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

Quentin Alamélou[1,2], Paul-Edmond Berthier[2] Stéphane Cauchie[2], Benjamin Fuller[3], and Philippe Gaborit[1]

[1] Xlim, Université de Limoges, quentin.alamelou@xlim.fr,gaborit@unilim.fr
[2] ITA, equensWorldline, firstname.lastname@worldline.com
[3] Connecticut University, benjamin.fuller@uconn.edu

Abstract. A Fuzzy Extractor (Dodis *et al.*, Eurocrypt 2004) is a two-step protocol 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 showed that any information-theoretically secure reusable fuzzy extractor is subject to strong limitations. Recently, Canetti *et al.* (Eurocrypt 2016) proposed a computationally secure reusable fuzzy extractor for the Hamming metric that corrects a *sublinear* fraction of errors.

We propose a generic framework to solve the reusability problem. We introduce 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. A reusable pseudoentropic isometry can be combined with a traditional fuzzy extractor to provide a reusable fuzzy extractor.

To show the promise of our framework, we construct a reusable pseudoentropic isometry for the set difference metric. Our work construction handles a *linear* fraction of errors and is secure in the nonprogrammable random oracle model. Furthermore it is efficient, requiring only hash function evaluations and decoding an error correcting code.

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). 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 pseudoentropic isometries.

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 [13,22], Physically Unclonable Functions (PUFs) [30] and quantum information [5]. Both PUFs and biometrics suffer the common issue of errors that prevent stable cryptographic key generation.

1.1 Randomness sources

PUFs. A physically unclonable function 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 using this a function of this key.

Biometrics. Biometrics are systems that recognize individuals based on their biological and/or their behavioral characteristics. Biometrics are evaluated by their uniqueness, collectability and permanence [21] where this last characteristic represents the period in which those traits are stable. 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]. Unlike passwords, biometrics suffer inevitable but minor variations. This prevents the traditional approach of storing a hash value in place of the original template. Dodis *et al.* [15] stated that Hamming distance looks like the "most natural metric to consider" [10,15,23]. However, with the exception of iris [13], set distance better suits biometric matchers [27,20]. The set difference metric is appropriate when the noisy secret is a set of features. Examples include digital fingerprints and the exotic movie liker problem [14,22,35,27].

1.2 Fuzzy Extractors

The field of *information reconciliation* [5] enables retrieving identical values from noisy data. *Privacy amplification* [5] aims at converting these values into uniform random strings. Fuzzy Extractors (FEs) [15][4], 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 a predefined distance metric, say $d(\omega, \omega') \leq t$.

Metrics Dodis *et al.* proposed FE constructions for the Hamming, set difference and edit metrics, sometimes adapting previous work [23,22]. In this work 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, respectively referred as the *small* and *large* universe settings. 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 [22]). The small universe setting benefits from a reduction to the Hamming metric, referred as the bin-set equivalence (described in Section 2). However, this transform is not applicable in the large universe setting on which we concentrate.

Reusability Boyen exhibited 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 and we have no evidence that practical sources obey this condition. Subsequent works showed that existing fuzzy extractors are prone to this reusability weakness in practice [6,34].

The recent work of Canetti *et al.* [10] constructs the first reusable fuzzy extractor that makes no such 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]. Their construction is secure for distributions with high entropy samples instead of global min-entropy. Their construction requires more structure of the

[4] In the following, we will refer to the journal version [14].

source but can be secure for lower entropy rates than existing constructions. 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 to correct an error rate (t/n) sublinear in n. Prior to this work, there were no known reusable fuzzy extractor correcting a linear error rate for any common metric.

1.3 Browser fingerprints as authentication source

Current industrial authentication solutions tend to be both smartphone and software only solutions (*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 the burgeoning field of browser fingerprinting as an authentication factor.

Eckersley [16] 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 some features may have changed over time. Subsequent studies [7,26,29,2] show such a system is deployable for personal computers. Early solutions for mobile devices were either too resources demanding [25,11] or focused only on device identification and not individual users [12,37].

The recent work of Kurtz *et al.* provides a comprehensive analysis in the mobile setting [24]. 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 detected 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 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 and genuine differences; more precisely, if we consider an enrollment value ω, it is very 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 ω^j 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. The notion of *adaptive* biometric system was 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 [16,2,24].

To avoid ambiguity, we will talk about the *instability* of a physical biometric to indicate that they undergo minor but stable differences and the *variability* of browser fingerprints to indicate that these latter ones may suffer deeper changes over time.

1.4 Our contribution

1. We propose a generic framework to address reusability by combining any nonreusable FE and a new primitive we call a *reusable pseudoentropic isometric* (RPI). Informally, a RPI pseudorandomly projects fuzzy secrets while maintaining distances between two noisy readings. When ρ enrollments have to be done from a fuzzy secret ω, the RPI outputs ρ unrelated values

$\Omega^1, \ldots, \Omega^\rho$ that can then be securely 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 Ω'^j and Ω^j is the same as between ω and ω'.
2. We construct a RPI for the set difference metric and thus design the first reusable FE for linear error rates. Our solution is for the set difference metric in the large universe setting.
3. We define *Adaptive* Fuzzy Extractors (AFEs) that recover a stable key R from noisy readings even if readings drift naturally over time. In addition to primitives Gen and Rep, an update primitive Upd is introduced. The reproduction algorithm should output 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 propose a generic methodology to design an AFE out of a RPI and a nonreusable FE. Once again, an efficient instantiation for the set difference is proposed. We are aware of no previous study that considers the problem of key derivation from drifting data such as browser fingerprints.

2 Preliminaries

Notation $\log$ denotes the logarithm in basis 2. $GF(n)$ 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.

Let $\mathsf{H} : \{0,1\}^* \times \{0,1\}^{l_1} \rightarrow \{0,1\}^\kappa$ denotes a cryptographic hash function modeled as a non-programable random oracle [4]. 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 $d : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{N}$ fulfilling the properties of symmetry, triangle inequality and zero distance between equal points.

2.1 Background

Set Difference Metric Let $\mathcal{M}$ consists in all subsets of a universe $\mathcal{U}$. For two sets ω and ω' belonging to $\mathcal{M}$, their symmetric difference is defined as $\omega \Delta \omega' \stackrel{def}{=} \{x \in \omega \cup \omega' | x \notin \omega \cap \omega'\}$. The set difference metric between ω and ω' is the defined as $d(\omega, \omega') \stackrel{def}{=} |\omega \Delta \omega'|$. Recall 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 applicable 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 the (non) ability (for an adversary) to guess the value of a random variable. In the information-theoretic case, we often 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:

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

HILL entropy is a commonly used computational notion of entropy [18]. It was extended to the conditional case by Hsiao, Lu, and Reyzin [19].

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.* [14], was information theory-based. We focus on the computational definition introduced in [17]. Fuller *et al.* extend their definition to any family of distributions and we adopt this convention.

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

- *The generate procedure Gen on input $\omega \in \mathcal{M}$ outputs an extracted string $R \in \{0,1\}^l$ and a helper string $P \in \{0,1\}^*$.*
- *The reproduction procedure Rep takes an element $\omega' \in \mathcal{M}$ and a bit string $P \in \{0,1\}^*$ as inputs.*
- *The correctness guarantees that if $d(\omega, \omega') \leqslant t$ and $(R, P) \leftarrow$ Gen(ω), then Rep$(\omega', P) = R$. If $d(\omega, \omega') > t$, then no guarantee is provided about the output of Rep.*
- *The security property guarantees that for any distribution $W \in \mathcal{W}$ on $\mathcal{M}$, the string R is pseudorandom conditioned on P 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 refer to fuzzy extractors 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.* design fuzzy extractors 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 Reusability of fuzzy extractors [8] can be stated as the possibility to call procedure Gen numerous times on the noisy readings of ω while retaining security. Let us 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 will then generate ρ couples $(R^1, P^1), \ldots, (R^\rho, P^\rho)$ where $(R^j, P^j) \leftarrow \mathsf{Gen}(\omega^j)$. Recalling that P^j's are meant to be public and that different servers should not trust each other, Canetti *et al.* [10] proposed a security model for which a given R^{i_0} is secure even if all the R^js (for $j \neq i_0$) 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 $\eta \leftarrow \{0,1\}^l$.*
- ***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

Symmetric encryption We use a symmetric encryption scheme denoted (Enc, Dec).We require (Enc, Dec) to fulfill 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. $c_b \leftarrow \mathsf{Enc}(\mathsf{k}, m_b)$
3. $b' \leftarrow D(c_b : \mathcal{O}^{\text{encrypt(k,.)}})$
4. Return b'.

Pseudoentropic Isometries We propose a new paradigm for designing reusable FEs. Given a fuzzy secret ω, a pseudoentropic isometry aims at randomly deriving Ω that retains the entropy of ω and the distances between inputs (at least locally). More precisely, a pseudoentropic isometriy consists in two randomized procedures (DerGen, DerRep) defined as follows.

Definition 4 (Pseudoentropic isometry). *Let $(\mathcal{M}_1, d_1)$ and $(\mathcal{M}_2, d_2)$ be two metric spaces. A $(\mathcal{M}_1, \mathcal{M}_2, \mathcal{W}, m_2, \epsilon_{sec}, s_{sec})$-pseudoentropic isometry is a pair of randomized procedures (DerGen, DerRep) with the following properties:*

1. *DerGen on input $\omega \in \mathcal{M}_1$ outputs $\Omega \in \mathcal{M}_2$ and some $F \in \{0,1\}^*$.*
2. *DerRep takes an element $\omega' \in \mathcal{M}_1$ and a bit string $F \in \{0,1\}^*$ as inputs to output $\Omega' \in \mathcal{M}_2$. The correctness property guarantees that if $(\Omega, F) \leftarrow \mathsf{DerGen}(\omega)$, then $d_2(\Omega, \Omega') \leq d_1(\omega, \omega')$. Else, no guarantee is provided about Ω'.*
3. *Let W a distribution, DerGen(W) is denoted (U, V). If $\omega \stackrel{\$}{\leftarrow} W$ and $(\Omega, F) = \mathsf{DerGen}(\omega)$, we denote $(\Omega, F) \stackrel{\$}{\leftarrow} (U, V)$. The security property guarantees that for any distribution $W \in \mathcal{W}$, we have $H^{\mathtt{HILL}}_{\epsilon_{sec}, s_{sec}}(W|V) \geqslant m_2$ and $H^{\mathtt{HILL}}_{\epsilon_{sec}, s_{sec}}(U|V) \geqslant m_2$.*

This notion is related to biometric embeddings used in [14]. A biometric embedding projects any fingerprint value into a metric space where a FE exists while loosely maintaining distances. On their own pseudoentropic isometries are not novel (the identity function is a pseudoentropic isometry). 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 DerGen. Drawing on the definition of reusability for FEs (Definition 3), we define a RPI as follows.

Definition 5 (RPI). *Let $W^* \in \mathcal{W}$ be a fixed 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 4, 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 $(\Omega^j, F^j) \leftarrow \mathsf{DerGen}(\omega^j)$ and $(\Omega^*, F^*) \leftarrow \mathsf{DerGen}(\omega^*)$.*
- ***Distinguishing*** *The advantage of D consists in:*

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

(DerGen, DerRep) 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.

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 fuzzy extractor. 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. 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 (DerGen, DerRep) 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 DerGen 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 $d_1(\omega, \omega') \leqslant t$.

Note: Even in the nonreusable setting this approach has benefits. For distributions W where the number of possible error patterns is larger than $H_\infty(W)$, the bounds of traditional fuzzy extractors provide little security (see [14, Theorem 5.1] and the discussion in [10]). However, it may be possible to use a pseudoentropic isometric to avoid these bounds.

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

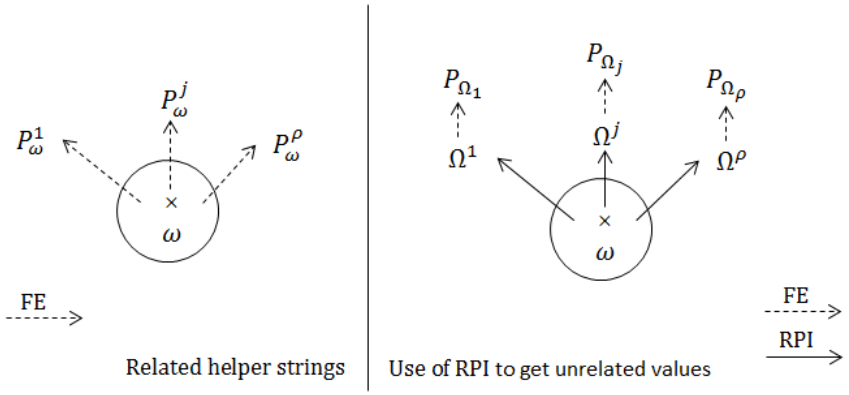

Fig. 1. Overview of reusability via RPI randomization

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

Fig. 2. A generic reusable 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 4, 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 fuzzy extractors 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 DerGen 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 5) .
3. Computes $(\Omega^*, F^*) \leftarrow \mathsf{DerGen}(\omega^*)$ 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 $\epsilon = \epsilon_{\text{RPI}}$ and $s = s_{\text{RPI}} - |\mathsf{Gen'}|$.

$\mathcal{G}_2$ 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 $H^{\mathtt{HILL}}_{\epsilon_{\text{RPI}}, s_{\text{RPI}}}(\Omega^*|F^*) \geq m_2$ then $H^{\mathtt{HILL}}_{\epsilon_{\text{RPI}}, s_{\text{RPI}}}(\Omega^*|F^{i_0}) \geq m_2$. Thus by Lemma 1, (U_l, P^*) and (R^*, P^*) are computationally indistinguishable. Hence, this game is indistinguishable from the previous one for $\epsilon = 2\epsilon_{\text{RPI}} + \epsilon_{\text{FE}}$ and $s = \min\{s_{\text{RPI}}, s_{\text{FE}}\}$.

$\mathcal{G}_3$ 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

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

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 fuzzy extractors in [14] 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.

To design our set difference-based RPI, we will use the hash function $\mathsf{H} : \{0,1\}^* \times \{0,1\}^{l_1} \rightarrow \{0,1\}^\kappa$. When modeled as a non-programmable random oracle [28], its outputs appear uniformly distributed and contain all the entropy of its input. Our set difference-based RPI, presented in Figure 3, essentially randomizes each set element using H.

Step 4 of Algorithm DerGen aims at avoiding collision. In such a case, a new seed salt is chosen and the protocol starts again. Choosing κ big enough, one can be sure that the case with

Algorithm DerGen
Input: $\omega = \{\omega_1, \ldots, \omega_s\}$,
$\forall 1 \leqslant i \leqslant s,\ \omega_i \in \mathcal{U}$.
1. $\mathsf{salt} \xleftarrow{\$} \{0,1\}^{l_1}$.
2. For $i = 1 \ldots s$,
$x_i \leftarrow \mathsf{H}(\omega_i; \mathsf{salt})$.
3. Set $\Omega = \{x_1, \ldots, x_s\}$.
4. If $|\Omega| < s$,
go to 1.
5. Return (salt, Ω).

Algorithm DerRep
Inputs: $\omega' = \{\omega'_1, \ldots, \omega'_s\}$,
$\forall 1 \leqslant i \leqslant s,\ \omega'_i \in \mathcal{M}$,
$\mathsf{salt} \in \{0,1\}^*$.
1. For $i = 1 \ldots s$,
$x'_i \leftarrow \mathsf{H}(\omega'_i; \mathsf{salt})$.
2. Set $\Omega' = \{x'_1, \ldots, x'_{s'}\}$.
3. While $|\Omega'| < s$,
$z \xleftarrow{\$} GF(2^\kappa)$.
$\Omega' \cup \{z\}$.
4. Return Ω'.

Fig. 3. A set difference-based RPI

no collision occurs with overwhelming probability. In the same vein, Step 3 of Algorithm DerRep maintains distances between original values (ω and ω') and randomly derived ones (Ω and Ω'). Indeed, because of step 4 of DerGen, a collision occurring in DerRep can only be due to an element ω'_i that did not appear in ω. We defer our proofs and analysis of this construction to Appendix B. The condition we require on distributions is that each set has superlogarithmic min-entropy.

Theorem 2. *Let λ be a security parameter, let $q = \mathtt{poly}(\lambda)$, $\rho = \mathtt{poly}(\lambda)$, $\kappa = \omega(\log \lambda)$, $|\mathsf{salt}| = \omega(\log \lambda)$ and $m_2 = \omega(\log \lambda)$. Define $\mathcal{W}$ as the set of all joint distributions W where for any i, $\tilde{H}_\infty(W^i | W^{-i}) \geq \kappa$ where W^{-i} represents all other elements of W. Then Figure 3 defines a $(\mathcal{M}_{\mathcal{U},s}, \mathcal{M}_\kappa, \mathcal{W}, m_2, \epsilon, q)$-RPI for the set difference metric where $m_2 = \kappa \cdot s$ for $\epsilon = q(q+1)2^{-\kappa} + q2^{-m_2} = \mathtt{ngl}(\lambda)$.*

Notes: In the above theorem the running time of the adversary does not matter only the number of oracle queries. So instead of listing the distinguisher size we list the number of queries q they can make to the random oracle. Second, the proof only relies on an attacker finding a preimage to a random oracle output so the reusability ρ does not effect security parameters.

Corollary 1. *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 reusable fuzzy extractor 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).

Our construction also is very efficient due to the use of a nonreusable FE. Efficient information theoretic FEs are proposed in [14] for which storage and time complexities are respectively of order $t \log n$ (with equivalent entropy loss) and $\mathtt{poly}(s \log n)$. Reaping benefits of such constructions, our work then enjoys the same complexities.

In addition, we note that it is possible to view the second construction of Canetti et al. [10, Construction 2] as the composition of a RPI and nonreusable fuzzy extractor. Their construction is called *Lock-and-Error-Correct*. They first apply a digital locker to each dimension and then a fuzzy extractor. If the input source W has high entropy in all dimensions, the digital locker serves as an RPI. Interestingly, the digital locker projects the set difference metric directly to the binary Hamming metric in contrast to our construction which remains in the large alphabet setting.

4 Adaptive Fuzzy Extractors

As discussed in the Introduction, we consider browser and device fingerprints [16,2,24] as possible 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 [2,16,24]. 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 fuzzy extractor 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 6. *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 $d(\omega^i, \omega^{i+1}) \leqslant u$.*

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. Plus, 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 6). A classic FE is meant to recover a previously extracted key R if and only if the reproduction value ω' belongs to $\mathcal{B}(\omega, t)$. In our context, we require an AFE to recover the actual key R as long as the reproduction value ω' has somewhat *naturally* drifted from ω although ω' does not belong $\mathcal{B}(\omega, t)$. Given parameters $0 \leqslant u \leqslant t$, we propose 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 adaptive fuzzy extractor 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 adaptive fuzzy extractors is depicted in Figure 4.

4.2 Definition and Security Model

Definition 7 (Adaptive Fuzzy Extractor). *A triple of randomized procedures "generate" (Gen), "update" (Upd), "reproduce" (Rep) is an $(\mathcal{M}, \mathcal{W}, l, u, t, \phi)$-adaptive fuzzy extractor that is $(\epsilon_{sec}, s_{sec})$ hard if the following holds:*

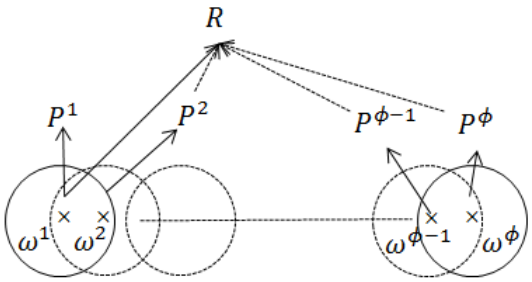

Fig. 4. Philosophy of Adaptive Fuzzy Extractors.

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* $P \leftarrow$ **Upd**(ω^*, P^*), $R \leftarrow$ **Rep**(ω^*, 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) $(R, P) \leftarrow$ **Gen**(ω) *and* $d(\omega, \omega') \leqslant t$;
   (b) there exists (ω^*, P^*) *such that* $P \leftarrow$ **Upd**(ω^*, P^*), $R \leftarrow$ **Rep**(ω^*, P^*) *and* $(\omega^*, \omega') \leqslant t$, *then* **Rep**$(\omega', P) = R$. *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 8. *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* $(\mathcal{M}, \mathcal{W}, \ell, u, t, \phi)$*-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.*
  - $(R^k, P^{1,k}) \leftarrow$ **Gen**$(\omega^{1,k})$
  - *for* $2 \leqslant i \leqslant \phi$ *:* $P^{i,k} \leftarrow$ **Upd**$(\omega^{i,k}, P^{i-1,k})$.

– ***Distinguishing*** *The advantage of D is*

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

(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 7).

5 From classic FE to Reusable AFE

To obtain a reusable AFE out of a classic FE, there are two key points to reach:

– computed helper strings should not leak information about user's fuzzy secret(s). As already seen in Section 3, RPIs address this point.
– a classic FE recovers 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 a classic (*i.e.* non adaptive) FE.

Let (Enc, Dec) be a symmetric encryption scheme $(\epsilon_{\text{CPA}}, s_{\text{CPA}})$-FTG-CPA secure. Let (DerGen, DerRep) be a $(\mathcal{M}_1, \mathcal{M}_2, \mathcal{W}, m_2, \epsilon_{\text{RPI}}, s_{\text{RPI}})$-RPI that is $2\rho \cdot \phi$ reusable. Let $(\mathsf{Gen}_u, \mathsf{Rep}_u)$ be an average-case $(\mathcal{M}_2, m_2, l, u)$-FE that is $(\epsilon_{\text{FE}}, s_{\text{FE}})$-hard. Let $(\mathsf{Gen}_t, \mathsf{Rep}_t)$ be an average-case $(\mathcal{M}_2, m_2, l, t)$-FE that is $(\epsilon_{\text{FE}}, s_{\text{FE}})$-hard. Figure 5 depicts how to design a reusable AFE out of these tools.

5.1 Generation procedure

Given an enrollment value ω, the first step of the generation algorithm is to use the RPI to randomly project it onto two unrelated derived fingerprints Ω_u and Ω_t. Procedure Gen_u of the nonreusable FE correcting u errors will then be applied on Ω_u. Similarly, Gen_t is applied to Ω_t. Two temporary keys, K_u and K_t will be extracted. The first one 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 consists in deriving ω' into Ω'_u and Ω'_t. Successful decryption of c_u under K_u recovers 1 and indicates that ω' is within distance u. It then makes sense to call update. If so, R can be unlocked by re-generating K_t. DerGen then randomizes ω' into Ψ_u and Ψ_t. Gen' computes new temporary keys K'_u and K'_t along with new helper strings. R is finally re-encrypted under K'_t.

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

1. *Derivation Step*
$(\Omega_u, F_u) \leftarrow \mathsf{DerGen}(\omega)$.
$(\Omega_t, F_t) \leftarrow \mathsf{DerGen}(\omega)$.

2. *Use of a classical FE*
$(K_u, Q_u) \leftarrow \mathsf{Gen'}(\Omega_u, u)$.
$(K_t, Q_t) \leftarrow \mathsf{Gen'}(\Omega_t, t)$.

3. *Key Generation*
$R \xleftarrow{\$} \{0,1\}^l$.

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

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

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

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

2. *Helper Data Re-generation*
$(\Psi_u, F'_u) \leftarrow \mathsf{DerGen}(\omega')$.
$(\Psi_t, F'_t) \leftarrow \mathsf{DerGen}(\omega')$.
$(K'_u, Q'_u) \leftarrow \mathsf{Gen'}(\Psi_u, u)$.
$(K'_t, Q'_t) \leftarrow \mathsf{Gen'}(\Psi_t, t)$.
$c'_u = \mathsf{Enc}(K'_u, 1)$, $c'_t = \mathsf{Enc}(K'_t, K)$.

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

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

1. *Parsing Helper Data*
Parse $P = (F_u, P_u, c_u), (F_t, P_t, c_t)$.
$\Omega'_t \leftarrow \mathsf{DerRep}(\omega', F_t)$.

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

3. Return R.

Fig. 5. Generation, Update, and Reproduce procedures

5.3 Reproduction procedure

The reproduction procedure is straightforward. Taking as inputs ω' and some P, DerRep generates the corresponding Ω'_t as previously described. If Ω'_t is within distance t of the previously and implicitly enrolled Ω_t, then Rep' recovers K_t which enables to finally unlock R.

Theorem 3. *With notation defined in 5, Figure 5 defines a* $(\mathcal{M}_1, \mathcal{W}, \ell, u, t, \phi)$*-AFE that is* (ϵ_{FE}, s_{sec}) *hard and* ρ*-reusable for* $\epsilon = \phi(6\epsilon_{RPI} + \epsilon_{CPA} + 2\epsilon_{FE})$, $s_{sec} = \min\{s_{RPI} - 2(|\mathsf{Gen'}| + |\mathsf{Enc}|), s_{CPA}, s_{FE}\}$.

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).

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.

$\mathcal{G}_1$ $\mathcal{C}$ instead of running DerGen on ω to get Ω_t runs on $\omega^* \leftarrow W^*$ (the distribution defined in Definition 5). Denote $(\Omega^*_t, F^*) \leftarrow \mathsf{DerGen}(\omega^*)$. The value Ω^*_t is substituted in the second Gen' process and the resulting encryption. This changes Q^*_t and c^*_t, all other values remain the same. By the reusability of the RPI, this game is indistinguishable from $\mathcal{G}_0$ for $\epsilon = \epsilon_{\text{RPI}}$ and $s = s_{\text{RPI}} - 2(|\mathsf{Gen}'| + |\mathsf{Enc}|)$.

$\mathcal{G}_2$ In the previous game, D was given R randomly sampled from $\{0,1\}^l$ and $P = (F_u, Q_u, c_u, F_t, Q_t^*, c_t^*)$. The only parts of P that are related to R are c_t^* and Q_t^* which are independent of the other values. Now, $\mathcal{C}$ samples some $\mu \xleftarrow{\$} \{0,1\}^l$ and computes $c^* = \mathsf{Enc}(K_t, \mu)$. $\mathcal{C}$ sets $P^* = (F_u, Q_u, c_u, F_t, Q_t^*, c^*)$ 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 $(R, c^* = \mathsf{Enc}(K_t, \mu))$. First note by Lemma 1, K_t^* and K_t are both $2\epsilon_{\text{RPI}} + \epsilon_{\text{FE}}$ close to uniform. Hence, K_t and K_t^* are $2(2\epsilon_{\text{RPI}} + \epsilon_{\text{FE}})$ close. Finally, by FTG-CPA security of $(\mathsf{Enc}, \mathsf{Dec})$, $\mathcal{G}_2$ is indistinguishable from $\mathcal{G}_1$ for $\epsilon = 2(2\epsilon_{\text{RPI}} + \epsilon_{\text{FE}}) + \epsilon_{\text{CPA}}$ and $s = \min\{s_{\text{CPA}}, s_{\text{RPI}}, s_{\text{FE}}\}$.

$\mathcal{G}_3$ In the previous game, D is given R and $(F_u, Q_u, c_u, F_t, Q_t^*, c^*)$. Then, $\mathcal{C}$ can also sample some $\eta \xleftarrow{\$} \{0,1\}^l$ to finally give (η, P^*) to D. Since R and P^* are independent, this game is the same as the previous one to D's view.

$\mathcal{G}_4$ In the previous game, D was given η and $P^* = (F_u, Q_u, c_u, F_t, Q_t^*, c^*)$ that are independent. $\mathcal{C}$ can now replace c^*, Q_t^* with the actual values, independent of η, by the same reasoning as in $\mathcal{G}_1$. This indistinguishability holds for $\epsilon = \epsilon_{\text{RPI}}$ and $s = s_{\text{RPI}} - 2(|\mathsf{Gen}'| + |\mathsf{Enc}|)$

By transitivity, $\mathcal{G}_4$ is indistinguishable from $\mathcal{G}_0$ which leads to the indistinguishability required by Definition 7 for $\epsilon = 6\epsilon_{\text{RPI}} + \epsilon_{\text{CPA}} + 2\epsilon_{\text{FE}}$ and $s = \min\{s_{\text{RPI}} - 2(|\mathsf{Gen}'| + |\mathsf{Enc}|), s_{\text{CPA}}, s_{\text{FE}}\}$.

Reusability In the previous argument we first replaced the distribution Ω_t with an uncorrelated distribution, then replaced the ciphertext, the key R, and reverted back. 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 $P^{i,k} = (F_u^{i,k}, Q_u^{i,k}, c_u^{i,k}, F_t^{i,k}, Q_t^{i,k}, c_t^{i,k})$ as specified in Figure 5.

$\mathcal{G}_1$ $\mathcal{C}$ instead of running DerGen on ω^j to get Ω_t^j runs on $\omega^* \leftarrow W^*$ (the distribution defined in Definition 5). Denote $(\Omega_t^{*,j}, F^{*,j}) \leftarrow \mathsf{DerGen}(\omega^*)$. The value $\Omega_t^{*,j}$ is substituted in the second Gen' process and the resulting encryption. This changes $Q_t^{*,j}$ and $c_t^{*,j}$, all other values remain the same. By the reusability of the RPI, this game is indistinguishable from $\mathcal{G}_0$ for $\epsilon = \epsilon_{\text{RPI}}$ and $s = s_{\text{RPI}} - 2(|\mathsf{Gen}'| + |\mathsf{Enc}|)$.

$\mathcal{G}_{2,\ldots\phi}$ In each of these games we replace $\omega^{i,j}$ with independent samples from the distribution W^*. $\Omega_t^{i,j}$ is updated to $\Omega_t^{*,i,j}$ as in the game above. This replacement effects $Q_t^{*,i,j}$ and $c_t^{*,i,j}$, and not the other values. Each of these games is also indistinguishable for $\epsilon = \epsilon_{\text{RPI}}$ and $s = s_{\text{RPI}} - 2(|\mathsf{Gen}'| + |\mathsf{Enc}|)$.

$\mathcal{G}_{\phi+1,\ldots,2\phi}$ 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 $\epsilon = \epsilon_{\text{CPA}} + 2(2\epsilon_{\text{RPI}} + \epsilon_{\text{FE}})$ and $s = \min\{s_{\text{CPA}}, s_{\text{RPI}}, s_{\text{FE}}\}$.

$\mathcal{G}_{2\phi+1}$ 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.

$\mathcal{G}_{2\phi+1,\ldots,3\phi+1}$ In each of these games a single pair of $Q_t^{*,i,j}$ and $c_t^{*,i,j}$ is replaced back with the actual values which are independent of η. Each game is indistinguishable from the previous for $\epsilon = \epsilon_{\text{RPI}}$ and $s = s_{\text{RPI}} - 2(|\mathsf{Gen}'| + |\mathsf{Enc}|)$.

By transitivity, this last game $\mathcal{G}_{3\phi+1}$ is indistinguishable from $\mathcal{G}_0$ for $\epsilon_{\text{sec}} = \phi(6\epsilon_{\text{RPI}} + \epsilon_{\text{CPA}} + 2\epsilon_{\text{FE}})$ and $s_{\text{sec}} = \min\{s_{\text{RPI}} - 2(|\mathsf{Gen}'| + |\mathsf{Enc}|), s_{\text{CPA}}, s_{\text{FE}}\}$ fulfilling Definition 8.

Corollary 2. *Using the RPI defined in Figure 3 for the family $\mathcal{W}$ defined above one can construct a reusable and adaptive FE for any $s_{sec} = \mathtt{poly}(\lambda), \rho = \mathtt{poly}(\lambda), \phi = \mathtt{poly}(\lambda)$ such that $\epsilon_{sec} = \mathtt{ngl}(\lambda)$ where $t = \Theta(n)$.*

6 Conclusion and Future Works

In this work, 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 construction is for the large universe setting and is a complement to the work Canetti *et al.* Their work can be extended to the small universe setting but only for sublinear error rates.

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 maintaining distances between original and randomized values: 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 the non-programmable random oracle in this work and the work of Canetti *et al.* can be viewed as a RPI using a digital locker.

In addition to tackling reusability issue, 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 an RPI, a fuzzy extractor, and a symmetric encryption scheme. While defining Adaptive Fuzzy Extractors increases the scope of Fuzzy Extractors, our definition of t-drift is a first pass. Accurate modeling requires better understanding of such fingerprints.

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. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM, 1993.
5. C. H. Bennett, G. Brassard, and J.-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.
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. 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.
12. 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.
13. J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14:21–30, 2002.
14. 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.
15. 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.
16. 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.
17. B. Fuller, X. Meng, and L. Reyzin. *Computational Fuzzy Extractors*, pages 174–193. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
18. 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.
19. 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.
20. A. K. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP J. Adv. Signal Process*, 2008:113:1–113:17, Jan. 2008.
21. A. K. Jain, K. Nandakumar, and A. Ross. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, 2016.
22. A. Juels and M. Sudan. A fuzzy vault scheme. *Des. Codes Cryptography*, 38(2):237–257, Feb. 2006.
23. 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.
24. 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.
25. 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.
26. K. Mowery and H. Shacham. Pixel perfect: Fingerprinting canvas in HTML5. In M. Fredrikson, editor, *Proceedings of W2SP 2012*. IEEE Computer Society, May 2012.
27. K. Nandakumar, A. K. Jain, and S. Pankanti. Fingerprint-based fuzzy vault: Implementation and performance. *IEEE Transactions on Information Forensics and Security*, 2(4):744–757, Dec 2007.

28. J. B. Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *Advances in Cryptology – CRYPTO*, pages 111–126. Springer, 2002.
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 Analysis of Set Difference Based RPI

Proposition 1. *Let λ be the security parameter and fix $\epsilon > 0$. Define $\mathcal{W}$ as the set of all joint distributions W where for any i, $\tilde{H}_\infty(W^i|W^1,...,W^{i-1},W^{i+1},...,W^s) \geq \kappa + 2\log(1/\epsilon) + O(1)$. Figure 3 defines a $(\mathcal{M}_{\mathcal{U},s}, \mathcal{M}_\kappa, \mathcal{W}, m_2, \epsilon, \infty)$-PI for the set difference metric where $m_2 = \kappa \cdot s$ with an unbounded number of queries to the random oracle where $\epsilon' = \epsilon + (1 - e^{-s^2/2^\kappa})$. In particular if $2^\kappa = \omega(\mathtt{poly}(\lambda))$ then $\epsilon' = \epsilon + \mathtt{ngl}(\lambda)$.*

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

1. *Isometry property.* By design, Ω is of size s and for any $\omega_i = \omega'_i$, then $x_i = x'_i$. Thus, $d(\Omega, \Omega') \leq d(\omega, \omega')$.
2. *Security.* Treating H modeled as a random oracle, first consider the case where the condition in step 4 is not triggered. Then H is a good extractor for each individual enrollment. With H a random oracle, the knowledge of the random salt does not impact any entropy loss: $\forall\, \mathsf{salt}, H_\infty(X = x_i | V = \mathsf{salt}) = m$. That is, for an individual enrollment

$$(X_i, W_1, ..., W_{i-1}, W_{i+1}, ..., W_s, salt) \approx_\epsilon (U_k, W_1, .., W_{i-1}, W_{i+1}, ..., W_s, \mathsf{salt}).$$

   To show the statement of the proposition we measure the probability of the condition in step four being triggered.

$$\begin{aligned}
\Pr[\text{no collision}] &= \prod_{i=1}^{s}(1 - \frac{i}{2^\kappa}) \\
&\geq \left(1 - \frac{s}{2^\kappa}\right)^s \\
&= \left(\left(1 - \frac{s}{2^\kappa}\right)^{2^k/s}\right)^{s^2/2^\kappa} \\
&\geq (1/e)^{s^2/2^\kappa} = e^{-s^2/2^\kappa}
\end{aligned}$$

Adding reusability We now proceed to show reusability of the construction in this setting we need a slightly more restrictive definition of the random oracle and the set of sources $\mathcal{W}$. We need to ensure that the output length of the random oracle is longer enough that collisions occur with negligible probability and the entropy of each item is high enough that the adversary can query an input to the random oracle with only negligible probability. In the below theorem, we bound the size of an adversary by the number of random oracle queries it is permitted to make. In this theorem, the adversary may have unlimited computational power only the number of queries affects security.

Proof (of Theorem 2). Fix some j that the adversary is trying to distinguish. The basic idea of the proof is that the adversary should have no information about the output of the random oracle on ω^j unless they have been able to find an input to the random oracle that matches its output.

Outputs of algorithms DerGen and DerRep consist in sets of random elements belonging to $GF(2^\kappa)$. By the use of H, these elements are uniformly distributed over $\mathcal{M}_\kappa$. Let $W^1, \ldots W^\rho$ be related distributions from which $\omega^1, \ldots, \omega^\rho$ are respectively sampled (some or all ω^js could then be equal). The adversary is given

$$(\mathsf{salt}^1, \Omega^1), \ldots, (\mathsf{salt}^{j-1}, \Omega^{j-1}), (\mathsf{salt}^j, z), (\mathsf{salt}^{j+1}, \Omega^{j+1}), \ldots, (\mathsf{salt}^\rho, \Omega^\rho)$$

where $(\mathsf{salt}^i, \Omega^i) \leftarrow \mathsf{DerGen}(\omega^j)$ and z is either Ω^j or the output of DerGen on an unrelated value.

The only way for an adversary D to learn about z is to find a input query $\omega^{i,*}$ such that $H(\omega^{i,*}, \mathsf{salt}^j) = z^i$ for some $1 \leq i \leq s$. Thus, we can consider strategies for generating input queries. Consider some enrollment value k. Each input symbol to the random oracle has input min-entropy $\omega(\log \lambda)$. Furthermore, while the random oracle returns a string with each query the only useful piece of information is whether the output value matches the provided output value in Ω^k. For convenience we assume that when an adversary receives a single matching value, they completely learn ω^*.

First note that the probability of a matching response on the first query is at most $\Pr[\text{match}] = \Pr[\text{correct point}] + \Pr[\text{collision}] \leq 2^{-\kappa} + 2^{-m_2}$. The probability of a collision does not change as D asks more queries but the probability of finding the correct point increases. For any $\omega_1, ..., \omega_q$ inputs to an oracle the adversary learns that at most one of these values matches (as they are assumed to win when an oracle input matches). Thus we have Let $A_1, A_2, \ldots A_q$ be the random variables representing whether the ω_i was a correct value. Each A_i is just a bit, and at most one of them is equal to 1. Thus, the total number of possible responses is $q + 1$. Thus, we have the following,

$$\begin{aligned}\forall i, j, \tilde{H}_\infty(W^{i,j}|View(D(\cdot))) &= \tilde{H}_\infty(W^{i,j}|A_1, \ldots, A_q) \\ &= H_\infty(W^{i,j}) - |A_1, \ldots, A_q| \\ &= \kappa - \log(q+1)\,,\end{aligned}$$

where the second line follows from the first by [14, Lemma 2.2]. Thus, after all its queries the probability that of a match on the qth query is less than $2^{-\kappa - \log(q+1)} + 2^{-m_2}$. Thus, by union bound across all queries the total probability of a match is:

$$\Pr[\text{match}] \leq q(q+1)2^{-\kappa} + q2^{-m_2}.$$

This probability is negligible with the parameters as specified in the theorem. We then have the following:

$$\begin{aligned}Adv(D) =& \Pr[D(\Omega^1, \ldots, \Omega^\rho, F^1, \ldots, F^\rho) = 1] \\ &- \Pr[D(\Omega^1, \ldots, \Omega^{j-1}, \Omega^*, \Omega^{j+1}, \ldots, \Omega^\rho, F^1, \ldots, F^\rho) = 1] \\ &= \Pr[D(\Omega^1, \ldots, \Omega^\rho, F^1, \ldots, F^\rho) = 1|\text{match}] \Pr[\text{match}] \\ &+ \Pr[D(\Omega^1, \ldots, \Omega^\rho, F^1, \ldots, F^\rho) = 1|\text{no match}] \Pr[\text{no match}] \\ &- (\Pr[D(\Omega^1, \ldots, \Omega^{j-1}, \Omega^*, \Omega^{j+1}, \ldots, \Omega^\rho, F^1, \ldots, F^\rho|\text{match})] \Pr[\text{match}] \\ &+ \Pr[D(\Omega^1, \ldots, \Omega^{j-1}, \Omega^*, \Omega^{j+1}, \ldots, \Omega^\rho, F^1, \ldots, F^\rho|\text{no match})] \Pr[\text{no match}]) \\ &= \Pr[\text{match}](\Pr[D(\Omega^1, \ldots, \Omega^\rho, F^1, \ldots, F^\rho) = 1|\text{match}] \\ &- (\Pr[D(\Omega^1, \ldots, \Omega^{j-1}, \Omega^*, \Omega^{j+1}, \ldots, \Omega^\rho, F^1, \ldots, F^\rho|\text{match})])) \\ &+ \Pr[\text{no match}](\Pr[D(\Omega^1, \ldots, \Omega^\rho, F^1, \ldots, F^\rho) = 1|\text{no match}] \\ &- \Pr[D(\Omega^1, \ldots, \Omega^{j-1}, \Omega^*, \Omega^{j+1}, \ldots, \Omega^\rho, F^1, \ldots, F^\rho|\text{no match})]) \\ &\leq \left(q(q+1)2^{-\kappa} + q2^{-m_2}\right) \cdot 1 + (1 - q(q+1)2^{-\kappa} + q2^{-m_2}) \cdot 0 \\ &= q(q+1)2^{-\kappa} + q2^{-m_2}.\end{aligned}$$