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

Quentin Alamélou[1,2] Paul-Edmond Berthier[2] Stéphane Cauchie[2]
Philippe Gaborit[1],
[1] Université de Limoges, XLIM-DMI, Limoges, France,
{quentin.alamelou,philippe.gaborit}@xlim.fr,
[2] equensWorldline, FPL-ITA and R&D Departments, Seclin, France,
{quentin.alamelou,paul-edmond.berthier,stephane.cauchie}@equensworldline.com.

November 21, 2016

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 inputs an enrollment value w to output R and an helper string P that will enable further reproduction of R from some close reading w'. Nevertheless, Boyen fast highlighted the need for *reusable* fuzzy extractors (CCS 2004) that remain secure even when numerous calls to the generation procedure are made on the same fuzzy secret. Only recently has been proposed such a fuzzy extractor by Canetti *et al.* (Eurocrypt 2016). Even if not explicitly mentioned in their work, their main construction based on Hamming distance can also handle the set difference metric, an important metric also used for fuzzy extractors.

In this work, we propose a new and generic framework to solve the reusability problem. Our approach is simple, reaps benefits of any nonreusable fuzzy extractor and naturally applies to set difference metric. When the universe size is polynomial in the set size, our work permits to handle, in polynomial time, the case where the number of errors is *linear* in the length of the code, when the recent work of Canetti *et al.* can only handle a *sublinear* number of errors in polynomial time. This last point makes our construction more efficient than previous constructions since the practical efficiency of a fuzzy extractor depends on its capacity to handle errors. Our scheme has also benefits better storing capacities. Now when the universe size is superpolynomial in the set size, our work enjoys satisfying complexities of existing and efficient nonreusable Fuzzy Extractors while the approach of Canetti *et al.* does not apply.

Besides considering the reusability issue, we also propose the field of browser and device fingerprinting as a new and promising playground for Fuzzy Extractors. Philosophically close to biometrics, this burgeoning field of fingerprinting mainly considers list of features but comes with deeper variations over time while still enabling users' identification. We then define *Adaptive* Fuzzy Extractors, meant to handle these changes. More precisely, such a Fuzzy Extractor enables to recover a key R from a reading w' as long as w' has *naturally* drifted from w.

1 Introduction

As cryptography traditionally relies on uniformly distributed and reproducible long-term secrets often dedicated to authentication or key derivation, reality comes with basic concerns about creating and storing such values. Numerous randomness sources such as biometrics and human-generated data [13, 20], Physically Unclonable Functions (PUFs) [27] or even quantum information [6] have long been studied.

1.1 Fuzzy Extractors

The field of *information reconciliation* enables retrieving identical values from data subject to genuine variations while *privacy amplification* [6] aims at converting these values into uniform random strings. Fuzzy Extractors (FEs), introduced by Dodis et al. [15][1], consist in a non-interactive two-step protocol $(\mathsf{Gen}, \mathsf{Rep})$ both fulfilling information reconciliation and privacy amplification. Taking as input ω from a random source, primitive Gen, used at enrollment, outputs a uniformly distributed key R and some public helper string P while Rep, later given the helper string and ω', can reproduce the secret key as long as ω' is close enough to ω relatively to a certain metric. Enabling key generation from noisy data without requiring sensitive storage, FEs fast appeared as an elegant solution to a wide set of applications such as key generation, key manipulation, exploitation and protection of private randomness sources such as biometrics or PUFs.

1.1.1 Different Metrics

If Hamming distance may appear as the most natural metric to study, Dodis *et al.* proposed numerous constructions also satisfying set difference and edit distances, either as original constructions or adaptations of previous works [21, 20]. The Set difference metric focuses on the case where inputs w are subsets of size s a universe $\mathcal{U}$ of cardinal n. For these metric, Dodis *et al.* distinguished two settings, respectively referred as the *small* and *large* universe settings. In the former case, we have that $n = \mathsf{poly}(s)$ while in the latter one n is superpolynomial in s. The large universe setting is the more natural and frequent case to consider: for example w can be a list of book titles, a list of movies (movie lover's problem due to [20]) or even a list of features coming from an unsamplable universe. The small universe setting benefits a straightforward reduction with binary vectors, referred as the $\mathsf{bin\text{-}set}$ equivalence that is described in Section 2.

1.1.2 Reusable Fuzzy Extractors

Boyen then fast exhibited the need for *reusable* fuzzy extractors [9] for which numerous helper string generations from the same fuzzy secret do not impact user's security and privacy. Indeed, he designed some FEs fitting the definition of [14] that leak the fuzzy secret ω when numerous, but polynomially bounded, calls to Gen were made. Follow-up works showed that all existing fuzzy extractors were prone to this weakness [33, 7]. Apart from the restrictive and unlikely model of Boyen [9] for which the exclusive or of user's fuzzy secrets should not leak any information, there existed no reusable fuzzy extractor before the recent work of Canetti *et al.* [10] based on Hamming distance. Making use of a particular encryption scheme, referred as *digital lockers*, for which the decryption can indicate if the decrypted value is the expected plaintext, they design a simple and elegant FE that fulfills computational security. Considering distributions with high entropy samples instead of global min-entropy, their scheme benefits lower entropy rates compared to existing constructions. Their main binary construction can be extended through $\mathsf{bin\text{-}set}$ equivalence to a FE

[1] In the following, we will refer to the expanded version [14].

in the set difference based metric but only in the small universe setting. Plus, in such a case, their scheme only allows to correct errors sublinear in n and if one wants to improve the error correction, it leads to improve some constant c which will then imply time and storage complexities linear in n^c. On the other hand, when instantiated with efficient error correcting codes such as LDPC, our scheme will benefit both error correction and time complexity linear in n.

1.2 Randomness sources and FEs

Because of aforesaid properties, fuzzy extractors have fast gained interest, notably in the fields of PUFs and biometrics. Indeed, both fields suffer the common issue of noisy errors that prevent stable and cryptographic key generation.

PUFs. A physically unclonable function is a physical entity that is easy to evaluate but hard to predict. Unique by manufacturing process, PUFs aim at implementing challenge-response authentication. Nevertheless, environmental variations may impact their quality so that FEs were proposed to correct response errors. Unhappily, numerous attacks on PUFs [30, 31] may deter the use of such an hardware solution.

Biometrics. Biometrics are systems that recognize individuals based on their biological and/or their behavioral characteristics. To be considered as a biometric traits, such characteristics have to be assessed in regard of: uniqueness, collectability and permanence [19] where this later represents the period in which those traits are stable.

Beyond ethical concerns [28], privacy of biometric templates is often questioned from a technical point of view [28, 34]. Contrary to static data such as passwords, they suffer inevitable but minor variations that have to be handled. FEs then appeared well-fitted for their protection [14, 25, 18]. As stated by Dodis *et al.* [14], Hamming distance looked like the "most natural metric to consider" so that many constructions rely on it [21, 14, 10]. Nevertheless, with the exception of iris [13], Hamming distance does not adjust to biometric matchers [25, 18].

On the contrary, the set difference metric appears well fitted whenever the noisy secret is a set of features such as for digital fingerprints or even the exotic movie liker problem [34, 25, 14, 20]. Unfortunately, Blanton and Aliasgari [7] have shown that available set difference-based constructions [14] do not fulfill reusability [9] so that, at this time, the only way to handle set difference metric is to adapt the work of Canetti *et al.* [10] using the equivalence recalled earlier. This could lead to a problematic state of affairs when set elements belong to a large universe.

Now, in addition to these theoretical constraints, some more practical considerations have to be taken into account. Current industrial authentication solutions tend to be both smartphone and (only) software based solutions. As an illustration, the very trendy use case of HCE-based payment solution [2] requires a strong authentication.In most cases, strong authentication is achieved by fulfilling both knowledge (*e.g.* knowing a password) and owning (*e.g.* owning a chip card) factors. Because of this purely software constraint, new authentication scenarios have to be found. Related specifications [1] suggest to exploit biometric fingerprints. Besides biometric related limitations recalled above, only brand new devices handle such signals while the smartphone fleet of the general public remains non-equipped.

Finally, for both theoretical and practical reasons, it is worth looking for an alternative randomness source that could be both be an answer to industrial needs and appears as an effective playground for FEs.

1.3 Fingerprinting as a new playground

FEs can theoretically apply to any noisy and identifying kind of data. We then propose the the burgeoning field of fingerprinting as an application. In our knowledge, no previous study mentions this context for FEs.

Brought to light by the seminal work of eckersley [16], browser fingerprinting consists in issuing a fingerprint from a browser (user agent, list of fonts, list of plug-ins,...). By visiting the associated website [16], anyone can be revealed the entropy contained in its browser configurations. Accessing this data enables to compare different values and to detect returning browsers even when some features may have changed over time. As consequent studies [8, 24, 26, 3], this solution was suitable for computers but looses its interest in the case of mobile devices. Later on, some solutions for mobile devices have been proposed but were either too resources demanding [23, 11] or focused on Identifying the device but did not take user's behavior into consideration [12, 36]. While the list of installed applications had already been considered to identify users [32, 4], this is only with the comprehensive work of [22] that a device oriented study equivalent to [16] has been provided. Under apple's iOS, Kurtz *et al.* show how to compute a device fingerprint using 29 different configuration features. From a dataset of 13,000 fingerprints, they show that all these fingerprints are unique and detected returning devices with an accuracy of 97%. Along their work, the list of installed applications and the top 50 songs appear are the most identifying values present on a device. This recent accumulation of such studies exhibits the great amount of entropy mobile devices may wear. Hence, these fingerprints, revealing user's habits and behavior, appear as a credible factor when designing a strong authentication protocol. Many of these device/browser fingerprints constitute list of features coming from potentially unsamplable universes (*e.g.* songs, applications, plug-ins, ...) and then naturally fit with the large universe. This enhances the need for a reusable FE in the large universe setting.

1.3.1 With new playground comes new problematic

However, a new limitation fast appears when dealing with such fingerprints: even if identifying, they may undergo deep variations over time so that extracted keys could suffer shorter lifetime when the stated goal of FEs is to generate stable keys. Drawing a parallel with behavioral biometrics, we then a propose a new definition for FEs that aims at addressing this latter limitation.

Physiological biometrics undergo minor and genuine differences which, as we saw, caught the attention of FEs. More precisely, if we consider an enrollment value ω, it is very likely that any new reading w' will stay within a certain distance. While these latter enjoy a comfortable permanence (even if concerned by intrinsic aging), some of the behavioral biometrics can suffer from a shorter permanence period that has to be handled. Moreover biometric systems are constrained to enroll the user with as few as possible learning templates (from 1 to 10). To solve both of the previous issues, the notion of *adaptive* biometric system has been introduced in [35] for which template database is updated whenever a successful authentication occurs.

From these behavioral biometric problematic, one could draw a a parallel with device and browser fingerprints. In the purpose of exhibiting privacy threats, recent studies highlight how identifying fingerprinting can be [16, 3, 4, 22]. To fairly identify users, current solutions have to handle variability of such fingerprints (cookies, canvas fingerprint, installed applications,...). This leads to the problematic of *uniquely identifying* users where the goal is to decide if a given profile is a former one that has undergone variations or is a new one [16, 3, 22].

To avoid any ambiguity, we will talk about *instability* of physiological biometrics to indicate that they may undergo minor differences while *variability* of behavioral, device and browser fingerprints indicates that these latter ones may suffer deeper changes.

Coping with the terminology of adaptive biometric systems, we define *adaptive* fuzzy extractors meant to output a stable key even if the authentication value w' is not that close to the enrollment one W, as long as w' has somewhat naturally drifted from w.

1.4 Our contribution

The contribution of this work is threefold:

1. We propose a generic framework to address reusability that can be instantiated with any nonreusable FE. The idea is to pre-process fuzzy secrets while maintaining distances between two noisy readings. More precisely, when ρ enrollments have to be done from the fuzzy secret w, such a process will output ρ unrelated values $\Omega^1, \ldots, \Omega^\rho$ that can then be securely used once by any nonreusable FE. Now, if a user wants to authenticate herself toward provider j from a noisy reading w', our pre-processing function will generate Ω'_j such as the distance between Ω'_j and Ω^j is the same as between w and w'. We formalize this fingerprint randomization by defining *Fingerprint derivation Functions* (FDFs).

2. Relying on the previously introduced framework, we design the first practical set difference-based reusable FE in the large universe setting that will enjoy efficient time and storage complexities of the best nonreusable FEs. Moreover, our instantiation in the small universe setting, with n denotes universe size, comes up with both time complexity and error correction in $O(n)$ while adapting Hamming-based construction of Canetti *et al.* through $\mathsf{bin\text{-}set}$ equivalence leads to time complexity of order $\omega(n^c.logn)$ and error correction only sublinear in n. Plus, by design of their scheme, increasing the correction capacity leads to increasing c.

3. Considering fingerprints with deeper variations (*e.g.* behavioral, device and browser fingerprints) but still identifying [35, 16, 3, 22] led us to define *Adaptive* Fuzzy Extractors (AFEs) that aim at recovering a stable key R from noisy readings even if these noisy readings are not that close. Taking into consideration more variable but still identifying fingerprints, AFEs enable the key recovery as long as the authentication value has naturally drifted from the enrollment one. In addition to primitives Gen and Rep, an update primitive Upd is introduced. We propose a generic methodology to design an AFE out of an FDF and a nonreusable FE. Once again, an efficient instantiation for the set difference is proposed.

2 Preliminaries

2.1 Notation

For a vector u, we denote $\mathsf{supp}(x) \overset{def}{=} \{i : x_i \neq 0\}$. 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 H denotes a cryptographic hash function modeled as a random oracle $\mathsf{H} : \{0,1\}^* \times \{0,1\}^{l_1} \rightarrow \{0,1\}^\kappa$ 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 $\delta^D(X,Y) = |\mathbb{E}[D(X)] - \mathbb{E}[D(Y)]|$. When X and Y indistinguishable given security parameter sec ($\delta^D(X,Y) \leqslant \epsilon_{\text{sec}}$), we denote $X \approx Y$. $\mathcal{D}_{s_{\text{sec}}}$ denotes the class of randomized circuits which output a single bit and have site at most s_{sec}. We say $W^1, \ldots, W^n$ random variables are *correlated by subset* if it exists $p \leqslant n$ such as $\{i_1, \ldots, i_p\} \subset \{1, \ldots, n\}$ and $W^{i_1}, \ldots, W^{i_p}$ are correlated. Let λ denotes a security parameter. Except stated otherwise, we have $l = l(\lambda)$, $\kappa = \kappa(\lambda)$, $m = m(\lambda)$, $s_{\text{sec}} = \mathsf{poly}(\lambda)$ and $\epsilon = \mathsf{negl}(\lambda)$.

2.2 Background

Metric Spaces A metric space is a finite set $\mathcal{M}$ equipped with a distance $d : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{N}$ fulfilling classic properties of symmetry, triangle inequality and zero distance between equal points.

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 $w \Delta w' \overset{def}{=} \{x \in w \cup w' | x \notin w \cap w'\}$. The set difference metric between w and w' is the defined as $d(w, w') \overset{def}{=} |w \Delta w'|$.
We now present the bin-set equivalence. If w denotes a set of size s, it can be viewed a binary vector in $\{0,1\}^n$, with 1 at position x if $x \in w$, and 0 otherwise.

Entropy Notions Entropy specifies the amount of information contained in some data. In security-related contexts, the (non) ability (for an adversary) to guess the value of a random variable is a measure of its quality. In the information-theoretic case, we often rely on the well-suited notion of *min-entropy*. We say that 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 $H_\infty(A) \overset{def}{=} -log_2 max_{a \in A} P[A = a]$. Another useful notion is the *average (conditional)* min-entropy of A given B is $\tilde{H}_\infty(\mathrm{A}|\mathrm{B}) = -log_2(\mathbb{E}_{b \in \mathrm{B}} \max_a \Pr[\mathrm{A} = a | \mathrm{B} = b])$. We do not recall computational entropy notions since Fuller *et al.* [17] exhibited that no interest arises when considering computational, instead of statistical, entropy in the field of Fuzzy Extractors.

Fuzzy Extractors Recalls

The original definition of FEs, due to Dodis *et al.* [14], was information theory-based. We focus on the computational one introduced in [17]. While Fuller *et al.* voluntary extend their definition to any family of distributions for a convenience purpose, we focus on distributions of given min-entropy as it was the case in [14]. Relying on the original definition, we study the security over distributions of min entropy m.

Definition 1 (Fuzzy Extractor)**.** *A pair of randomized procedures "generate" (Gen) and "reproduce" (Rep) is a $(\mathcal{M}, m, l, t)$-computational fuzzy extractor (resp. average case) that is (ϵ, 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 property guarantees that if $d(w, w') \leqslant t$ and $(R, P) \leftarrow$ Gen(ω), then* Rep$(\omega', P) = R$. *If $d(w, w') > t$, then no guarantee is provided about the output of* Rep.

- *The security property guarantees that for any distribution W on $\mathcal{M}$ of min-entropy m, the string R is pseudorandom conditioned on P i.e. $\delta^{s_{sec}}((R, P), (U_l, P)) \leqslant \epsilon$.*

Dodis *et al.* also define *average-case* FEs for which the security property requires that for any auxiliary variable I such as $\tilde{H}_\infty(W|I) \geqslant m$, then $((R, P, I), (U_l, P, I))$ appear indistinguishable. As recalled in [17], any information-theory based FE [14] can be relaxed into a computational FE with the same parameters. Along with their work, 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) such as on input ω, the "sketch" SS procedure 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 [9] can be stated as the possibility to call procedure Gen numerous times on the same fuzzy secret without impacting security or privacy. 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 so that Gen will then generate ρ couples $(R^1, P^1), \ldots, (R^\rho, P^\rho)$. Recalling that P^js are meant to be public and that different servers should not trust each other, Canetti *et al.* [10] proposed an adapted 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 2 (Reusable Fuzzy Extractor [10])**.** *Let (Gen, Rep) be a $(\mathcal{M}, m, l, t, \epsilon)$-FE and $W^1, W^2, \ldots, W^\rho$ be ρ correlated random variables over $\mathcal{M}$. Let D be an adversary. Define the following game for all $j = 1, \ldots, \rho$:*

- ***Sampling*** *The challenger $\mathcal{C}$ samples $\omega^j \leftarrow W^j$ and $\eta \leftarrow \{0,1\}^l$.*

- ***Generation*** *$\mathcal{C}$ computes $(R^j, P^j) \leftarrow$ Gen(ω^j).*

- ***Distinguishing*** *The advantage of D consists in:*

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

(Gen, Rep) is $(\epsilon, \rho, s_{sec})$-reusable if for all $D \in \mathcal{D}$ and for all $j = 1, \ldots, \rho$, the advantage $Adv(D)_{s_{sec}}$ is at most ϵ.

2.3 Tools

In the work of Canetti et al. [10], the secret key R outputted by Gen consists in a randomly generated value while the helper string P contains numerous ciphertexts of R encrypted under substrings of the noisy secret ω. To securely encrypt numerous times the key R, they made use of digital lockers that are reusable symmetric encryption schemes with correlated and weak keys. The reusability of digital lockers implies the FE reusability. We now describe the tools demanded to design reusable FEs fitting our methodology.

Symmetric encryption scheme We will also require the use of a symmetric encryption scheme that will be denoted $(\mathsf{Enc}, \mathsf{Dec})$. Correctness ensures that $\mathsf{Dec}(K, \mathsf{Enc}(K, D)) = D$. From a security point of view, we require $(\mathsf{Enc}, \mathsf{Dec})$ to fulfill the "find-then-guess" chosen plaintext attack (FTG-CPA) security due to Bellare *et al.* [5]. This notion is the closest notion to the classic CPA security defined for public key schemes. Since an adversary cannot encrypt messages on its own in the private key paradigm, the FTG-CPA security let her access to an encryption oracle. The ensuing game then consist for an adversary with such a polynomially bounded oracle access to send m_0 and m_1 to a challenger that will compute c_b for $b \in \{0, 1\}$. $(\mathsf{Enc}, \mathsf{Dec})$ is said to be FTG-CPA if any computational adversary has negligible advantage in guessing b.

Fingerprint Derivation Function *Fingerprint Derivation Functions* (FDFs), that can be seen as one-way isometry functions, constitute a new direction to solve FEs' reusability. Given a fuzzy secret w, a FDF aims at randomly deriving Ω retaining much of the entropy of w and enabling to maintain distances between inputs and outputs. More precisely, a FDF is as a couple of randomized procedures $(\mathsf{DerGen}, \mathsf{DerRep})$ working as follows:

- DerGen on input ω randomly returns some unrelated Ω and some public string F.
- DerRep on input a noisy version ω' and some F returns Ω' such the distance between Ω and Ω' is equal to the distance between ω and ω'.

This notion could be related to secure sketches and biometric embeddings both used in [14]. In fact, a biometric embedding projects any fingerprint value into a metric space where a FE exists while loosely maintaining distances. On the other hand, a secure sketch enables to recover ω from ω' and some public helper string. The idea here is to *precisely* maintain distances while not enabling *any recovery* of original fingerprint values.

Adapting definition of secure sketches [14], the security property requires that both original and derived distributions to be of high entropy even in presence of the outputted public string F.

Definition 3 (Fingerprint Derivation Function)**.** *Let $(\mathcal{M}_1, d_1)$ and $(\mathcal{M}_2, d_2)$ be two metric spaces. A $(\mathcal{M}_1, \mathcal{M}_2,$
$m_1, m_2)$-fingerprint derivation function (FDF) with error 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 $d_2(\Omega, \Omega') = d_1(w, w')$. Else, no guarantee is provided about Ω'.*
3. *Let W a distribution, DerGen(W) is denoted (U, V). If $\omega \overset{\$}{\leftarrow} W$ and $(\Omega, F) = \mathsf{DerGen}(\omega)$, we denote $(\Omega, F) \overset{\$}{\leftarrow} (U, V)$. The security property guarantees that for any distribution W of*

min-entropy m_1, the values of W and U can be recovered by the adversary who observe V with probability no greater than 2^{-m_2}.

That is, we have $\tilde{H}_\infty(W|V) \geqslant m_2$ and $\tilde{H}_\infty(U|V) \geqslant m_2$.

The intuition behind FDF-reusability is that the knowledge of previous derived fingerprints will not help any computational adversary D to distinguish a random value from a newly derived fingerprint obtained via DerGen. Relying on the reusability of FEs (Definition 2), we define a reusable FDF as follows.

Definition 4 (Reusable FDF)**.** *Let $W^1, W^2, \ldots,$
W^ρ be ρ correlated random variables over $\mathcal{M}_1$. Let D an adversary. Using notation of Definition 3, we define the following game for all $j = 1, \ldots, \rho$:*

- ***Sampling** The challenger $\mathcal{C}$ samples $w^j \leftarrow W^j$ and $u \overset{\$}{\leftarrow} \mathcal{M}_1$.*
- ***Generation** $\mathcal{C}$ generates $(\Omega^j, F^j) \leftarrow$ DerGen(ω^j) and $(\Omega^*, F^*) \leftarrow$ DerGen(u).*
- ***Distinguishing** The advantage of D consists in:*

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

(DerGen, DerRep) is said to be $(\epsilon, \rho, s_{sec})$-reusable if for all $D \in \mathcal{D}_{s_{sec}}$ and for all $j = 1, \ldots, \rho$, the advantage $Adv(D)$ is negligible in s_{sec}.

3 From nonreusable to reusable Fuzzy Extractors

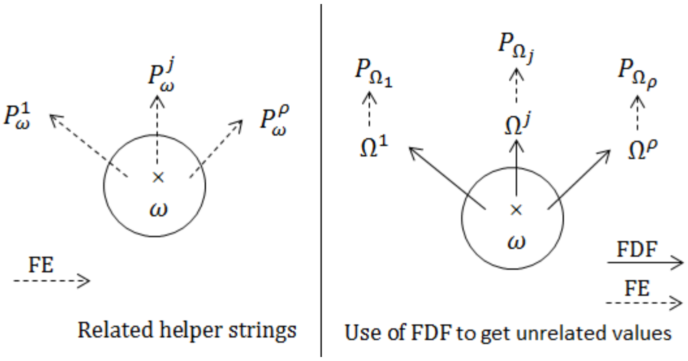
Before the recent work of Canetti *et al.*, all Hamming-based [33] and set difference-based [7] constructions, were prone to reusability issue. In this section, we introduce a new and generic way to address reusability. The idea is to first use a FDF to randomize fuzzy secrets and then apply a nonreusable FE on unrelated derived fingerprints. Based on nonreusable FEs due to Dodis *et al.* [14], we propose a reusable FE that enjoys better storage, time and correction complexities compared to the work of Canetti *et al.* [10] adapted to the set difference metric.

3.1 High Level Overview

Let (Gen', Rep') denote a (*average-case*) nonreusable FE. Taking as input a fuzzy secret ω, the generation procedure Gen' implicitly draws a ball $\mathcal{B}(\omega, t)$ centered in 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 $\mathcal{B}(\omega, t)$. In most cases and notably for constructions based on secure sketches, if numerous helper strings $P^1_\omega, \ldots, P^\rho_\omega$ are generated (Gen') from the same fuzzy secret $\in \mathcal{B}(\omega, t)$, all the P^is inherently contain information about ω which leads to reusability issue.

Let us consider the situation where a user wants to enroll its fuzzy secret ω on ρ authentication servers. To address reusability, the key idea is to randomly project the ρ fuzzy versions of ω onto unrelated values. By using an adapted ρ-reusable FDF, 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 FDF to get Ω'^j verifying $d(\Omega'^j, \Omega'^j) = d(w, w')$. Since the Ω^js are -a priori- uncorrelated while each Ω^j undergoes only one generation procedure. Hence, one should be able to enroll herself from the same fuzzy secret without fearing reusability issue. This idea is summed up in Figure 3.1.



Related helper strings

Use of FDF to get unrelated values

Figure 1: Overview of reusability via FDF randomization

3.2 Generic methodology

In this subsection, we detail our generic methodology to design a reusable FE (Gen, Rep) out of a nonreusable one and an adapted FDF. Let (DerGen, DerRep) be a $(\mathcal{M}_1, \mathcal{M}_2, m_1, m_2)$-FDF that is $(\epsilon, \rho, s_{\text{sec}})$-reusable. Let (Gen', Rep') be an average-case $(\mathcal{M}_2, m_2, l, t)$-FE that is $(\epsilon, s_{\text{sec}})$ hard.

Designing a reusable FE out of these latter tools is rather straightforward. As depicted in Figure 3.1, FDF reusability will enable to generate numerous unrelated values that will imply FE reusability. The generation procedure Gen will first call DerGen to randomize the input w into Ω. The nonreusable FE is then applied on Ω. Because of the FDF reusability, numerous projected fingerprints Ω^js can be derived without impacting security of the original fuzzy secret w. Finally, the generated helper strings P_{Ω_i}s ($P_{\Omega_i} \leftarrow$ Gen'(Ω_i)) can be publicly revealed assuming security of (Gen', Rep') in the single use case. The overall helper string consists in the helper strings of both the FDF and the nonreusable FE.

Generation procedure Gen

```
Input: $w \in \mathcal{M}_1$,
1. $(\Omega, F) \leftarrow$ DerGen$(w)$.
2. $(R, Q) \leftarrow$ Gen'$(\Omega)$.
3. Set $P = (F, Q)$.
4. Return $(R, P)$.
```

Reproduction procedure Rep

```
Inputs: $w' \in \mathcal{M}_1$,
        Helper data $P \in \{0,1\}^*$.
1. Parse $P = (F, Q)$
2. $\Omega' \leftarrow$ DerRep$(w', F)$.
3. $R \leftarrow$ Rep'$(\Omega', Q)$.
4. Return $R$.
```

Figure 2: A generic reusable FE

The FDF ensures that $d_2(\Omega, \Omega') = d_1(w, w')$ 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 $d_1(w, w') \leqslant t$. One should notice that this methodology also applies to secure sketches.

Security Analysis Correctness and security properties rely on those of (DerGen, DerRep) and ($\mathsf{Gen'}$,$\mathsf{Rep'}$).

Theorem 1. *Figure 2 defines a $(\mathcal{M}_1, m_1, l, t)$-FE that is $(\epsilon, \rho, s_{sec})$-reusable.*

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 3, we have that Ω and F respectively come from distribution U and V such as $\tilde{H}_\infty(U|V) \geqslant m_2$. Now, with ($\mathsf{Gen'}$,$\mathsf{Rep'}$) an average-case FE, we have that $(R, Q, I) \approx (U_l, Q, I)$ for any distribution I such as $\tilde{H}_\infty(U|I) \geqslant m_2$. In particular, $\delta^D((R, Q, F), (U_l, Q, F)) \leqslant \epsilon_{\text{sec}}$ which leads to $\delta^D((R, P), (U_l, P)) \leqslant \epsilon_{\text{sec}}$ denoting $P = (F, Q)$.

Let $W^1, \ldots, W^\rho$ be correlated distributions over $\mathcal{M}_1$, all of min-entropy m_1. Through a sequence of games, We will now demonstrate that the reusability of the FDF and security of the FE ($\mathsf{Gen'}$,$\mathsf{Rep'}$) in the single use case lead to the overall reusability of (Gen, Rep). The following games consist in a challenger $\mathcal{C}$ trying to fool a computationally bounded adversary D whose goal is to attack the security of the key R^{i_0}.

$\mathcal{G}_0$ $\mathcal{C}$ honestly samples fingerprint values as prescribed in definition 2 and sends $(F^1, Q^1, K^1), \ldots, (F^{i_0}, Q^{i_0}, K^{i_0}), \ldots, (F^\rho, Q^\rho, K^\rho)$ to D.

$\mathcal{G}_1$ In this game, there is one change compared to the previous one. $\mathcal{C}$ first honestly samples the w^js and then uses DerGen to obtain $\Omega^1, F^1, \ldots, \Omega^\rho, F^\rho$. Now, $\mathcal{C}$ replaces w^{i_0} with some random $w^* \overset{\$}{\leftarrow} \mathcal{M}_1$. He computes $(\Omega^*, F^*) \leftarrow \mathsf{DerGen}(w^*)$ and $(R^*, Q^*) \leftarrow \mathsf{Gen'}(\Omega^*)$ to finally set $P^* = (F^{i_0}, Q^*)$. In the end, D is given with the actual P^js and R^js except for $j = j_0$ for which he receives P^*, R^*. If D could distinguish this game from the previous one, he would then be able to distinguish the distribution with Ω^{j_0} from the one with Ω^*. This is impossible assuming reusability of the FDF. Then, $\mathcal{G}_1$ appears indistinguishable $\mathcal{G}_0$.

$\mathcal{G}_2$ In this game, after computing, $R^*, Q^* \leftarrow \mathsf{Gen'}(\Omega^*)$, the challenger $\mathcal{C}$ discards the value R^* and replaces it with some $\eta \overset{\$}{\leftarrow} \{0,1\}^l$ randomly sampled. Now, assuming security of the average case FE ($\mathsf{Gen'}$, $\mathsf{Rep'}$), we have that (U_l, P^*) and (R^*, P^*) are computationally indistinguishable so that this game appears indistinguishable from the previous one.

$\mathcal{G}_3$ In the previous game, D was given finally $(F^1, Q^1, K^1), \ldots, (F^{i_0}, Q^*, \eta), \ldots, (F^\rho, Q^\rho, K^\rho)$ where η is random and as such, does not depend on P^*. Here, $\mathcal{C}$ will then send 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 $(F^1, \ldots, F^{i_0}, \ldots, F^\rho, ..., \Omega^1, \ldots, \Omega^{i_0}, \ldots, \Omega^\rho)$ from $(F^1, \ldots, F^{i_0}, \ldots, F^\rho, ..., \Omega^1, \ldots, \Omega^{i_0-1}, \Omega^*, \Omega^{i_0+1}, \ldots, \Omega^\rho)$. In other words, D would have finally broken reusability of the underlying FDF. Then $\mathcal{G}_3$ is indistinguishable from $\mathcal{G}_2$.

In the end of $\mathcal{G}_3$, D is given $(P^1, K^1), \ldots, (P^{i_0}, \eta), \ldots, (P^\rho, K^\rho)$ where η was randomly sampled. By transitivity, this latter game is indistinguishable from $\mathcal{G}_0$. This latter indistinguishability is exactly the one required by Definition 2.

■

3.3 A Set Difference-based FDF

In this subsection, we present a set difference-based FDF that will enable us to instantiate our generic methodology described in previous subsection. Our instantiation naturally fits both small and large universe settings while the Hamming-based work of Canetti *et al.* can only be extended to the small universe setting. We will see that even in this latter case, our work benefits better storage and running time complexities.

Environment and Notation Set difference based fuzzy extractors presented in [14] take as inputs subsets of a universe $\mathcal{U}$ such as $n = |\mathcal{U}|$. We denote $(\mathcal{M}_{\mathcal{U}}, d)$, the metric space where $\mathcal{M}_{\mathcal{U}}$ consist in all the subsets of $\mathcal{U}$ and d is the set difference metric. Plus, $\mathcal{M}_{\mathcal{U},s}$ denotes the restriction of $\mathcal{M}_{\mathcal{U}}$ to s-elements subsets. In the following, $\mathcal{M}_{\kappa}$ 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. Let W^s be defined as $W^s \overset{def}{=} \{\{x_1, x_2, \ldots, x_s\} | \forall 1 \leqslant i \leqslant s, x_i \overset{\$}{\leftarrow} W\}$; W^s then represents a distribution over $\mathcal{M}_{\mathcal{U},s}$ and we denote its entropy $\tilde{m}$.

Proposition 1. *With the above notation, the min-entropy $\tilde{m}$ of W^s is such as:*

$$\tilde{m} \geqslant s.m - log(s!)$$

Proof. Since $H_{\infty}(W) = m$, we have $Pr[W = x] \leqslant 2^{-m}$. Following the associated tree diagram, any tuple $(x_1, \ldots, x_s)$ has probability to occur less than $2^{-s.m}$. When it comes to sets (no order consideration), $\{x_1, \ldots, x_s\}$ has then an occurring probability less than $2^{-s.m}.s!$. ■

To design our set difference-based FDF, we will use the hash function $\mathsf{H} : \{0,1\}^* \times \{0,1\}^{l_1} \to \{0,1\}^{\kappa}$. Modeled as a random oracle, its outputs appear uniformly distributed and contain all the entropy of its input. In a nutshell, our set difference-based FDF, presented in Figure 3, consists in randomizing each set element through the use of H.

```
Algorithm DerGen
Input: $w = \{w_1, \ldots, w_s\}$,
        $\forall 1 \leqslant i \leqslant s,\ w_i \in \mathcal{U}$.
1. $\mathsf{salt} \overset{\$}{\leftarrow} \{0,1\}^{l_1}$.
2. For $i = 1 \ldots s$,
     $x_i \leftarrow \mathsf{H}(w_i; \mathsf{salt})$.
3. Set $\Omega = \{x_1, \ldots, x_s\}$.
4. If $|\Omega| < s$,
     go to 1.
5. Return $(\mathsf{salt}, \Omega)$.
```

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

Figure 3: A set difference-based FDF

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 no collision occurs with non negligible probability. In the same vein, Step 3 of Algorithm DerRep enables to maintain distances between original values (w and w') and randomly derived ones (Ω and Ω'). Indeed, thank to step 4 of DerGen, a collision occurring while proceeding DerRep on w' can only be due to an element w'_i that did not appear in w.

Proposition 2. *Figure 3 defines a* $(\mathcal{M}_{\mathcal{U},s}, \mathcal{M}_\kappa, m_1, m_2)$*-FDF for the set difference metric where* $m_1 = \tilde{m}$ *and* $m_2 = s.m - log(s!)$.

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

1. *Isometry property.* By design, Ω was derived from w ensuring that no collision occurs when generating the x_is belonging to Ω (Step 4). To prove the isometry of our FDF, there are two cases to consider:

   - If no collision occurs, isometry is straightforward: elements common to w' and w lead to common elements in Ω' and Ω; elements that differ from w' and w lead to different elements in Ω and Ω' so that distances are maintained.
   - Else, we have that $|\Omega'| = s' < s$. It means that some collision occurred within during step 1. By design of DerGen, every time a collision occurs, it can only be due to some w'_i not belonging to w. Hence, if $|\Omega'| < s'$, missing elements of Ω' are necessarily due to differences between w' and w so that adding the expected number of random elements to Ω' (Step 3 of DerRep) ensure isometry property of (DerGen, DerRep).

2. *Security.* Because of H modeled as a random oracle, we have that:

   - It keeps all the entropy of its inputs. That is, if $H_\infty(w_i) = m$ and $x_i = \mathsf{H}(w_i; \mathsf{salt})$, then $H_\infty(x_i) = m$.
   - Its outputs are uniform: if $x_i = \mathsf{H}(w_i; \mathsf{salt}) \in \{0,1\}^\kappa$, then its κ bits are randomly distributed.

   Thanks to the properties described above H, $\Pr[X = x_i] \leqslant 2^{-m}$. Relying on notation introduced in Definition 3 and the reasoning of proposition 1, we have $H_\infty(U) \geqslant s.m - log(s!)$. With H a random oracle, the knowledge of the random salt does not impact any entropy loss: $\forall\, \mathsf{salt}, H_\infty(X = x_i | V = \mathsf{salt}) = m$. It leads to $H_\infty(U = \Omega | V = \mathsf{salt}) \geqslant s.m - log(s!)$ to finally give $\tilde{H}_\infty(U|V) \geqslant s.m - log(s!)$. Now, since the value salt is chosen randomly and independently of distribution W (algorithm DerGen, Step 1) It leads to $\tilde{H}_\infty(W|V) = H_\infty(W) = \tilde{m}$.

■

Proposition 3. *Figure 3 defines a set difference-based FDF that is* $(\epsilon, \rho, s_{sec})$*-reusable.*

Proof. 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 $\rho \in \mathbb{N}$. Let $W^1, \ldots W^\rho$ be related distributions from which $w^1, \ldots, w^\rho$ are respectively sampled (some or all w^js could then be equal). Let us assume that one is given $(\mathsf{salt}^1, \Omega^1), \ldots, (\mathsf{salt}^\rho, \Omega^\rho)$ where $(\mathsf{salt}^j), \Omega^j \leftarrow \mathsf{DerGen}(w^j)$. Now, let us replace some $(\mathsf{salt}^{j_0}, \Omega^{j_0})$ with $(\mathsf{salt}^*, \Omega^*) \leftarrow \mathsf{DerGen}(w^*)$ where w^*, a priori unrelated to w^js, is randomly sampled from $\mathcal{M}_\mathcal{U}$. Since public salts salt^js are generated randomly and independently from any other values, they can be replaced by other random values. Plus derived values Ω^js are uniformly distributed over $\mathcal{M}_{m,s}$ through the use of H modeled as a random oracle. Hence, none adversary D can distinguish aforesaid distributions assuming security of H. ■

Corollary 1. *Let (Gen', Rep') be an average-case $(\mathcal{M}_\kappa, m_1, l, t)$ nonreusable FE based for the set difference metric that is (ϵ, s_{sec}) hard. Let (DerGen, DerRep) be the $(\epsilon, \rho, s_{sec})$-reusable $(\mathcal{M}, \mathcal{M}_\kappa, m_1, m_2)$-FDF defined in Figure 3. From Theorem 1, one can design (Gen,Rep) a $(\mathcal{M}, m_1, l, t)$-FE that is $(\epsilon, \rho, s_{sec})$-reusable based on set difference metric.*

This result is straightforward due to Theorem 1 and Propositions 2 and 3. Let us notice that with previous notation, $m_1 = \tilde{m}$ and $m_2 \geqslant s.m - log(s!)$.

3.3.1 Comparison with work of Canetti *et al.*

We now compare our construction with the work of Canetti *et al.* [10] adapted to set difference metric. While the set-bin equivalence cannot be applied in the more interesting large universe setting, we consider that it has no cost in the small universe setting where $n = \mathsf{poly}(s)$. Let t denotes the number of errors tolerated by a FE. The work of Canetti *et al* only achieves a correction capacity sublinear in n. If one wants to increase t, it is required to increase some constant c for which both generation and reproduction procedures are linear in n^c ($O(n^c)$). For similar reasons, their storage complexity is equivalent to $\omega(n^c.log^2(n))$. From efficiency of our FDF, our asymptotic performances are mainly due to the underlying nonreusable FE. Such efficient constructions are proposed in [14] for which time and storage complexities are respectively of order $t.logn$ and poly(n). Plus the time complexity is the consequence of running the decoding algorithm of an error correcting code $[n, k, 2t + 1]$. With an instantiation based on LDPC codes [29], we get that both error correction and time complexity can be linear in n. Nevertheless, contrary to the work of Canetti *et al.*, our construction is instantiated with secure skecthes implying some entropy loss, estimated around $tlogn$.

One should notice that -still basing our instantiations on [14]- our work enjoys storage and time complexities of order $tlogn$ (with equivalent entropy loss) and $\mathsf{poly}(slogn)$ in the large universe setting.

4 Adaptive Fuzzy Extractors

While Fuzzy Extractors were originally designed to handle any kind of noisy and keying material, they fast gained a lot of interest in the field of biometrics and PUFs. Unhappily as exhibited in Section 1, these applications suffer numerous limitations and in particular, biometrics still lack of practicability with respect to existing fuzzy extractors. This state of affairs led us to consider the burgeoning alternative of browser and device fingerprints [16, 3, 22] that will naturally derive over time leading to changes much more important than genuine instability of classic biometrics. We then define *Adaptive* Fuzzy Extractors, meant to address this drifting problematic.

4.1 Environment: Alternative randomness and Drift problematic

Device and browser fingerprints (*e.g.* list of installed applications, list of cookies, ...) are much more versatile than biometrics. It raises the problematic of *uniquely* identifying users. More precisely, given a fingerprint value w', 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 should not be assimilated to a new profile. Recent works have come with satisfying results with respect to this problematic [16, 3, 22]. Transposed in the context of authentication, this rises the problematic of authenticating users that might present numerous differences. Even if important, these differences do not prevent fingerprint values to be reliable [16, 3, 22]. In our context, this amounts to deal with a fuzzy extractor that can recover the actual key R even if the authentication value w' and the enrollment one w present more than t errors. The idea is to say that w' should have naturally drifted from w. Drawing a parallel with the *concept drift* problematic of predictive analytics, we refer to these fingerprint derivation as *fingerprint drift* and formalize it through the following definition

Definition 5. *Let $(\mathcal{M}, d)$ a metric space. Let $w^1, \dots, w^\phi$ elements of $\mathcal{M}$ and an integer t. We say that $(\mathcal{M}, w^1, w^\phi, t)$ is a t-drift of length ϕ on $\mathcal{M}$ if for all $j = 1, \dots, \phi - 1$, we have that $d(w^j, w^{j+1}) \leqslant t$.*

This definition is simple and may not capture all the subtleties a fingerprint may fit. This a first stone that should be improved at the light of further studies on fingerprinting. In the context of fuzzy extractors, a naive answer to the fingerprint drift issue could be frequent re-enrollments. Nevertheless, in practice, enrollment sessions always constitute critical sessions that organizations want to avoid. Plus, FEs were originally designed to enable the use of long term secrets so that frequent re-enrollment sessions would annihilate their primary goal.

4.2 High Level Overview

To cope with this drifting paradigm, we define *Adaptive* Fuzzy Extractors (AFEs) that enjoy a third primitive Upd compared to classic FEs. Without re-enrolling herself, a user should be able reproduce the same secret R than the one computed by Gen as long as variations between the enrollment value w and the authentication one w' follow an expected t-drift (Definition 5). A classic FE is meant to recover a previously extracted key R if and only if the reproduction value w' belongs to $\mathcal{B}(w, t)$. In our context, we require an AFE to recover the actual key R as long as the reproduction value w' has somewhat *naturally* drifted from w although w' does not belong $\mathcal{B}(w, t)$. The terms *naturally* refers here to following a t-drift.

More precisely, 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 w' is still close enough to w ($d(w, w') \leqslant u$).
- *Recovering Zone.* w' is close enough to w to enable key recovery but too far away to enable any helper string update ($u < d(w, w') \leqslant t$); further reproduction values w' should draw near w to enable an update.

In particular, we have that an adaptive fuzzy extractor can recover R as long as the fuzzy fingerprint w defines an u-drift. Without representing updating zones for the sake of clarity, the philosophy of adaptive fuzzy extractors is depicted in Figure 4.2.

4.3 Definition and Security Model

We now formally define Adaptive Fuzzy Extractors.

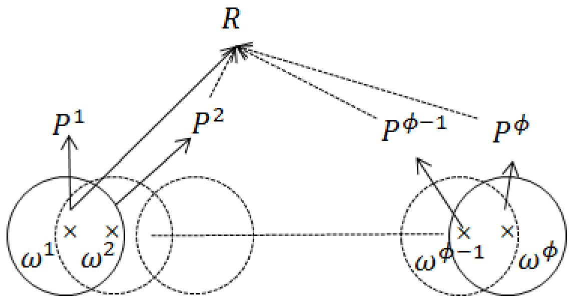


Figure 4: Philosophy of Adaptive Fuzzy Extractors.

Definition 6 (Adaptive Fuzzy Extractor)**.** *A triple of randomized procedures "generate" (*Gen*), "update" (*Upd*), "reproduce" (*Rep*) is an* $(\mathcal{M}, m, l, u, t, \phi)$*-adaptive fuzzy extractor that is* (ϵ, s_{sec}) *hard if the following holds:*

1. *The generation procedure* Gen *on input* $w \in \mathcal{M}$ *outputs an extracted string* $R \in \{0,1\}^l$ *and an helper string* $P \in \{0,1\}^*$.

2. *The updating procedure* Upd *takes as inputs some* $w' \in \mathcal{M}$ *and a bit string* $P \in \{0,1\}^*$*. The correctness guarantees that if either of conditions is satisfied:*

   (a) $(K, P) \leftarrow \mathsf{Gen}(w)$ *and* $d(w, w') \leqslant u$;

   (b) *there exists some* (w^*, P^*) *such as* $P \leftarrow \mathsf{Upd}(w^*, P^*)$ *while* $K \leftarrow \mathsf{Gen}(w)$ *and* $(w^*, w') \leqslant u$,

   then Upd *outputs an updated helper string* P'*. In any other case, no guarantee is provided about the output of* Upd.

3. *The reproduction procedure* Rep *takes as inputs* $w' \in \mathcal{M}$ *and a bit string* $P \in \{0,1\}^*$*. The correctness guarantees that if either of conditions is satisfied:*

   (a) $(K, P) \leftarrow \mathsf{Gen}(w)$ *and* $d(w, w') \leqslant t$;

   (b) *there exists some* (w^*, P^*) *such as* $P \leftarrow \mathsf{Upd}(w^*, P^*)$ *while* $K \leftarrow \mathsf{Gen}(w)$ *and* $(w^*, w') \leqslant t$,

   then $\mathsf{Rep}(w', P) = R$*. In any other case, no guarantee is provided about the output of* Rep.

4. *The security property guarantees that for correlated distributions* $W^1, \ldots, W^{\phi}$ *(or correlated by subsets) on* $\mathcal{M}$ *with min-entropy* m*, the string* R *is pseudorandom even for those whose observe the* P^i*s for* $i = 1 \ldots \phi$ *where* P^i *consists in one of the helper strings generated, through* Gen *or* Upd*, from* $w^i \xleftarrow{\$} W^i$*. That is,* $\mathcal{D}^{\mathcal{D}_{sec}}((R, \{P^i\}_{i=1}^{\phi}), (U_l, \{P^i\}_{i=1}^{\phi})) \leq \epsilon$.

Security Model

We now define reusability of AFEs. One can consider that the newly introduced drifting problematic brings another dimension to the security model due to Canetti *et al.* 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 of its mercurial fingerprints. In such a context, helper strings generations (via Gen or Update) can be made either on correlated or uncorrelated random variables according to user's activity:

- when she subscribes to different services on a short period of time, her fingerprint values are close and belong to correlated variables. This scenario reduces to the previous one (Definition 2);
- she may have to update her fingerprints toward a particular subscriber. With a fingerprint following a u-drift, the user may then generate helper strings from correlated and/or uncorrelated variables;
- when she subscribes to two different services at distant time periods for respective fingerprint values w_{i_0} and w_{i_1}, these values may appear as providing from uncorrelated variables.

Adapting reusability definition of [10] (Definition 2), we propose the following game-based definition for security of reusable AFEs.

Definition 7. *Let $(W^1, W^2, \ldots, W^\rho)$ be -potentially correlated by subsets- random variables over $\mathcal{M}$. Let (Gen, Upd, Rep) be a $(\mathcal{M}, m, l, u, t, \phi)$-AFE that is (ϵ, s_{sec}) hard. Let D an adversary. Let the following game such as for all $1 \leqslant j \leqslant \rho$: and $1 \leqslant i \leqslant \phi$ respectively denoting the number of calls to Gen and Upd.*

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

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

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

One should notice that taking $\phi = 0$ and restricting the game to correlated variables W^js leads to the reusability game of classic FEs (Definition 2).

5 From classic FE to Reusable AFE

In this section, we design a reusable AFE out of a FE, a FDF and a symmetric encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$

5.1 Overview

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, FDFs address this point.
- a classic FE recovers an extracted K from w' as long as this latter is close enough to the enrollment value w. When too many differences appear between the reproduction value and the enrollment one, K cannot be recovered. 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. In fact, we will see that such temporary keys will be the ones outputted by a classic (*i.e.* non adaptive) FE.

Environment and Notation With notation of Subsection 3.3, we work with the following tools. Let $(\mathsf{Enc}, \mathsf{Dec})$ be a symmetric encryption scheme fulfilling FTG-CPA security. We have that $\mathsf{Enc} : \{0,1\}^* \times \{0,1\}^l \to \{0,1\}^*$ and $\mathsf{Enc} : \{0,1\}^* \times \{0,1\}^l \to \{0,1\}^*$. Let $(\mathsf{DerGen}, \mathsf{DerRep})$ be a $(\mathcal{M}_1, \mathcal{M}_2, m_1, m_2)$-FDF that is $(2\rho.\phi, s_{\text{sec}})$ reusable . Let $(\mathsf{Gen'}, \mathsf{Rep'})$ be an average-case $(\mathcal{M}_2, m_2, l, u(\text{resp. } t))$-FE $(\epsilon, s_{\text{sec}})$ hard.

Next subsections are dedicated to precisely detail how to design a reusable AFE out of $(\mathsf{Gen'}, \mathsf{Rep'})$, $(\mathsf{DerGen}, \mathsf{DerRep})$ and $(\mathsf{Enc}, \mathsf{Dec})$.

5.2 Generation procedure

Given an enrollment value w, the first step of the generation algorithm is to use the FDF to randomly project it onto two unrelated derived fingerprints Ω_u and Ω_t. Procedure $\mathsf{Gen'}$ of the nonreusable FE correcting u (*resp.* t) errors will then be applied on Ω_u (*resp.* 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. This latter is meant to be the key outputted by the AFE. In addition to helper strings outputted by DerGen and $\mathsf{Gen'}$, the overall helper string of our AFE contains encryptions of "1" and of R, respectively under K_u and K_t. string.

5.3 Update procedure

The update procedure takes as inputs a fuzzy version w' allegedly close to the enrollment value w, some helper data $P \in \{0,1\}^*$ to be updated into P', meant to be more relevant relatively to the current fingerprint value w'. Once again, the first step consists in deriving w' into Ω'_u and Ω'_t. Algorithm Gen implicitly defined updating and recovering zones from Ω_u and Ω_t. Hence, successful decryption of c_u under K_u recovers 1 and indicates that w' is within distance u. It then makes sense to call update. If so, R can be unlocked by re-generating K_t. $\mathsf{Gen'}$ then randomizes

w' into Ψ_u and Ψ_t along with ensuing and new helper strings. Finally, new temporary keys K'_u and K'_t are computed and R is re-encrypted under K'_t.

Generation and Update procedures are depicted in Figure 5.

```
**Generation procedure Gen**
Input: $w \in \mathcal{M}_1$,

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

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: $w' \in \mathcal{M}_1$, $P \in \{0,1\}^*$

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

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

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

Figure 5: Generation and Update procedures

5.4 Reproduction procedure

The reproduction procedure presented in Figure 6 is straightforward. Taking as inputs w', 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 $\mathsf{Rep'}$ recovers K_t which enables to finally unlock R.

```
**Reproduction Procedure Gen**
Input: $w' \in \mathcal{M}_1$, $P \in \{0,1\}^*$.

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

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

3. Return $R$.
```

Figure 6: Reproduction procedure

Theorem 2. *Figures 5 and 6 define a $(\mathcal{M}_1, m_1, u, t, \phi)$-AFE that is $(\epsilon, \rho, s_{sec})$-reusable.*

Remark 1. *Enabling $2.\rho.\phi$ reusability for a FDF means that it 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.*

Proof. The correctness is straightforward given those of (DerGen, DerRep), (Gen',Rep') and (Enc, Dec). There are to security notions to prove: pseudorandomness of the extracted key R even in presence of related helper strings P^js and reusability.

We begin with the pseudorandomness of the extracted key (point 4 of Definition 6). Let $W^1, \ldots,$ W^ϕ random variables over $\mathcal{M}$ with min-entropy m_1 either correlated or correlated by subsets. Now let $w^1, \ldots, w^\phi$ respectively sampled from $W^1, \ldots, W^\phi$ and defining a u-drift.

- $\mathcal{G}_0$ In this game, the challenger $\mathcal{C}$ first computes $(\Omega^i, F^i) \leftarrow \mathsf{Gen'}(w^i)$ for all $1 \leqslant i \leqslant \phi$. $\mathcal{C}$ then enrolls Ω^1: he computes $(P'^1_t, K^1_t) \leftarrow \mathsf{Gen'}(\Omega^1)$, randomly samples $R \leftarrow \{0,1\}^l$ and sets $c^1_t = \mathsf{Enc}(K^1_t, R)$. He also computes (P'^1_u, K^1_u, c^1_u) as defined in Figure 5. Now, for all $2 \leqslant i \leqslant, \mathcal{C}$ computes $P^i \leftarrow \mathsf{Upd}(w^i, P^{i-1})$ where $P^i = (F^i_t, P'^i_t, c^i_t), (F^i_u, P'^i_u, c^i_u)$. He finally sends the P^is and R to D.

- $\mathcal{G}_1$ In this game, the challenger $\mathcal{C}$ makes one modification compared to $\mathcal{G}_0$. Indeed, $\mathcal{C}$ makes the exact same computations but does not send R to D in the end. More precisely, he sends the actual P^is computed as in previous game but he replaces the key R by a randomly sampled value $\eta \xleftarrow{\$} \{0,1\}^l$. Let us recap: in previous game, D was given c^i_ts constituting ϕ encryptions of R under ϕ different secret keys K^i_t and R. In the end of $\mathcal{G}_1$, D is given the same c^i_ts and η. Now, let us assume that D can distinguish these two games. In particular, it means that it exists i_0 such as given $c^{i_0}_t = \mathsf{Enc}(K^{i_0}_t, R)$ and η, D can tell that $c^{i_0}_t$ is not the encryption of η which leads to breaking FTG-CPA of the underlying encryption scheme.

Hence $\mathcal{G}_0$ and $\mathcal{G}_1$ are indistinguishable to D's view. This terminates the proof for pseudorandomness.

Reusability The proof for reusability consists in adapting the methodology used for Theorem 1 to our adaptive context. Modifications mainly consist in technical details while the reasoning itself relies in combining proofs of Theorem 1 and of the previous security of R. We refer the reader to Appendix 7 for a detailed proof. ∎

Instantiation for the set difference metric

Let (DerRep, DerGen) be the $(\mathcal{M}_1, \mathcal{M}_2, m_1, m_2)$ FDF defined through Figure 3 that is $(\epsilon, 2.\phi.\rho, s_{\text{sec}})$-reusable. Let (Gen', Rep') be an average-case nonreusable $(\mathcal{M}_2, m_2, l, u(\text{resp. } t))$-FE $(\epsilon, s_{\text{sec}})$-hard and based on set difference metric. Such FEs can be found in [14].

Corollary 2. *Figure 5 and 6 define a $(\mathcal{M}_1, m_1, l, u, t, \phi)$-adaptive fuzzy extractor that is $(\epsilon, \rho, s_{sec})$-reusable.*

Proof. This is straightforward assuming Theorem 2. ∎

6 Conclusion and Future Works

In this work, we solve the reusability issue of Fuzzy Extractors for the set difference metric. In particular, our work is the first to handle the frequent case where set elements are sampled from a big universe (*i.e.* when the universe size is superpolynomial in the set size). While Canetti *et al.* recently designed the first reusable FE based on Hamming metric (Eurocrypt 2016), their main construction can be extended to the set difference metric but only in the small universe setting of lesser interest. In fact, our set difference-based solution is an instantiation -relying on a hash function modeled as a random oracle- of a more 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 *Fingerprint Derivation Functions* (FDFs) for such a purpose. We then designed Reusable Fuzzy Extractors out of any efficient nonreusable Fuzzy Extractors and adapted FDFs. Consequently, our work benefits satisfying asymptotic complexities both in small and large universe settings. As a comparison, the work of Canetti *et al.* extended to set difference metric in the small universe setting can only correct a number of errors sublinear in n while their time complexity is equivalent to $\omega(n^c.log(n))$, where n is the universe size (or equivalently the length of binary vectors in Hamming case) . Now instantiating nonreusable constructions of Dodis *et al.* with LDPC codes lead to time and correction complexities in $O(n)$. Furthermore, in the large universe setting where t can be considered as negligible with respect to n, our work enjoys time complexity in $O(\mathsf{poly}(slogn))$.

In addition to tackling reusability issue, we also propose the notion of Adaptive Fuzzy Extractors, particularly well-fitted in the burgeoning context of device and browser fingerprinting. 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 strengthening, if needed, our previous result. Such an application raises the problematic of identifying values with shorter lifetime period. We then defined *Adaptive* Fuzzy Extractors meant to capture these variations while still enabling generation of a long-term stable key. While defining Adaptive Fuzzy Extractors increase the scope of Fuzzy Extractors, this could also lead to sharper studies about such device and browser fingerprints. Indeed, our t-drift definition is a first stone that should not be seen as an ultimate goal. Better understanding of such fingerprints should lead to more accurate and fair models. Hence, one might expect a better knowledge of such fingerprints to predict when update procedures will have to be called.

References

[1] Device fingerprinting and fraud protection. Device-Fingerprinting-and-Online-Fraud-Protection-Whitepaper.pdf.

[2] Host-based card emulation. https://developer.android.com/guide/topics/connectivity/nfc/hce.html.

[3] 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.

[4] J. P. Achara, G. Acs, and C. Castelluccia. On the unicity of smartphone applications. In *Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society*, WPES '15, pages 27–36. ACM, 2015.

[5] 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.

[6] C. H. Bennett, G. Brassard, and J.-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[7] 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.

[8] 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.

[9] 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.

[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] A. K. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP J. Adv. Signal Process*, 2008:113:1–113:17, Jan. 2008.

[19] A. K. Jain, K. Nandakumar, and A. Ross. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, 2016.

[20] A. Juels and M. Sudan. A fuzzy vault scheme. *Des. Codes Cryptography*, 38(2):237–257, Feb. 2006.

[21] 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.

[22] 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.

[23] 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.

[24] K. Mowery and H. Shacham. Pixel perfect: Fingerprinting canvas in HTML5. In M. Fredrikson, editor, *Proceedings of W2SP 2012*. IEEE Computer Society, May 2012.

[25] 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.

[26] 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.

[27] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.

[28] S. Prabhakar, S. Pankanti, and A. K. Jain. Biometric recognition: Security and privacy concerns. *IEEE Security & Privacy*, 1(2):33–42, 2003.

[29] T. J. Richardson and R. L. Urbanke. Efficient encoding of low-density parity-check codes. *IEEE Trans. Information Theory*, 47(2):638–656, 2001.

[30] 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.

[31] 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.

[32] S. Seneviratne, A. Seneviratne, P. Mohapatra, and A. Mahanti. Predicting user traits from a snapshot of apps installed on a smartphone. *SIGMOBILE Mob. Comput. Commun. Rev.*, 18(2):1–8, June 2014.

[33] K. Simoens, P. Tuyls, and B. Preneel. Privacy weaknesses in biometric sketches. In *2009 30th IEEE Symposium on Security and Privacy*, pages 188–203.

[34] 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.

[35] U. Uludag, A. Ross, and A. Jain. Biometric template selection and update: a case study in fingerprints. *Pattern Recognition*, 37(7):1533–1542, 2004.

[36] 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.

7 Proof of Reusability of Theorem 5.1

In the following we prove that Figures 5 and 6 define an AFE that is $(\epsilon, \rho, s_{\text{sec}})$-reusable.

Proof. Let $W^1, \dots, W^\rho \in \mathcal{W}$, potentially correlated by subsets and of min-entropy m. Through a sequence of games, we will prove reusability of $(\mathsf{Gen}, \mathsf{Upd}, \mathsf{Rep})$ according to Definition 7.

$\mathcal{G}_0$ From Definition 7, the challenger $\mathcal{C}$ acts as follows. For $1 \leqslant j \leqslant \rho$, we have :

- $\mathcal{C}$ samples $w^{j,1} \leftarrow W^j$.
- $\mathcal{C}$ computes helper strings via "generation" and "update" procedures.
  - $(P^{j,1}, R^j) \leftarrow \mathsf{Gen}(w^{j,1})$
  - for $2 \leqslant i \leqslant \phi$: $\mathcal{C}$ chooses $w^{j,i}$ such as $d(w^{j,i}, w^{j,i-1}) \leqslant u$ and computes $P^{j,i} \leftarrow \mathsf{Upd}(w^{j,i})$.

In the end, D is given all the helper strings $P^{j,i}$s along with the secret keys R^js.

$\mathcal{G}_1$ In this game, $\mathcal{C}$ will challenge D on some index j_0. A close look to Figure 5 ensures that $P^{j,1}$ and $P^{j,i}$ $(i \geqslant 2)$ are generated the same way. To begin, the value $w^{j,i}$ $(i \geqslant 1)$ is randomized twice via a FDF to get some $\Omega_u^{j,i}$ and $\Omega_t^{j,i}$. FE (Gen',Rep') is then applied to get respectively $(K_u'^{j,i}, Q_u'^{j,i})$ and $(K_t'^{j,i}, Q_t'^{j,i})$. The role of $K_t'^{j,i}$ is to encrypt the secret outputted or unlocked key R^j. $\mathcal{C}$ chooses to cheat when generating an helper string related to R^{j_0}. He then chooses a random index i_0 to attack R^{j_0}. Now, he samples $w^* \overset{\$}{\leftarrow} \mathcal{M}_1$ to obtain $\Omega_t^* \leftarrow \mathsf{DerGen}(w^*)$ and finally computes $(P^*, K^*) \leftarrow \text{Gen'}(\Omega^*)$. The actual value P^{j_0,i_0} is then replaced with $P^* = (F_u^{j_0,i_0}, Q_u'^{j_0,i_0}, c_u^{j_0,i_0}), (F_t^{j_0,i_0}, P^*, c^* = \mathsf{Enc}(K^*, R^{j_0}))$. $\mathcal{C}$ finally transmits P^* instead of P^{j_0,i_0} along with the R^js and the other $P^{j,i}$s. Assuming FDF $2.\rho.\phi$-reusability, $\mathcal{G}_1$ is indistinguishable from $\mathcal{G}_0$ to D's view.

$\mathcal{G}_2$ In this game, $\mathcal{C}$ replaces K^* with a random value $\eta \overset{\$}{\leftarrow} \{0,1\}^l$. Under security of the classic and nonreusable FE (Gen', Rep'), we have that $(K^*, P^*) \approx (U_l, P^*)$. This game then appears indistinguishable from the previous one. Note that this replacement leads to getting $\mathsf{Enc}(\eta, R^{j_0})$ instead of $\mathsf{Enc}(K^*, R^{j_0})$ as part of the corrupted helper string P^*.

$\mathcal{G}_3$ In this game, $\mathcal{D}$ has received actual secret keys R^js and helper strings $P^{j,i}$s except for (i_0, j_0) for which the right part of P^* is $(F_t^{j_0,i_0}, P^*, c^* = \mathsf{Enc}(\eta, R^{j_0}))$. Since η and R^{j_0} were randomly and generated independently from any values, $\mathcal{C}$ could then then replaces R^{j_0} by some $\eta \overset{\$}{\leftarrow} \{0,1\}^l$ when sending the secret keys. If D can distinguish this game from the previous one, he is in particular able to attack the FTG-CPA security notion since he would be able to distinguish $(\mathsf{Enc}(\eta, R^{j_0}), R^{j_0})$ from $(\mathsf{Enc}(\eta, R^{j_0}), \eta)$. Assuming FTG-CPA security, D cannot distinguish and these two games appear indisinguishable.

$\mathcal{G}_3$ $\mathcal{C}$ can now replaces P^* and η by values Q'^{j_0,i_0} and K^{j_0,i_0}. Assuming again $2.\rho.\phi$ reusability of the underlying FDF, this game is indistinguishable of the previous one. In the end of this game, D has been given all the actual helper strings $P^{j,i}$s and the secret R^js except for index æ$_0$ which is replaced with some random η

The sequence of indistinguishable games permits to conclude that $\mathcal{G}_0$ and $\mathcal{G}_3$ are indistinguishable. This terminates the proof for reusability.

■