. This RPI allows us to design the first reusable FE for linear error rates. Our construction applies for the set difference metric in the large universe setting. Our construction proceeds as follows for each element of the input set:
   (a) We sample a random point in a new metric space,
   (b) We lock the random point using the element of the input set as the key,
   When Rep is run, the fraction of unlockable points is the same as the overlap between the sets. This construction does no error-correction, it projects "randomly" while preserving distance.
4. Working with device fingerprints led to us define *Adaptive* Fuzzy Extractors (AFEs) that recover a stable key R from noisy readings that naturally drift over time. In addition to primitives Gen and Rep, an update primitive Upd is introduced. The reproduction algorithm outputs R even if the authentication value ω^i is not close to the enrolled ω as long as ω^i has drifted from ω and Upd has been run regularly. We show how to design an AFE out of a RFE and a symmetric encryption scheme. The idea is to keep a long-term key that is decrypted using the current reading of ω and reencrypted using the new reading to account for long-term drift.

2 Preliminaries

Notation $\log$ denotes the base 2 logarithm. $GF(n)$ denotes the finite field of n elements. $x \leftarrow f(.)$ denotes that x is an output of a function f. If f is randomized, we use the semicolon to make the randomness explicit. $f(x;\mu)$ is the result of f computed on x with randomness μ.

For any entity $\mathcal{E}$, we denote by $\mathcal{E}(z)$ the fact that $\mathcal{E}$ has knowledge of z. U_ℓ denotes the uniformly distributed random variable on $\{0,1\}^l$. For a distinguisher D (or a class of distinguishers $\mathcal{D}$), we write the computational distance between X and Y as $\delta^D(X,Y) = |\mathbb{E}[D(X)] - \mathbb{E}[D(Y)]|$. $\mathcal{D}_{s_{\text{sec}}}$ denotes the class of randomized circuits which output a single bit and have size at most s_{sec}. Let λ denote a security parameter. Except stated otherwise, we have $l = l(\lambda)$, $\kappa = \kappa(\lambda)$, $m = m(\lambda)$, $m_1 = m_1(\lambda)$,$m_2 = m_2(\lambda)$, $s_{\text{sec}} = \mathtt{poly}(\lambda)$ and $\epsilon_{\text{sec}} = \mathsf{negl}(\lambda)$. A metric space is a finite set $\mathcal{M}$ equipped with a distance $d : \mathcal{M} \times \mathcal{M} \to \mathbb{N}$ fulfilling the properties of symmetry, triangle inequality and zero distance between equal points.

2.1 Background

Set Difference Metric Let $\mathcal{M}$ consist of all subsets of a universe $\mathcal{U}$. For two sets ω and ω' belonging to $\mathcal{M}$, their symmetric difference is defined as $\omega \Delta \omega' \stackrel{def}{=} \{x \in \omega \cup \omega' | x \notin \omega \cap \omega'\}$. Symmetric difference is a metric that we denote by d.

Dodis *et al.* [16] noted the bin-set equivalence: if ω denotes a set, it can be viewed a binary vector in $\{0,1\}^n$, with 1 at position x if $x \in w$, and 0 otherwise. Viewed in this way, set difference can be expressed as Hamming distance between these associated vectors. This transform is not efficient when the universe size n is superpolynomial.

Entropy Notions Entropy specifies the amount of information contained in some data. In security-related contexts, we care about how well an adversary can guess the value of a random variable. In the information-theoretic case, we rely on the notion of *min-entropy*. A random variable A has min-entropy m, denoted $H_\infty(A) = m$, if A has predictability 2^{-m} *i.e.* $\max_a Pr[A = a] = 2^{-m}$. Put another way, we have $H_\infty(A) \stackrel{def}{=} -\log(\max_{a \in A} P[A = a])$. The *average* min-entropy of A given B is:

$$\tilde{H}_\infty(\mathrm{A}|\mathrm{B}) \stackrel{\text{def}}{=} -\log(\mathbb{E}_{b \in \mathrm{B}} \max_a \Pr[\mathrm{A} = a|\mathrm{B} = b]).$$

HILL entropy is a commonly used computational notion of entropy [20]. It was extended to the conditional case by Hsiao, Lu, and Reyzin [21].

Definition 1. *Let (W, S) be a pair of random variables. W has* HILL entropy *at least k conditioned on S, denoted $H^{\mathtt{HILL}}_{\epsilon_{sec}, s_{sec}}(W|S) \geq k$ if there exists a collection of distributions X_s giving rise to joint distribution (X, S), such that $\tilde{H}_\infty(X|S) \geq k$ and $\delta^{\mathcal{D}_{s_{sec}}}((W, S), (X, S)) \leq \epsilon_{sec}$.*

Fuzzy Extractors The original definition of FEs, due to Dodis *et al.* [15], was information theory-based. We focus on the computational definition introduced by Fuller *et al.* [18]. They extend their definition to an explicit family of distributions and we adopt this convention.

Definition 2 (Fuzzy Extractor). *A pair of randomized procedures "generate" (Gen) and "reproduce" (Rep) is a $(\mathcal{M}, \mathcal{W}, l, t, \delta)$-computational fuzzy extractor that is $(\epsilon_{sec}, s_{sec})$-hard if Gen and Rep satisfy the following properties:*

- *Gen on input $\omega \in \mathcal{M}$ outputs an extracted string $R \in \{0,1\}^l$ and a helper string $P \in \{0,1\}^*$.*
- *Rep takes an element $\omega' \in \mathcal{M}$ and a bit string $P \in \{0,1\}^*$ as inputs.*
- ***Correctness:** if $d(\omega, \omega') \leqslant t$ and $(R, P) \leftarrow$ Gen(ω), then $\Pr[$Rep$(\omega', P) = R] \geq 1 - \delta$ where the probability is over the coins of Gen and Rep.*
- ***Security:** for any $W \in \mathcal{W}$ on $\mathcal{M}$, $R|P$ is pseudorandom i.e. $\delta^{s_{sec}}((R, P), (U_l, P)) \leqslant \epsilon_{sec}$.*

Dodis *et al.* also define *average-case* FEs for which the security property requires that for any auxiliary variable I, $((R, P, I), (U_l, P, I))$ appear indistinguishable. We also consider FEs that are secure for all distributions of (average) min-entropy m, in this case we replace $\mathcal{W}$ with the parameter m.

Dodis *et al.* have designed FEs based on three different metrics which are Hamming, set difference and edit distances. All their constructions rely on *secure sketches*. Such a primitive is a pair of procedures (SS, Rec) where, the "sketch" procedure SS takes in ω and outputs a public string P. Later given ω' and P, procedure Rec recovers ω as long as ω' is close to ω. Coupled with an *average-case extractor*, Dodis *et al.* design FEs out of such a primitive. Since P enables to recover ω from ω', it necessarily leads to what the authors define as *entropy loss*.

Reusable Fuzzy Extractor RFEs [8] allow multiple calls to Gen on the noisy readings of ω while retaining security. Consider ρ readings $\omega^1, \ldots, \omega^\rho$ of the same fuzzy secret from which the user will be enrolled on ρ different authentication servers. Gen independently generates ρ pairs $(R^1, P^1), \ldots, (R^\rho, P^\rho)$ where $(R^j, P^j) \leftarrow \mathsf{Gen}(\omega^j)$. Canetti *et al.* [10] proposed a security model where a given R^{j_0} is secure even if all P^j and all other R^js (for $j \neq j_0$) are given to an adversary.

Definition 3 (Reusable Fuzzy Extractor [10]). *Let* $(\mathsf{Gen}, \mathsf{Rep})$ *be a* $(\mathcal{M}, \mathcal{W}, l, t)$*-FE that is* $(\epsilon_{sec}, s_{sec})$*-hard and* $W^1, W^2, \ldots, W^\rho$ *be* ρ *correlated random variables over* $\mathcal{M}$ *where* $W^j \in \mathcal{W}$ *for all* $1 \leq j \leq \rho$*. Let* D *be an adversary. Define the following game for all* $j = 1, \ldots, \rho$*:*

- ***Sampling*** *The challenger* $\mathcal{C}$ *samples* $\omega^j \leftarrow W^j$ *for all* j *and* $\eta \xleftarrow{\$} \{0,1\}^l$*.*
- ***Generation*** $\mathcal{C}$ *computes* $(R^j, P^j) \leftarrow \mathsf{Gen}(\omega^j)$ *for all* j*.*
- ***Distinguishing*** *The advantage of* D *consists in:*

$$Adv(D) \stackrel{def}{=} Pr[D(R^1, \ldots, R^\rho, \{P^j\}_{1 \leqslant j \leqslant \rho}) = 1]$$
$$-Pr[D(R^1, \ldots, R^{j-1}, \eta, R^{j+1}, \ldots, R^\rho, \{P^j\}_{1 \leqslant j \leqslant \rho}) = 1]$$

$(\mathsf{Gen}, \mathsf{Rep})$ *is* $(\epsilon_{sec}, \rho, s_{sec})$*-reusable if for all* $D \in \mathcal{D}_{s_{sec}}$ *and for all* $j = 1, \ldots, \rho$*,* $Adv(D) \leq \epsilon_{sec}$*.*

2.2 Tools

Digital Lockers Digital lockers are computationally secure symmetric encryption schemes that retain security even when used multiple times with correlated and weak (i.e., nonuniform) keys [11]. They have the additional feature that the wrong key can be recognized as such (with high probability). We use notation $\mathsf{c} = \mathsf{lock}(\mathsf{key}, \mathsf{val})$ for the algorithm that performs the locking of the value val using key, and $\mathsf{unlock}(\mathsf{key}, \mathsf{c})$ for the algorithm that performs the unlocking (which will output val if key is correct and $\perp$ with high probability otherwise).

Digital lockers can be easily constructed in the random oracle (see Lynn, Prabhakaran, and Sahai [27, Section 4]). Bitansky and Canetti [5], building on the work of [9,11], show how to obtain composable digital lockers based on a strong version of the Decisional Diffie-Hellman assumption without random oracles.

The security of digital lockers is defined via virtual-grey-box simulatability [5], a simulator is allowed unbounded running time but only a bounded number of queries to an ideal locker. Intuitively, the definition says if the keys to the ideal locker are hard to guess, the simulator will not be able to unlock the ideal locker and thus neither will the real adversary. Formally, let $\mathsf{idealUnlock}(\mathsf{key}, \mathsf{val})$ be the oracle that returns val when given key, and $\perp$ otherwise.

Definition 4. *The pair of algorithm* $(\mathsf{lock}, \mathsf{unlock})$ *with security parameter* λ *is an* ℓ*-composable secure digital locker with error* γ *if the following hold:*

- ***Correctness*** *For all* key *and* val*,* $\Pr[\mathsf{unlock}(\mathsf{key}, \mathsf{lock}(\mathsf{key}, \mathsf{val})) = \mathsf{val}] \geq 1 - \gamma$*. Furthermore, for any* $\mathsf{key}' \neq \mathsf{key}$*,* $\Pr[\mathsf{unlock}(\mathsf{key}', \mathsf{lock}(\mathsf{key}, \mathsf{val})) = \perp] \geq 1 - \gamma$*.*
- ***Security*** *For every PPT adversary* A *and every positive polynomial* p*, there exists a (possibly inefficient) simulator* S *and a polynomial* $q(\lambda)$ *such that for any sufficiently large* s*, any polynomially-long sequence of values* $(\mathsf{val}_i, \mathsf{key}_i)$ *for* $i = 1, \ldots, \ell$*, and any auxiliary input* $z \in \{0,1\}^*$*,*

$$\left| \Pr\left[A\left(z, \{\mathsf{lock}\left(\mathsf{key}_i, \mathsf{val}_i\right)\}_{i=1}^{\ell}\right) = 1 \right] - \Pr\left[S\left(z, \{|\mathsf{key}_i|, |\mathsf{val}_i|\}_{i=1}^{\ell}\right) = 1 \right] \right| \leq \frac{1}{p(\mathsf{s})}$$

where S is allowed $q(\lambda)$ oracle queries to the oracles $\{\mathsf{idealUnlock}(\mathsf{key}_i, \mathsf{val}_i)\}_{i=1}^{\ell}$.

Pseudoentropic Isometries We propose a new paradigm for designing reusable FEs. Given a fuzzy secret ω, a PI derives a random Ω that retains the entropy of ω and the distances between inputs. More precisely, a PI is a pair (RPIGen, RPIRep) defined as follows.

Definition 5 (Pseudoentropic isometry). *Let $(\mathcal{M}_1, d_1)$ and $(\mathcal{M}_2, d_2)$ be two metric spaces. A $(\mathcal{M}_1, \mathcal{M}_2, \mathcal{W}, m_2, \epsilon_{sec}, s_{sec}, \delta)$-pseudoentropic isometry is a pair of randomized procedures (RPIGen, RPIRep) with the following properties:*

1. *RPIGen on $\omega \in \mathcal{M}_1$ outputs $\Omega \in \mathcal{M}_2$ and $F \in \{0,1\}^*$.*
2. *RPIRep takes an element $\omega' \in \mathcal{M}_1$ and a bit string $F \in \{0,1\}^*$ as inputs to output $\Omega' \in \mathcal{M}_2$.*
3. ***Correctness:** if $(\Omega, F) \leftarrow$ RPIGen(ω), then $\Pr[d_2(\Omega, \Omega') \leq d_1(\omega, \omega')] \geq 1 - \delta$. where the probability is over the randomness of (RPIGen, RPIRep).*
4. ***Security:** for any distribution $W \in \mathcal{W}$, for $(U, V) \leftarrow$ RPIGen(W) we have $H^{\mathtt{HILL}}_{\epsilon_{sec}, s_{sec}}(U|V) \geqslant m_2$.*

Security implies that $H^{\mathtt{HILL}}(W|V) \geqslant m_2$ with a slight loss in parameters as the adversary can run RPIRep if they recover W.

This notion is related to biometric embeddings used in [15]. A biometric embedding projects any fingerprint value into a metric space where a FE exists while loosely maintaining distances. On their own PIs are not novel (the identity function is a PI). A reusable pseudoentropic isometry or RPI is the key to our approach. In an RPI the knowledge of previous derived values does not help D to distinguish a random value from a newly derived projection obtained via RPIGen. Drawing on the definition of reusability for FEs (Definition 3), we define a RPI as follows.

Definition 6 (RPI).

Let $W^* \in \mathcal{W}$ be a distribution. Let $W^1, W^2, \ldots, W^\rho$ be ρ correlated random variables over $\mathcal{M}_1$. Let D an adversary. Using notation of Definition 5, we define the following game for all $j = 1, \ldots, \rho$:

- ***Sampling** The challenger $\mathcal{C}$ jointly samples $\omega^j \leftarrow W^j$. Then independently samples $\omega^* \xleftarrow{\$} W^*$.*
- ***Generation** $\mathcal{C}$ generates $(\Omega^j, F^j) \leftarrow$ RPIGen(ω^j) and $(\Omega^*, F^*) \leftarrow$ RPIGen(ω^*).*
- ***Distinguishing** The advantage of D consists in:*

$$\begin{aligned} Adv(D) \stackrel{def}{=} & Pr[D(\Omega^1, \ldots, \Omega^\rho, F^1, \ldots, F^\rho) = 1] \\ & - Pr[D(\Omega^1, \ldots, \Omega^{j-1}, \Omega^*, \Omega^{j+1}, \ldots, \Omega^\rho, F^1, \ldots, F^\rho) = 1] \end{aligned}$$

(RPIGen, RPIRep) is said to be ρ-reusable if for all $D \in \mathcal{D}_{s_{sec}}$ and for all $j = 1, \ldots, \rho$, the advantage $Adv(D) \leq \epsilon_{sec}$.

3 From nonreusable to reusable Fuzzy Extractors

In this section, we introduce a new and generic way to address reusability. The idea is to first use a RPI to randomize fuzzy secrets and then apply a nonreusable FE on the unrelated projected values.

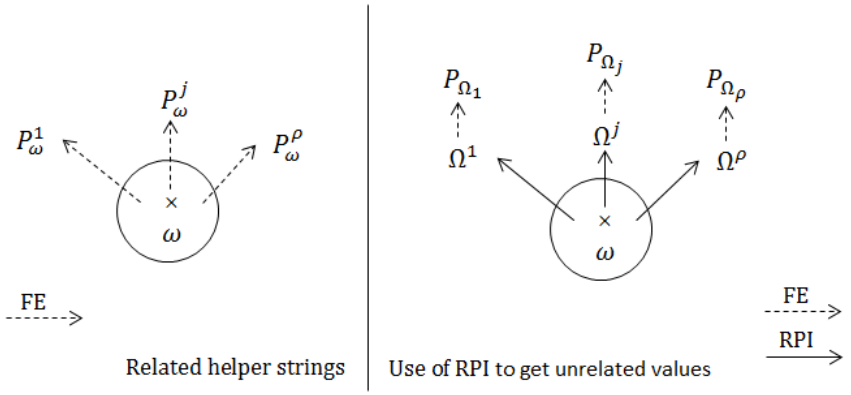

Fig. 1. Overview of reusability via RPI randomization

Generation procedure Gen	**Reproduction procedure Rep**
Input: $\omega \in \mathcal{M}_1$.	Inputs: $\omega' \in \mathcal{M}_1$,
1. $(\Omega, F) \leftarrow$ RPIGen(ω).	Helper data $P \in \{0,1\}^*$.
2. $(R, Q) \leftarrow$ Gen'(Ω).	1. Parse $P = (F, Q)$
3. Set $P = (F, Q)$.	2. $\Omega' \leftarrow$ RPIRep(ω', F).
4. Return (R, P).	3. $R \leftarrow$ Rep'(Ω', Q).
	4. Return R.

Fig. 2. A generic reusable FE

3.1 Approach

Let (Gen', Rep') denote a (*average-case*) nonreusable FE. The generation procedure Gen' implicitly draws a ball $\mathcal{B}(\omega, t)$ centered on its input ω where the radius t consists in the error tolerance of the FE. Whenever a noisy reading ω' is given to procedure Rep', the secret key will be recovered as long as ω' belongs to $\mathcal{B}(\omega, t)$.

To address reusability, we randomly project the ρ fuzzy versions of ω onto unrelated values so that each of these latter retains original entropy independently of others. By using a ρ-RPI, the user gets unrelated values $\Omega^1, \ldots, \Omega^\rho$ that will be each enrolled once, respectively toward servers $1, \ldots, \rho$. Now whenever she wants to authenticate herself toward server j from ω', the user uses the aforesaid RPI to get Ω'^j (where $d(\Omega^j, \Omega'^j) \leq d(\omega, \omega')$). This idea is illustrated in Figure 1.

Let (RPIGen, RPIRep) be a ρ-RPI from $\mathcal{M}_1$ to $\mathcal{M}_2$. Let (Gen', Rep') be an average-case FE over $\mathcal{M}_2$ correcting t errors. The generation procedure Gen will first call RPIGen to randomize the input ω into Ω. The nonreusable FE is then applied on Ω. The RPI ensures that $d_2(\Omega, \Omega') \leq d_1(\omega, \omega')$ while the correctness of the underlying nonreusable FE ensures that Rep' recovers R from Ω' and the associated helper string as long as $d_2(\Omega, \Omega') \leqslant t$. Overall this leads to recovering R as long as $d_1(\omega, \omega') \leqslant t$.

Theorem 1. *Let (RPIGen, RPIRep) be a* $(\mathcal{M}_1, \mathcal{M}_2, \mathcal{W}, m_2, \epsilon_{RPI}, s_{RPI}, \delta_{RPI})$*-RPI that is* ρ*-reusable and (Gen', Rep') be an average-case* $(\mathcal{M}_2, m_2, l, t, \delta_{FE})$*-FE that is* (ϵ_{FE}, s_{FE})*-hard. Then Figure 2 defines a* $(\mathcal{M}_1, \mathcal{W}, l, t, \delta_{RPI} + \delta_{FE})$*-FE that is* $(\rho, \epsilon_{sec}, s_{sec})$*-reusable for* $\epsilon_{sec} = 4\epsilon_{RPI} + \epsilon_{FE}$ *and* $s_{sec} = \min\{s_{RPI} - |\mathsf{Gen}'|, s_{FE}\}$.

Proof. The correctness is straightforward and follows from aforesaid explanations. To ensure security, we first show that R appears pseudorandom even in presence of P and then treat reusability.

Under notation of Definition 5, we have that Ω and F respectively come from distribution U and V such as $H^{\mathtt{HILL}}_{\epsilon_{\mathrm{RPI}},s_{\mathrm{RPI}}}(U|V) \geqslant m_2$. We first show that FEs work on distributions with HILL entropy. The proof is delayed until Appendix A.

Lemma 1. *Let U, V be a joint distribution where $H^{\mathtt{HILL}}_{\epsilon_{RPI},s_{RPI}}(U|V) \geqslant m_2$ and let (Gen', Rep') be an average-case $(\mathcal{M}_2, m_2, l, t)$-FE that is (ϵ_{FE}, s_{FE})-hard. Define $R, P \leftarrow \mathsf{Gen}'(U)$, then*

$$\delta^{\mathcal{D}_s}((R, P, V), (U_l, P, V)) \leqslant \epsilon.$$

for $\epsilon = 2\epsilon_{RPI} + \epsilon_{FE}$ and $s = \min\{s_{RPI}, s_{FE}\}$.

Lemma 1 allows us to conclude that $\delta^{\mathcal{D}_s}((R, Q, F), (U_l, Q, F)) \leqslant \epsilon$. That is,

$$\delta^{\mathcal{D}_s}((R, P), (U_l, P)) \leqslant \epsilon$$

for $P = (F, Q)$, and aforesaid parameters $\epsilon = 2\epsilon_{\mathrm{RPI}} + \epsilon_{\mathrm{FE}}$, $s = \min\{s_{\mathrm{RPI}}, s_{\mathrm{FE}}\}$.

Reusability Let $W^1, \ldots, W^\rho$ be correlated distributions over $\mathcal{M}_1$, where $W^j \in \mathcal{W}$ for all j. The following games consist in a challenger $\mathcal{C}$ trying to fool D for some distinguished i_0:

$\mathcal{G}_0$ $\mathcal{C}$ honestly samples values as prescribed in Definition 3 and sends

$$(R^1, F^1, Q^1), \ldots, (R^{i_0}, F^{i_0}, Q^{i_0}), \ldots, (R^\rho, F^\rho, Q^\rho)$$

to D.

$\mathcal{G}_1$ In this game, there is one change compared to the previous one. $\mathcal{C}$:

1. Samples the ω^js and then uses RPIGen to obtain $(\Omega^1, F^1), \ldots, (\Omega^\rho, F^\rho)$.
2. Replaces ω^{i_0} with random $\omega^* \leftarrow W^*$ (where W^* as prescribed in Definition 6).
3. Computes $(\Omega^*, F^*) \leftarrow \mathsf{RPIGen}(\omega^*)$ and $(R^*, Q^*) \leftarrow \mathsf{Gen}'(\Omega^*)$.
4. Sets $P^* = (F^{i_0}, Q^*)$.
5. Gives D the actual R^js and P^js except for $j = i_0$ for which he receives (R^*, P^*).

If D can distinguish this game from the previous one, he would then be able to distinguish the distribution with Ω^{i_0} from the one with Ω^*. This breaks the reusability of the RPI. That is, $\mathcal{G}_1$ appears indistinguishable from $\mathcal{G}_0$ for $\epsilon = \epsilon_{\mathrm{RPI}}$ and $s = s_{\mathrm{RPI}} - |\mathsf{Gen}'|$.

$\mathcal{G}_2$ In this game, after computing $(R^*, Q^*) \leftarrow \mathsf{Gen}'(\Omega^*)$, $\mathcal{C}$ discards the value R^* and replaces it with some $\eta \xleftarrow{\$} \{0,1\}^l$ randomly sampled. Since $H^{\mathtt{HILL}}_{\epsilon_{\mathrm{RPI}},s_{\mathrm{RPI}}}(\Omega^*|F^*) \geq m_2$ then $H^{\mathtt{HILL}}_{\epsilon_{\mathrm{RPI}},s_{\mathrm{RPI}}}(\Omega^*|F^{i_0}) \geq m_2$. Thus by Lemma 1, (U_l, P^*) and (R^*, P^*) are computationally indistinguishable. Hence, this game is indistinguishable from the previous one for $\epsilon = 2\epsilon_{\mathrm{RPI}} + \epsilon_{\mathrm{FE}}$ and $s = \min\{s_{\mathrm{RPI}}, s_{\mathrm{FE}}\}$.

$\mathcal{G}_3$ In the previous game, D was given $(R^1, F^1, Q^1), \ldots, (\eta, F^{i_0}, Q^*), \ldots, (R^\rho, F^\rho, Q^\rho)$ where η is random and does not depend on P^*. In this game, $\mathcal{C}$ sends the actual Q^{i_0} (obtained via computed $\mathsf{Gen}'(\Omega^{i_0})$ instead of Q^*.

If D can distinguish that Q^{i_0} has been given instead of Q^* (obtained via computed $\mathsf{Gen}'(\Omega^*)$, he can in particular distinguish Ω^{i_0} from Ω^*. Hence, he can distinguish

$$(\Omega^1, \ldots, \Omega^{i_0}, \ldots, \Omega^\rho, F^1, \ldots, F^{i_0}, \ldots, F^\rho)$$

Algorithm RPIGen
Input: $\omega = \{\omega_1, \ldots, \omega_s\}$,
$\forall 1 \leqslant i \leqslant s,\ \omega_i \in \mathcal{U}$.
1. For $i = 1 \ldots s$,
$x_i \xleftarrow{\$} \mathcal{M}_\kappa$.
$c_i = \mathsf{lock}(\omega_i, x_i)$.
2. Set $\Omega = \{x_1, \ldots, x_s\}$
and $c = c_1, \ldots, c_s$.
3. Return (c, Ω).

Algorithm RPIRep
Inputs: $\omega' = \{\omega'_1, \ldots, \omega'_s\}$,
$c = c_1, ..., c_s$.
1. For $i = 1 \ldots s$,
$x'_i \leftarrow \mathsf{unlock}(\omega'_i, c_i)$.
2. Set $\Omega' = \{x'_1, \ldots, x'_{s'}\} \setminus \{\perp\}$.
3. While $|\Omega'| < s$,
$z \xleftarrow{\$} \mathcal{M}_\kappa$.
$\Omega' \cup \{z\}$.
4. Return Ω'.

Fig. 3. A set difference-based RPI

from

$$(\Omega^1, \ldots, \Omega^{i_0-1}, \Omega^*, \Omega^{i_0+1}, \ldots, \Omega^\rho, F^1, \ldots, F^{i_0}, \ldots, F^\rho).$$

This contradicts the reusability of the RPI. Thus, $\mathcal{G}_3$ is indistinguishable from $\mathcal{G}_2$ for $\epsilon = \epsilon_{\text{RPI}}$ and $s = s_{\text{RPI}} - |\mathsf{Gen}'|$.

In $\mathcal{G}_3$, D is given $(R^1, P^1), \ldots, (\eta, P^{i_0}), \ldots, (R^\rho, P^\rho)$ where η is randomly sampled. By transitivity, this latter game is indistinguishable from $\mathcal{G}_0$. This latter indistinguishability is exactly the one required by Definition 3.

3.2 A Set Difference-based RPI

In this subsection, we present a set difference-based RPI that will enable us to instantiate our methodology described in previous subsection.

Environment and Notation Set difference based FEs in [15] take as inputs subsets of a universe $\mathcal{U}$ with $n = |\mathcal{U}|$. We denote $(\mathcal{M}_\mathcal{U}, d)$, the metric space $\mathcal{M}_\mathcal{U}$ consisting of all the subsets of $\mathcal{U}$ with the set difference metric d. Let $\mathcal{M}_{\mathcal{U},s}$ denote the restriction of $\mathcal{M}_\mathcal{U}$ to s-elements subsets. $\mathcal{M}_\kappa$ denotes $(GF(2^\kappa), d)$ equipped with the set difference metric d. Similarly $\mathcal{M}_{\kappa,s}$ denotes the restriction to sets of sizes s. Let W be a probability distribution over $\mathcal{U}$ with min-entropy m. We use digital lockers to construct our set difference-based RPI. Our construction, presented in Figure 3, randomizes each set element using a digital locker.

It is possible to have a collision in step 2 of Algorithm RPIGen, however by choosing κ big enough this occurs with negligible probability. Step 3 of Algorithm RPIRep adds additional elements to ensure that the output set is of size s. This step can be triggered if there was a collision in RPIGen or if unlock outputs $\perp$. The only time this step makes $d(\Omega, \Omega') \geq d(\omega, \omega')$ is when unlock outputs $\perp$ when the two values actually match. The condition we require on distributions is that each set has superlogarithmic min-entropy.

Theorem 2. *Let λ be a security parameter and let $\kappa = \omega(\log \lambda)$. Let $\mathcal{W}$ be the set of all joint distributions $W_1, W_2, \ldots W_s$ where, for any $i \leqslant s$, $H(W_i) \geq m$. Let $(\mathsf{lock}, \mathsf{unlock})$ be a $(s \cdot \rho)$-composable digital locker with error δ. Then for any $s_{sec} = \mathtt{poly}(\lambda)$ there exists a $\epsilon_{sec} = \mathtt{ngl}(\lambda)$ such that Figure 3 defines a $(\mathcal{M}_{\mathcal{U},s}, \mathcal{M}_\kappa, \mathcal{W}, s \cdot \kappa, \epsilon_{sec}, s_{sec}, s \cdot \delta)$-RPI for the set difference metric where $m_2 = \kappa \cdot s$ for $\epsilon = q(q+1)2^{-\kappa} + q2^{-m_2} = \mathtt{ngl}(\lambda)$.*

Our proof is similar in spirit to the proof of Canetti *et al.* [10]. We first prove a simpler proposition that the construction is a PI and then consider reusability.

Proposition 1. *Let λ be a security parameter and let $\kappa = \omega(\log \lambda)$. Let $\mathcal{W}$ be the set of all joint distributions $W_1, W_2, \ldots W_s$ where, for any $i \leqslant s$, $H(W_i) \geq m$. Let* $(\mathsf{lock}, \mathsf{unlock})$ *be a s-composable digital locker with error δ. Then for any $s_{sec} = \mathtt{poly}(\lambda)$ there exists a $\epsilon_{sec} = \mathtt{ngl}(\lambda)$ such that Figure 3 defines a $(\mathcal{M}_{\mathcal{U},s}, \mathcal{M}_\kappa, \mathcal{W}, s \cdot \kappa, \epsilon_{sec}, s_{sec}, s \cdot \delta)$-PI for the set difference metric.*

Proof. We have to prove both isometric and security properties.

Isometry property. Ω is of size s. For any $\omega_i = \omega_i'$, if no digital locker outputs $\perp$ then $x_i = x_i'$. Thus,

$$\Pr[d(\Omega, \Omega') \leq d(\omega, \omega')] \geq 1 - s\delta.$$

Security. Our goal is to show that for all $s_{sec} = \mathtt{poly}(\lambda)$ there exists $\epsilon_{sec} = \mathtt{ngl}(\lambda)$ such that $\delta^{\mathcal{D}_{s_{sec}}}((R, P), (U, P)) \leq \epsilon_{sec}$. Fix some polynomial s_{sec} and let D be a distinguisher of size at most s_{sec}. We want to bound

$$|\,\mathbb{E}[D(\Omega, P)] - \mathbb{E}[D(U_{\mathcal{M}_\kappa}, P)]|$$

by a negligible function.

We proceed by contradiction: suppose this difference is not negligible. That is, suppose that there is some polynomial $p(\cdot)$ such that for all λ_0 there exists some $\lambda > \lambda_0$ such that

$$|\,\mathbb{E}[D(\Omega, P)] - \mathbb{E}[D(U_{\mathcal{M}_\kappa}, P)]| > 1/p(\lambda).$$

Note that λ is a function of λ_0 but we omit this notation for the remainder of the proof. By the security of digital lockers (Definition 4), there is a polynomial q and an unbounded time simulator S (making at most $q(\lambda)$ queries to the oracles $\{\mathsf{idealUnlock}(\omega_i, x_i)\}_{i=1}^s$) such that

$$\left|\mathbb{E}[D(\Omega, C_1, ..., C_s)] - \mathbb{E}\left[S^{\{\mathsf{idealUnlock}(\omega_i, x_i)\}_{i=1}^s}(\Omega, \kappa)\right]\right| \leq \frac{1}{3p(\lambda)}. \tag{1}$$

The same is true if we replaced Ω above by an independent uniform random variable U over $\mathcal{M}_\kappa$. We now prove the following lemma, which shows that S cannot distinguish between Ω and $U_{\mathcal{M}_\kappa}$.

Lemma 2. *Let U denote the uniform distribution over $\mathcal{M}_\kappa$. Then*

$$\left|\,\mathbb{E}\left[S^{\{\mathsf{idealUnlock}(\omega_i, x_i)\}_{i=1}^s}(R, \kappa)\right] - \mathbb{E}\left[S^{\{\mathsf{idealUnlock}(\omega_i, x_i)\}_{i=1}^s}(U_{\mathcal{M}_\kappa}, \kappa)\right]\,\right| \leq \frac{q(q+1)}{2^m} \leq \frac{1}{3p(\lambda)}, \tag{2}$$

where q is the maximum number of queries S can make.

Proof. Fix any $u \in \mathcal{M}_\kappa$ (the lemma will follow by averaging over all u). Let Ω^* be the correct value of Ω. The only information that S can learn about whether the value is Ω^* or u is through the query responses. First, modify S slightly to quit immediately if it gets a response not equal to $\perp$ (we assume such as soon as S gets back a non-$\perp$ response it distinguishes with probability 1). There are $q + 1$ possible values for the view of S on a given input (q of those views consist of some number of $\perp$ responses followed by the first non-$\perp$ response, and one view has all q responses equal to $\perp$). By [15, Lemma 2.2b], $\tilde{\mathrm{H}}_\infty(V_i | View(S), \{j_{ik}\}) \geq \tilde{\mathrm{H}}_\infty(V_j | \{j_{ik}\}) - \log(q+1) \geq m - \log(q+1)$. Therefore, at each query, the probability that S gets a non-$\perp$ answer is at most $(q+1)2^{-m}$. Since there are q queries of S, the overall probability is at most $q(q+1)/2^m$. Then since 2^m is $\mathtt{ngl}(\lambda)$, there exists some λ such that for all $\lambda > \lambda_0$, $q(q+1)/2^m \leq 1/(3p(\lambda))$. This completes the proof of Lemma 2.

Adding together Equation 1, Equation 2, and Equation 1 in which Ω is replaced with $U_{\mathcal{M}_\kappa}$, we obtain that

$$\delta^D((\Omega, P), (U_{\mathcal{M}_\kappa}, P)) \leq \frac{1}{p(\lambda)}.$$

This is a contradiction and completes the proof of Proposition 1.

Reusability Reusability follows from the security of digital lockers. For each $i \in \{1, ..., \rho\}$, we can treat the outputs $\Omega^1, \ldots, \Omega^{i-1}, \Omega^{i+1}, \ldots, \Omega^\rho$ as auxiliary input to the digital locker adversary. The result follows by simulatability of this adversary, but requires additional composability from the digital locker.

Corollary 1. *Let λ be a security parameter and suppose there exists $(\mathsf{lock}, \mathsf{unlock})$ with that is ℓ composable for any $\ell = \mathtt{poly}(\lambda)$ with error $\delta = \mathtt{ngl}(\lambda)$. Using the RPI defined in Figure 3 for the family $\mathcal{W}$ defined above one can construct a reusable FE for any $s_{sec} = \mathtt{poly}(\lambda), \rho = \mathtt{poly}(\lambda)$ such that $\epsilon_{sec} = \mathtt{ngl}(\lambda)$ and where $t = \Theta(n)$.*

Discussion Our instantiation of an RPI for the set difference metric (large universe) allows construction of the first RFE correcting a linear error rate that makes no assumption about how individual readings are correlated. The previous work of Boyen [8] assumed that the exclusive OR of two repeated enrollments leaked no information. The recent work of Canetti *et al.* [10] only achieves a sublinear error rate (and works for Hamming or set difference in the small universe setting).

The efficiency of our construction is bounded by the efficiency of digital lockers, we do not expect the use of information-theoretic FEs to be a roadblock to practical efficiency.

It is also worth noting that Figure 3 defines a RPI that could handle sets of variable sizes; if ω' is such as $|\omega'| = s' \neq s$, it suffices to replace s by s' in while loop (step 3 of RPIRep). Then, one could couple this RPI with any nonreusable FE that can handle sets of variable sizes (see [15]) to obtain such a RFE.

4 Adaptive Fuzzy Extractors

As discussed in the Introduction we seek to use browser and device fingerprints [17,2,25] as authentication sources. Browser fingerprints naturally evolve over time leading to drift of values. We define *Adaptive* Fuzzy Extractors to address this problem.

Given a fingerprint value ω', the goal is to decide if this fingerprint value is a new user or a previously encountered one that has undergone variations. In the latter case, the user has to be recognized and a new profile should not be created. Recent works have shown this can be accomplished using matching techniques [2,17,25]. Considered in the context of authentication, it is a natural question if a stable key can be derived from these values. In our context, this amounts to creating a FE that can recover the actual key R even if the authentication value ω' and the enrollment one ω present more than t errors. The idea is to say that ω' should have naturally drifted from ω.

Definition 7. *Let $(\mathcal{M}, d)$ a metric space. Let $\omega^1, \ldots, \omega^\phi$ elements of $\mathcal{M}$ and an integer u. We say that $(\mathcal{M}, \omega^1, \omega^\phi, u)$ is a u-drift of length ϕ on $\mathcal{M}$ if for all $i = 1, \ldots, \phi - 1$, we have that $d(\omega^i, \omega^{i+1}) \leqslant u$.*

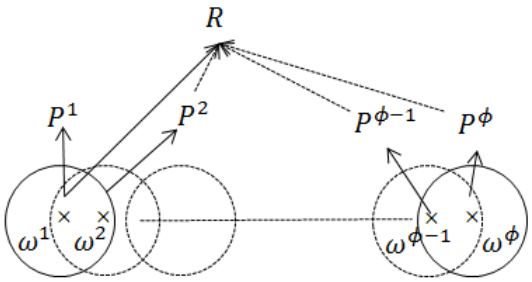

Fig. 4. Philosophy of Adaptive Fuzzy Extractors.

Since this definition is a first attempt to capture fingerprinting drift, alternative definitions may be better suited for other settings.

A naive answer to the fingerprint drift issue could be frequent re-enrollments. In practice, enrollment sessions constitute critical sessions that organizations want to avoid. FEs were designed to enable the use of long term secrets; frequent re-enrollment sessions annihilate their primary goal.

4.1 High Level Overview

We define *Adaptive* Fuzzy Extractors (AFEs) that add a third primitive Upd to classic FEs. Without re-enrolling herself, a user should be able reproduce the same secret R as the one computed by Gen as long as variations between the enrollment value ω and the authentication one ω' follow an expected u-drift (Definition 7). In other words, an AFE must recover the actual R as long as ω' has somewhat *naturally* drifted from ω although ω' does not belong to $\mathcal{B}(\omega, t)$. Given parameters $0 \leqslant u \leqslant t$, we define AFEs to work according to two zones:

- *Updating Zone.* It can update the helper string value P before too many errors occur i.e. while ω' is still close enough to ω ($d(\omega, \omega') \leqslant u$).
- *Recovering Zone.* ω' is close enough to ω to enable key recovery but too far away to enable any helper string update ($u < d(\omega, \omega') \leqslant t$).

An AFE can recover R as long as the fuzzy values define an u-drift (if an update is performed for each ω^i). Without representing updating zones for the sake of clarity, the philosophy of AFEs is depicted in Figure 4.

4.2 Definition and Security Model

Definition 8 (Adaptive Fuzzy Extractor). *A triple of randomized procedures "generate" (Gen), "update" (Upd), "reproduce" (Rep) is an $(\mathcal{M}, \mathcal{W}, l, u, t, \delta, \phi)$-adaptive fuzzy extractor that is $(\epsilon_{sec}, s_{sec})$ hard if the following holds:*

1. *On input $\omega \in \mathcal{M}$, Gen outputs an extracted string $R \in \{0,1\}^l$ and helper string $P \in \{0,1\}^*$.*
2. *On input $\omega' \in \mathcal{M}$ and $P \in \{0,1\}^*$, if either:*

   (a) $(R, P) \leftarrow$ Gen(ω) and $d(\omega, \omega') \leqslant u$;

(b) there exists (ω^*, P^*) *such that* $P \leftarrow \mathsf{Upd}(\omega^*, P^*)$, $R \leftarrow \mathsf{Rep}(\omega^*, P^*)$ *and* $(\omega^*, \omega') \leqslant u$, *then* Upd *outputs an updated helper string* P'.

3. *The reproduction procedure* Rep *takes as inputs* $\omega' \in \mathcal{M}$ *and a bit string* $P \in \{0,1\}^*$. *The correctness guarantees that if either of conditions is satisfied:*
   (a) $(R, P) \leftarrow \mathsf{Gen}(\omega)$ *and* $d(\omega, \omega') \leqslant t$;
   (b) there exists (ω^*, P^*) *such that* $P \leftarrow \mathsf{Upd}(\omega^*, P^*)$, $R \leftarrow \mathsf{Rep}(\omega^*, P^*)$ *and* $(\omega^*, \omega') \leqslant t$, *then*
   $$\Pr[\mathsf{Rep}(\omega', P) = R] \geq 1 - \delta.$$
   In any other case, no guarantee is provided about the output of Rep.
4. *The security property guarantees that for correlated distributions* $W^1, \ldots, W^\phi$ *on* $\mathcal{M}$ *where* $W^i \in \mathcal{W}$, *the string* R *is pseudorandom even for those whose observe the* P^i*s for* $i = 1 \ldots \phi$, *generated through* Gen *or* Upd *from* $\omega^i \xleftarrow{\$} W^i$. *That is,*
   $$\delta^{\mathcal{D}_{s_{sec}}}((R, \{P^i\}_{i=1}^{\phi}), (U_l, \{P^i\}_{i=1}^{\phi})) \leq \epsilon_{sec}.$$

Security Model We now define reusability of AFEs. By taking the previous scenario where a user subscribes to ρ providers, this user should now be able to update ϕ times its profile on each server to handle variations. Adapting reusability definition of [10] (Definition 3), we propose the following definition for security of reusable AFEs.

Definition 9. *Let* $(W^{1,1}, W^{2,1}, \ldots, W^{\phi,\rho})$ *be potentially correlated random variables over* $\mathcal{M}$ *where each* $W^{i,j} \in \mathcal{W}$. *Let (*Gen, Upd, Rep*) be a* $(\mathcal{M}, \mathcal{W}, \ell, u, t, \delta, \phi)$*-AFE that is* $(\epsilon_{sec}, s_{sec})$ *hard. Let* D *an adversary. Let the following game for all* $1 \leqslant j \leqslant \rho$*:*

- ***Sampling*** *The challenger jointly samples* $\omega^{1,k} \leftarrow W^{1,k}$ *for* $1 \leq k \leq \rho$, $\eta \in \{0,1\}^l$.
- ***Helper string Generation and Drifting*** *The challenger computes helper strings via "generation" and "update" procedures.*
  - $(R^k, P^{1,k}) \leftarrow \mathsf{Gen}(\omega^{1,k})$
  - *for* $2 \leqslant i \leqslant \phi$ *:* $P^{i,k} \leftarrow \mathsf{Upd}(\omega^{i,k}, P^{i-1,k})$.
- ***Distinguishing*** *The advantage of* D *is*

$$\begin{aligned} Adv(D) = &\Pr[D(R^1, \ldots, R^\rho, \{P^{i,k}\}_{i \leqslant \phi, k \leqslant \rho}) = 1] \\ &- \Pr[D(R^0, \ldots, R^{j-1}, \eta, R^{j+1}, \ldots, R^\rho, \{P^{i,k}\}_{i \leqslant \phi, k \leqslant \phi}) = 1] \end{aligned}$$

(Gen, Upd, Rep*) is* ρ*-reusable if for all* $D \in \mathcal{D}_{s_{sec}}$, *the advantage is at most* ϵ_{sec}.

Taking $\phi = 1$ leads to the reusability game of classic FEs (Definition 3) while taking $\rho = 1$ leads to the adaptive fuzzy extractor game (Definition 8).

5 From reusable Fuzzy Extractors to reusable Adaptive Fuzzy Extractors

To obtain a reusable AFE, the computed helper strings should not leak information about the user's fuzzy secret(s). To address this we use a RFE (that may be instantiated using an RPI). Most FEs recover an extracted R from ω' as long as this latter is close enough to the enrollment value ω. To continuously recover a key in spite of fingerprint derivation, the key idea is to generate a random

stable key R that will be locked under keys with shorter lifespan. The temporary keys will be the ones outputted by the (non adaptive) RFE.

We use a symmetric encryption scheme (Enc, Dec). The notion of security we require is the "find-then-guess" chosen plaintext attack (FTG-CPA) security due to Bellare *et al.* [3]. This notion is analogous to public key CPA security and defines an encryption oracle ($\mathcal{O}^{\text{encrypt}}$) since one cannot encrypt messages on its own in the private key paradigm.
Let a challenger $\mathcal{C}$ with secret key k. Adversary D queries encryptions of its choice to $\mathcal{O}^{\text{encrypt}}$. At some point, he sends m_0,m_1 to $\mathcal{C}$ that will encrypt m_b. D is asked to recover b. (Enc, Dec) is said to be $(\epsilon_{\text{CPA}}, s_{\text{CPA}})$-hard if for all $D \in \mathcal{D}_{s_{\text{CPA}}}$, $\text{Adv}^{\text{FTG-CPA}}_{\mathcal{C},D}(\lambda) \stackrel{\text{def}}{=} \Pr[D(\text{Exp}^{\text{FTG-CPA-1}}_{\mathcal{C},D}(\lambda)) = 1] - \Pr[D(\text{Exp}^{\text{FTG-CPA-0}}_{\mathcal{C},D}(\lambda)) = 1] \leqslant \epsilon_{\text{CPA}}$.

Experiment $\text{Exp}^{\text{FTG-CPA-b}}_{\mathcal{C},D}(\lambda)$
1. $(m_0, m_1) \leftarrow D(\mathcal{O}^{\text{encrypt(k,.)}})$
2. $c_b \leftarrow \text{Enc}(\text{k}, m_b)$
3. $b' \leftarrow D(c_b : \mathcal{O}^{\text{encrypt(k,.)}})$
4. Return b'.

Let Enc, Dec be a $(\epsilon_{\text{CPA}}, s_{\text{CPA}})$-FTG-CPA secure and let $(\text{Gen}'_{\text{u}}, \text{Rep}'_{\text{u}})$ be a $(\mathcal{M}, \mathcal{W}, \ell, u, \delta_{FE})$-RFE that is $(2\rho \cdot \phi, \epsilon_{FE}, s_{FE})$ reusable. We assume that Gen′ can be instantiated with distance u or t and this does not effect reusability. Figure 5 depicts how to design a reusable AFE out of these tools.

5.1 Generation procedure

Given an enrollment value ω, Gen'$_u$ of the RFE correcting u errors will then be applied to produce K_u and Gen'$_t$ to produce K_t. The key K_u will be used to detect if a fingerprint still belongs to the updating zone while the second one will be used to lock the randomly generated stable key R. In addition to helper strings, the overall helper string of our AFE contains encryptions of "1" and of R, respectively under K_u and K_t.

5.2 Update procedure

The update procedure takes as inputs a fuzzy version ω' and some helper data $P \in \{0,1\}^*$ to be updated into P'. The first step rederives K_u and K_t. Successful decryption of c_u under K_u indicates that ω' is within distance u. If so, R can be unlocked. Gen' then computes new temporary keys K'_u and K'_t along with new helper strings. R is finally re-encrypted under K'_t.

5.3 Reproduction procedure

The reproduction procedure is straightforward. Taking as inputs ω' and some P, Rep'$_t$ recovers K_t which enables to finally unlock R. We defer analysis to Appendix B.

Theorem 3. *Figure 5 defines a $(\mathcal{M}_1, \mathcal{W}, \ell, u, t, \phi, \delta)$-AFE that is $(\epsilon_{sec}, s_{sec})$ hard and ρ-reusable for $\epsilon = 2\phi(\epsilon_{CPA} + \epsilon_{FE}), s_{sec} = \min\{s_{CPA}, s_{FE}\}, \delta = 2\rho\delta_{FE}$.*

Remark 1. $2.\rho.\phi$ reusability for a RPI means that there exists $2.\rho.\phi$ balls of radius t that lead (or have led during a certain period of time) to a successful authentication. Parameters have to be chosen so that such values remain very unlikely to be randomly predicted by any adversary. Recall that each of these balls is usually of exponential size in the distance parameter (either u or t).

Generation procedure Gen
Input: $\omega \in \mathcal{M}_1$.

1. *Use a reusable FE*
$(K_u, Q_u) \leftarrow \mathsf{Gen}'_u(\omega)$.
$(K_t, Q_t) \leftarrow \mathsf{Gen}'_t(\omega)$.

2. *Generate Key*
$R \xleftarrow{\$} \{0,1\}^l$.

3. *Generate Helper Data*
$c_u = \mathsf{Enc}(K_u, 1)$,
$c_t = \mathsf{Enc}(K_t, R)$,

Set $P = (Q_u, c_u), (Q_t, c_t)$.
Return (R, P).

Update procedure Upd
Inputs: $\omega' \in \mathcal{M}_1$, $P \in \{0,1\}^*$.

1. *Check Fingerprint*
Parse $P = (Q_u, c_u), (Q_t, c_t)$.
$K_u \leftarrow \mathsf{Rep}'(\omega', Q_u)$.
$K_t \leftarrow \mathsf{Rep}'(\omega', Q_t)$.
$b \leftarrow \mathsf{Dec}(K_u, c_u)$
If b $\neq 1$, return $\perp$.
$R \leftarrow \mathsf{Dec}(K_t, c_t)$.

2. *Regenerate Helper Data*
$(K'_u, Q'_u) \leftarrow \mathsf{Gen}'_u(\omega')$.
$(K'_t, Q'_t) \leftarrow \mathsf{Gen}'_t(\omega')$.
$c'_u = \mathsf{Enc}(K'_u, 1)$,
$c'_t = \mathsf{Enc}(K'_t, R)$.

3. Set $P' = (Q'_u, c'_u), (Q'_t, c'_t)$.
Return P'.

Reproduction Procedure Rep
Inputs: $\omega' \in \mathcal{M}_1$, $P \in \{0,1\}^*$.

1. *Parse Helper Data*
Parse $P = (P_u, c_u), (P_t, c_t)$.

2. *Reproduce Key*
$K_t \leftarrow \mathsf{Rep}'(\omega', P_t)$.
$R \leftarrow \mathsf{Dec}(K_t, c_t)$.

Return R.

Fig. 5. Generation, Update, and Reproduce procedures

6 Conclusion and Future Works

We show the first reusable fuzzy extractor for the set difference metric. Our construction is also the first reusable fuzzy extractor handling a linear error rate that makes no assumption about how repeated readings are correlated.

Our set difference-based solution is an instantiation of a general framework in which we propose to randomize fuzzy secrets before applying fuzzy extractors. Since fuzzy secrets may come from correlated distributions, the idea is to decorrelate them while preserving entropy and distances: we introduced the concept of *Reusable Pseudoentropic Isometries* (RPIs) for such a purpose. We then designed Reusable Fuzzy Extractors out of any efficient nonreusable Fuzzy Extractors and RPIs. We use digital lockers to construct a RPI for the set difference metric.

We also propose the notion of *Adaptive* Fuzzy Extractors, which make sense for sources that drift over time including device and browser fingerprints. Device fingerprinting is an expanding field for which values (*e.g.* favorite songs, installed applications, plug-ins, general settings, fonts, ...) often appear in the form of lists with elements coming from a big universe. Adaptive Fuzzy Extractors are meant to capture these variations while still enabling generation of a long-term stable key. We also construct a set difference adaptive fuzzy extractor out of a reusable fuzzy extractor and a symmetric encryption scheme.

References

1. Host-based card emulation. https://developer.android.com/guide/topics/connectivity/nfc/hce.html.
2. G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 674–689, New York, NY, USA, 2014. ACM.
3. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, FOCS '97, pages 394–, Washington, DC, USA, 1997. IEEE Computer Society.
4. C. H. Bennett, G. Brassard, and J.-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.
5. N. Bitansky and R. Canetti. On strong simulation and composable point obfuscation. In *Advances in Cryptology–CRYPTO 2010*, pages 520–537. Springer, 2010.

6. M. Blanton and M. Aliasgari. Analysis of reusability of secure sketches and fuzzy extractors. *IEEE Trans. Information Forensics and Security*, 8(9):1433–1445, 2013.
7. K. Boda, A. M. Földes, G. G. Gulyás, and S. Imre. User tracking on the web via cross-browser fingerprinting. In *Proceedings of the 16th Nordic Conference on Information Security Technology for Applications*, NordSec'11, pages 31–46, Berlin, Heidelberg, 2012. Springer-Verlag.
8. X. Boyen. Reusable cryptographic fuzzy extractors. In *Proceedings of the 11th ACM Conference on Computer and Communications Security*, CCS '04, pages 82–91. ACM, 2004.
9. R. Canetti and R. R. Dakdouk. Obfuscating point functions with multibit output. In *Advances in Cryptology–EUROCRYPT 2008*, pages 489–508. Springer, 2008.
10. R. Canetti, B. Fuller, O. Paneth, L. Reyzin, and A. Smith. *Advances in Cryptology – EUROCRYPT 2016*, chapter Reusable Fuzzy Extractors for Low-Entropy Distributions, pages 117–146. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
11. R. Canetti, Y. Tauman Kalai, M. Varia, and D. Wichs. *On Symmetric Encryption and Point Obfuscation*, pages 52–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
12. M. Chen, J. Fridrich, M. Goljan, and J. Lukás. Determining image origin and integrity using sensor noise. *IEEE Transactions on Information Forensics and Security*, 3(1):74–90, 2008.
13. A. Das, N. Borisov, and M. Caesar. Do you hear what i hear?: Fingerprinting smart devices through embedded acoustic components. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 441–452, 2014.
14. J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14:21–30, 2002.
15. Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, Mar. 2008.
16. Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540. Springer Berlin Heidelberg, 2004.
17. P. Eckersley. How unique is your web browser? In *Privacy Enhancing Technologies, 10th International Symposium, PETS 2010, Berlin, Germany, July 21-23, 2010. Proceedings*, pages 1–18, 2010.
18. B. Fuller, X. Meng, and L. Reyzin. *Computational Fuzzy Extractors*, pages 174–193. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
19. J. Galbally, A. Ross, M. Gomez-Barrero, J. Fierrez, and J. Ortega-Garcia. Iris image reconstruction from binary templates: An efficient probabilistic approach based on genetic algorithms. *Computer Vision and Image Understanding*, 117(10):1512–1525, 2013.
20. J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
21. C.-Y. Hsiao, C.-J. Lu, and L. Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 169–186. Springer, 2007.
22. A. K. Jain, K. Nandakumar, and A. Ross. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, 2016.
23. A. Juels and M. Sudan. A fuzzy vault scheme. *Des. Codes Cryptography*, 38(2):237–257, Feb. 2006.
24. A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, CCS '99, pages 28–36. ACM, 1999.
25. A. Kurtz, H. Gascon, T. Becker, K. Rieck, and F. Freiling. Fingerprinting mobile devices using personalized configurations. *Proceedings on Privacy Enhancing Technologies*, 2016(1):4–19, 2016.
26. J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, June 2006.
27. B. Lynn, M. Prabhakaran, and A. Sahai. Positive results and techniques for obfuscation. In *Advances in Cryptology–EUROCRYPT 2004*, pages 20–39. Springer, 2004.
28. K. Mowery and H. Shacham. Pixel perfect: Fingerprinting canvas in HTML5. In M. Fredrikson, editor, *Proceedings of W2SP 2012*. IEEE Computer Society, May 2012.
29. N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, pages 541–555, Washington, DC, USA, 2013. IEEE Computer Society.
30. R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.

31. S. Prabhakar, S. Pankanti, and A. K. Jain. Biometric recognition: Security and privacy concerns. *IEEE Security & Privacy*, 1(2):33–42, 2003.
32. U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber. Modeling attacks on physical unclonable functions. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 237–249, New York, NY, USA, 2010. ACM.
33. U. Ruhrmair and M. van Dijk. Pufs in security protocols: Attack models and security evaluations. *2012 IEEE Symposium on Security and Privacy*, pages 286–300, 2013.
34. K. Simoens, P. Tuyls, and B. Preneel. Privacy weaknesses in biometric sketches. In *2009 30th IEEE Symposium on Security and Privacy*, pages 188–203.
35. U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain. Biometric cryptosystems: issues and challenges. *Proceedings of the IEEE*, 92(6):948–960, June 2004.
36. U. Uludag, A. Ross, and A. Jain. Biometric template selection and update: a case study in fingerprints. *Pattern Recognition*, 37(7):1533–1542, 2004.
37. Z. Zhou, W. Diao, X. Liu, and K. Zhang. Acoustic fingerprinting revisited: Generate stable device id stealthily with inaudible sound. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 429–440. ACM, 2014.

A Proof of Lemma 1

Proof (Proof of Lemma 1). Suppose not, that is suppose that there exists some D of size at most s such that $\delta^s((R,P,V),(U_l,P,V)) > \epsilon$. Let X_v be a set of distributions giving rise to a joint distribution such that $\tilde{H}_\infty(X|V) \geq m_2$. Consider a D_1 that does the following:

1. Receive input α, β.
2. Run $\gamma, \nu \leftarrow \mathsf{Gen}'(\alpha)$.
3. Output $D(\gamma, \nu, \beta)$.

Also consider a D_2 that does the following:

1. Receive input α, β.
2. Run $\gamma, \nu \leftarrow \mathsf{Gen}'(\alpha)$.
3. Sample random string $u \leftarrow U_\ell$.
4. Output $D(u, \nu, \beta)$.

Denote $R', P' \leftarrow \mathsf{Gen}'(X)$. By the triangle inequality we have the following:

$$\begin{aligned}
&\delta^{D_1}((U,V),(X,V)) + \delta^{D_2}((U,V),(X,V)) \\
&\quad = \delta^D((R,P,V),(R',P',V)) + \delta^D((U_\ell,P,V),(U_\ell,P',V)) \\
&\quad \geq \delta^D((R,P,V),(U_\ell,P,V)) - \delta^D((U_\ell,P',V),(R',P',V)) \\
&\quad \geq \epsilon - \epsilon_{FE} = 2\epsilon_{RPI}
\end{aligned}$$

Thus, either D_1 or D_2 distinguishes U, V from X, V with advantage at least ϵ_{RPI}. Either of these distinguishers contradict the HILL entropy of U, V. This completes the proof of Lemma 1.

B Proof of Theorem 3

Proof. Correctness is straightforward. Once again, we separate pseudorandomness and reusability to deal with security. We begin by recalling that FTG-CPA security ensures that an adversary with an encryption oracle cannot distinguish between encryptions of two chosen messages.

Pseudorandomness As exhibited by following games, pseudrandomness of R comes from security of the encryption scheme (Enc,Dec).

$\mathcal{G}_0$ $\mathcal{C}$ samples $\omega \xleftarrow{\$} W$ where W is a distribution from $\mathcal{W}$. He then generates $(R, P) \leftarrow \mathsf{Gen}(w)$ as prescribed in Figure 5. $\mathcal{C}$ gives (R, P) to D.

$\mathcal{G}_1$ In the previous game, D is given R and $P = (Q_u, c_u, Q_t, c_t)$. The only parts of P that are related to R are c_t and Q_t. Now, $\mathcal{C}$ samples some $\mu \xleftarrow{\$} \{0,1\}^l$ and computes $c^* = \mathsf{Enc}(K_t, \mu)$. $\mathcal{C}$ sets $P^* = (Q_u, c_u, Q_t, c^*)$ and sends (R, P^*). If D can distinguish this game from the previous one, he can in particular distinguish $(R, c_t^* = \mathsf{Enc}(K_t, R))$ from $(R, c^* = \mathsf{Enc}(K_t, \mu))$. By Lemma 1, K_t is ϵ_{FE} close to uniform. By FTG-CPA security of (Enc, Dec), $\mathcal{G}_1$ is indistinguishable from $\mathcal{G}_0$ for $\epsilon = \epsilon_{\text{FE}} + \epsilon_{\text{CPA}}$ and $s = \min\{s_{\text{CPA}}, s_{\text{FE}}\}$.

$\mathcal{G}_2$ In the previous game, D is given R and $P^* = (Q_u, c_u, Q_t, c^*)$. In this game $\mathcal{C}$ samples $\eta \xleftarrow{\$} \{0,1\}^l$ and gives (η, P^*) to D. Since R and P^* are independent, this game is the same as the previous one to D's view.

$\mathcal{G}_3$ In the previous game, D was given η and $P^* = (Q_u, c_u, Q_t, c^*)$. $\mathcal{C}$ now replaces c^* with the actual values, independent of η, by the same reasoning as in $\mathcal{G}_1$. This indistinguishability holds for $\epsilon = \epsilon_{\text{FE}} + \epsilon_{\text{CPA}}$ and $s = \min\{s_{\text{CPA}}, s_{\text{FE}}\}$.

By transitivity, $\mathcal{G}_4$ is indistinguishable from $\mathcal{G}_0$ which leads to the indistinguishability required by Definition 8 for $\epsilon = 2(\epsilon_{\text{FE}} + \epsilon_{\text{CPA}})$ and $s = \min\{s_{\text{CPA}}, s_{\text{FE}}\}$.

Reusability In the previous argument we first replaced the ciphertext, the key R, and reverted back the ciphertext. Our strategy here is the same but it involves a hybrid argument where each update for a single enrollment has those values replaced. This leads to slightly worse parameters but the same overall structure.

Let $W^{1,1}, \ldots, W^{\phi,1}, W^{1,2}, \ldots, W^{\phi,\rho}$ be correlated distributions over $\mathcal{M}_1$, where $W^{i,k} \in \mathcal{W}$ for all i, k. Consider some fixed $1 \leq j \leq \rho$. The following games consists in a challenger $\mathcal{C}$ trying to fool D.

$\mathcal{G}_0$ $\mathcal{C}$ honestly samples values as prescribed in Definition 3 and sends

$$(R^1, P^{1,1}, \ldots, P^{\phi,1}), \ldots, (R^j, P^{1,j}, \ldots, P^{\phi,j}), \ldots, (R^\rho, P^{1,\rho}, \ldots, P^{\phi,\rho})$$

to D. Throughout this argument we will not modify R^i or $P^{k,i}$ for any $i \neq j$. Thus, we write this expression (reordering variables) as

$$(R^{-j}, P^{-j}, R^j, P^{1,j}, ..., P^{\phi,j}).$$

For all i, k, $P^{i,k}$ can be written $P^{i,k} = (Q_u^{i,k}, c_u^{i,k}, Q_t^{i,k}, c_t^{i,k})$ as specified in Figure 5.

$\mathcal{G}_{1,\ldots,\phi}$ At this point we replace the encrypted value $c^{*,i,j}$ one by one with random values. As in the pseudorandomness argument each game is indistinguishable for $\epsilon = \epsilon_{\text{CPA}} + \epsilon_{\text{FE}}$ and $s = \min\{s_{\text{CPA}}, s_{\text{FE}}\}$.

$\mathcal{G}_{\phi+1}$ The key R is now replaced with a random η. Since R and the modified P are independent this game is statistically identical to the previous game.

$\mathcal{G}_{\phi+2,\ldots,2\phi+1}$ In each of these games a single value $c_t^{*,i,j}$ is replaced with the actual values which are independent of η. Each game is indistinguishable from the previous for $\epsilon = \epsilon_{\text{CPA}} + \epsilon_{FE}$ and $s = \min\{s_{\text{CPA}}, s_{\text{FE}}\}$.

By transitivity, this last game $\mathcal{G}_{2\phi+1}$ is indistinguishable from $\mathcal{G}_0$ for $\epsilon_{\text{sec}} = 2\phi(\epsilon_{\text{CPA}} + \epsilon_{\text{FE}})$ and $s = \min\{s_{\text{CPA}}, s_{\text{FE}}\}$.