Public Key Cryptosystems with Noisy Secret Keys

Charles Herder[1], Benjamin Fuller[2], Marten van Dijk[2], and Srinivas Devadas[1]

[1] Computer Science and Artificial Intelligence (CSAIL), Massachusetts Institute of Technology
[2] University of Connecticut

Abstract. Passwords bootstrap symmetric and asymmetric cryptography, tying keys to an individual user. Biometrics are intended to strengthen this tie. Unfortunately, biometrics exhibit noise between repeated readings. Fuzzy extractors (Dodis et al., Eurocrypt 2004) derive stable symmetric keys from noisy sources.

We ask if it is also possible for noisy sources to directly replace private keys in asymmetric cryptosystems. We propose a new primitive called *public-key cryptosystems with noisy keys*. Such a cryptosystem functions when the private key varies according to some metric. An intuitive solution is to combine a fuzzy extractor with a public key cryptosystem. Unfortunately, fuzzy extractors need static helper information to account for noise. This helper information creates fundamental limitations on the resulting cryptosytems.

To overcome these limitations, we directly construct public-key encryption and digital signature algorithms with noisy keys. The core of our constructions is a computational version of the fuzzy vault (Juels and Sudan, Designs, Codes, and Cryptography 2006). Security of our schemes is based on graded encoding schemes (Garg et al., Eurocrypt 2013, Garg et al., TCC 2016). Importantly, our public-key encryption algorithm is based on a weaker model of grading encoding. If functional encryption or indistinguishable obfuscation exist in this weaker model, they also exist in the standard model.

In addition, we use the computational fuzzy vault to construct the first reusable fuzzy extractor (Boyen, CCS 2004) supporting a linear fraction of errors.

1 Introduction

Cryptography relies on long-term secrets for key derivation, authentication, and private key storage. Passwords are the traditional mechanism to derive keys to secure such applications. There is considerable research on the insecurity of passwords [4]. Biometrics [12, 47, 6, 16, 37, 39] are an alternative source for long-term secrets. At its core, the ideal vision of biometrics is simple: to use biometrics in place of secret keys for modern public/symmetric key cryptosystems. Informally, we want the following:

– Good reliability − biometrics are noisy (linear error rates), and have a fixed amount of entropy. Therefore, we need to accept a linear fraction of errors efficiently while minimizing the cost to security.
– Re-provisionable keys − a user should be able to enroll his/her identity with multiple authorities.
– Ease of use − a user should be able to use *only* his/her biometrics to identify him/herself. A protocol should not require the user to carry around additional factors of authentication.

This vision has not been achieved, and biometrics are not widely used in Internet applications [4]. Significant progress has been made via fuzzy extractors, which derive stable long-term symmetric keys from biometrics [15]. A fuzzy extractor is a pair of algorithms. Generate or Gen that takes an initial value c from the biometric and outputs a uniform key r and a template, p. For security, p should contain little information about the biometric or the derived key (either information-theoretically [15] or computationally [17]). The second algorithm reproduce or Rep takes a noisy biometric reading c' and p to reproduce the key if the two readings are close enough (according to some metric d).

A strawman construction At first glance, fuzzy extractors should enable the vision of biometric public key cryptography. Consider the combination of a fuzzy extractor $(\mathsf{Gen}', \mathsf{Rep}')$ and a public-key encryption scheme, denoted $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. The idea is to run $\mathsf{Gen}'(w)$ (where w is the measured biometric) to create a stable key r. This value r is used as the randomness for the encryption key generation algorithm, $(pk, sk) \leftarrow \mathsf{Gen}(r)$. At decryption time, the value r is regenerated (from w', a second measurement of the biometric) and used to regenerate (pk, sk). There are two limitations to this approach.

1. Current efficient[3] fuzzy extractors require p to be public, unique to the user, and constant forever (each user can only have one p over his/her entire lifetime). Therefore, one of the following two scenarios must be the case:
   – The user personally manages p. This is problematic at scale, as average users will not have the expertise to personally manage keys.
   – There is a trusted centralized authority that manages all users' biometrics for the *entire lifetime* of its users. This does not scale, creates a single-point-of-failure, and the authority cannot expire or be revoked, as its users' keys cannot expire (or be revoked) during their lifetime.
2. Even if a unique p is acceptable from a scalability perspective it hurts usability as p is required for decryption. Therefore, one of three scenarios is required:

[3] Some recent fuzzy extractor constructions [8, 23] can be provisioned more than once, known as a reusable fuzzy extractor [5]. However, prior to our work, there were no known reusable fuzzy extractors for a linear fraction of errors (without assuming virtual grey box obfuscation for $\mathbf{NC}^1$ circuits [3]). Practical biometrics have a linear fraction of errors so we don't consider these algorithms here.

- When the user receives a ciphertext, he/she downloads p from a central authority. This requires a centralized always-on service for anyone to decrypt. Furthermore, traffic patterns at this service reveal substantial information about user behavior.
- The user carries p with him/her. This hinders ease of use and requires users to carry a token for decryption (this is the same usage model as multi-factor authentication). This p cannot be lost by the user, as re-provisioning p compromises security.
- p is transmitted alongside the ciphertext. This de-anonymizes the ciphertext − every ciphertext is now attributable to the user over their entire lifetime. It would be possible for an adversary to link all ciphertexts over the life of an individual user.

We propose a new primitive: *public-key cryptosystems with noisy secrets* that is a significant step towards realizing the vision for biometrics described at the beginning of the introduction. We propose a public-key encryption scheme and a digital signature scheme where a noisy value can be used as a private key. Informally, key generation takes as input a value sk and creates a public key such that decryption (resp. signing) is possible from all nearby sk' (that is, where $d(sk, sk') \leq t$). This is in contrast to fuzzy IBE where the public key is noisy [43].
Our encryption scheme rectifies the above problems and accomplishes the following:

1. An embodiment of a public key encryption system, where the biometric directly replaces the secret key (and there is no helper information).[4]
2. A user can create multiple public keys corresponding to their noisy biometric. It is infeasible to determine if public keys correspond to the same private key.
3. Public keys are anonymous, containing no information about the individual's biometric.
4. It is not possible to determine if two ciphertexts correspond to the same public key. No public randomness is necessary for decryption, only the ciphertext and the biometric.

Our approach draws on the fuzzy vault of Juels and Sudan [30] which we now describe:

Gen A randomly sampled value r is viewed as a polynomial $Q(x)$. A set of x-values is determined from the input value c. Interpolating points on this polynomial are published x_i, y_i. In addition, random values x'_i, y'_i are added to this set. These random points are known as chaff. The true points together with chaff points are the public value p.

[4] We also construct the first reusable fuzzy extractor that corrects a linear fraction of errors. We stress that achieving the desired properties requires more than a reusable fuzzy extractor.

Rep The user inputs p and a nearby value c'. Using w' the user is able to create a similar set of x_i values. This set x_i will contain mostly points that are on the polynomial $Q(x)$, thus the user can interpolate the polynomial using these points to recover r.

The security argument is information theoretic − given a high-enough degree polynomial, the chaff points are information theoretically indistinguishable from the points interpolating $Q(x)$. This is because for a given set of real and chaff points, there are (statistically) many false polynomials that happen to have enough interpolating points. These false polynomials can't be distinguished from the real polynomial.

1.1 Overview of our Approach

Our approach is similar to the fuzzy vault but replaces an arbitrary field with points on a graded encoding scheme [18]. Importantly, our security assumption is weaker than the hybrid graded encoding scheme assumption of [20]. We show that our security assumption is unlikely to imply either functional encryption or indistinguishability obfuscation (Lemma 1).

Graded Encodings Our construction leverages recently development graded encoding schemes (GES), first described in [18]. A GES allows addition of values at the same level ℓ_i. Multiplying two values changes the level of the output to be the sum of the levels. So multiplying values at two encoding levels ℓ_i and ℓ_j yields a value at a different encoding level ℓ_k. In addition, it is possible to check whether an encoded value at a distinguished level ℓ^* is equal to zero (known as the zero-test). During initialization, we choose which encoding levels must be multiplied to obtain an ℓ^* encoding. We denote the "multilinearity level," the number of encodings levels that are multiplied to obtain the zero-test level as κ.

Graded encodings have seen extensive application since their introduction. They can be used to construct indistinguishability obfuscation [19] for NC^1 circuits (and all of P assuming the security of learning with errors). Recent cryptanalytic "zeroizing" attacks have been used to break constructions using the public zero-test parameter [18, 9, 28, 38].

As a warmup we construct a fuzzy extractor based on graded encodings. That is, we show how to derive a stable symmetric key using graded encodings. We will then transform this basic approach to construct our public key algorithms.

A computational fuzzy vault We consider noisy values C taking values in $\mathbb{R}^{m\times\kappa}$. In addition, we consider a binary version of this matrix $\mathtt{sign}(C)$, that we denote as O. For simplicity, we refer to the pair (O, C) but O is completely determined by C. (See Section 2.2 for a discussion of biometric distributions.)

On taking subsequent samples, we observe that for a good biometric, $C' \approx C$. If we consider a bit $O_{i,j} = \mathtt{sign}(C_{i,j})$ and $O'_{i,j} = \mathtt{sign}(C'_{i,j})$, then informally, we know that if $|C'_{i,j}|$ is large with respect to the expectation of $|C_{i,j} - C'_{i,j}|$, then it is unlikely that $O_{i,j} \neq O'_{i,j}$. Recent studies have shown that this approach is a

Gen

1. Input: $1^\lambda, O \in \{0,1\}^{m \times \kappa}$
2. Sample $\{pp, sp\} = \mathsf{GES.Setup}(1^\lambda, \kappa)$
3. For $i \in [m], j \in [\kappa]$
   (a) $\boldsymbol{H}_{i,j} = \mathsf{GES.Encode}(0, l_i, sp, pp)$
   (b) If $O_{i,j} = 0$:
   $\boldsymbol{H}_{i,j} = \mathsf{GES.SProd}(h_{i,j}, U(R), pp)$
4. Sample $seed$.
5. Set $K = Ext(O, seed)$.
6. Set $P = \{\boldsymbol{H}, pp, seed\}$
7. Output (K, P).

Rep

1. Input: $C' \in \mathbb{R}^{m \times \kappa}, P$.
2. Parse P as $\{\boldsymbol{H}, pp, seed\}$.
3. Set $C''_j = \arg\max_{i \in [m]} |C'_{i,j}|$, $j \in [\kappa]$
4. $prod = \mathsf{GES.Prod}(\{\boldsymbol{H}_{C''_i, i}\}, pp)$
5. If $0 = \mathsf{GES.ZT}(prod, pp)$, return $\perp$.
6. For $i \in [m], j \in [\kappa]$:
   (a) Set $C^T = C''$.
   (b) Set $C^T_j = i$.
   (c) $prod = \mathsf{GES.Prod}(\{\boldsymbol{H}_{C^T_j, j}\}, pp)$
   (d) If $1 = \mathsf{GES.ZT}(prod, pp)$,
   $O_{i,j} = 1$.
   (e) Else $O_{i,j} = 0$.
7. Output $K = Ext(O, seed)$.

Algorithm 1. Pseudocode of the computational fuzzy vault. The algorithms Setup, Encode, SProd, Prod, and ZT are operations on the graded encoding scheme.

good model for human and silicon biometrics, and has resulted in performance improvements in fuzzy extractors [23, 25, 46]. We now describe the algorithm:

Gen The Gen algorithm uses O. A 1 in the i-th column of O is translated into an encoding of 0 at level i. A 0 is converted into a random value in the level i encoding space. With high probability this value is not an encoding of any value. This is described pictorially in Figure 2. This creates a matrix of GES values H that is stored as the public value P. The key K is an appropriate randomness extractor applied to O [2, 40].

Rep The Rep algorithm uses the matrix C' (a noisy version of C). For each column i, find the row j_i that contains the maximum value in C. The values $H[j_i, i]$ are then multiplied. If C' was close to C this creates a level ℓ^* encoding of 0. The zero test of the GES is used to check correctness. If the zero test succeeds, the remainder of O can be recovered bit by bit. This is depicted in Figure 3.

Similar to the original fuzzy vault, we use C' to select a subset of locations in the matrix that are likely to be error-free (cf. Section 2). This subset is interspersed with chaff points that are not valid encodings. Including a single invalid encoding in the chosen subset is enough to prevent extraction. Security relies on the inability of an adversary to select a valid encoding at each level. Analysis of this fuzzy vault is formalized in Section A, where it is shown that Algorithm 1 is the first reusable fuzzy extractor that can correct a constant fraction of errors.

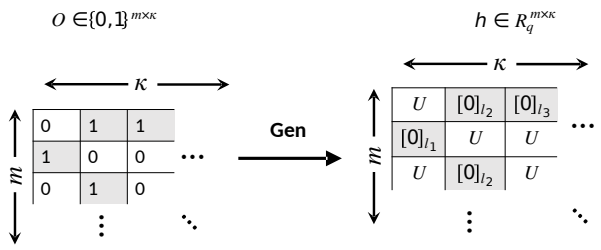

Fig. 2. Graphical representation of an example of how the Gen algorithm computes $\boldsymbol{H}$. A '1' in the i'th column of O is translated to a valid encoding of 0 under l_i. A '0' is translated to a uniformly random value in the encoding space.

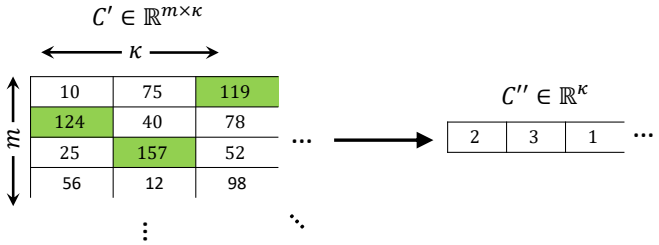

Fig. 3. Graphical representation of an example of the first step of Rep, selecting C'' by using the maximal confidence information from each column of C'. This vector is used to compute a valid level-ℓ^* encoding of 0 using $\boldsymbol{H}$, which is in turn used to compute O by swapping entries of of the matrix and re-testing if the product is a valid zero encoding.

1.2 Public-key encryption with noisy keys

The mathematical flexibility of the graded encoding scheme allows us to create a public key encryption and a digital signature algorithm directly. As discussed above, these cryptosystems allow a *noisy private key*.

The public-key encryption scheme operates as follows. In Gen, the computational fuzzy vault is executed a large number of times to create a number of $\mathbf{H}_i$. The Enc algorithm adds a random subset of these $\mathbf{H}_i$ to create a matrix $\mathbf{H}^*$. If the bit to be encrypted is 1, all elements are replaced by random elements in the encoding space. Decryption uses the Rep algorithm on $\mathbf{H}^*$. If Rep succeeds the plaintext is 0 otherwise it is 1. Our construction satisfies traditional IND-CPA security (cf. Section 5).

Unique features This construction achieves the vision described at the beginning of this section. One powerful advantage of this encryption approach is that *biometric templates may be reused*. That is, one may construct polynomially many

unique public keys from a single (noisy) biometric template. These public keys, even when considered together, reveal nothing about the underlying input value C. Further, an adversary cannot distinguish two public keys derived from the same private key versus two public keys derived from unique private keys. Ciphertexts do not reveal which public key was used to generate them. Finally, because the private key consists entirely of the biometric template, the individual is not required to carry any additional storage or memorize any PIN in order to decrypt the message.

Security assumption We require a strictly weaker assumption than is required for the construction of indistinguishability obfuscation (iO) from a GES [20]. We assume a complete break of the system if the adversary is able to find a single element where the zero-test returns true. We show that if functional encryption exists for any complexity class in the presence of our GES, then it exists without the use of the GES (Lemma 1).

1.3 Digital signatures with noisy signing keys

We also construct an existentially unforgeable signature system (cf. Section 6). However, the proposed biometric signature system requires the *full hybrid graded encoding* assumption [20] that implies VBB obfuscation and functional encryption. The stronger assumption is required in this construction because there is a public zero-test parameter (in contrast to our encryption scheme). Finally, the signature system requires the signer to have access to his/her public key at the time of signing. Thus, we note that such a construction can also exist by obfuscating a distance check and a signature algorithm. It would be interesting to further explore the possibility of constructing biometric digital signatures *without* the above limitations.

We present this scheme for two reasons: 1) to contrast with the weak GES that is sufficient for encryption 2) the construction uses a zeroization attack as a part of normal operation. The signature system uses the zeroization attack on [18] and similar GES constructions. To the best of our knowledge this is the first time that a zeroization attack has been used in a cryptographic construction.

Our public key encryption and signature systems use graded encoding schemes in a fundamentally different way. The noisy secret (in this case the biometric template) can be used to obfuscate encodings of zero − with access to the secret biometric, the individual may access low-level encodings of zero that enable encoding re-randomization and extraction of encoding plaintexts through zeroization. Further, in the encryption algorithm, the secret is required in order to successfully use the zero-test parameter − weakening the security assumption.

1.4 Organization

Section 2 describes the properties of the biometric distribution required for security and efficiency of the proposed algorithms. We give background for graded encoding schemes and present our model in Section 3. We define and provide a

construction for a biometric public key cryptosystem in Section 4. We construct a biometric public-key encryption in Section 5. We similarly define and construct a biometric digital signature system in Section 6. Appendix A formally presents the computational fuzzy vault described here and compares it to state of the art fuzzy extractors.

2 Requirements on Biometric Source

Cryptographic algorithms using biometrics need two properties of the biometric. These two properties (informally) are stability and entropy. Stability has been formalized in previous work as an error rate when modeling the noisy value O as a stable value with i.i.d. noise [36, 17, 30, 15, 31]. Further, information theoretic constructions normally require O to have a prescribed amount of min-entropy [36, 15]. We require a slightly stricter requirement on O than min-entropy, but significantly looser than i.i.d. (which is used in prior constructions, e.g. [45])

2.1 Formal Requirements on Biometric Distributions

Consider a matrix $O \in \{0,1\}^{m \times \kappa}$ of bits output by the measurement of a human biometric or a physical unclonable function [41, 42] (which have similar characteristics).[5] Interpret this bitstring as a binary matrix $O_{i,j} \in \{0,1\}$ with $i \in [m]$, $j \in [\kappa]$. Our constructions rely on the presence of a non-binary version of O that exists in biometrics called *confidence information*.

That is, we assume the existence of an additional matrix $C \in \mathbb{R}^{m \times \kappa}$ that is used to derive $O_{i,j} = \mathtt{sign}(C_{i,j})$. One can either think of sampling C and deriving O or jointly sampling these values. We consider two different distributions: $\{O, C\} \leftarrow\!\!\shortmid \chi_{\text{Inter}}$ representing a sample of a biometric from a random individual, and $\{O', C'\} \leftarrow\!\!\shortmid \chi_{\text{Intra}}$ representing repeated measurements of a biometric from a single individual.

Definition 1 (Biometric with Confidence Information). *A biometric $O \in \{0,1\}^{m \times \kappa}$ has confidence information $C \in \mathbb{R}^{m \times \kappa}$ if $O_{i,j} = \mathtt{sign}(C_{i,j})$, and there exists a decreasing function $f(\cdot) \geq 0$ such that:*

$$\Pr(O'_{i,j} \neq O_{i,j} : |C_{i,j}| > C_{\text{Thresh}}) = \epsilon_2 \leq f(C_{\text{Thresh}})$$

where the probability is taken over $\{O, C\}, \{O', C'\} \leftarrow\!\!\shortmid \chi_{\text{Intra}}$.

A large $|C_{i,j}|$ corresponds to a bit that has $\Pr(O_{i,j} = O'_{i,j}) > 1 - \epsilon_2$ for some small failure probability ϵ_2. The asymptotic behavior of f in Definition 1 directly impacts the efficiency of a confidence-based system. Ideally, $f =$

[5] We suggestively use the variable κ, as it will be the same as the multilinearity level in the final cryptosystems, and will be correlated to the security level of the cryptosystem. m will be seen to be a parameter for the "level of redundancy," and will be correlated to the error correction capability of the cryptosystem.

$O(1/\mathsf{Poly}(C_{\text{Thresh}}))$, so that one may choose C_{Thresh} large enough to suppress erroneous measurements.

It may not be possible to suppress the failure probability to be negligible in the security parameter. However, biometric scanning systems are not expected to have a negligible failure probability so this is acceptable. Their performance is assumed to degrade with environmental conditions and the behavior of the user.

However, many fuzzy sources (e.g., ring oscillator PUFs [44]) do exhibit sufficient error suppression with increasing confidence information [23]. Using confidence information, we define the requirements on key stability for protocols described in this paper. Similar to [23], Definition 2 requires that a bit whose confidence information is larger than some threshold (C_{Thresh}) has a probability of flipping between measurements less than some ϵ_2.

Definition 2 (Biometric Stability). *A biometric $O \in \{0,1\}^{m\times\kappa}$ with confidence information $C \in \mathbb{R}^{m\times\kappa}$ is (ϵ_1, ϵ_2)-stable if there exists a C_{Thresh} such that:*

- $\Pr(O'_{i,j} \neq O_{i,j} : |C_{i,j}| > C_{\text{Thresh}}) \leq \epsilon_2$ *over* $\{O, C\}, \{O', C'\} \hookleftarrow \chi_{\text{Intra}}$
- $\Pr_{\{O,C\}\hookleftarrow\chi_{\text{Inter}}}(\exists\, i \in [\kappa] \text{ s.t. } |C_{i,j}| > C_{\text{Thresh}}) > 1 - \epsilon_1$ *for all* $j \in [m]$

Definition 2 informally states that a high-confidence bit $O_{i,j}$ with $|C_{i,j}| > C_{\text{Thresh}}$ has less than ϵ_2 probability of flipping between measurements. Further, the biometric has enough bits (large enough m) such that at least one stable bit is found in each group of m bits. Note that m is purely determined by the size of the biometric, and determines the level of redundancy and therefore will contribute to the error correction capability of the cryptosystem. We will see that κ corresponds to security and key size.

The relationship of κ to key size is formalized in Definition 3. This notion of biometric entropy has a natural physical interpretation which we discuss next.

Definition 3 (Biometric Grouped Entropy). *A biometric $O \in \{0,1\}^{m\times\kappa}$ has λ-lower bounded m-grouped entropy if:*

$$\lambda < \min_{S\in\mathbb{Z}_m^\kappa} H_\infty(\{O_{S_i,i}\} : i \in [\kappa])$$

Definition 3 requires that the an adversary cannot pick a set of κ bits (one bit from each of the κ buckets of size m) and be correct with greater than $2^{-\lambda}$ probability. Formally, Definition 3 requires that the min-entropy of all sets containing one bit from each bin is lower-bounded by λ.

Observe that Definition 3 is strictly stronger than a min-entropy requirement, but is strictly weaker than an i.i.d. assumption. Further, Definition 3 is weaker than many other intermediate assumptions, such as the assumption in recent fuzzy extractor constructions [23].

2.2 A Primer on Biometric Distributions

The distributions of human biometric data such as fingerprints, irises, are not (and likely cannot be) known precisely. Therefore, one cannot prove that the biometric data obeys any formal Definitions (including Definitions 2 and 3).

Empirical observations to date do not contradict our definitions. The notion of confidence information improves both performance and security of silicon and human biometric systems [23, 25, 46]. In particular, empirical studies of the human iris [13] find high amounts of entropy providing evidence they satisfy Definition 3.

Definition 3 also has a natural physical interpretation in the context of biometrics. For many biometrics, neighboring bits are more strongly correlated than bits that are further away. Definition 3 describes a biometric that has κ groups of m bits. Among the m bits of a single group, there is no requirement on the distribution (i.e., all m bits in a group could be equal to each other without impacting security). Physically, this can be interpreted as sampling a set of neighboring bits. Next, each group of m bits must have a distribution that is sufficiently independent from each other group. This corresponds to requiring that the groups of m bits are sufficiently far away from each other.

In fact the output of an iris sensor is a matrix derived from applying imaging processing at different polar coordinates. This matrix is correlated but entropic in for a single fixed polar angle with weak correlation across polar angles [22]. This matches the requirements of Definition 3.

3 Multilinear maps and graded encodings

This work will leverage the recent work on instantiating multilinear maps using noisy graded encodings on ideal lattices [18] as well as the recent approaches to rectify the discovered vulnerabilities [20]. It will be shown that the constructions in this paper are at least as secure as those presented in [20], and are qualitatively more secure.

3.1 Multilinear Maps

A multilinear map is parameterized by the security parameter λ, the multilinearity level $\kappa = \mathsf{Poly}(\lambda)$. The multilinear cryptosystem is written as a graded encoding scheme (GES) that encodes plaintext values of ring elements at different levels denoted by a label $l \in S$ for some set S. Notationally, we refer to an encoding of plaintext $\boldsymbol{a}$ under label l as $[\boldsymbol{a}]_l$. When not confusing, we use the same font (e.g., $\boldsymbol{x}$) to denote an encoding where the underlying plaintext and/or encoding level are unspecified.

There is an additive homomorphism for encodings at the same level, and scalar multiplication of any encoding may also be computed homomorphically.

The multiplicative homomorphism changes the encoding level. This work will use a fully asymmetric graded encoding of multilinearity level $\kappa = O(\lambda)$. The zero-test level l^* will be the product of single encodings from each level:

$$\prod_{i \in [\kappa]} [\boldsymbol{a}_i]_{l_i} = \left[\prod_{i \in [\kappa]} \boldsymbol{a}_i \right]_{l^*}$$

This work will use exclusively *noisy* encodings, as they allow multiple unique encodings of the same plaintext at the same encoding level, which will be crucial to the algorithm. As a result, all plaintexts and encodings, regardless of encoding level will live in a single ring R.

The GGH encoding scheme [18] encodes values as elements of the quotient ring $R/\langle \boldsymbol{g} \rangle$, where $\langle \boldsymbol{g} \rangle$ is prime, and $\boldsymbol{g}$ is a *secret* short vector in R. A short vector $\boldsymbol{e}$ is encoded at level 0 as $\boldsymbol{e} + \boldsymbol{g} \cdot \boldsymbol{r}$ for some short random vector $\boldsymbol{r}$. Level-0 encodings are analogous to plaintext.

Higher level encodings are obtained through additional secret vectors $\boldsymbol{z}$ drawn uniformly at random in R. An encoding (using $\boldsymbol{z}$) is obtained by computing $\frac{\boldsymbol{e}+\boldsymbol{g}\cdot\boldsymbol{r}}{\boldsymbol{z}}$.

Encodings have the additive and multiplicative homomorphisms described above so long as the accumulated noise does not overflow. The public zero-test parameter is defined as $\boldsymbol{p}_{\text{zt}} = \boldsymbol{h} \cdot \boldsymbol{z}_{l^*}/\boldsymbol{g} \mod q$, where $\boldsymbol{h}$ is "somewhat small". One computes whether a level-l^* encoding is zero (i.e., equal to $\boldsymbol{u} = \boldsymbol{g} \cdot \boldsymbol{r}/\boldsymbol{z}_{l^*}$ for small $\boldsymbol{r}$) if $\boldsymbol{u} \times \boldsymbol{p}_{\text{zt}}$ is "small."

One extension of the graded encoding scheme that will be used in this paper is to use multiple secret vectors $\boldsymbol{z}_i$. Specifically, we will choose uniform $\boldsymbol{z}_i$ for $i \in [\kappa]$, and compute the zero test parameter as $\boldsymbol{p}_{\text{zt}} = \boldsymbol{h} \cdot \boldsymbol{z}_{l^*}/\boldsymbol{g}$ with $\boldsymbol{z}_{l^*} = \prod_{i \in [\kappa]} \boldsymbol{z}_i$.

Mitigating Vulnerabilities The GGH multilinear map construction is still new, and is not based on a well understood hardness assumption. However, its widespread potential applications (through the use of indistinguishability obfuscation) have inspired a significant cryptologic research effort. A full review of the recent efforts to break/repair GGH and its variants is beyond the scope of this paper. Instead, we focus on a single encoding scheme - the original GGH encoding using ideal lattices, and focus on a recent security argument [20], which as of the writing of this paper, is unbroken. The construction in this paper uses a strictly weaker security argument than [20] which does not imply indistinguishability obfuscation or functional encryption. We will now briefly (and informally) review the vulnerabilities identified in the GGH multilinear map and the new security model.

Most vulnerabilities discovered with the GGH multilinear maps construction (and variants) has been through the zero test parameter $\boldsymbol{p}_{\text{zt}}$. This is not surprising, as this parameter (informally) removes the obfuscating encoding of the $\boldsymbol{z}_i$ terms. If the zero-test procedure succeeds (results in a short vector), one is left with a level-0 encoding of some form, which is analogous to plaintext. "Zeroizing" attacks exploit correlations among these "plaintext" values to compute the secret parameters of the system [38, 10, 27].

Specifically, if one is able to obtain multiple l^* encodings of 0 (such that multiplication by $\boldsymbol{p}_{\text{zt}}$ results in a short vector in R), then one may compute (in some cases) the secret parameter $\boldsymbol{g}$ that generates the principal ideal corresponding to zero-encodings [38, 10].

A new security model for GGH has been proposed in [20] that captures this class of vulnerabilities (termed "hybrid graded encoding"). This model is

a stateful oracle model, that we describe further in Section 3.2. Further, [20] uses this model of GGH to build VBB obfuscation, where l^* encodings of 0 are decorrelated, so the above zeroizing attacks don't apply.

This paper uses an oracle model similar to [20]. The key difference, as detailed in Section 3.2 is that the adversary wins if it constructs *any* valid l^* encoding of 0 with non-negligible probability in the security parameter.

3.2 Hardness Assumption

The GGH13 graded encoding construction (and other similar constructions) have been broken and repaired several times in the past few years [18, 9, 27, 28, 10, 11, 38, 21]. This paper follows and extends the security argument in [20]. We assume that the noisy graded encoding scheme (GES) has the following interface:

Definition 4. *A Noisy GES has the following functions*

$$\begin{aligned}
\{pp\} &\leftarrow \mathsf{GES.Setup}(1^\lambda, \kappa) \\
\{\boldsymbol{p}_{\mathrm{zt}}, sp\} &\leftarrow \mathsf{GES.Gen}(pp) \\
[\boldsymbol{a}]_l &\leftarrow \mathsf{GES.Encode}(\boldsymbol{a}, l, sp, pp) \\
[\boldsymbol{a}+\boldsymbol{b}]_l &\leftarrow \mathsf{GES.Add}([\boldsymbol{a}]_l, [\boldsymbol{b}]_l, pp) \\
\left[\prod_{k=i_1}^{i_j} \boldsymbol{a}_{\boldsymbol{i_k}}\right]_{l_{i_1+\ldots i_j}} &\leftarrow \mathsf{GES.Prod}(\{[\boldsymbol{a}_{\boldsymbol{i_1}}]_{l_{i_1}}, [\boldsymbol{a}_{\boldsymbol{i_2}}]_{l_{i_2}}, \ldots, [\boldsymbol{a}_{\boldsymbol{i_j}}]_{l_{i_j}}\}, pp) \\
[\boldsymbol{c}\cdot\boldsymbol{a}]_l &\leftarrow \mathsf{GES.SProd}([\boldsymbol{a}]_l, \boldsymbol{c}, pp) \\
\boldsymbol{p}'_{\mathrm{zt}} &\leftarrow \mathsf{GES.ZTSProd}(\boldsymbol{c}, pp, \boldsymbol{p}_{\mathrm{zt}}) \\
\{0,1\} &\leftarrow \mathsf{GES.ZT}([\boldsymbol{a}]_{l^*}, \boldsymbol{p}_{\mathrm{zt}}, pp)
\end{aligned}$$

The noisy GES is correct if GES.ZT *returns 1 iff* $\boldsymbol{a} = 0$ *with probability* $1 - \mathsf{neg}(\lambda)$.[6]

There are two sets of public parameters. The value pp is a universal public parameter for security parameter λ and multilinearity κ (e.g., a description of the ring of encodings). The per use parameters are sampled in GES.Gen, which also computes the public zero-test parameter $\boldsymbol{p}_{\mathrm{zt}}$.[7]

Most GES schemes do not include functionality to multiply the zero test parameter (GES.ZTSProd). We will use this functionality to randomize the zero test parameter. This is a trivial operation for most GES instantiations, including [18]. One simply chooses random, invertible $\boldsymbol{r} \in R$ and multiplies the zero-test parameter by this value. However, for zero test to succeed after this operation requires changing some handles as well. Importantly, this change does not require any change to the hybrid security model we introduce next. We return to this functionality in Section 5.

[6] This API defines correctness. We model the information leaked by GES.ZT in Definitions 5, 6 for security.

[7] Other GES models including [20] merge GES.Setup and GES.Gen, and consider $\boldsymbol{p}_{\mathrm{zt}}$ to be a member of pp. This does not affect the security definition.

The primary interest in graded encoding schemes has been to construct functional encryption (FE) and indistinguishability obfuscation (iO). Recent work has made significant progress in reducing the multilinearity level required of a graded encoding scheme to construct these primitives [34, 35, 1].

This work weakens the assumption on the graded encoding scheme by effectively hiding the level-l^* encodings from an adversary, thereby rendering the zero-test parameter useless to such an adversary. We show in Lemma 1 and Corollary 1 that it is very unlikely that our security assumption can be used to construct either functional encryption or indistinguishability obfuscation.

Further, we observe that the GGH graded encoding with a *secret* zero-test is secure under the well-established NTRU assumption [24]. We do not quite reduce to NTRU since the adversary has access to the zero-test parameter, but we show that the adversary cannot compute a valid encoding of zero at the zero-test level.

Our assumption is similar in form to the hybrid graded encoding system proposed in [20], they model the GGH13 construction using a stateful oracle $\mathcal{M}$ with the following functions:

Initialization $\mathcal{M}$ takes the parameters of the hybrid graded encoding system. It initializes the GGH public/private parameters internally, and initializes an empty table of handle-encoding pairs and an empty table of handle-ring element pairs.

Algebraic Operations $\mathcal{M}$ takes two handles as parameters, looks up their corresponding encodings in its internal handle-encoding table. If either lookup fails, $\mathcal{M}$ fails. It performs the desired algebraic operation on the encodings. It determines if the new encoding exists in the table and returns its handle if it exists. Otherwise, it returns a new, random handle that is stored in the table with the new, computed encoding.

Zero-testing $\mathcal{M}$ takes a handle as parameter, looks up its encoding in the internal table, and fails if the handle is not found. It performs a zero-test on the encoding. If zero-test fails, $\mathcal{M}$ fails. Otherwise, $\mathcal{M}$ adds the resulting ring element to the internal handle-ring element table (if it isn't already present) and returns a handle to this element.

Post-zeroizing Computation $\mathcal{M}$ is given a bounded-degree polynomial p and a sequence of ring element handles (i.e., generated by the zero-test algorithm). $\mathcal{M}$ fails if any handle is not present in the table. Otherwise, it returns 1 if the p evaluates to 0 and returns 0 otherwise.

Definition 5 (Hybrid Graded Encoding Model). *A GES is λ-secure in the Hybrid Graded Encoding Model if an adversary, given access to the GES through the oracle $\mathcal{M}$, and the initial set of encodings, cannot generate a call to the post-zeroizing computation of $\mathcal{M}$ to return 1 with probability greater than* $\mathsf{neg}(\lambda)$.

Intuitively, the attack model assumes that (1) encodings are indistinguishable until they are zero-tested, and (2) only successful zero-tests are useful to an adversary. We weaken the definition of the oracle, which we call $\mathcal{M}'$:

– The oracle is modified such that the handles live in some ring R' (the encodings live in a ring R) where $1/|R'| = \mathsf{negl}(\lambda)$.
– Initialization, and zero-testing occur exactly as before (handles for new encodings are sampled uniformly at random in R').
– For algebraic operations, the handle of the new encoding is computed as the appropriate algebraic operation of the input handles (instead of generating new random handles each time).[8]
– For algebraic operations with invalid handles as arguments, the new handle (computed algebraically) is still returned, but no update to the internal table is made.
– For post-zeroizing computation, the oracle returns 1 if it is provided *any* valid ring-element handle.

We then modify the security definition as follows:

Definition 6 (Hidden Zero-Test Graded Encoding System). *A GES is λ-secure in the Hidden Zero-Test Hybrid Graded Encoding Model if an adversary, given the oracle $\mathcal{M}'$ and the initial set of encodings, one cannot compute a valid ring-element handle (i.e., cause the post-zeroizing computation of $\mathcal{M}'$ to accept) with probability greater than $\mathsf{neg}(\lambda)$.*

The attack model does not make any assumption regarding *which encodings are revealed* to the adversary at the beginning of the game. The security argument of the system depends on this choice. The assumption in Definition 6 does not help one to construct functional encryption.

Lemma 1. *If Definition 6 implies functional encryption for any (randomized) complexity class, then functional encryption exists for that complexity class without assuming Definition 6.*

Proof. Consider the following definition of functional encryption (for an arbitrary circuit class, arbitrary number of keys, etc.).

$$\begin{aligned} \{pk, msk\} &\leftarrow \mathsf{FE.Setup}(1^\lambda) \\ sk_k &\leftarrow \mathsf{FE.Keygen}(msk, k) \\ ct &\leftarrow \mathsf{FE.Enc}(pk, pt) \\ F(k, pt) &\leftarrow \mathsf{FE.Dec}(sk_k, ct) \end{aligned}$$

Typically one considers F to be a universal circuit and k choosing which circuit is executed on pt. Definition 6 has secret parameters (e.g., the [18] secret parameters) that enable the encoding, algebraic operations, and zero testing. We assume in Definition 6 that *any* successful use of the zero-test algorithm reveals all secret parameters, and unsuccessful zero-testing reveals nothing.

Since successful zero-test reveals all secret parameters, it must be the case that zero-test may only be used in FE.Setup and FE.Keygen, else this represents

[8] Observe that this is similar to a "generic group" idealized model.

a security break. Thus, FE.Enc and FE.Dec do not use any zero-test. FE.Setup and FE.Gen share state with FE.Enc and FE.Dec via pk and sk_k.

Further, FE.Setup and FE.Keygen cannot reveal any information about a zero-tested handle to FE.Enc or FE.Dec (or enable them to compute zero-test), since this would also constitute an immediate break of the cryptosystem by Definition 6.

Without zero-test, the only information that can be obtained about underlying encodings shared from FE.Setup, FE.Keygen to FE.Enc, FE.Dec is through testing *direct equivalence of handles*, since they are sampled at random for each encoding that is sampled.[9]

We observe that if the above functional encryption exists using the oracle in Definition 6, then functional encryption exists in the hybrid game where FE.Setup and FE.Keygen calls to the oracle are routed to Oracle 1 (the oracle from Definition 6), and FE.Enc and FE.Dec calls to the oracle are routed to Oracle 2 defined below:

Encoding Returns a random handle in R'.
Algebraic Operations Returns the appropriate algebraic operation on the handle in R'.
Zero-Testing Returns $\perp$.

We observe that FE.Enc, FE.Dec cannot statistically distinguish between Oracle 1 and Oracle 2, so the functional encryption scheme must still work in this new model.

We next observe that Oracle 2 contains no secret information, and its internal description may be revealed to the adversary. Oracle 1, on the other hand is only used in FE.Setup and FE.Gen and is not accessible to the adversary. This case is equivalent to a second hybrid where Oracle 1 is a part of the master secret key msk. In this second hybrid, internal state and description of Oracle 1 may be revealed without affecting security, as this description would be a part of msk, and not accessible to the adversary.

Therefore, by simulating the oracle with random samples and arithmetic, there are no hidden values behind each encoding and and no private parameters in Oracle 2. Since the adversary can not distinguish these two worlds, the functional encryption scheme must exist when presented with Oracle 2 and thus functional encryption exists without the original security assumption.

Corollary 1. *If Definition 6 implies indistinguishability obfuscation for* NC1*, then* LWE *implies functional encryption.*

Proof. From [19], indistinguishability obfuscation (iO) for NC1 and LWE imply iO for all circuits. Further, [19] shows that iO for all circuits implies functional encryption (using public key encryption and NIZK). Corollary immediately follows from Lemma 1.

[9] Recall that handles for arithmetic combinations of sampled encodings are also arithmetically generated from handles for the sampled encodings. This does not affect the argument.

Both of the above results seem unlikely. Significant work has been undertaken to construct both functional encryption and indistinguishability obfuscation for all circuits with little success under standard assumptions.

Another key limitation of the multilinear graded encoding assumption is that current embodiments (e.g., [18]) have asymptotic complexity (both computational and storage) that are high-degree polynomials in the security parameter, and as a result are impractical for use in modern computing systems (this work is no exception in this regard). We believe that the hidden zero-test hybrid graded encoding scheme in Definition 6 may not require the same asymptotic complexity, as the adversary cannot use zero-test successfully. However, we do not explore this idea further at this time, as it is outside the scope of this work.

4 Biometric Cryptosystems

We now create a public key encryption and signature scheme using the hybrid graded encoding scheme. First, we provide formal definitions for these constructions, allowing a noisy private key. We abstract the definitions away from biometrics and refer to the biometric as a "fuzzy source."

4.1 PKCS with Fuzzy Private Key

Definition 7 (Public Key Encryption with Fuzzy Secret Keys). *A Public Key Encryption System with Fuzzy Secret Keys (PKE-FSK) is a set of PPT algorithms:*

$$\begin{aligned} pp &\leftarrow \mathsf{Setup}(1^\lambda) \\ pk &\leftarrow \mathsf{Gen}(O, pp) \\ ct &\leftarrow \mathsf{Enc}(pt, pk, pp) \\ pt &\leftarrow \mathsf{Dec}(ct, C', pp) \end{aligned}$$

$\{O, C\}, \{O', C'\}$ *are samples from a fuzzy source (only* O *and* C' *are used).* $\{\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}\}$ *is IND-CPA secure with security parameter* λ*. The cryptosystem is* δ*-correct if:*

$$\Pr(pt = \mathsf{Dec}(\mathsf{Enc}(pt, pk, pp), C', pp) : pp \leftarrow \mathsf{Setup}(1^\lambda),\ pk \leftarrow \mathsf{Gen}(O, pp)) \geq 1 - \delta$$

where the probability is taken over $\{O, C\} \hookleftarrow \chi_{\text{Intra}}$ *and* $\{O', C'\} \hookleftarrow \chi_{\text{Intra}}$*.*

Definition 7 formalizes the notion of public key encryption with fuzzy secret keys. The success probability is defined over the distribution of $\{O, C\}$, $\{O', C'\}$ (i.e., the randomness of the fuzzy key source), and the internal randomness of Gen, Enc, and Dec. The parameter pp is a global constant for security parameter λ, while pk is the per-user public key.

For our construction an individual user may provision polynomially many public keys with a single fuzzy key source without compromising security or privacy (i.e., an adversary cannot detect whether two or more private keys were provisioned from the same individual or different individuals).

We next formalize the notion of digital signatures with fuzzy secret keys in Definition 8.

Definition 8 (Digital Signatures with Fuzzy Secret Keys). *A Digital Signature Algorithm with Fuzzy Secret Keys (DSA-FSK) is the following set of PPT algorithms:*

$$\begin{aligned} pp &\leftarrow \mathsf{Setup}(1^\lambda) \\ pk &\leftarrow \mathsf{Gen}(O, pp) \\ sig &\leftarrow \mathsf{Sign}(msg, C', pk, pp) \\ \{0,1\} &\leftarrow \mathsf{Ver}(msg, sig, pk, pp) \end{aligned}$$

$\{O, C\}, \{O', C'\}$ *are samples from a fuzzy source.* $\{\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver}\}$ *are existentially unforgeable with security parameter* λ*. The signature system is* δ*-correct if:*

$$\Pr(\mathsf{Ver}(msg, \mathsf{Sign}(msg, C', pk, pp), pk, pp) = 1 : \\ pp \leftarrow \mathsf{Setup}(1^\lambda),\ pk \leftarrow \mathsf{Gen}(O, pp)) \geq 1 - \delta$$

where the probability is taken over $\{O, C\} \hookleftarrow \chi_{\text{Intra}}$ *and* $\{O', C'\} \hookleftarrow \chi_{\text{Intra}}$*.*

Similar to PKE-FSK two samples are taken for a given signature ($\{O, C\}$ during Gen, and $\{O', C'\}$ during Sign). The success probability is defined over these random variables and over the randomness of Gen, Sign, and Ver.

5 Biometric Public Key Encryption

Our public key algorithm uses a simplified version computational fuzzy vault as a sub-step within the algorithm.

Simplified computational fuzzy vault We present the simplified algorithm in Algorithm 4. There are three key differences between Algorithm 4 and the computational fuzzy vault in the Introduction.

- $\mathsf{FV.Gen}'$ uses encodings of $\boldsymbol{a}$ instead of encodings of 0.
- $\mathsf{FV.Rep}'$ doesn't check if the high-confidence indices C_i'' are correct or recover the other indices.
- We remove the randomness extractor.

[10] In this step we are multiplying a valid encoding by a random value. This may seem pointless, as the encoding is of 0, but by multiplying by large enough value, we overflow the noise of the GES. This makes the encoding "bad." We choose this scalar product approach (as opposed to just replacing the encoding with random garbage), because it is compatible with our oracle model.

$\boldsymbol{H} \leftarrow \mathsf{FV.Gen}'(O, \boldsymbol{a}, pp, sp)$
Input: $O \in \{0,1\}^{m \times \kappa}$
Output: $\boldsymbol{H} \in R^{m \times \kappa}$

1. For $i \in [m]$, $j \in [\kappa]$
   (a) $\boldsymbol{H}_{i,j} = \mathsf{GES.Encode}(\boldsymbol{a}, l_i, sp, pp)$
   (b) If $O_{i,j} = 0$
      i. $\boldsymbol{H}_{i,j} = \mathsf{GES.SProd}(h_{i,j}, U(R), pp)$[10]
2. return $\boldsymbol{H}$

$C'' \leftarrow \mathsf{FV.Rep}'(C')$
Input: $C' \in \mathbb{R}^{m \times \kappa}$
Output: $C'' \in \mathbb{Z}_m^{\kappa}$

1. return $C''_i = \arg\max_{i \in [m]} |C'_{i,j}|$ for $j \in [\kappa]$

Algorithm 4. Modified computational fuzzy vault used as a subroutine in the Encrypt/Decrypt and Sign/Verify algorithms.

Encryption Scheme We now summarize our public key encryption scheme (the complete algorithm is described in Algorithm 5). Let R be the ring of encodings. The $\mathsf{PKE.Gen}$ procedure runs the computational fuzzy vault $\log^2 |R|$ times ($\mathsf{FV.Gen}'$), where $|R|$ is the cardinality of the ring of encodings.

Next, $\mathsf{PKE.Enc}$, encrypts a single bit as follows. If $b = 1$, it re-randomizes each of the $m \times \kappa$ elements of H by summing random subset of the $\log^2 |R|$ encodings of $H_{i,j}$. Observe that re-randomization is possible, as there are $\log |R|$ bits in each encoding, and $\log^2 |R| >> \log |R|$.

Valid encodings of 0 remain encodings of 0 during this process, while bad encodings remain bad encodings. If $b = 0$, the system further transforms all $m \times \kappa$ encodings into bad encodings by taking arithmetic combinations of these encodings. It sends this ciphertext alongside a *randomized* zero-test parameter – described below.

Finally, $\mathsf{PKE.Dec}$ runs $\mathsf{FV.Rep}$ on the ciphertext and returns 1 if it succeeds and 0 if it fails.

Zero-test Randomization An individual is able to successfully decrypt with his/her biometric is also able to construct a level-l^* zero and zero-test. Therefore, by Definition 6, this user may obtain all secret parameters of the GES.

Therefore, a given set of GES parameters must be *unique to a single user*. $\mathsf{GES.Add}$, $\mathsf{GES.Prod}$, $\mathsf{GES.SProd}$ may all be computed with universally constant public parameters pp (i.e., a description of R). In GES instances such as [18], $\mathsf{GES.ZT}$ depends on $\boldsymbol{p}_{\mathrm{zt}}$ which depends on the underlying secret parameters. Because $\boldsymbol{p}_{\mathrm{zt}}$ is required for decryption, it must be accessible to the decryptor. However, it is unique per-user, so this results in the same de-anonymization as the original fuzzy extractor construction.

Therefore, we must *randomize* the zero-test parameter. To do so we select a random invertible element r. The zero test parameter is multiplied by $\boldsymbol{r}$ using $\mathsf{GES.ZTSProd}$. We then multiply all encodings at a single level (called ℓ_α) by $\boldsymbol{r}^{-1}$ using $\mathsf{GES.SProd}$. The output of $\mathsf{GES.ZT}$ is unchanged (as it is a product of encodings by the zero-test parameter), and the new zero-test parameter $\boldsymbol{p}'_{\mathrm{zt}}$ is uniformly distributed.

$pp \leftarrow \mathsf{PKE.Setup}(1^\lambda)$

1. return $pp \leftarrow \mathsf{GES.Setup}(1^\lambda, \kappa)$

$pk \leftarrow \mathsf{PKE.Gen}(O, pp)$
Input: $O \in \{0,1\}^{m \times \kappa}$
Output: $pk \in R^{m \times \kappa \times \log^2 |R|}$

1. $\{\boldsymbol{p}_{\text{zt}}, sp\} \leftarrow \mathsf{GES.Gen}(pp)$
2. For $i \in [\log^2 |R|]$
   (a) $H_i = \mathsf{FV.Gen}'(O, 0, pp, sp)$
3. return $\{\boldsymbol{p}_{\text{zt}}, H\}$

$ct \leftarrow \mathsf{PKE.Enc}(pt, pk, pp)$
Input: $pt \in \{0,1\}$, $pk = \{\boldsymbol{p}_{\text{zt}}, H\}, H \in R^{m \times \kappa \times \log^2 |R|}$
Output: $ct = \{u, \boldsymbol{p}_{\text{zt}}\}, u \in R^{m \times \kappa}$

1. Choose uniformly random subsets $S \subseteq [\log^2 |R|]$.
2. for $i \in [m], j \in [\kappa]$
   (a) $u_{i,j} = \sum_{k \in S} H_{i,j,k}$
   (b) if $pt = 0$
   $u_{i,j} = \mathsf{GES.SProd}(u_{i,j}, U(R), pp)$
3. Choose random $\boldsymbol{r} \in R$, inverse $\boldsymbol{r}^{-1}$
4. $\boldsymbol{p}'_{\text{zt}} = \mathsf{GES.ZTSProd}(\mathbf{r}, \boldsymbol{p}_{\text{zt}}, pp)$.
5. For all $i \in [m]$
   (a) $u_{i,\alpha} = \mathsf{GES.SProd}(u_{i,\alpha}, \boldsymbol{r}^{-1}, pp)$.[11]
6. $ct = (u, \boldsymbol{p}'_{\text{zt}}, pp)$.

$pt \leftarrow \mathsf{PKE.Dec}(ct, C', pp)$
Input: $ct = \{u, \boldsymbol{p}_{\text{zt}}\}, u \in R^{m \times \kappa}$
Output: $pt \in \{0,1\}$

1. Run $C'' \leftarrow \mathsf{FV.Rep}'(C')$
2. $prod = \mathsf{GES.Prod}(\{u_{C''_1,1}, \ldots, u_{C''_\kappa,\kappa}\}, pp)$
3. if $\mathsf{GES.ZT}(prod, \boldsymbol{p}_{\text{zt}}, pp)$, return 1
4. return 0

Algorithm 5. Public Key Encryption leveraging the Computational Fuzzy Vault of Algorithm 4.

We are now ready to provide the correctness and security proofs for the procedures presented in Algorithm 5.

Lemma 2. *Let* $\{O, C\} \hookleftarrow \chi_{\text{Intra}}$*, and* $\{O', C'\} \hookleftarrow \chi_{\text{Intra}}$*, where* χ_{Intra} *is* ϵ_1, ϵ_2*-stable according to Definition 2.*

Algorithm 5 is δ*-correct – that is:*

$$\Pr(pt = \mathsf{PKE.Dec}(\mathsf{PKE.Enc}(pt, pk, pp), C', pp) : pp \leftarrow \mathsf{PKE.Setup}(1^\lambda),\ pk \leftarrow \mathsf{PKE.Gen}(O, pp)) \geq 1 - \delta$$

with $\delta = \kappa(\epsilon_1 + \epsilon_2) + \mathsf{negl}(\lambda)$*. The randomness is taken over* $\{O, C\}, \{O', C'\} \hookleftarrow \chi_{\text{Intra}}$ *and over the randomness of* PKE.Gen.

Proof. PKE.Setup, PKE.Gen, and PKE.Enc always succeed, so we consider the probability of success for PKE.Dec.

[11] We select $\alpha \in [\kappa]$ arbitrarily.

First, by Definition 2, each column of C' contains a value greater than C_{Thresh} with probability $1 - \epsilon_1$. Therefore, by the union bound, all columns of C' have an element greater than C_{Thresh} (that are then included in C'') with probability $> 1 - \kappa \cdot \epsilon_1$.

Again by Definition 2, the elements of u selected by $\mathsf{FV.Rep}'$ (i.e., C''), each have probability $1 - \epsilon_2$ of being a valid encoding of 0. Therefore, $\prod_{i \in [\kappa]} u_{C''_i, i}$ is a valid level-l^* encoding of 0 with probability $1 - \kappa \cdot \epsilon_2$. By the definition of the GES, the zero test procedure operating on a valid level-l^* encoding succeeds with probability $1 - \mathsf{negl}(\lambda)$

The probability of success of $\mathsf{PKE.Dec}$ is therefore $1 - \kappa(\epsilon_1 + \epsilon_2) - \mathsf{negl}(\lambda)$

Theorem 1. *Let $\{O, C\} \hookleftarrow \chi_{\text{Inter}}$, where χ_{Inter} has λ-lower bounded m-grouped entropy (Definition 3). Then Algorithm 5 is IND-CPA secure with parameter λ.*

Proof. The adversary chooses msg_0, msg_1, and receives an encryption ct_b of plaintext msg_b for random $b \in \{0, 1\}$.

First, note that $ct_b = \{u, \boldsymbol{p}_{\text{zt}}\}$ and $u \in R^{m \times \kappa}$ contains handles, rather than encodings, in the hybrid graded encoding scheme. Further, handles for uniformly random elements in R are (by definition) indistinguishable from the handles corresponding to "valid" encodings. Therefore, the adversary gains no information regarding the bits of O from pk or u by itself.

Any valid query to the zero-test function in $\mathcal{M}$ must be of the form:

$$\prod_{j \in [\kappa]} \sum_{i \in S_j \subseteq [m]} u_{i,j}$$

where i, j index one of the encodings in u, and for some arbitrarily chosen S_i. This is because $\mathcal{M}$ implements all of the homomorphisms, and the above expression represents the most general construction of a level-l^* encoding.

However, the above expression will fail if *any* of the $u_{i,j}$ are "invalid" (i.e., handles for uniformly random elements in R). (Here we ignore that the probability that random elements happen to be zero or that a set of random elements add or multiply to a zero element. This probability is proportional to the inverse of the size of the encoding field and negligible.)

By Definition 3, the adversary cannot choose such a set with probability greater than $\mathsf{negl}(\lambda)$. Therefore, the zero-test function will return $\perp$ with probability $1 - \mathsf{negl}(\lambda)$. As a result, A cannot compute a valid level-l^* encoding from ct_b with better than probability $> \mathsf{negl}(\lambda)$.

By the same argument, no adversary can construct a valid level-l^* encoding from pk.

Therefore, the above game using $\mathcal{M}$ is computationally indistinguishable from the game using hybrid $\mathcal{M}'$, whose zero-test function always returns $\perp$.

Since the zero-test is unavailable, the above hybrid game is statistically indistinguishable from a second hybrid where pk contains encodings of random values (still using $\mathcal{M}'$). We choose each encoding in $u \in ct$ as a random subset-sum of $\log^2 |R|$ encoding, which is much larger than the number of bits per-encoding ($\log |R| > \lambda$). Therefore, the second hybrid is statistically indistinguishable from

a third hybrid where $u \in ct$ consists of encodings of random values (decorrelated from pk).

In this third hybrid, a ciphertext for $msg = 0$ is statistically indistinguishable from a ciphertext for $msg = 1$.

Corollary 2. *Algorithm 5 remains IND-CPA secure if the adversary obtains* $\{pp\}_i$*, and* $\{pk\}_{i,j}$*, where* $pp_i \leftarrow \mathsf{PKE.Setup}(1^\lambda)$ *and* $pk_{i,j} \leftarrow \mathsf{PKE.Gen}(O, pp_i)$*.*

Proof. Follows immediately from the proof of Theorem 1, since the third hybrid can be simulated by the adversary, and therefore each pk gives zero knowledge to the adversary.

5.1 Discussion

Algorithm 5 only requires the *hidden zero-test* hybrid graded encoding assumption. An adversary cannot construct a valid level-l^* encoding, and therefore, the zero-test algorithm will with high probability never succeed.

Theorem 1 holds even without using GES.PZSProd. However, we include it in Algorithm 5, because using GES.PZSProd results in a ciphertext that cannot be attributed to any user (observe that u is already random because of the re-randomization step in PKE.Enc,and $\boldsymbol{p}_{\mathrm{zt}}$ is randomized by GES.PZSProd, so $ct = \{u, \boldsymbol{p}_{\mathrm{zt}}\}$ is also random).

The algorithm trivially extends to multi-bit messages, with each bit corresponding to a pair of encodings. This corresponds to $\kappa \cdot m \log^3 |R|$ bits per ciphertext for a graded encoding scheme in ring R. We have not explored CCA security in this section − the multi-bit encryption scheme is trivially malleable by bit-swapping.

We note that a decryption error is not detected as $\perp$ in the decryption algorithm − the decryption algorithm rather returns $pt = 0$. This is not optimal, but is easily overcome, as an error decrypting multi-bit messages would result in all message bits being locked to 0 − a condition easily detected.

As stated above, the above cryptosystem remains secure even when the user provisions multiple public keys for a single fuzzy secret. Further, the public key(s) provisioned by a user cannot be distinguished from uniform random, and cannot be correlated to an individual user. Finally, the ciphertexts are similarly indistinguishable from uniform random and are therefore *anonymous* and cannot be attributed to an individual user or public key.

6 Biometric Signatures

Next, we present the biometric digital signatures (Algorithm 8) that embodies Definition 8. Similarly to the public key encryption algorithm, the digital signature algorithm leverages the computational fuzzy vault. Also similarly to the public key encryption algorithm, the biometric signature algorithm does not use the recovered O as a key, but instead leverages the structure present in the

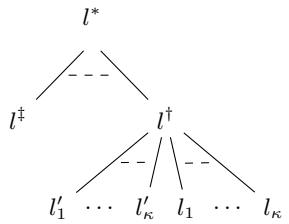

Fig. 6. Graph G of the encodings for the tree-GES used by Algorithm 8. The dashed lines between branches indicate which children encodings are paired to form the parent encoding. Observe that $l'_1, \ldots, l'_\kappa$ pair to form $l^\dagger$, $l_1, \ldots, l_\kappa$ pair to form $l^\dagger$, none of l'_i, l_j pair to form any valid label for any i, j, and $l^\dagger$ and $l^\ddagger$ pair to form l^*, the level at which the zero-test parameter works.

multilinear system to implement digital signatures without requiring entropy accumulation of the fuzzy private key *or* leveraging a secondary cryptosystem or assumption.

However, unlike biometric encryption, the signature system presented requires a the full hybrid graded encoding assumption (Definition 5). As such, this construction is already implied by the existence of VBB obfuscation under Definition 5.

However, we believe that the construction is still interesting − first, it is significantly more efficient, as it does not require circuit-level obfuscation. Further, it seems unlikely that a construction of digital signatures with fuzzy secret keys (Definition 8) must require such a strong computational assumption. The digital signature algorithm is instructive in this regard. In particular, Algorithm 8 fundamentally requires the use of the zero-test as a part of the verification algorithm, which is at odds with the hidden zero-test GES (Definition 6). This suggests that a substantially different approach would be required, and we leave this as an open problem.

Finally, this construction demonstrates several novel techniques that are possible within Definition 6 to construct interesting and unique cryptosystems beyond encryption and digital signatures.

6.1 Algorithm Description

This construction will use a tree-GES, accessed through an extension of the GES API (Definition 4): $pp \leftarrow \mathsf{GES.TreeSetup}(1^\lambda, G)$. This function takes a tree G (specifically, a description of the tree in Figure 6), and constructs an associated encoding scheme with encoding labels defined by G.

Further, it requires the use of the "zeroizing attack" discussed in [18]. We refer to this algorithm as $\boldsymbol{v} \leftarrow \mathsf{GES.Zeroize}(\{[0]_{l^\dagger}\}_i, [\boldsymbol{v}]_{l^\ddagger}, [1]_{l^\ddagger}, pp)$, where $\{[0]_{l^\dagger}\}_i$ is a set of $> \lambda$ zeros encoded at $l^\dagger$ (the level $l^\ddagger$ pairs with level $l^\dagger$ to form an encoding at level l^*). $\mathsf{GES.Zeroize}$ is shown in Algorithm 7. Observe that

$\boldsymbol{v} \leftarrow \mathsf{GES.Zeroize}(\{[0]_{l^\dagger}\}_i, [\boldsymbol{v}]_{l^\ddagger}, [1]_{l^\ddagger}, pp)$
Input: $\{[0]_{l^\dagger}\} \in R^\alpha$ for some $\alpha > \lambda$, $[\boldsymbol{v}]_{l^\ddagger}, [1]_{l^\ddagger} \in R$
Output $\boldsymbol{v} \in R$

1. for $i \in [\lambda]$
   (a) Compute $\boldsymbol{a}_i = \mathsf{GES.ZT}(\mathsf{GES.Prod}(\{[0]_{l^\dagger,i}, [1]_{l^\ddagger}\}, pp), \boldsymbol{p}_{\mathrm{zt}}, pp)$
   (b) Compute $\boldsymbol{b}_i = \mathsf{GES.ZT}(\mathsf{GES.Prod}(\{[0]_{l^\dagger,i}, [\boldsymbol{v}]_{l^\ddagger}\}, pp), \boldsymbol{p}_{\mathrm{zt}}, pp)$
2. return $\boldsymbol{v} = \mathsf{PolyGCD}(\{\boldsymbol{b}_i\}) \times \mathsf{PolyGCD}(\{\boldsymbol{a}_i\})^{-1}$

Algorithm 7. Plaintext extraction algorithm given access to λ valid encodings of 0 at level $l^\dagger$.

in the GGH construction [18] this algorithm is simply a polynomial GCD of zero-tested values. The algorithm is correct because the polynomial GCD of λ random samples from principal ideal $\mathcal{I}$ will be the generator of $\mathcal{I}$ with high probability. Therefore, in the GGH setting, where $\boldsymbol{p_{zt}} = \boldsymbol{h} \cdot (\prod_i \boldsymbol{z}_i) / \boldsymbol{g}$, it is true that $\mathsf{PolyGCD}(\{\boldsymbol{b}_i\}) = \boldsymbol{v} \cdot \mathsf{PolyGCD}(\{\boldsymbol{a}_i\})$ (where $\boldsymbol{a}$, $\boldsymbol{b}$ are as in Algorithm 7).

We leave it as an open problem whether a biometric signature scheme may be constructed using the hidden zero-test hybrid graded encoding assumption (Definition 6) and without $\mathsf{GES.Zeroize}$.

The biometric digital signature algorithm, similarly extends the modified computational fuzzy vault (Algorithm 4) with encoding levels $l_1, \ldots, l_\kappa$.

Levels $l'_1, \ldots, l'_\kappa$ are used to construct *pairs* of random encodings: $\boldsymbol{R} = \{\boldsymbol{R}_{0,i}, \boldsymbol{R}_{1,i}\}$ for $i \in [\kappa]$. There is also one additional encoding level $l^\ddagger$ (for a total multilinearity of $\kappa + 1$). It then provides one encoding $\boldsymbol{s}$ of a plaintext secret $\boldsymbol{v}$ under $l^\ddagger$.[12]

We now describe the functionality of $\mathsf{DSA.Sign}$ and $\mathsf{DSA.Ver}$. We assume that messages are first passed through a random oracle. This can be achieved by signing an encoding instead of the message. In both $\mathsf{DSA.Sign}$ and $\mathsf{DSA.Ver}$, msg is mapped to a unique level-$l^\dagger$ encoding by computing

$$\prod_{i \in [\kappa]} \boldsymbol{R}_{msg_i, i}$$

where $msg_i \in \{0, 1\}$ indicates the i'th bit of msg. Recall that $\boldsymbol{R} \in R^{2 \times \kappa}$ are random encodings, so two messages msg, msg' are unlikely to result in the same level-$l^\dagger$ encoding.

For this description, denote the level-$l^\dagger$ encoding associated with msg as $[\boldsymbol{w}]_{l^\dagger}$. $\mathsf{DSA.Sign}$ computes $[\boldsymbol{w}]_{l^\dagger}$, and using $\mathsf{GES.Zeroize}$, computes the secret level-0 encoding $\boldsymbol{v}$ from the public level-$l^\ddagger$ encoding $\boldsymbol{s} = [\boldsymbol{v}]_{l^\ddagger}$, and multiplies them to compute $sig = [\boldsymbol{w}]_{l^\dagger} \cdot \boldsymbol{v} = [\boldsymbol{w} \cdot \boldsymbol{v}]_{l^\dagger}$. Since $\boldsymbol{v}$ is a level-zero encoding (a "plaintext"), this product *does not change the encoding level.*

No one except the individual with the correct biometric can access the low-level zeros, so no one else can compute the secret $\boldsymbol{v}$. However, they can compute $[\boldsymbol{w}]_{l^\dagger} \cdot \boldsymbol{s}$, to obtain a level-$l^*$ encoding $[\boldsymbol{w} \cdot \boldsymbol{v}]_{l^*}$.

[12] In the GGH13 encoding, $\boldsymbol{v} \in R/\langle \boldsymbol{g} \rangle$

$pp \leftarrow \mathsf{DSA.Setup}(1^\lambda)$

1. Construct graph G as in Figure 6.
2. return $pp \leftarrow \mathsf{GES.TreeSetup}(1^\lambda, G)$.

$\{0,1\} \leftarrow \mathsf{DSA.Ver}(sig, msg, pk, pp)$
Input: $sig \in R$, $msg \in \{0,1\}^\kappa$, $pk = \{\boldsymbol{H}, \boldsymbol{R}, \boldsymbol{s}, \boldsymbol{y}, \boldsymbol{p}_{\text{zt}}\}$

1. $\boldsymbol{u} = \mathsf{GES.Prod}(\{\boldsymbol{R}_{msg_1,1}, \ldots, \boldsymbol{R}_{msg_\kappa,\kappa}, \boldsymbol{v}, \}, pp)$
2. $\boldsymbol{u'} = \mathsf{GES.Prod}(\{sig, \boldsymbol{y}\}, pp)$
3. return $\mathsf{GES.ZT}(\boldsymbol{u} - \boldsymbol{u'}, \boldsymbol{p}_{\text{zt}}, pp)$

$pk \leftarrow \mathsf{DSA.Gen}(O, pp)$
Input: $O \in \{0,1\}^{m \times \kappa}$
Output: $pk = \{\boldsymbol{H}, \boldsymbol{R}, \boldsymbol{s}, \boldsymbol{y}\}$ with $\boldsymbol{H} \in R^{m \times \kappa}$ and $\boldsymbol{R} \in R^{2 \times \kappa}$
Note that $\mathsf{FV.Gen'}$ is called with $\boldsymbol{a} = 0$ and uses $l_1, \ldots, l_\kappa$ to compute $\boldsymbol{H}$

1. Run $\{\boldsymbol{p}_{\text{zt}}, sp\} \leftarrow \mathsf{GES.Gen}(pp)$
2. Run $\boldsymbol{H} = \mathsf{FV.Gen'}(O, 0, pp, sp)$ (Algorithm 4).
3. for $i \in [\kappa]$
   (a) Sample $\boldsymbol{U}_{0,i}, \boldsymbol{U}_{1,i}$ uniformly at random from plaintexts (level-0 encodings).
   (b) $\boldsymbol{R}_{0,i} = \mathsf{GES.Encode}(\boldsymbol{U}_{0,i}, l'_i, sp, pp)$
   (c) $\boldsymbol{R}_{1,i} = \mathsf{GES.Encode}(\boldsymbol{U}_{1,i}, l'_i, sp, pp)$
4. Sample $\boldsymbol{v}$ uniformly at random from plaintexts (level-0 encodings).
5. Set $\boldsymbol{s} = \mathsf{GES.Encode}(\boldsymbol{v}, l^\ddagger, sp, pp)$
6. Set $\boldsymbol{y} = \mathsf{GES.Encode}(1, l^\ddagger, sp, pp)$
7. return $pk = \{\boldsymbol{H}, \boldsymbol{R}, \boldsymbol{s}, \boldsymbol{y}, \boldsymbol{p}_{\text{zt}}\}$

$sig \leftarrow \mathsf{DSA.Sign}(msg, pk, C', pp)$
Input: $C' \in \mathbb{R}^{m \times \kappa}$, $msg \in \{0,1\}^\kappa$, $pk = \{\boldsymbol{H}, \boldsymbol{R}, \boldsymbol{s}, \boldsymbol{y}\}$
Output: $dig \in R$

1. Run $C'' \leftarrow \mathsf{FV.Rep'}(C')$ (in Algorithm 4).
2. Reconstruct O by iteratively replacing one element of C'' with a different index ($< m$), and zero-testing
3. for $i \in \log^2 |R|$
   (a) Randomly choose $T_i \in \mathbb{Z}^\kappa$ such that $O_{T_{i,j},j} = 0$ for all $j \in [\kappa]$
   (b) $\boldsymbol{x}_i = \mathsf{GES.Prod}(\{\boldsymbol{H}_{T_{i,1},1}, \ldots, \boldsymbol{H}_{T_{i,\kappa},\kappa}\}, pp)$
4. Set $\boldsymbol{v} = \mathsf{GES.Zeroize}(\{\boldsymbol{x}_i\}, \boldsymbol{s}, \boldsymbol{y}, pp)$
5. $sig = \mathsf{GES.SProd}(\mathsf{GES.Prod}(\{\boldsymbol{R}_{msg_1,1}, \ldots, \boldsymbol{R}_{msg_\kappa,\kappa}\}, pp), \boldsymbol{v}, pp)$
6. Sample random subset $S \subseteq [\log^2 |R|]$
7. return $sig = sig + \sum_{i \in S} \boldsymbol{x}_i$

Algorithm 8. Digital Signatures Leveraging the Computational Fuzzy Vault.

If this signature is valid, this value must match $sig \cdot \boldsymbol{y}$ (recall $\boldsymbol{y} = [1]_{l^\ddagger}$). $\mathsf{DSA.Ver}$ does precisely this.

Finally, we observe that if the $\mathsf{DSA.Sign}$ algorithm published $sig = [\boldsymbol{w}]_{l^\dagger} \cdot \boldsymbol{v}$ directly, an adversary can compute the inverse (assuming a GGH13 encoding) of $[\boldsymbol{w}]_{l^\dagger}$ to compute $\boldsymbol{v}$. Therefore, we must randomize the encoding of sig by adding a random set of level-$l^\dagger$ zeros. This is possible because the biometric in combination with $\boldsymbol{H}$ gives $\mathsf{DSA.Sign}$ access to many level-$l^\dagger$ encodings of zero.

The proofs of correctness and security for Algorithm 8 are straightforward and are included in the appendix for completeness.

6.2 Discussion

This algorithm requires the full hybrid graded encoding assumption (Definition 5). This definition already implies VBB obfuscation, so the existence this signature system is already implied. However, this construction does not require the overhead of circuit-level obfuscation.

The full hybrid graded encoding assumption is required because the verification algorithm involves the computation of a zero at level l^*. The randomization of this zero is sufficient such that the results of the zero-test are decorrelated, which is sufficient for security in the hybrid graded encoding scheme model.

Because neither the computational fuzzy vault nor encryption schemes presented in this paper require the full assumption, it is interesting to ask whether signatures do indeed require this stronger assumption. We leave this as an open problem for now, but we do note that if such a signature scheme were to exist, it would have to be fundamentally different than the presented algorithm. Namely, such a signature algorithm must not use the zero-test parameter in the verification algorithm. As such, the only mechanism to compare encodings is through *direct equivalence* (testing equivalence of handles in the hybrid GES model).

The digital signatures algorithm uses O to identify *low-level zeros*, which are used to decode a level-$l^\ddagger$ encoding via zeroization (Algorithm 7). Further, it uses these secret low-level zeros to re-randomize encodings as seen in the public key encryption algorithm. This suggests a line of inquiry that may be of independent interest outside biometric applications − in the past, low-level encodings of zero have been avoided in constructions leveraging the noisy GES from [18] due to the resulting zeroization attacks. However, this construction hides the low-level encodings of zero and leverages zeroization attacks as a part of the security protocol. This suggests that there may still be useful methods of using low-level zeros in an obfuscated manner to construct interesting cryptographic primitives.

References

1. Ananth, P., and Sahai, A. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. http://eprint.iacr.org/2016/1097.
2. Barak, B., Shaltiel, R., and Wigderson, A. Computational analogues of entropy. In *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*. Springer, 2003, pp. 200–215.
3. Bitansky, N., Canetti, R., Kalai, Y. T., and Paneth, O. On virtual grey box obfuscation for general circuits. In *International Cryptology Conference* (2014), Springer, pp. 108–125.
4. Bonneau, J., Herley, C., Van Oorschot, P. C., and Stajano, F. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *2012 IEEE Symposium on Security and Privacy* (2012), IEEE, pp. 553–567.
5. Boyen, X. Reusable cryptographic fuzzy extractors. In *Proceedings of the 11th ACM conference on Computer and communications security* (2004), ACM, pp. 82–91.
6. Brostoff, S., and Sasse, M. Are passfaces more usable than passwords?: A field trial investigation. *People and Computers* (2000), 405–424.
7. Canetti, R., and Dakdouk, R. R. Obfuscating point functions with multibit output. In *Advances in Cryptology – EUROCRYPT* (2008), Springer, pp. 489–508.
8. Canetti, R., Fuller, B., Paneth, O., Reyzin, L., and Smith, A. Reusable fuzzy extractors for low-entropy distributions. In *Advances in Cryptology–EUROCRYPT 2016* (2016), Springer, pp. 117–146.

9. Cheon, J. H., Han, K., Lee, C., Ryu, H., and Stehlé, D. Cryptanalysis of the multilinear map over the integers. In *Advances in Cryptology – EUROCRYPT 2015* (Berlin, Heidelberg, 2015), Springer Berlin Heidelberg, pp. 3–12.
10. Coron, J.-S., Gentry, C., Halevi, S., Lepoint, T., Maji, H. K., Miles, E., Raykova, M., Sahai, A., and Tibouchi, M. Zeroizing without low-level zeroes: New MMAP attacks and their limitations. In *Advances in Cryptology – CRYPTO 2015* (Berlin, Heidelberg, 2015), R. Gennaro and M. Robshaw, Eds., Springer Berlin Heidelberg, pp. 247–266.
11. Coron, J.-S., Lee, M. S., Lepoint, T., and Tibouchi, M. Cryptanalysis of GGH15 multilinear maps. Cryptology ePrint Archive, Report 2015/1037, 2015. http://eprint.iacr.org/2015/1037.
12. Daugman, J. How iris recognition works. *Circuits and Systems for Video Technology, IEEE Transactions on 14*, 1 (January 2004), 21 – 30.
13. Daugman, J. Information theory and the iriscode. *IEEE Transactions on Information Forensics and Security 11*, 2 (2016), 400–409.
14. Delvaux, J., Gu, D., Verbauwhede, I., Hiller, M., and Yu, M.-D. M. Efficient fuzzy extraction of PUF-induced secrets: Theory and applications. *Cryptographic Hardware and Embedded Systems* (2016).
15. Dodis, Y., Reyzin, L., and Smith, A. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In *Advances in Cryptology - Eurocrypt 2004* (2004).
16. Ellison, C., Hall, C., Milbert, R., and Schneier, B. Protecting secret keys with personal entropy. *Future Generation Computer Systems 16*, 4 (2000), 311–318.
17. Fuller, B., Meng, X., and Reyzin, L. Computational fuzzy extractors. In *Advances in Cryptology-ASIACRYPT 2013*. Springer, 2013, pp. 174–193.
18. Garg, S., Gentry, C., and Halevi, S. Candidate multilinear maps from ideal lattices. In *Advances in Cryptology–EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013, Proceedings* (2013), vol. 7881, Springer, p. 1.
19. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., and Waters, B. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on* (2013), IEEE, pp. 40–49.
20. Garg, S., Miles, E., Mukherjee, P., Sahai, A., Srinivasan, A., and Zhandry, M. Secure obfuscation in a weak multilinear map model. In *Theory of Cryptography Conference* (2016), Springer, pp. 241–268.
21. Garg, S., Mukherjee, P., and Srinivasan, A. Obfuscation without the vulnerabilities of multilinear maps. Tech. rep., Cryptology ePrint Archive, Report 2016/390, 2016. http://eprint. iacr. org, 2016.
22. Gentile, J. E., Ratha, N., and Connell, J. SLIC: Short-length iris codes. In *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on* (2009), IEEE, pp. 1–5.
23. Herder, C., Ren, L., van Dijk, M., Yu, M. D., and Devadas, S. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. *IEEE Transactions on Dependable and Secure Computing*, 99 (2016).
24. Hoffstein, J., Pipher, J., and Silverman, J. H. NTRU: A ring-based public key cryptosystem. In *International Algorithmic Number Theory Symposium* (1998), Springer, pp. 267–288.

25. Hollingsworth, K. P., Bowyer, K. W., and Flynn, P. J. The best bits in an iris code. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*, 6 (2009), 964–973.
26. Hsiao, C.-Y., Lu, C.-J., and Reyzin, L. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (2007), Springer, pp. 169–186.
27. Hu, Y., and Jia, H. A comment on gu map-1. *IACR Cryptology ePrint Archive 2015* (2015), 448.
28. Hu, Y., and Jia, H. *Cryptanalysis of GGH Map*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 537–565.
29. Huth, C., Becker, D., Guajardo, J., Duplys, P., and Güneysu, T. Securing systems with scarce entropy: LWE-based lossless computational fuzzy extractor for the IoT.
30. Juels, A., and Sudan, M. A fuzzy vault scheme. *Designs, Codes and Cryptography 38*, 2 (2006), 237–257.
31. Juels, A., and Wattenberg, M. A fuzzy commitment scheme. In *Proceedings of the* 6^{th} *ACM Conference on Computer and Communications Security* (New York, NY, USA, 1999), CCS '99, ACM, pp. 28–36.
32. Kanukurthi, B., and Reyzin, L. Key agreement from close secrets over unsecured channels. In *Advances in Cryptology – EUROCRYPT* (2009), Springer, pp. 206–223.
33. Koeberl, P., Li, J., Rajan, A., and Wu, W. Entropy loss in PUF-based key generation schemes: The repetition code pitfall. In *Hardware-Oriented Security and Trust (HOST)* (May 2014), pp. 44–49.
34. Lin, H. Indistinguishability obfuscation from constant-degree graded encoding schemes. In *Advances in Cryptology – Eurocrypt* (2016), Springer, pp. 28–57.
35. Lin, H., and Vaikuntanathan, V. Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on* (2016), IEEE, pp. 11–20.
36. Maes, R., Van Herrewege, A., and Verbauwhede, I. PUFKY: A Fully Functional PUF-based Cryptographic Key Generator. In *Proceedings of the 14th International Conference on Cryptographic Hardware and Embedded Systems* (2012), CHES'12, pp. 302–319.
37. Mayrhofer, R., and Gellersen, H. Shake well before use: Intuitive and secure pairing of mobile devices. *IEEE Transactions on Mobile Computing 8*, 6 (2009), 792–806.
38. Miles, E., Sahai, A., and Zhandry, M. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. Cryptology ePrint Archive, Report 2016/147, 2016. http://eprint.iacr.org/2016/147.
39. Monrose, F., Reiter, M. K., and Wetzel, S. Password hardening based on keystroke dynamics. *International Journal of Information Security 1*, 2 (2002), 69–83.
40. Nissan, N., and Zuckerman, D. Randomness is Linear in Space. *Journal for Computer and System Sciences (JCSS) 52*, 1 (1996), 43–52.
41. Pappu, R. *Physical One-Way Functions*. PhD thesis, Massachusetts Institute of Technology, 2001.
42. Pappu, R. S., Recht, B., Taylor, J., and Gershenfeld, N. Physical one-way functions. *Science 297* (2002), 2026–2030.
43. Sahai, A., and Waters, B. Fuzzy identity-based encryption. In *Advances in Cryptology – Eurocrypt* (2005), Springer, pp. 457–473.

44. Suh, G. E., and Devadas, S. Physical unclonable functions for device authentication and secret key generation. In *ACM/IEEE Design Automation Conference (DAC)* (2007).
45. Yu, M.-D. M., and Devadas, S. Secure and robust error correction for physical unclonable functions. In *Design and Test of Computers* (2010), Institute of Electrical and Electronics Engineers.
46. Zhang, L., Li, H., and Niu, J. Fragile bits in palmprint recognition. *IEEE Signal Processing Letters 19*, 10 (2012), 663–666.
47. Zviran, M., and Haga, W. J. A comparison of password techniques for multilevel authentication mechanisms. *The Computer Journal 36*, 3 (1993), 227–237.

A A Fuzzy Extractor from a Graded Encoding

In this appendix, we review fuzzy extractors and compare the construction in Figure 1 to state of the art. The goal of a fuzzy extractor is to provide a stable (symmetric) key from a noisy source. We consider computational security for an arbitrary family of distributions as in [17]. The original definition of Dodis et al. [15] considered the family of distributions with a particular amount of min-entropy.

Definition 9. *[17, Definition 4] Let $\mathcal{W}$ be a family of probability distributions over $\mathcal{M}$. A pair of randomized procedures "generate" (Gen) and "reproduce" (Rep) is an* $(\mathcal{M}, \mathcal{W}, \ell, t)$-computational fuzzy extractor *that is $(\epsilon_{sec}, s_{sec})$-hard with error δ if Gen and Rep satisfy the following properties:*

- *The generate procedure Gen on input $w \in \mathcal{M}$ outputs an extracted string $r \in \{0,1\}^\ell$ and a helper string $p \in \{0,1\}^*$.*
- *The reproduction procedure Rep takes an element $w' \in \mathcal{M}$ and a bit string $p \in \{0,1\}^*$ as inputs. The* correctness *property guarantees that if $d(w, w') \leq t$ and $(r, p) \leftarrow \mathsf{Gen}(w)$, then $\Pr[\mathsf{Rep}(w', p) = r] \geq 1 - \delta$, where the probability is over the randomness of $(\mathsf{Gen}, \mathsf{Rep})$.*
- *The* security *property guarantees that for any distribution $W \in \mathcal{W}$, the string r is pseudorandom conditioned on p, that is, the statistical distance $\delta^{\mathcal{D}_{s_{sec}}}((R, P), (U_\ell, P)) \leq \epsilon_{sec}$.*

Most fuzzy extractors are built by combining a secure sketch which recovers the original value w and a randomness extractor [40]. This approach was shown to provide a fuzzy extractor in [15]. We roughly follow this approach in this work. We describe a weak version of a *computational secure sketch*. We refer to a measurement of a noisy value as C and the sign of C as O. A subsequent measurement of the biometric is referred to C', O' respectively. The algorithms in this work *only* require O, and C'.

Since we do not recover both variables O and C our approach actually uses a computational version of a fuzzy conductor [32]. The proof showing that a secure sketch and extractor implies a fuzzy extractor also applies for a fuzzy conductor.

Definition 10 (Computational Fuzzy Conductor). *A computational fuzzy conductor is pair of randomized functions $(X, \boldsymbol{H}) \leftarrow \mathsf{Gen}(O, 1^\lambda)$, $X \leftarrow \mathsf{Rep}(C', \boldsymbol{H})$. The pair $\{\mathsf{Gen}, \mathsf{Rep}\}$ is an efficient δ-correct, λ-secure computational fuzzy conduction for inter-key distribution χ_{Inter} and intra-key distribution χ_{Intra} if the following are true:*

- Gen *and* Rep *both run in probabilistic polynomial time.*
- Gen *outputs values $X, \boldsymbol{H}$.*
- $\mathsf{Pr}_{\{O,C\},\{O',C'\} \leftrightarrow_{\text{Intra}}}(\mathsf{Rep}(C', \boldsymbol{H})) = X) > 1 - \delta$ *where $(X, \boldsymbol{H}) \leftarrow \mathsf{Gen}(O, 1^\lambda)$.*
- *For all PPT adversary $\mathcal{A}$ there exists a negligible function ϵ with*

$$\mathsf{Pr}_{\{O,C\} \leftarrow \chi_{\text{Inter}}, (X,\boldsymbol{H}) \leftarrow Gen(O)}(\mathcal{A}(\boldsymbol{H}, \chi_{\text{Inter}}) = X) \leq \epsilon(\lambda).$$

The above definition provides computational unpredictability. Standard randomness extractors do not extract from computational unpredictability. However, reconstructive extractors [2] were shown to extract from unpredictability in [26, Lemma 6]. In the remainder of our discussion we focus on the conductor and assume that there exists a good method for converting X to a pseudorandom output.

A.1 Construction Using Hybrid Graded Encoding Attack Model

We now formalize the intuition provided in Figure 1. We focus on proving that the construction without the extractor is a good computational fuzzy conductor. We assume the fuzzy key has the properties in Definitions 2 and 3. As a reminder Definition 2 says that there exists bits where C is large and for these bits O will be 1 with high probability. Recall we view the input value $O \in \{0,1\}^{m \times \kappa}$, Definition 3 says it is, difficult to simultaneously predict a bit from each column of O.

We show that the scheme is secure in the hybrid graded encoding scheme presented in Definition 6. The Hybrid Graded Encoding System described in Section 3.2 represents a given encoding by a random "handle." All multilinear operations are performed by an oracle on these handles. We use the noisy GES model from Definition 4.

Recall that in the computational fuzzy vault 0s are translated to invalid encodings while 1s are translated to encodings of 0 at the appropriate level. The zero-test parameter is not used to test for "zero" at all. It is used to determine whether the computed encoding is indeed *valid*. The method was presented in Figure 1. We reiterate the intuition here:

- $(\boldsymbol{H}, pp) \leftarrow \mathsf{Gen}(1^\lambda, O)$: Constructs a GES with security parameter λ with multilinearity level κ (with κ unique encoding levels l_i and zero-test level l^*). It publishes the public parameters of this GES alongside $\boldsymbol{H}$. An entry of $\boldsymbol{H}$ is an encoding of 0 if the corresponding element of O is 1. If the corresponding element is 0, a random value in the encoding space is chosen. The output of Gen is $\boldsymbol{H}$ and the public parameters of the GES.
- $O \leftarrow \mathsf{Rep}(C', \boldsymbol{H}, pp)$: C' is a real-valued input. Rep chooses the maximally confident bit C''_j in each column, $C'_{*,j}$, and computes the product of these elements in $\boldsymbol{H}$. All of these elements are valid encodings of 0 with high probability. Therefore, this product is a valid level-l^* encoding of 0, so the zero-test will succeed. This is shown in Figure 3. To reconstruct O, Rep repeatedly replaces one element of C'' with a different index ($< m$), and checks if zero-testing succeeds until all elements are checked.

We show the correctness and soundness of the proposed algorithm under the hidden zero-test hybrid graded encoding model (Definition 6), and the biometric assumptions stated in Section 2.

Lemma 3. *Assuming the biometric is ϵ_1, ϵ_2-stable according to Definition 2, Figure 1 outputs O with probability $\delta = 1 - \kappa(\epsilon_1 + \epsilon_2) - \mathsf{negl}(\lambda)$.*

Proof. Gen always succeeds and generates $\{\boldsymbol{H}, pp\}$. Therefore, the success of the algorithm is determined by the success probability of Rep. By Definition 2, each column of C' contains a value greater than C_{Thresh} with probability $1 - \epsilon_1$. Therefore, by the union bound, all elements of C'' are greater than C_{Thresh} with probability $> 1 - \kappa \cdot \epsilon_1$.

Again by Definition 2, the elements of $\boldsymbol{H}$ selected by Rep (i.e., C''), each have probability $1 - \epsilon_2$ of being a valid encoding of 0. Therefore, the $\prod_{i \in [\kappa]} \boldsymbol{H}_{C''_i, i}$ is a valid level-l^* encoding of 0 with probability $1 - \kappa\epsilon_2$. By the definition of the GES, the zero test procedure operating on a valid level-l^* encoding succeeds with probability $1 - \mathsf{negl}(\lambda)$.

Finally, Rep scans through the remaining elements of $\boldsymbol{H}$ to identify valid encodings. By similar argument, this scan will detect all of the valid encodings with probability $1 - \mathsf{negl}(\lambda)$. Therefore, the overall probability of success of the secure sketch is $1 - \kappa(\epsilon_1 + \epsilon_2) - \mathsf{negl}(\lambda)$.

Lemma 4. *Let $\{O, C\} \hookleftarrow \chi_{\text{Inter}}$, where χ_{Inter} has λ-lower bounded m-grouped entropy (Definition 3). All PPT algorithms $O \leftarrow \mathsf{A}(\boldsymbol{H}, pp)$ reconstruct O with $\mathsf{negl}(\lambda)$ probability over the randomness of O and the encoding oracle.*

Proof. This follows from the same argument as in Theorem 1.

Theorem 2. *Let inter-key distribution χ_{Inter} and intra-key distribution χ_{Intra} obey Definitions 3 and 2. Then Figure 1 describes an efficient δ-correct, λ-secure computational fuzzy conductor according to Definition 10 for inter-key distribution χ_{Inter} and intra-key distribution χ_{Intra}, and for $\delta = 1 - \kappa(\epsilon_1 + \epsilon_2) - \mathsf{negl}(\lambda)$*

Proof. Follows from Lemmas 3 and 4.

Discussion The construction requires the weak hybrid graded encoding assumption (Definition 6). Further, the conductor only encodes 0s and random values in the encoding space. Therefore, it may be possible to weaken the GES assumption. Lemma 4 can be strengthened to say that the adversary learns nothing about O with overwhelming probability. This property is crucial for reusability.

Finally, we recognize that the conductor recovers O. However, this is not the only "secret" parameter recovered. Namely, access to the biometric enables the computation of low-level encodings of zero which can be multiplied to obtain a level-l^* encoding of 0. This additional information may be used *instead* of the biometric value (to directly build a fuzzy extractor). We would recommend this mode, we described the algorithm as reproducing O to be consistent with prior work and to improve exposition.

A.2 Reusability

In this section we show that the construction is "reusable." A conductor is reusable if a user can enroll their biometric multiple times without a loss of security. The problem was introduced by Boyen [5] and we adapt the definition of Canetti et al. [8].

Definition 11 (Reusable Computational Secure Sketch). *Let χ_I be a family of distributions. Let $\{\mathsf{Gen}, \mathsf{Rep}\}$ be an efficient δ-correct, λ-secure computational fuzzy conductor for all inter-key distribution $X \in \chi_I$ (and correct for the corresponding intra-key distribution). Then let $X^1, ..., X^\rho$ be ρ correlated random variables such that each $X^i \in \chi_I$. Let D be an adversary whose size is at most $\mathsf{poly}(\lambda)$. Define the following game for all $j = 1, ..., \rho$:*

- ***Sampling*** *For all $i = 1, ..., \rho$, the challenger samples $w^i \leftarrow W^i$.*
- ***Generation*** *For all $i = 1, ..., \rho$, the challenger computes $(x^i, p^i) \leftarrow \mathsf{Gen}(w^i)$.*
- ***Distinguishing*** *The advantage of D is*

$$\Pr[D(p^1, ..., p^\rho) = x^j].$$

$(\mathsf{Gen}, \mathsf{Rep})$ is ρ reusable if the advantage is at most $\mathsf{negl}(\lambda)$ for all j.

Lemma 5. *Assuming the biometric distribution for subjects* A *and* B *obey Definition 3, Figure 1 is a reusable computational secure sketch for any $\rho = \mathsf{poly}(\lambda)$.*

Proof. As before our proof is in the hidden zero-test hybrid encoding model. We show that if there exists a D that succeeds with probability $\alpha = 1/\mathsf{poly}(\lambda)$ in the reusability game then there exists a D' that succeeds in breaking the original fuzzy extractor with probability $\alpha - \mathsf{negl}(\lambda)$.

That is, assume there exists some D of size $\mathsf{poly}(\lambda)$ such that $\Pr[D(P^1, ..., P^\rho) = X^j] = \alpha$ for some $1 \leq j \leq \rho$. First note that D is presented with ρ independent oracles and the probability that a handle from one oracle is meaningful to another oracle is proportional to the inverse of the size of the encoding field. By the same argument in Theorem 1, D is unlikely to find a value where the zero test succeeds. Thus consider the following, D'.

- Input $p \in \{0,1\}^*$.
- Sample $u^1, ..., u^\rho \leftarrow \{0,1\}^{\kappa \times m}$.
- Compute $(x^i, p^i) \leftarrow \mathsf{Gen}(u^i)$ for $i = 1, ..., \rho$.
- Output $D(p^1, ..., p^{j-1}, p, p^{j+1}, ..., p^\rho)$.

By a hybrid argument, each substitution of a random value is only noticeable with negligible probability (when an adversary makes a zero-test succeed). Thus, we have

$$\Pr[D'(P^j) = W^j] \geq \Pr[D(P^1, ..., P^\rho) = W^j] - \mathsf{negl}(\lambda).$$

This contradicts the underlying security of the computational secure sketch. This completes the proof.

Notes The above proof also shows an additional property known as unlinkability. Unlinkability says it is hard to determine if two enrollments are from the same user. Reusability and unlinkability do not imply one another but our construction satisfies both properties. Also it is crucial to regenerate the parameters of the encoding in this construction. This implies that the oracles between different enrollments respond independently.

Also it might be tempting to set a single set of parameters to construct an IBE-like scheme with a single master secret key (e.g., if this is constructed using the [18] GES, this would correspond to using a single $\boldsymbol{g}$ across multiple individuals). However, there is no security guarantee in this case. Indeed, in the case of [18], knowledge of the biometric allows for the construction of low-level zeros, which allow for the computation of secret parameters via zeroization.

A.3 Prior Fuzzy Extractor Constructions

Constructions for extracting cryptographic keys from fuzzy biometric data in general falls into the theoretical framework provided by Dodis et al. in [15]. Several approaches have been proposed in the past. These approaches can be broadly classified into information-theoretic [31, 30, 15] and computational [17, 23, 8, 29] security. Approaches are summarized in Table A.3.

In general, information-theoretic techniques provide security for a broad class of probability distributions. Usually these constructions are secure for all distributions with a set amount of min-entropy (recent work has improved analysis for certain distributions [14]). However, this generality comes at a cost to security. Information-theoretic security losses often prevent use in real sources [33].

Delvaux et al. [14] showed that most coding-based secure sketches in the literature are equivalent. The variety of secure sketch constructions comes from the choice of

Construction	Correction	Security	Conditioned needed for reusability
Fuzzy Vault [31, 30, 15]	$O(\|W\|)$	$H_\infty(W)$	XOR of mult. enrollments leaks nothing
LPN/LWE [17, 23, 29]	$O(\|W\|)$	Hard to decode code w/ noise W	Each W^i derived from linear system of i.i.d bits
Digital Lockers [8]	$o(\|W\|)$	Samples of W have entropy	Security of individual enrollments (optimal)
Graded encoding This work	$O(\|W\|)$	Grouped entropy Def. 3	Security of individual enrollments (optimal)

Table 1. Current approaches to fuzzy extraction. Let W denote the input source.

code (concatenated, repetition, etc.), however these constructions are trying to find the optimal code for the setting. Thus, to provide some intuition we present the fuzzy vault construction of Juels and Sudan [30].

Fuzzy Vault The approach proposed in this paper is most reminiscent of the "Fuzzy Vault" [30]. In the original fuzzy vault, a secret is stored as a polynomial P. Interpolating points on this polynomial are published, randomly interspersed with "chaff points." The biometric allows you to identify which points are truly points on P, and which are chaff. The security argument for the original fuzzy vault is information theoretic − given a high-enough degree polynomial, the chaff points are information theoretically indistinguishable from the points interpolating P. Informally, this is because for a given set of real and chaff points, there are (statistically) many false polynomials that happen to have enough interpolating points. These false polynomials can't be distinguished from the real polynomial.

Boyen showed that reusability is unlikely to be achievable [5, Theorem 11] for information-theoretic fuzzy extractors except if the individual enrollments are correlated in specific and unrealistic ways [5, Theorem 9].

Computational Approaches Computational constructions are often tailored to specific types of distributions. Reusable fuzzy extractors have been constructed for arbitrary correlations in the computational model.

The majority of these constructions are based on the hardness of decoding random linear codes [17, 23, 29]. Their hardness comes from the hardness of learning with errors (LWE) or the learning parity with noise (LPN) problem. All of these constructions use the input value O as the noise term for the random code. They are secure if LWE/LPN is hard for this distribution. The work of Herder et al. [23] does not explicitly mention reusability but they show they construction is secure if Gen is called multiple times. Their definition implies security for multiple enrollment values. Their approach does require that each individual enrollment is derived from a linear system of independent and identically distributed bits.

The work of Canetti et al. [8] uses digital lockers [7]. These are obfuscated functions that output a multi-bit string when provided a specified input value and nothing otherwise. It is reusable for an arbitrary correlation between repeated readings. However, their approach only supports a subconstant fraction of errors.

B Proofs of Biometric Signature Correctness and Security

The proofs of correctness and security for the digital signature algorithm with fuzzy keys follows the same approach as the public key encryption and secure sketch algorithms. We begin with a proof of correctness.

Lemma 6. *Let* $\{O, C\} \hookleftarrow \chi_{\text{Intra}}$*, and* $\{O', C'\} \hookleftarrow \chi_{\text{Intra}}$*, where* χ_{Intra} *is* ϵ_1, ϵ_2*-stable according to Definition 2.*

Algorithm 8 is δ*-correct − that is:*

$$\Pr(1 = \mathsf{DSA.Ver}(\mathsf{DSA.Sign}(msg, pk, C'), msg, pk)) \geq 1 - \delta$$

with $\delta = \kappa(\epsilon_1 + \epsilon_2) + \mathsf{negl}(\lambda)$*,* $pk \leftarrow \mathsf{DSA.Gen}(O, pp)$*, and* $pp \leftarrow \mathsf{DSA.Setup}(1^\lambda, G)$*. The probability is taken over* $\{O, C\}, \{O', C'\} \hookleftarrow \chi_{\text{Intra}}$ *and over the randomness of* DSA.Gen *and* DSA.Setup*.*

Proof. By similar techniques as Lemmas 2 and 3, DSA.Sign's subroutine FV.Rep returns O with probability greater or equal to $1 - \kappa(\epsilon_1 + \epsilon_2)$.

Observe that GES.Zeroize succeeds, as $\log^2 |R| > \lambda$, as required by Algorithm 7.

Next, observe that if FV.Rep returns valid encodings, then the zero test will succeed with high probability due to the properties of the GES, as the two encodings computed by DSA.Ver are of the same value.

Lemma 7. *Let* $\{O, C\} \hookleftarrow \chi_{\text{Inter}}$*, where* χ_{Inter} *has* λ*-lower bounded* m*-grouped entropy (Definition 3).*

There does not exist a PPT Algorithm that can win the game in Definition 5 with non-negligible probability in λ *given* $pk \leftarrow \mathsf{DSA.Gen}(O, pp)$ *and* $pp \leftarrow \mathsf{DSA.Setup}(1^\lambda, G)$*, and access to an oracle that computes* $sig \leftarrow \mathsf{DSA.Sign}(msg, pk, C')$*, where* msg *is randomly chosen.*

Proof. Note that we require msg to be randomly chosen, simulating the result of a known message passed through a random oracle.

We assume the underlying encoding behind the oracle is the [18] graded encoding scheme, so we may define the encoding of $\boldsymbol{a}$ at level-i as

$$\frac{\boldsymbol{a} + \boldsymbol{g} \cdot \boldsymbol{r}}{\boldsymbol{z}_i}$$

where each element is a member of the appropriate cyclotomic field of degree n mod prime q, and $\boldsymbol{g}$, $\boldsymbol{r}$, $\boldsymbol{z}_i$ are all chosen according to [18].

Recall that in the hybrid graded encoding model of [20] (Definition 5) an adversary wins if it is able to compute a zero ring element (i.e., an encoding of the form $\boldsymbol{g} \cdot \boldsymbol{r}$ for sufficiently short $\boldsymbol{r}$). That is, after the adversary successfully runs zero-test to compute a set S of ring elements, it must compute a polynomial P on this set that returns a zero ring element.

By the same argument as in Lemma 1, the adversary cannot compute a valid $l^\dagger$ encoding from $\boldsymbol{H}$. Further, since elements of $\boldsymbol{R}$ are chosen at random, any level-$l^\dagger$ encoding computed from elements of $\boldsymbol{R}$ are also random.[13] Due to the randomness of

[13] Further, observe that elements of $\boldsymbol{R}$ may not be paired with elements of $\boldsymbol{H}$.

these encodings at level $l^\dagger$, they will not give the adversary any advantage in computing a level-l^* encoding of zero.

Each valid $\{msg, sig\}$ reveals *one* level l^* zero (recall that verification computes zero-test on Equation 1 and succeeds iff the expression is indeed an l^* encoding of zero):

$$sig \cdot \boldsymbol{y} - \boldsymbol{s} \cdot \prod_{i \in [\kappa]} \boldsymbol{R}_{msg_i, i} \tag{1}$$

We can write out the encoding using the following *secret* GGH parameters:

$\boldsymbol{g}$ Principal ideal generator zero encodings (drawn from discrete Gaussian distribution)
$\boldsymbol{z}_{l_i}$ Encoding polynomial for l_i
$\boldsymbol{z}_{l^\ddagger}$ Encoding polynomial for $l^\ddagger$
$\boldsymbol{r_s}$, $\boldsymbol{r_y}$, $\boldsymbol{r}''_{i,j}$ Randomness for encoding of $\boldsymbol{s}$, $\boldsymbol{y}$, $\boldsymbol{R}_{j,i}$ respectively (drawn from discrete Gaussian distribution).

Each encoding may then be written:

$$\begin{aligned} \boldsymbol{s} &= \frac{\boldsymbol{v} + \boldsymbol{g} \cdot \boldsymbol{r_s}}{\boldsymbol{z}_{l^\ddagger}} \\ \boldsymbol{y} &= \frac{1 + \boldsymbol{g} \cdot \boldsymbol{r_y}}{\boldsymbol{z}_{l^\ddagger}} \\ \boldsymbol{R}_{j,i} &= \frac{\boldsymbol{r}'_{i,j} + \boldsymbol{g} \cdot \boldsymbol{r}''_{i,j}}{\boldsymbol{z}_{l_i}} \end{aligned}$$

where $\boldsymbol{r}_{\text{rerand}}$ is uniquely random for each signature due to the re-randomization process in DSA.Sign. This variable is also short and distributed according to a discrete Gaussian.

The GGH encoding of Equation 1 may be written:

$$\left(\left(\boldsymbol{v} \cdot \prod_{i \in [\kappa]} \boldsymbol{R}_{msg_i, i} \right) + \frac{\boldsymbol{g} \cdot \boldsymbol{r}_{\text{rerand}}}{\boldsymbol{z}_{l^\dagger}} \right) \cdot \frac{1 + \boldsymbol{g} \cdot \boldsymbol{r_y}}{\boldsymbol{z}_{l^\ddagger}} - \frac{\boldsymbol{v} + \boldsymbol{g} \cdot \boldsymbol{r_s}}{\boldsymbol{z}_{l^\ddagger}} \cdot \prod_{i \in [\kappa]} \boldsymbol{R}_{msg_i, i} \tag{2}$$

The simplified, zero-tested value is then:

$$\boldsymbol{h} \cdot \left(\boldsymbol{r}_{\text{rerand}} + (\boldsymbol{v} \cdot \boldsymbol{r_y} - \boldsymbol{r_s}) \cdot \prod_i \boldsymbol{r}'_{msg_i, i} + \boldsymbol{g} \cdot \ldots \right) \tag{3}$$

Observe that $\boldsymbol{r}_{\text{rerand}}$ is the randomness computed by the re-randomization procedure executed in DSA.Sign, and is therefore unique and highly random per-signature.[14] In the GGH13 [18] model, $\boldsymbol{g} \cdot \boldsymbol{r}_{\text{rerand}} / \boldsymbol{z}_{l^\dagger}$ can be written as

$$\sum_{i \in S} \boldsymbol{x}_i$$

[14] In order to guarantee randomness, we observe that there must be significantly more than $\log |R|$ encodings of zero, so we choose $\log^2 |R|$.

where S is selected uniformly at random, and $\boldsymbol{x}_i$ are $l^\dagger$ encodings of zero computed from $\boldsymbol{H}$ given with knowledge of $\{O', C'\}$ as in Algorithm 8.[15]

By the same argument as the original re-randomization process in [18], $\boldsymbol{r}_{\text{rerand}}$ has high entropy and cannot be estimated by an adversary. Therefore, by the Schwartz-Zippel Lemma, an adversary has negligible probability of guessing a polynomial whose arguments are of the form in Equation 3.

Finally, recall that signatures can be added − this results (with high probability) in an invalid signature, but it is a valid encoding, so it might help an adversary to construct zero ring elements. However, note that group signatures under addition is isomorphic to groups of ring elements under addition, so without loss of generality, we consider only the ring elements generated by running $\mathsf{DSA.Ver}$ on individual signatures, not sums of signatures.

Since we have shown that an adversary cannot break the Hybrid GES scheme in Definition 5, we are now ready to show the unforgeability of signatures generated by Algorithm 8.

Theorem 3. *Let* $\{O, C\} \hookleftarrow \chi_{\text{Inter}}$*, where* χ_{Inter} *has* λ*-lower bounded* m*-grouped entropy (Definition 3).*

Algorithm 8 is existentially unforgeable by PPT adversaries − all PPT adversaries have advantage at most $\mathsf{negl}(\lambda)$*.*

Proof. Recall in $\mathsf{DSA.Setup}$, $pk = \{\boldsymbol{H}, \boldsymbol{R}, \boldsymbol{s}, \boldsymbol{y}, \boldsymbol{p}_{\text{zt}}\}$. The adversary may obtain polynomially many signatures for (adaptively chosen) messages msg_j. The adversary wins if it computes $\{msg', sig'\}$ for $msg' \notin \{msg_j\}$.

First note that since the message goes through a random oracle prior to being used in Algorithm 8, any adversary's strategy for choosing messages yields uniform random input (regardless if it is adaptive or not).

Next, note that by the same argument as in Lemmas 4 and 1, no adversary can construct a valid level-$l^\dagger$ encoding of 0 from $\boldsymbol{H}$ (since level $l'_1, \ldots, l'_\kappa$ encodings are random, and all other encodings are given at level $l^\ddagger$). Further, no adversary can construct a valid level $l^\dagger$ encoding of 0 from $\boldsymbol{R}$, as all encodings in $\boldsymbol{R}$ are uniformly random.

Therefore, following a similar argument to Lemma 1, the hybrid graded encoding scheme oracle $\mathcal{M}$ is computationally indistinguishable from $\mathcal{M}'$ whose zero-test function always returns $\perp$. This hybrid game is statistically indistinguishable from a second hybrid where the encodings in $\boldsymbol{H}$ are chosen uniformly at random. Finally, an adversary with access only to $\{\{msg, sig\}_l, \boldsymbol{R}, \boldsymbol{y}, \boldsymbol{s}$ can simulate accesses to $\boldsymbol{H}$.

Due to the Hybrid GES model, the only way to combine signatures with each other or with the other public parameters is through multiplication/addition/scalar multiplication. Because msg is random when input to Algorithm 8, any arithmetic combinations of signatures with each other or other public parameters are invalid with high probability.

[15] Observe that the same $\log |R|$ encodings of zero at level $l^\dagger$ may be computed deterministically during $\mathsf{DSA.Sign}$ once O is recovered. Further, if O does not contain sufficient bits to construct $\log |R|$ encodings of 0, O may be augmented with random bits that are never chosen in $\mathsf{FV.Rep}'$ may then be used to construct more encodings of 0.