Pseudoentropic Isometries: A New Framework for Fuzzy Extractor Reusability

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

[1] equensWorldline, FPL-ITA-MSC, Seclin, France,
firstname.lastname@equensworldline.com,
[2] University of Connecticut, Storrs, CT, United States,
benjamin.fuller@uconn.edu,
[3] Université de Limoges, XLIM-DMI, Limoges, France,
philippe.gaborit@xlim.fr,

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

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

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

We show that building a reusable fuzzy extractor from a reusable pseudoentropic isometry is straightforward by 1) randomizing the noisy secret ω into Ω and 2) applying a traditional fuzzy extractor to derive a secret key from Ω.

To show the promise of our framework, we propose instantiations that handle the set difference and Hamming metrics. The first one is an original construction based on composable digital lockers (Canetti and Dakdouk, Eurocrypt 2008) yielding the first reusable fuzzy extractor that corrects a *linear* fraction of errors. For the second one, we show that *Construction 2* proposed by Canetti *et al.* in Eurocrypt 2016 (Section 5.1) can be seen as an instantiation of our framework. In both cases, the pseudoentropic isometry's reusability requires noisy secrets distributions to have entropy in each symbol of the alphabet.

At last, we describe two practical solutions that reap benefits of our results while dealing with the aforementioned limitation.

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 [20,33], Physically Unclonable Functions (PUFs) [46] and quantum information [8]. Both PUFs and biometrics demonstrate errors that prevent stable cryptographic key generation.

Randomness sources A PUF is a physical entity that is easy to evaluate but hard to predict. Unique by manufacturing process, PUFs are used to implement challenge-response authentication. Recently, researchers have attacked PUFs, creating software models for the PUF behavior [48,49]. These attacks can be avoided by first deriving stable cryptographic key from the PUF output and then creating a challenge response protocol using a function of this key.

Biometrics are unique characteristics of individuals based on either biological or behavioral characteristics. Biometrics are evaluated by their uniqueness, collectability and permanence [32]. Unlike passwords, biometrics suffer inevitable but minor variations. These variations are assumed to correspond to a bounded distance between repeated readings according to some metric. Dodis *et al.* [22] stated that Hamming distance looks like the "most natural metric to consider" [16,22,34]. However, with the exception of iris [14], set distance better suits concrete cases such as biometric matchers (e.g. digital fingerprints) or even the more exotic movie lover's problem [33]. Typical systems create a *template* reading from an initial reading; subsequent readings are directly compared to this initial template. These templates have privacy concerns [47,51], in the worst case a matching biometric can be reverse engineered from the template [27].

1.1 Fuzzy Extractors

Information reconciliation [8] enables retrieving identical values from noisy data. *Privacy amplification* [8] converts values with entropy into uniform random strings. Fuzzy Extractors (FEs) [22][4], are a pair of non-interactive algorithms $(\mathsf{Gen}, \mathsf{Rep})$ 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 P and ω' and reproduces the secret key R as long as ω' is close enough to ω relatively to the distance metric, say $d(\omega, \omega') \leq t$. FEs exist with security against information-theoretic [21] or computational adversaries [26].

Metrics Dodis *et al.* proposed FE constructions for the Hamming, set difference and edit metrics drawing on prior work [34,33]. We focus on the set difference metric: inputs ω are subsets of size s of a universe $\mathcal{U}$ whose cardinality is n. For this metric, Dodis *et al.* distinguished two settings, referred to as the *small* and *large* universe settings. Let s be some security parameter. In the former case, we have that $n = \mathtt{poly}(s)$ 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 [33]). The small universe setting benefits from a reduction to the Hamming metric, referred to as the $\mathsf{bin\text{-}set}$ equivalence (described in Section 2). We concentrate on the large universe setting where this transform is not applicable.

Reusability Boyen stated the need for *reusable* fuzzy extractors [13] for which numerous helper strings P^j from a user's fuzzy secret do not impact user's security. Boyen showed that information-theoretic FEs must leak substantial information about ω when numerous calls to Gen are made. On the positive side, Boyen demonstrated reusable security when the exclusive OR of the user's fuzzy secrets reveal no sensitive information. This is a restrictive class of correlations; we have no evidence that practical sources obey this condition. Subsequent works showed that existing FEs are not reusable in practice [10,50].

[4] In the following, we will refer to the journal version [21].

Recently, Canetti *et al.* [16] constructed a reusable fuzzy extractor (RFE) that makes no assumption about how repeating readings are correlated. It works for the Hamming distance and provides security against computationally bounded adversaries. It uses a strong form of symmetric encryption, called *digital lockers* [15] (our construction also uses digital lockers but in a different way).

Their construction is secure for distributions with high entropy samples instead of global min-entropy. This is in contrast to traditional constructions that only assume the source has min-entropy. Their main binary construction can be extended through $\mathsf{bin\text{-}set}$ equivalence to an FE in the set difference metric but only in the small universe setting. Their scheme only allows an error rate (t/n) sublinear in n. Plus, Apon *et al.* [6] turned the *learning-with-errors* based FE of Fuller *et al.* [26] into an RFE. Hence, their work inherits the relative inefficiency of the latter construction [26]; in particular, it only corrects a sublinear number of errors.

Prior to this work, there were no known RFE correcting a linear error rate for any common metric for arbitrary correlation between enrollments. Most biometrics demonstrate an error rate between $10 - 30\%$.

1.2 Strong Mobile Authentication

Companies are moving to *multi-factor authentication* (also called strong authentication), with factors falling in at least two of the following categories: *knowledge* ("what you know"), *possession* ("what you own") and *inherence* ("what you are") [3]. Mobile applications usually use possession and knowledge factors to achieve strong authentication:

- *Possession* The ownership of the mobile device is often proved via a *One Time Password* (OTP) sent by SMS. Once having received the OTP, the user enters it on the authentication web page to prove that she indeed has the device in her possession.
- *Knowledge* A password chosen by the user.

Both of these means are subject to strong limitations. The SMS OTP is vulnerable to numerous attacks (see [42] and references therein) and the SMS channel has been deprecated by authorities such as NIST, which recommend to move to more secure means of authentication [45]. Human memorable passwords do not achieve sufficient entropy, indeed, recent estimates place the password entropy at 34 bits [35].

Alternative methods of authentication have emerged such as biometrics and PUFs, which could respectively fulfill inherence and possession proofs. While these solutions have received attention in the authentication literature, they rely on dedicated hardware sensors and components and, as such, cannot be implemented using only software. This is problematic in the case of mobile authentication as the availability of hardware components (e.g. biometric sensors) varies greatly between devices. For example, a fingerprint reader can be available on some devices while on others its usage is restricted to the operating system or even totally absent. All the above makes the design of a generic authentication solution difficult.

1.3 New Trends for Mobile Authentication

With the goal of generality on mobile devices, the only solution seems to be harvesting software collectable data. This line of research was initiated by Eckersley [23] who showed how to create a fingerprint from characteristics of a web browser (user agent, list of fonts, list of plug-ins,…).

Servers use this data to detect returning browsers even when features have changed over time. Subsequent studies [11,41,43,4] show such a system is deployable for personal computers. This research naturally led to studying the practicability of using similar fingerprints on mobile devices.

While early mobile solutions were insufficient [38,18,19,54], recent work of Kurtz *et al.* provides a comprehensive analysis in the mobile setting [36]. On Apple's iOS, they show how to compute a device fingerprint using 29 different configuration features. Using a dataset of 13,000 fingerprints, they show that fingerprints are unique and allow detection of returning devices with an accuracy of 97%. In their work, the list of installed applications and the top 50 songs are among the most identifying values present on a device. These fingerprints are candidates to be an inherent authentication factor. Many of these device fingerprints draw on features coming from large universes with variation according to the set difference metric (*e.g.* songs, applications, ...).

The usage of device fingerprints and behavioral fingerprints, respectively as possession (of the mobile device) and inherence factors enable strong authentication of a user. Device fingerprints can be constructed from the following values:

- *IMEI* (International Mobile Equipment Identity) [53] This value is 15 digits long, the first ones identifying the manufacturer while the 6 last digits are randomly chosen to produce a serial number that identifies the device. The IMEI then carries more than $6 \times \log(10) \approx 20$ bits of entropy.
- *IMSI* (International Mobile Subscriber Identity) [24] The IMSI's first digits are specific to the country and the mobile network, while the remaining 8 digits are randomly chosen. Based on the latter, we can assume that the IMSI carries more than $8 \times \log(10) \approx 26$ bits of entropy.
- *AndroidID* [28] The AndroidID is a 64 bits random number which used to be generated at the device's first boot and remain constant throughout its lifetime, enabling its identification. As of Android 8, it is specific to an application and randomly generated at its installation.

This list is far from being exhaustive, other values could also be used such as the device's Wifi and Bluetooth MAC addresses, the battery's serial number, etc. While these values are static, looking ahead, in our solution we seek to tie the possession and inherence factors, necessitating noise tolerance even if we use some static values. Behavioral sources include the following:

- *Apps* The user's applications list [5,36].
- *Music* The user's most listened to songs [5,36].
- *Contacts* The user's favorite contacts. The most common first name in the United States is 'James' [2] with a frequency of 3.318% which yields an approximate min-entropy of $-\log(3.318 \times 10^{-2}) \approx 5$ bits. The first name of the top 20 contacts list has a min-entropy of around 100 bits.
- *TopNetworks* The Wi-Fi networks with the strongest signal at given time and given location. More precisely, the n strongest networks -whose main identifiers are the service set identifiers (SSIDs) and the basic service set identifiers (BSSIDs) define a kind of trusted place (*e.g.* the user's home or workplace). Alternatively we also consider *RegSSID*, the SSIDs of Wifi networks already registered on the device.

Discussion These authentication factors do not require user interaction, contrary to passwords and SMS OTP. This improves user experience which is critical in some use cases (*e.g.* mobile payment applications).

However, some of these values may require specific permissions to be available. Other applications may ask for the same permissions, hence, malevolent ones may gain access to some of the above mentioned values. Yet, such applications are steadily and more and more efficiently hunted and removed from the Play Store [29]. Users are more and more conscious of permissions and privacy issues and grant permissions sparingly [37].

1.4 Our contributions

Secure sketches [21], which will be introduced later, are often used to build FEs. Both FEs and secure sketches are meant to recover a secret value from noisy inputs, the recovered value being either the noisy secret itself or an extracted cryptographic key. In both cases, noise elimination is performed using some helper data which leaks information.

FEs are mainly based on information-theoretic secure sketches and as such, prone to reusability issues [50,10]. Fuller *et al.* showed that computationally secure sketches are subject to many of the same limitations [26, Theorem 3.6]. To circumvent these negative results, we propose a new framework to achieve FE reusability. The main idea is to separate the task of reusability from the task of noise elimination. Our contributions are as follows:

1. We define a formal framework to turn any traditional fuzzy extractor into a reusable one:
   (a) We introduce a randomization stage captured by a new primitive we call a *pseudoentropic isometry* (PI). Informally, a PI pseudorandomly projects fuzzy secrets while maintaining distances between two noisy readings and entropy of the original secret. To be *reusable*, a PI must generate ρ uncorrelated values $\Omega^1, \ldots, \Omega^\rho$ from ρ enrollments values $\omega^1, \ldots, \omega^\rho$ drawn from the fuzzy secret ω. The reusability property is then defined as long as each Ω^{j_0} carries sufficient entropy even in presence of other Ω^js ($j_0 \neq j$). *Reusable pseudoentropic isometries* (RPIs), contrary to both FEs and secure sketches, do not perform any form of error correction and are not subject to many bounds from coding theory.
   (b) We show that combining an RPI and a traditional FE yields an RFE.
2. We propose two instantiations of our framework. The first one is an original construction with an RPI based on digital lockers [15]. This RPI allows us to design the first reusable FE for linear error rates. This construction applies for the set difference metric in the large universe setting and proceeds as follows for each element of the input set:
   (a) We sample a random point in a new metric space,
   (b) We lock the random point using the element of the input set as the key,
   When Rep is run, the fraction of unlockable points is the same as the overlap between the sets. This construction does no error-correction, it projects "randomly" while preserving distance, the actual fuzzy extractor is applied afterward. We also show that Construction 2 of Canetti *et al.* [16, Section 5.1] based on Hamming distance does fit our framework. Both instantiations require sources with superlogarithmic entropy (in each alphabet symbol) to achieve reusability.
3. Based on the widespread availability of entropy on mobile devices (Subsection 1.3), we describe two applications to securely instantiate our set-difference RFE.

2 Preliminaries

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

For any entity $\mathcal{E}$, we denote by $\mathcal{E}(z)$ the fact that $\mathcal{E}$ has knowledge of z. U_ℓ denotes the uniformly distributed random variable on $\{0,1\}^l$. For a distinguisher D (or a class of distinguishers $\mathcal{D}$), we write the computational distance between X and Y as $\delta^D(X,Y) = |\mathbb{E}[D(X)] - \mathbb{E}[D(Y)]|$. $\mathcal{D}_{s_{\text{sec}}}$ denotes the class of randomized circuits which output a single bit and have size at most s_{sec}. Let λ denote a security parameter. We denote the following security parameters $l = l(\lambda)$, $\kappa = \kappa(\lambda)$, $m = m(\lambda)$, $m_1 = m_1(\lambda)$,$m_2 = m_2(\lambda)$, $s_{\text{sec}} = \texttt{poly}(\lambda)$ and $\epsilon_{\text{sec}} = \mathsf{negl}(\lambda)$.

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

2.1 Background

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

Dodis *et al.* [22] noted the $\mathsf{bin\text{-}set}$ equivalence: if ω denotes a set, it can be viewed a binary vector in $\{0,1\}^n$, with 1 at position x if $x \in \omega$, and 0 otherwise. Viewed in this way, set difference can be expressed as Hamming distance between these associated vectors. This transform is not efficient when the universe size n is superpolynomial.

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

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

HILL entropy is a commonly used computational notion of entropy [30]. It was extended to the conditional case by Hsiao, Lu, and Reyzin [31].

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.* [21], was information-theory based. We focus on the computational definition introduced by Fuller *et al.* [26]. They extend their definition to an explicit family of distributions and we adopt this convention.

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

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

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

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

Reusable Fuzzy Extractor RFEs [13] allow multiple calls to Gen on the noisy readings of ω while retaining security. Consider ρ readings $\omega^1,\ldots,\omega^\rho$ of the same fuzzy secret from which the user will be enrolled on ρ different authentication servers. Gen independently generates ρ pairs $(R^1,P^1),\ldots,(R^\rho,P^\rho)$ where $(R^j,P^j) \leftarrow \mathsf{Gen}(\omega^j)$. Canetti *et al.* [16] proposed a security model where a given R^{j_0} is secure even if all P^j and all other R^js (for $j \neq j_0$) are given to an adversary.

Definition 3 (Reusable Fuzzy Extractor [16]). *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* $1 \le j \le \rho$. *Let* D *be an adversary. Define the following game for all* $j = 1,\ldots,\rho$*:*

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

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

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

2.2 Tools

Digital Lockers Digital lockers are secure symmetric encryption schemes that retain security even when used multiple times with correlated and nonuniform keys [17]. An incorrect key can also be recognized with high probability. We use notation $\mathsf{c} = \mathsf{lock}(\mathsf{key},\mathsf{val})$ for the algorithm that performs the locking of the value val using key, and $\mathsf{unlock}(\mathsf{key},\mathsf{c})$ for the algorithm that performs the unlocking (which will output val if key is correct and $\perp$ with high probability otherwise).

Digital lockers can be easily constructed in the random oracle (see Lynn, Prabhakaran, and Sahai [39, Section 4]). Bitansky and Canetti [9], building on the work of [15,17], show how to obtain composable digital lockers based on a strong version of the Decisional Diffie-Hellman assumption without random oracles.

The security of digital lockers is defined via virtual-grey-box simulatability [9], a simulator is allowed unbounded running time but only a bounded number of queries to an ideal locker. Intuitively, the definition says if the keys to the ideal locker are hard to guess, the simulator will not be able to unlock the ideal locker and thus neither will the real adversary. Formally, let $\mathsf{idealUnlock}(\mathsf{key},\mathsf{val})$ be the oracle that returns val when given key, and $\perp$ otherwise.

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

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

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

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

Point Functions Canetti *et al.*'s construction for large alphabets [16, Section 5.1] uses a weaker primitive called an obfuscated point function. This primitive can be viewed as a digital locker without a plaintext: it outputs 1 if the key is correct, 0 otherwise. Such a function can be constructed from the above mentioned digital locker with a single possible plaintext, or from a strong version of the Decisional Diffie-Hellman assumption [14]. We use notation $c = \mathsf{lockPoint}(key)$ and $\mathsf{unlockPoint}(key, c)$. Point functions security is defined the same way as for digital lockers with a fixed plaintext.

3 Reusability Framework

In this section, we present a new and generic way to address reusability. Reusable fuzzy extractors combine entropy extraction and error-correction with reusability property, we propose addressing them separately. The key idea is to randomize the fuzzy secrets before applying a nonreusable FE (that will do key recovery and error correction) on the unrelated projected values, hence handling the reusability property beforehand. We note that splitting entropy extraction and error-correction already exists for fuzzy extractors, with a secure sketch performing error correction and a randomness extractor [44] performing entropy extraction.

This randomization stage is performed by a pseudoentropic isometry (Def 5), a new primitive projecting fuzzy secrets while maintaining distances and entropy of the original noisy secret. To be reusable, a PI must be able to generate ρ uncorrelated values $\Omega^1, \ldots, \Omega^\rho$ from ρ enrollments values $\omega^1, \ldots, \omega^\rho$ drawn from the fuzzy secret ω. The reusability property (Def 6) says that each Ω^{j_0} carries sufficient entropy with respect to an adversary who knows Ω^js ($j_0 \neq j$). Following this randomization stage, we can apply a traditional FE on the uncorrelated randomized values Ω^js. This idea is depicted in Figure 1.

Within this framework, the FE is always applied on unrelated values Ω^js. Even when re-enrolling a fingerprint ω, the generated helper values P^js only yield information on the decorrelated values Ω^js and not on the original noisy secret. Thus, the combination of an RPI with a traditional, nonreusable, FE yields an RFE.

3.1 Definitions

Pseudoentropic Isometries A PI is a pair (RPIGen, RPIRep) defined as follows.

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

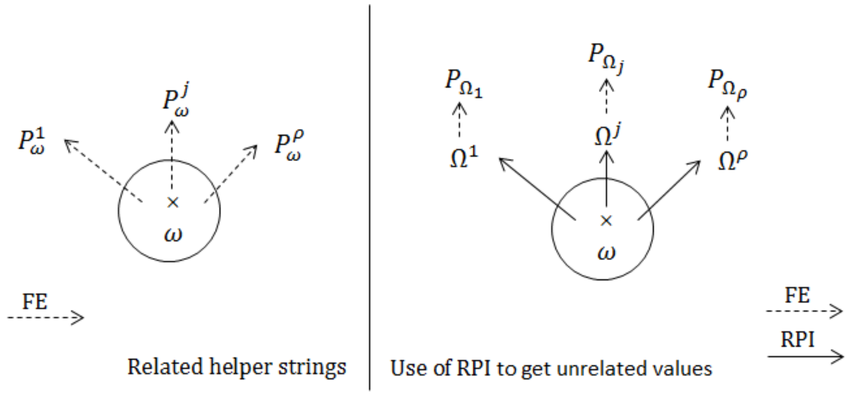

Fig. 1. Overview of reusability framework

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

Security implies that $H^{\mathtt{HILL}}(W|V) \geqslant m_2$ with a slight loss in parameters as any adversary that can recover W can use RPIRep to recover U.

Relation to other fuzzy extractor primitives This notion is related to biometric embeddings used in [21]. A biometric embedding projects any fingerprint value into a metric space where an FE exists while loosely maintaining distances. Furthermore, every (computational) secure sketch and fuzzy extractor is a pseudoentropic isometry. This verifies the intuition that this object is weaker than either a secure sketch or a fuzzy extractor.

On their own PIs are not novel (the identity function is a PI). A reusable pseudoentropic isometry or RPI is the key to our approach. In an RPI the knowledge of previous derived values does not help D to distinguish a random value from a newly derived projection obtained via RPIGen. Drawing on the definition of reusability for FEs (Definition 3), we define an RPI as follows.

Definition 6 (RPI). *Let $W^* \in \mathcal{W}$ be a distribution. Let $W^1, W^2, \ldots, W^\rho$ be ρ correlated random variables over $\mathcal{M}_1$. Let D be an adversary. Using notation of Definition 5, we define the following game for all $j = 1, \ldots, \rho$:*

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

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

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

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

Fig. 2. A generic reusable FE

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

Relation to other fuzzy extractor primitives Any (computational) reusable fuzzy extractor is a reusable pseudoentropic isometry, confirming the intuition that this primitive is weaker. Reusable secure sketches are defined with only the values F being available to the adversary (and not any of Ω^i). Thus, a reusable secure sketch does not imply a reusable pseudoentropic isometry. Furthermore, the reusable secure sketch of Boyen [13] is not an RPI. Thus, these two notions seem incomparable.

3.2 RPIs imply reusable fuzzy extractors

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 is the error tolerance of the FE. Whenever a noisy reading ω' is given to procedure Rep', the secret key will be recovered as long as ω' belongs to $\mathcal{B}(\omega, t)$.

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

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

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

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

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

Lemma 1. *Let U, V be a joint distribution where $H^{\mathtt{HILL}}_{\epsilon_{RPI}, s_{RPI}}(U|V) \geqslant m_2$ and let (*$\mathsf{Gen'}$*,* $\mathsf{Rep'}$*) 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 $\epsilon = 2\epsilon_{\text{RPI}} + \epsilon_{\text{FE}}$, $s = \min\{s_{\text{RPI}}, s_{\text{FE}}\}$.

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

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

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

to D.

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

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

If D can distinguish this game from the previous one, he would then be able to distinguish the distribution with Ω^{i_0} from the one with Ω^*. This breaks the reusability of the RPI. That is, $\mathcal{G}_1$ appears indistinguishable from $\mathcal{G}_0$ for $\epsilon = \epsilon_{\text{RPI}}$ 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 $\eta \overset{\$}{\leftarrow} \{0,1\}^l$ 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 $\mathcal{G}_0$. Indistinguishability between $\mathcal{G}_0$ and $\mathcal{G}_3$ satisfies the requirements of Definition 3.

4 Instantiating the framework

We propose two instantiations of the above mentioned framework, the first one for the set difference metric and the second one for the Hamming distance.

4.1 Set Difference-based Instantiation

Environment and Notation Set difference based FEs in [21] take as inputs subsets of a universe $\mathcal{U}$ with $n = |\mathcal{U}|$. We denote $(\mathcal{M}_{\mathcal{U}}, d)$, the metric space $\mathcal{M}_{\mathcal{U}}$ consisting of all the subsets of $\mathcal{U}$ with the set difference metric d. Let $\mathcal{M}_{\mathcal{U},s}$ denote the restriction of $\mathcal{M}_{\mathcal{U}}$ to s-elements subsets. $\mathcal{M}_{\kappa}$ denotes $(GF(2^{\kappa}), d)$ equipped with the set difference metric d. Similarly $\mathcal{M}_{\kappa,s}$ denotes the restriction to sets of sizes s. Let W be a probability distribution over $\mathcal{U}$ with min-entropy m. We use digital lockers to construct our set difference-based RPI. Our construction, presented in Figure 3, randomizes each set element using a digital locker.

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

Algorithm RPIRep
Inputs: $\omega' = \{\omega'_1, \ldots, \omega'_s\}$,
        $c = c_1, ..., c_s$.
1. $n = s$,
2. For $i = 1 \ldots s$,
     For $j = 1 \ldots n$,
       a. $x'_i \leftarrow \mathsf{unlock}(\omega'_i, c_j)$.
       b. if $x'_i = \perp \wedge j = n$: $x'_i \xleftarrow{\$} \mathcal{M}_{\kappa}$.
       c. else if $x'_i = \perp \wedge j \neq n$: continue.
       d. else: remove $c_j$; $n$- -; break.
3. Set $\Omega' = \{x'_1, \ldots, x'_{s'}\}$.
4. Return $\Omega'$.
```

Fig. 3. A set difference-based RPI

It is possible to have a collision in step 2.a. of Algorithm RPIGen, however this occurs with negligible probability (κ must be super-logarithmic for security). Step 2.b. of Algorithm RPIRep adds additional elements to ensure that the output set is of size s. This step can be triggered if there was a collision in RPIGen or if unlock outputs $\perp$ in Step 2.a. The only time this makes $d(\Omega, \Omega') \geq d(\omega, \omega')$ is when unlock outputs $\perp$ when the two values actually match. Considering the worst case scenario, where ω and ω' are disjoint sets, we end up with s^2 calls to the unlock function. This construction is secure when each element of the input set has superlogarithmic min-entropy.

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

Our proof is similar in spirit to the proof of Canetti *et al.* [16]. We first prove a simpler proposition that the construction is a PI and then consider reusability.

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

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

Isometry property. $\mathit{\Omega}$ is of size s. For any $\omega_i = \omega_i'$, if no digital locker outputs $\perp$ then $x_i = x_i'$ and the total number of calls to function unlock is . Thus,

$$\Pr[d(\mathit{\Omega}, \mathit{\Omega}') \leq d(\omega, \omega')] \geq 1 - s^2\delta.$$

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

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

by a negligible function.

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

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

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

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

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

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

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

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

Proof. Fix any $u \in \mathcal{M}_\kappa$ (the lemma will follow by averaging over all u). Let $\mathit{\Omega}^*$ be the correct value of $\mathit{\Omega}$. The only information that S can learn about whether the value is $\mathit{\Omega}^*$ or u is through the query responses. First, modify S slightly to quit immediately if it gets a response not equal to $\perp$ (we assume such as soon as S gets back a non-$\perp$ response it distinguishes with probability 1). There are $q + 1$ possible values for the view of S on a given input (q of those views consist of some number of $\perp$ responses followed by the first non-$\perp$ response, and one view has all q responses equal to $\perp$). By [21, Lemma 2.2b], $\tilde{\mathrm{H}}_\infty(V_i | View(S), \{j_{ik}\}) \geq \tilde{\mathrm{H}}_\infty(V_j | \{j_{ik}\}) - \log(q + 1) \geq m - \log(q + 1)$. Therefore, at each query, the probability that S gets a non-$\perp$ answer is at most $(q + 1)2^{-m}$. Since there are q queries of S, the overall probability is at most $q(q + 1)/2^m$. Then since 2^m is $\mathtt{ngl}(\lambda)$, there exists some λ such that for all $\lambda > \lambda_0$, $q(q + 1)/2^m \leq 1/(3p(\lambda))$. This completes the proof of Lemma 2.

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

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

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

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

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

Discussion Our instantiation of an RPI for the set difference metric (large universe) allows construction of the first RFE correcting a linear error rate that makes no assumption about how individual readings are correlated. The previous work of Boyen [13] assumed that the exclusive OR of two repeated enrollments leaked no information. The recent work of Canetti *et al.* [16] only achieves a sublinear error rate (and works for Hamming or set difference in the small universe setting).

The efficiency of our construction is bounded by the efficiency of digital lockers, we do not expect the use of information-theoretic FEs to be a roadblock to practical efficiency. For example, natural candidates to embed in our framework can be found in [21, Subsections 6.2, 6.3].

It is easy to adapt this construction to handle variable sizes. To do so, the RPIGen algorithm needs to pad with random elements up to a maximal set size to hide the actual number of elements in ω. If ω' is not the same size as ω it suffices to loop over the size of ω' in step 2 of RPIRep. Then, one could couple this RPI with any nonreusable FE that can handle sets of variable sizes (see [21, Subsection 6.3]) to obtain such an RFE.

4.2 Hamming Distance Instantiation

The work of Canetti *et al.* [16] presents three constructions of fuzzy extractors. They only claim reusability for Construction 1. However, under certain conditions Construction 2 is reusable. Furthermore, it implicitly uses the RPI framework, it first computes a map that preserves distance then applies an error-correcting code.[5]

Adapted construction of Canetti et al. Let $\mathcal{Z}$ be an alphabet and let $W = W_1, ..., W_s$ be a distribution over $\mathcal{Z}^s$. Let $C \subset \{0, 1\}^s$ a (t, δ_{code}) Hamming error correcting code that corrects t errors. Let $(\mathsf{lockPoint}, \mathsf{unlockPoint})$ be an s-composable secure obfuscated point function with error δ (for keys over $\mathcal{Z}$). The algorithms $\mathsf{Gen}, \mathsf{Rep}$ are defined in Figure 4.

The construction in Figure 4 does not produce a uniformly random r, it is necessary to apply a randomness extractor (technically, an average-case computational randomness extractor) to r, see [16, Section 5] for more information.

[5] We change the construction of Canetti *et al.* slightly to illustrate the connection to RPI but our construction is secure under the same conditions.

```
Algorithm Gen
Input: $w = w_1, ..., w_s$
1. Sample $c \leftarrow C, r \leftarrow \mathcal{U}_s$.
2. For $j = 1, ..., s$:
    a. If $r_j = 0$: Let $p_j = \mathsf{lockPoint}(w_j)$.
    b. Else:
        $t_j \overset{\$}{\leftarrow} \mathcal{Z}$.
        Let $p_j = \mathsf{lockPoint}(t_j)$.
3. Output $(c, p)$, where $p = p_1 \ldots p_s, o = r \oplus c$.
```

```
Algorithm Rep
Input: $(w', p)$
1. For $j = 1, ..., s$:
    a. If $\mathsf{unlockPoint}(w'_j, p_j) = 1$: set $r'_j = 0$.
    b. Else: set $r'_j = 1$.
2. Set $c = \mathsf{Decode}(o \oplus r')$.
3. Output $c \oplus r$.
```

Fig. 4. A Hamming distance-based RPI

In the work of Canetti *et al.* [16], this construction is not presented as reusable because not all symbols w_i are assumed to have entropy. If each symbol w_i individually has entropy (there is requirement on the correlation between symbols), the construction leaks no information: the string r is uniformly random from the adversaries points of view. The only information about r is revealed by $c \oplus r$ because c is chosen from a code. The digital locker is an RPI mapping w to a uniformly random point r in the output space. In Rep, the computed string $d(r, r') \leq d(w, w') \leq t$ (assuming the locker never has an error). The string c is functioning as a secure sketch (specifically, the code-offset secure sketch). Thus, this construction is actually a combination of a RPI and a (non-reusable) secure sketch.

5 Use cases

As previously mentioned, the industry standard is multifactor authentication. In the mobile world, this is often achieved through the combination of a password and an SMS OTP (respectively as knowledge and possession proofs), both being subject to strong limitations (see Subsection 1.2). Our use cases both consider a bank authenticating a user on their Android device [6]. For practical use cases, we consider a security level of 100 bits to be sufficient.

5.1 Use Case 1

We now construct a multifactor authentication system that proves possession of a phone and inherence of the user. For the first authentication factor, we use a source derived from hardware and software identifiers. For the second authentication factor, we replace the weak entropy password with alternative software fingerprints of an individual [36,12,5].

Notation and examples We denote ω_P and ω_I respectively as sources intended to prove possession and inherence. We will consider ω_I as a source that varies according to the set difference metric in the large universe setting. An example ω_P is described in Figure 5. An example ω_I is the first name of the user's top 20 contacts.

Both ω_P and ω_I individually carry enough entropy for strong authentication purposes (see respectively Figure 5 and Subsection 1.3). However, while the value ω_P is static, ω_I is a noisy secret that fits set difference metric. At first glance, we could use our reusable FE construction on just ω_I. However, each element of ω_I has little entropy so it does not fulfill the condition of Theorem 2 for reusability.

[6] Similar solutions could also be deployed for iOS devices [36].

Data	Estimated Entropy
IMEI	20 bits
IMSI	26 bits
AndroidID	64 bits
$\omega_P = \textit{IMEI}\|\textit{IMSI}\|\textit{AndroidID}$	110 bits

Fig. 5. Device fingerprint's entropy

To solve this problem, we propose to concatenate ω_P to each element ω_I. We denote this augmented source as $\tilde{\omega}_i = \omega_P||\omega_{I,i}$. This augmented source has min-entropy which is the sum of the individual sources. Furthermore, it ties together the two sources in a cryptographic way. Recall that our RPI instantiation (Figure 3) is built from digital lockers which only require min-entropy for security. Let $(\mathsf{Gen}, \mathsf{Rep})$ be the RFE from Theorem 1.Then we describe the instantiation of a fuzzy extractor for ω_P and ω_I in Figure 6.

```
Generation procedure Gen
Input: $\omega_P, \omega_I = \{\omega_{I,1}, \dots, \omega_{I,s}\}$.
1. For $i = 1 \dots s$:
     Set $\tilde{\omega}_i = \omega_P||\omega_{I,i}$.
2. Set $\tilde{\omega} = \{\tilde{\omega}_1, \dots, \tilde{\omega}_s\}$.
3. $(R, P) \leftarrow \mathsf{Gen}(\tilde{\omega})$.
4. Return $(R, P)$.
```

```
Reproduction procedure Rep
Inputs: $\omega'_P, \omega'_I, P$
1. For $i = 1 \dots s$:
     Set $\tilde{\omega}'_i = \omega'_P||\omega'_{I,i}$.
2. Set $\tilde{\omega}' = \{\tilde{\omega}'_1, \dots, \tilde{\omega}'_s\}$.
3. $R \leftarrow \mathsf{Rep}(\tilde{\omega}', P)$.
4. Return $R$.
```

Fig. 6. A practical use case for set difference based metric

Security Considering a normal utilization of the device, ω_P remains the same whereas ω_I can be subject to changes. Therefore, $\omega'_P = \omega_P$ and $\tilde{\omega}'_i = \omega_P||\omega'_{I,i}$. As a result, $d(\omega_I, \omega'_I) = d(\tilde{\omega}, \tilde{\omega}')$ and the correctness of the underlying RFE $(\mathsf{Gen}, \mathsf{Rep})$ ensures the correctness of use case 1. The designed fingerprint enables to apply Theorem 2.

5.2 Use Case 2

Our second use case uses identifying data that leads to distributions under the set difference metric with sufficient entropy for security. This construction can be seen as an extension of use case 1. We use the following sources of data:

- ω_{hard}, a device fingerprint based on the device's hardware elements (*e.g.* IMEI, MAC addresses).
- ω_{soft}, a software fingerprint relying on the device's software components.
- ω_{user}, a user fingerprint based on the user's personal information.
- $\omega_{RegSSID}$, a fingerprint based on the SSIDs registered on the device (see subsection 1.3).
- ω_{misc}, a fingerprint that can be based on miscellaneous data.

Some of these sources may not directly achieve sufficient entropy (e.g. ω_{user}). In this case, we augment the sources by using *AndroidID* in the same way as in the last use case. Potential instantiations of these sources are proposed in Figure 7.

Fingerprint	Description
ω_{hard}	*IMEI*$\|\|$*IMSI*$\|\|$*WifiMAC*$\|\|$*BluetoothMAC*$\|\|$*BatterySerialNumber*
ω_{soft}	*AndroidID*$\|\|$*HashOfLibraries*$\|\| \ldots \|\|$*OS version*
ω_{user}	*LastName*$\|\|$*FirstName*$\|\|$*BirthDate*$\|\| \ldots \|\|$*PhoneNumber*$\|\|$*UserPreferences*
$\omega_{RegSSID}$	*SSID*$_1 \|\| \ldots \|\|$*SSID*$_n$
ω_{misc1}	-
ω_{misc2}	-

Fig. 7. Potential fingeprints

Potential Fingerprints for ω_{misc} We propose some behavioral fingerprints as potential candidates for ω_{misc}. The idea would be to apply an FE on a noisy secret $\tilde{\omega}$ and use the resulting secret $\tilde{R}$ as the value for ω_{misc}. These potential fingerprints include, but are not limited to those introduced in Subsection 1.3:

- $\tilde{\omega}_{Ctcs}$, a top contacts fingerprint based on the user's n_1 favorite contacts.
- $\tilde{\omega}_{Song}$, a top songs fingerprint based on the user's n_2 most listened to songs.
- $\tilde{\omega}_{Apps}$, a top applications fingerprint based on the user's n_3 most used applications.

Setting n_1, n_2, n_3 accordingly should enable to fulfill the targeted security level (*e.g.* $n_1 = 20$).

Security Policy Based on these sources, we assume a server that authenticates a user only if a few ones have changed. This policy's threshold corresponds here to the RFE correction capacity t.

We then define the noisy secret $\omega = \{\omega_{hard}, \omega_{soft}, \omega_{user}, \omega_{RegSSID}, \omega_{misc1}, \omega_{misc2}\}$ for which ω_{hard} and ω_{user} should be static while the others are prone to changes. Here, the authentication server could set $t = 2, 3$.

Security As long as the number of errors remains inferior or equal to t, the correctness of the underlying RFE ensures the correctness of the construction. Each element of ω has sufficient entropy for our set difference instantiation to be reusable (see Theorem 2).

5.3 Discussion

In the case of HCE-based payment [1], major payment schemes [40,52,7] require the usage of device fingerprinting for authentication purposes. However, new regulations, such as GDPR (General Data Protection Regulation) in the European Union [25], will soon strictly regulate the usage of such data and limit their sharing with a server. Our use cases cope with these restrictions by executing on the mobile device a security policy decided by the server and thus avoiding the latter collecting the data. As mentioned in Subsection 1.3, the usage of software accessible fingerprints which do not require user interactions can be seen as a strong asset in mobile payment applications such as HCE-based ones.

6 Conclusion and Future Works

We present a reusability framework in which we propose to randomize fuzzy secrets before applying fuzzy extractors. Since fuzzy secrets may come from correlated distributions, the idea is to decorrelate them while preserving entropy and distances: we introduced the concept of *Reusable*

Pseudoentropic Isometries (RPIs) for such a purpose. We show how to build reusable fuzzy extractors out of any efficient nonreusable fuzzy extractors and RPIs.

Relying on this new framework, we use digital lockers to construct an RPI and design 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. We also show that the framework can be applied to the Hamming distance through the example of Canetti *et al.*'s second construction.

In the last section, we propose two practical use cases for our set difference instantiation, that reap benefits from the numerous device and behavioral fingerprints available on mobile phones. These use cases show the applicability of our RPI construction in the context of industrial mobile authentication.

References

1. Host-based card emulation. https://developer.android.com/guide/topics/connectivity/nfc/hce.html.
2. Most common first names and last names in the u.s. `https://names.mongabay.com/male_names.htm`.
3. Multi-factor authentication. https://www.pcisecuritystandards.org/pdfs/Multi-Factor-Authentication-Guidance-v1.pdf.
4. 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.
5. 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.
6. D. Apon, C. Cho, K. Eldefrawy, and J. Katz. Efficient, reusable fuzzy extractors from lwe. Cryptology ePrint Archive, Report 2017/755, 2017. `http://eprint.iacr.org/2017/755`.
7. Bancontact. SEPA Rulebooks Scheme Manuals Remote Domain 46D0 Mobile App Security Guidelines, 2016.
8. C. H. Bennett, G. Brassard, and J.-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.
9. N. Bitansky and R. Canetti. On strong simulation and composable point obfuscation. In *Advances in Cryptology–CRYPTO 2010*, pages 520–537. Springer, 2010.
10. 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.
11. 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.
12. H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh. Mobile device identification via sensor fingerprinting. *CoRR*, abs/1408.1416, 2014.
13. 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.
14. R. Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, pages 455–469, 1997.
15. R. Canetti and R. R. Dakdouk. Obfuscating point functions with multibit output. In *Advances in Cryptology–EUROCRYPT 2008*, pages 489–508. Springer, 2008.
16. 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.
17. R. Canetti, Y. Tauman Kalai, M. Varia, and D. Wichs. *On Symmetric Encryption and Point Obfuscation*, pages 52–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
18. 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.

19. 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.
20. J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14:21–30, 2002.
21. 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.
22. 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.
23. 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.
24. ETSI. Digital cellular telecommunications system (phase 2+). *Security related network functions*, 1992.
25. European Parliament. General Data Protection Regulation (GDPR). `http://ec.europa.eu/justice/data-protection/reform/files/regulation_oj_en.pdf`, 2016.
26. B. Fuller, X. Meng, and L. Reyzin. *Computational Fuzzy Extractors*, pages 174–193. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
27. J. Galbally, A. Ross, M. Gomez-Barrero, J. Fierrez, and J. Ortega-Garcia. Iris image reconstruction from binary templates: An efficient probabilistic approach based on genetic algorithms. *Computer Vision and Image Understanding*, 117(10):1512–1525, 2013.
28. Google. Android Developer Reference, Settings.Secure, ANDROID_ID. `https://developer.android.com/reference/android/provider/Settings.Secure.html#ANDROID_ID`.
29. Google. Android Security 2016 Year in Review. `https://source.android.com/security/reports/Google_Android_Security_2016_Report_Final.pdf`, 2017.
30. 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.
31. 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.
32. A. K. Jain, K. Nandakumar, and A. Ross. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, 2016.
33. A. Juels and M. Sudan. A fuzzy vault scheme. *Des. Codes Cryptography*, 38(2):237–257, Feb. 2006.
34. 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.
35. S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer̃, N. Christin, L. F. Cranor, and S. Egelman. Of passwords and people: measuring the effect of password-composition policies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2595–2604. ACM, 2011.
36. 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.
37. J. Lin, B. Liu, N. M. Sadeh, and J. I. Hong. Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings. In *Tenth Symposium on Usable Privacy and Security, SOUPS 2014, Menlo Park, CA, USA, July 9-11, 2014*, pages 199–212, 2014.
38. 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.
39. B. Lynn, M. Prabhakaran, and A. Sahai. Positive results and techniques for obfuscation. In *Advances in Cryptology–EUROCRYPT 2004*, pages 20–39. Springer, 2004.
40. MasterCard. MasterCard CloudBased Payments Security Guidelines for MPA Development Version 1.1, 2015.
41. K. Mowery and H. Shacham. Pixel perfect: Fingerprinting canvas in HTML5. In M. Fredrikson, editor, *Proceedings of W2SP 2012*. IEEE Computer Society, May 2012.
42. C. Mulliner, R. Borgaonkar, P. Stewin, and J. Seifert. Sms-based one-time passwords: Attacks and defense - (short paper). In *Detection of Intrusions and Malware, and Vulnerability Assessment - 10th International Conference, DIMVA 2013, Berlin, Germany, July 18-19, 2013. Proceedings*, pages 150–159, 2013.
43. 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.

44. N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.
45. NIST. Digital identity guidelines: Authentication and lifecycle management. `http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf`.
46. R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.
47. S. Prabhakar, S. Pankanti, and A. K. Jain. Biometric recognition: Security and privacy concerns. *IEEE Security & Privacy*, 1(2):33–42, 2003.
48. 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.
49. 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.
50. K. Simoens, P. Tuyls, and B. Preneel. Privacy weaknesses in biometric sketches. In *2009 30th IEEE Symposium on Security and Privacy*, pages 188–203.
51. 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.
52. VISA. Security Requirements and Evaluation Guidance for Mobile Applications. Visa Digital Solutions. Version 1.0, 2014.
53. Wikipedia. International Mobile Equipment Identity (IMEI). `https://en.wikipedia.org/wiki/International_Mobile_Equipment_Identity`.
54. 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.