Generating Secret Keys from Biometric Body Impedance Measurements

Marc Roeschlin
Dept. of Computer Science
University of Oxford
marc.roeschlin@cs.ox.ac.uk

Ivo Sluganovic
Dept. of Computer Science
University of Oxford
ivo.sluganovic@cs.ox.ac.uk

Ivan Martinovic
Dept. of Computer Science
University of Oxford
ivan.martinovic@cs.ox.ac.uk

Gene Tsudik
Dept. of Computer Science
University of California, Irvine
gts@ics.uci.edu

Kasper B. Rasmussen
Dept. of Computer Science
University of Oxford
kasper.rasmussen@cs.ox.ac.uk

ABSTRACT

Growing numbers of ubiquitous electronic devices and services motivate the need for effortless user authentication and identification. While biometrics are a natural means of achieving these goals, their use poses privacy risks, due mainly to the difficulty of preventing theft and abuse of biometric data. One way to minimize information leakage is to derive *biometric keys* from users' raw biometric measurements. Such keys can be used in subsequent security protocols and ensure that no sensitive biometric data needs to be transmitted or permanently stored.

This paper is the first attempt to explore the use of human *body impedance* as a biometric trait for deriving secret keys. Building upon Randomized Biometric Templates as a key generation scheme, we devise a mechanism that supports consistent regeneration of unique keys from users' impedance measurements. The underlying set of biometric features are found using a feature learning technique based on Siamese networks. Compared to prior feature extraction methods, the proposed technique offers significantly improved recognition rates in the context of key generation.

Besides computing experimental error rates, we tailor a known key guessing approach specifically to the used key generation scheme and assess security provided by the resulting keys. We give a very conservative estimate of the number of guesses an adversary must make to find a correct key. Results show that the proposed key generation approach produces keys comparable to those obtained by similar methods based on other biometrics.

1. INTRODUCTION

Measuring the intrinsic electrical property of the human body, also known as body impedance, has been proposed as an effective physiological biometric trait for both user authentication and identification [7, 8, 23]. Due to its unobtrusiveness, body impedance is well-suited for a wide range of scenarios. Given that the user is in physical contact with two conductive surfaces, multiple biometric measurements can be acquired rapidly and without additional user participation. Thus, embedding the measurement apparatus into a variety of devices can support the use of body impedance in many security contexts, as a means of primary or supplementary authentication. Examples include: customer authentication on ATMs equipped with conductive PIN-pads, car driver authentication via conductive steering wheels, and continuous user authentication with conductive keyboards.

In terms of pros and cons, body impedance is similar to most other biometrics. While biometrics offer usability advantages over authentication schemes based on secrets or possession of items, arguably their main drawback is immutability—once a biometric is compromized, it can not be changed or replaced. Two recent examples are a leak where hackers stole 5.6 million fingerprints of United States federal employees [13], and revelations about insecure storage of fingerprint authentication data by a major mobile phone manufacturer [12]. This is particularly worrying because the research community has shown that original biometric input can be reconstructed from such stored reference data (i.e., the biometric template), if not adequately protected [11, 31].

Many biometric authentication systems therefore resort to storing biometric templates on a Trusted Platform Module or in a Trusted Execution Environment to prevent unauthorized access. In an ideal situation, however, biometric templates could be protected in a way similar to best practice in password-based user authentication. When handling passwords, it is the *de facto* standard to store only hard-to-invert, uniquely salted hashes; anything else is considered unfit for secure authentication.

Biometric template protection [18]—transforming raw biometric input into a representation which does not leak any sensitive information if acquired by an adversary—is an intricate problem and remains one of the main obstacles to widespread deployment of biometric authentication. One approach to template protection is *biometric key generation*, whereby raw biometric measurements are used to consistently generate the same unique value for each individual user. This allows strong biometric data protection, since in a case of compromise, an adversary can not infer information about the user from a stored key or template. Also, biometric keys

WPES'16, October 24 2016, Vienna, Austria

ISBN 978-1-4503-4569-9/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2994620.2994626

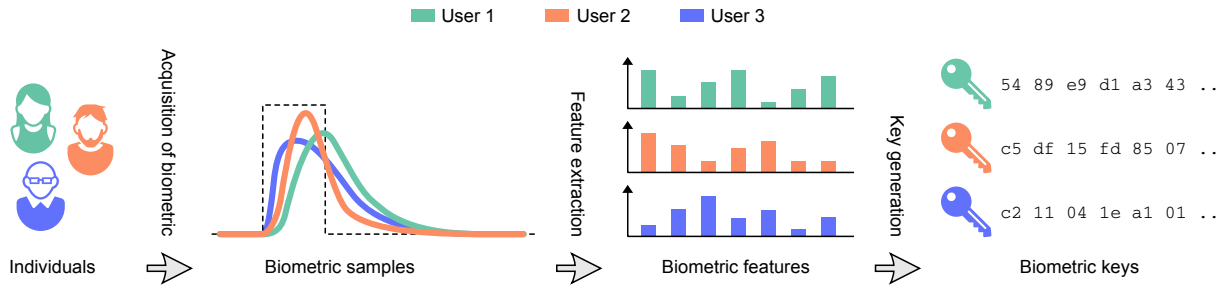

Figure 1: Procedure for key derivation from body impedance.

can alleviate humans from remembering strong secrets, since the keys can be used to either add entropy to an existing cryptographic scheme or serve as a replacement for password- or token-based authentication.

Unfortunately, generating consistent biometric keys from volatile biometric data is a challenging task, both in terms of achieved performance, as well as applying and tailoring existing methods to different biometric modalities. While prior work investigated using body impedance in the context of user authentication and identification [7, 8, 23], the issue of deriving biometric keys or otherwise protecting this biometric has been unexplored so far, despite being an important step towards practicality. This paper represents the first attempt to derive keys and secure biometric templates from body impedance data. It makes the following contributions:

- As a basis for deriving biometric keys, in Section 3, we implement and compare two methods for extracting stable and idiosyncratic features: (1) an approach based on *feature learning* using multilayer convolutional networks in a Siamese architecture, and (2) feature extraction based on frequency analysis.
- In Section 4, we improve the key evaluation approach of Ballard et at. [2] to analyze strength of derived biometric keys, and discuss security guarantees of key generation based on body impedance.
- Finally, our experimental evaluation in Section 5 shows that the electrical property of the human body can be used to derive biometric keys and sets a baseline for future work in body impedance-based key generation methods.

2. BIOMETRIC KEY GENERATION

In this paper, we design and implement a procedure to extract consistent secrets, *biometric keys*, from body impedance measurements and experimentally assess security provided by such keys in the context of protecting user's sensitive biometric data. Stated simply, our goal is to remove the need for storing an individual's biometric measurements or any derivative representation that an adversary could use in case of a breach. To this end, we aim to generate a secret value from the raw biometric measurements and subsequently only use the derived biometric key for user authentication or other protocols.

Before describing our approach, we overview previous work in biometric template protection techniques and state general goals and requirements for biometric key generation schemes.

2.1 Background

Biometric template protection techniques can be divided into two main groups [26–28, 30]: biometric cryptosystems and cancellable biometrics. The latter focus on (non-invertible) transforms on biometric features, which allows privacy-preserving template matching of biometric samples. On the other hand, biometric cryptosystems rely on generating (or binding) biometric keys by storing user-specific biometric information as a secure biometric template, in order to retrieve or generate keys.

Randomized Biometric Templates (RBT) [1]—the key generation scheme we use in this work—falls into the subcategory of key-generating schemes, along with other methods such as fuzzy extractors [9] and secure sketches [22, 33]. RBT builds upon quantization schemes which have been successfully applied to different biometric modalities, including iris [29], face [33], voice [4], fingerprint [21] and handwriting [1, 10, 36]. For more background information, we refer to a comprehensive survey of biometric template protection methods [30].

2.2 Requirements and Goals

A biometric key generation scheme is considered sound if it is: *correct* and *secure* [18].

To achieve correctness, the scheme must output consistent keys, allowing a legitimate user of the biometric to provide a recent biometric measurement and recreate the biometric key established during enrollment. The rate of failure can be measured by the false reject rate: the percentage of key generation trials that do not lead to the original key.

To be considered secure, the following three necessary requirements need to hold [2, 18]:

1. **Biometric Uncertainty:** measurement values used to generate the key (i.e., biometric input) must be difficult to guess.
2. **Key Randomness:** the resulting key must appear random.
3. **Irreversibility:** it must be difficult to deduce information about the biometric from the generated key, the template and any auxiliary data.

We further elaborate on these requirements in Section 4 and experimentally validate our proposed key generation scheme against them in Section 5.

3. PROPOSED APPROACH

The proposed approach, overviewed in Figure 1, consists of three parts: *1)* acquisition of biometric measurements, *2)* extraction of a stable feature-set, and *3)* derivation of a biometric key from that set. We describe each step in detail.

3.1 Acquisition of Impedance Measurements

We use pulse-response [23] as an instance of a particular body impedance biometric. Pulse-response measures body impedance from one hand to the other and is acquired by sending a low-voltage electric signal (in the form of a short square pulse) from the palm of one hand to the other. When the signal travels through the human body, it is affected and modified by the body's impedance, yielding a unique signature (see Figure 1) which was shown to be sufficient for user authentication and identification [7, 8, 23].

Impedance measurements are imperceptible and medically safe to humans. The voltage of the emitted signal is sufficiently low (1V for pulse-response) that the resulting current is on the order of $10 - 100\mu$A, less than the current flow induced by touching the terminals of a standard 1.5V battery. This is well below the human sensation threshold and does not pose health implications, even in case of long term exposure. For a more detailed discussion about ethics and user safety we refer to [23].

The electrodes, which emit the pulse and receive the altered response signal, can be realized as conductive handles, or incorporated into different front-end devices, e.g., conductive keyboards, PIN-pads, handlebars, steering wheels or push plates on doors. Pulse-response is fast to acquire and unobtrusive. Measurements can be acquired continuously, as long as the user maintains hand contact with two electrodes. These features make pulse-response suitable for many scenarios and a good candidate to serve as a basis base for our analysis. However, we believe that the proposed method generalizes to other body impedance biometrics.

3.2 Feature Extraction

Feature extraction is the process of transforming raw input data into a set of intermediate feature values which serve as input to subsequent machine learning tasks, such as classification, identification, or (as in our case) key derivation. For successful key derivation, features extracted from the underlying biometric must be: (1) *stable*, i.e., maximally similar for multiple measurements of the same individual, and (2) *idiosyncratic*, i.e., maximally distinct for different users. One traditional approach to feature extraction is to have domain experts manually *engineer* a set of potentially strong features. Feature values are then generated on a subset of data, to allow selection of those features likely to fit their intended purpose.

Feature Learning.

We follow a relatively new approach called *feature learning* that is currently at the forefront of state-of-the-art in machine learning domains, such as speech recognition [16], machine translation [34], and image recognition, where computer performance is for the first time surpassing those of humans [15]. The main advantage of feature learning is in automating manual, labor-intensive and biased tasks of feature engineering using multilayer convolutional networks as feature extraction models. Parameters of these networks are programmatically optimized along clear optimization criteria during a feature learning stage. As a result, the network *learns* a hierarchical nonlinear representation which can map raw input data into a lower-dimensional feature space with desired characteristics [5].

In order to find an optimal mapping, we use a *Siamese* architecture [3, 5], shown in Figure 2, where two identical

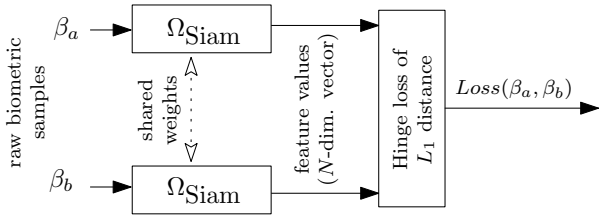

Figure 2: Siamese architecture: Two copies of convolutional networks Ω_{Siam} share internal parameters. They are trained on pairs of inputs (β_a, β_b) by optimizing the Hinge loss of the distance of feature value outputs $\Omega_{\text{Siam}}(\beta_a)$ and $\Omega_{\text{Siam}}(\beta_b)$.

convolutional networks Ω_{Siam} share the same set of parameters. During optimization, the networks accept pairs of raw biometric inputs β_a and β_b (belonging to users a and b) and compute feature values as their outputs $\Omega_{\text{Siam}}(\beta_a)$, $\Omega_{\text{Siam}}(\beta_b)$. Ideally, feature values should be similar (have small distance) if the inputs belong to the same user, and sufficiently different (distance of at least m) if they belong to different users. Such an optimization criterion can be directly represented with a Hinge loss function applied to a distance measure:

$$Loss(\beta_a, \beta_b) = \begin{cases} |\Omega_{\text{Siam}}(\beta_a) - \Omega_{\text{Siam}}(\beta_b)| & \text{if } a = b \\ \max\left(0, m - |\Omega_{\text{Siam}}(\beta_a) - \Omega_{\text{Siam}}(\beta_b)|\right) & \text{if } a \neq b \end{cases}$$

We optimize the shared parameters of convolutional networks Ω_{Siam} to minimize the loss function of the described Siamese architecture using iterative stochastic gradient descent (SGD) [16]. Pairs of raw biometric measurements are randomly sampled from a pool of data reserved for training (70% of the dataset), and the remainder of the dataset is used to estimate the Hinge loss on unseen data. We run iterative SGD until each pair of samples has been encountered five times on average.

The number of features extracted by Ω_{Siam} is varied by parameter N that specifies the dimensionality of the resulting nonlinear representation, i.e., a N-dimensional vector. Further details about the structure of used multilayer convolutional network Ω_{Siam} are given in Appendix A.

Baseline Feature Extraction Method.

As a baseline for comparison, we also implemented the feature extraction method used in the paper that introduced pulse-response as a biometric [23]. That feature extraction approach, denoted by Ω_{FFT}, consists of transforming the raw input signal to the frequency domain by computing the **Fast Fourier Transform (FFT)** to obtain frequency components. We compare performance of the two feature extraction methods in Section 5.3 and show that Ω_{Siam} outperforms Ω_{FFT} by a considerable margin.

The intuition behind strong performance of Ω_{Siam} can be found in previous work, which has shown that multilayer convolutional networks generalize remarkably well on many tasks [15,34], even to the level of supporting *transfer learning*, where the features are trained on one task, and then slightly adapted and used on another [37]. Unlike other dimensionality reduction techniques, such as Linear Discriminant

Algorithm 1: Enroll

Global: features $\phi_1, \ldots, \phi_N$, quantization widths $\delta_1, \ldots, \delta_N$
Input: enrollment samples $\beta_1, \ldots, \beta_j$, PIN π
Output: key K, template T_U, token z

$\Psi \leftarrow \mathsf{SelectStrongFeatures}(\beta_1, \ldots, \beta_j)$
$\Delta \leftarrow [\mathsf{Permute}\{\phi_i \in \Psi\},\ \mathsf{Permute}\{\phi_i \notin \Psi\}]$
$T_U \leftarrow [\,],\ \kappa \leftarrow \epsilon$
foreach ϕ_i in the order of Δ **do**
  $\mu_i \leftarrow \mathsf{Median}\{\phi_i(\beta_1), \ldots, \phi_i(\beta_j)\}$
  $\alpha_i \leftarrow \lfloor \mu_i - \delta_i/2 \rfloor \bmod \delta_i$
  $c_i \leftarrow (\ \mathsf{Enc}_1^\pi(i), \mathsf{Enc}_2^\pi(\alpha_i)\ )$
  $T_U \leftarrow T_U.\mathsf{Append}(c_i)$
  if $\phi_i \in \Psi$ **then**
    $x_i \leftarrow \lfloor \mu_i - \delta_i/2 \rfloor$
    $\kappa \leftarrow \kappa \parallel i \parallel x_i$
$z \leftarrow \mathsf{Hash}^{\text{token}}(\ \pi \parallel \kappa\ )$
$K \leftarrow \mathsf{Hash}^{\text{key}}(\ \pi \parallel \kappa\ )$
return K, T_U, z

Figure 3: Simplified enrollment procedure of Randomized Biometric Templates. Enroll outputs the user's biometric template and biometric key. ϵ denotes the empty string and $\parallel$ represents string concatenation.

Algorithm 2: KeyGen

Global: features $\phi_1, \ldots, \phi_N$, quantization widths $\delta_1, \ldots, \delta_N$
Input: biometric sample β, PIN π, template T_U, token z
Output: key K or *nil* on failure

$\kappa \leftarrow \epsilon$
foreach c as appearing in T_U **do**
  $i \leftarrow \mathsf{Dec}_1^\pi(c[0])$
  $\alpha_i \leftarrow \mathsf{Dec}_2^\pi(c[1])$
  $x_i \leftarrow \phi_i(\beta) - (\ (\phi_i(\beta) - \alpha_i) \bmod \delta_i)$
  $\kappa \leftarrow \kappa \parallel i \parallel x_i$
  if $\mathsf{Hash}^{\text{token}}(\ \pi \parallel \kappa\ ) = z$ **then**
    $K \leftarrow \mathsf{Hash}^{\text{key}}(\ \pi \parallel \kappa\ )$
    return K
return *nil*

Figure 4: Simplified key generation procedure of Randomized Biometric Templates. KeyGen outputs the user's biometric key if token z can be regenerated.

Analysis, Siamese architecture allows us to train the feature extraction model on pairs of training data, thus increasing the number of different inputs seen during training. The model never explicitly takes the number of different classes into account and is therefore well-suited for learning features in open set applications, such as biometric key generation.

3.3 Key and Template Generation

We now overview Randomized Biometric Templates [1] (RBT), the key generation scheme used in this paper. The scheme performs error-correction by quantizing the values of features. It defines two procedures: Enroll derives the randomized biometric template from a set of biometric measurements, and KeyGen uses the biometric template to recreate the key from a single biometric sample. Figures 3 and 4 shows a simplified version of these two procedures in pseudocode. Terms "sample" and "measurement" are used interchangeably.

Enroll.

Taking into account that the same feature can have different variability across individuals, Enroll determines which features can be used for consistent key generation and therefore should be included in the set of "strong" features Ψ for a particular user. Then, Enroll computes necessary information to perform error-correction and encodes it in a user-specific *randomized biometric template*.

As input, Enroll requires a set of user's biometric measurements $\beta_1, \ldots, \beta_j$, user's PIN π, access to the biometric features $\phi_1, \ldots, \phi_N$, and the corresponding global quantization widths for each feature $\delta_1, \ldots, \delta_N$. As a first step, Ψ is computed using the function $\mathsf{SelectStrongFeatures}(\ldots)$, which is overviewed in Figure 5; a feature ϕ_i is considered "strong" for a particular user if the range of its values for all enrollment samples $(\beta_1, \ldots, \beta_j)$ is smaller than δ_i, the global width of quantization for feature i. Only "strong" features are used to generate user's biometric key. This ensures that the user can reproduce a key at a later time and results in reduced false reject rates.

After Ψ is determined, Enroll reorders feature indices such that all "strong" features are located before all other features and then both parts are pseudorandomly permuted:

$$\Delta \leftarrow [\mathsf{Permute}\{\phi_i \in \Psi\},\ \mathsf{Permute}\{\phi_i \notin \Psi\}]$$

Enroll continues to calculate a quantization offset α_i for every feature i. Each α_i is computed such that the median μ_i of the feature values collected during enrollment $\phi_i(\beta_1), \ldots, \phi_i(\beta_j)$ falls in the middle of the corresponding quantization interval.

Finally, the user's biometric template T_U consists of a list of all N pairs of feature indices and corresponding quantization offsets (i, α_i) in the order as they appear in Δ. The pairs are encrypted to $c_i \leftarrow (\mathsf{Enc}_1^\pi(i), \mathsf{Enc}_2^\pi(\alpha_i))$ by two different pseudorandom permutations, Enc_1^π and Enc_2^π, determined from the user's PIN π.

Given only T_U with a permutation of pairs (i, α_i), KeyGen does not know the size of the set of "strong" features Ψ. In order to encode the number of "strong" features, a token z is generated by Enroll to serve as a stopping criterion for KeyGen. z is computed as a hash of the concatenation of feature indices i and quantized feature values x_i of all "strong" features using hash function $\mathsf{Hash}^{\text{token}}$.

KeyGen.

The key generation algorithm KeyGen requires template T_U, a token z, PIN π, and a new biometric measurement β as input, and outputs a user's biometric key K. After decrypting the template with the PIN (i.e., reversing Enc_1^π and E_2^π), KeyGen computes the quantized feature values for the given β. It subtracts the quantization offsets α_i from the feature values $\phi_i(\beta)$ and uses the global quantization widths δ_i to reconstruct the corresponding quantized value x_i for each feature. The final biometric key K is constructed by applying $\mathsf{Hash}^{\text{key}}$ to the concatenation of i and x_i for all "strong" features according to the order of how they appear in the decrypted template.

Recall from the description of Enroll that KeyGen does not know the number of features in Ψ, so it iteratively computes

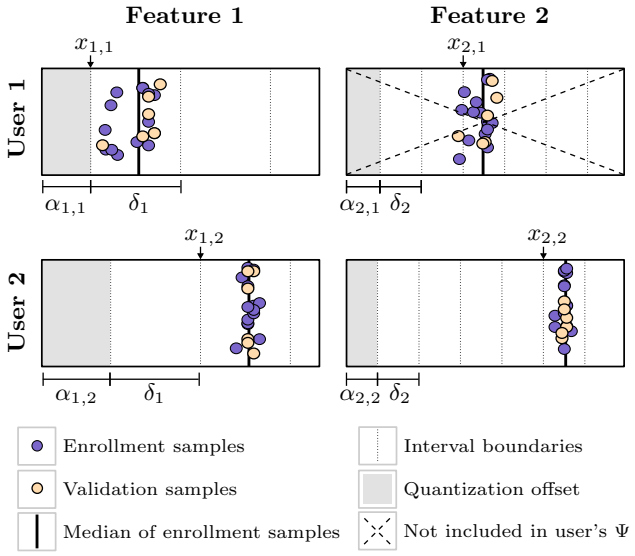

Figure 5: Example of quantization and feature assignment for two users and two features. The horizontal position represents raw feature values for enrollment and validation samples. Samples are distributed vertically to improve readability. $\alpha_{i,j}$ denotes the quantization offset of feature i for user j. $x_{i,j}$ represents the quantized value of feature i for user j. Feature 2 is not included in the set of "strong" features (Ψ) for user 1 since the spread of enrollment samples is higher than the quantization width (δ_2).

the value of the temporary token z' with more and more features, until the values of tokens collide: $z' = z$. It is important to note that even though K and z are generated as hashes of the same values, the scheme uses different hash functions $\mathsf{Hash}^{\text{token}}$ and $\mathsf{Hash}^{\text{key}}$ to ensure that no information about K can be deduced from z.

Using z as the stopping criterion provides additional benefit against the adversary who does not have the user's PIN since they have no way of knowing neither which features, nor in which order should be included when calculating the user's biometric key. Finally, this is a protection even against an adversary who is in possession of user's template and PIN, as he will still not know how many features need to be included when trying to guess the key. We discuss this in Section 4.2.

Quantization Widths δ_i and Level of Quantization k.

The values of δ_i are controlled by the level of quantization k. Since they are global values used in generation of all users' biometric templates which should be representative of the overall population statistics, we estimate them on a subset of data from multiple users. δ_i for each feature is determined by selecting k-th percentile of the spread of the feature's values. Choosing the k-th percentile for δ_i results in quantization intervals that are expected to quantize feature values of k% of the population consistently.

As a consequence, k controls the granularity of quantization: the larger k is, the more features on average are found to be "strong" for a particular user. This affects the chances of consistently replicating the biometric key in two ways. On one hand, as the error-correction intervals increase, KeyGen is more likely to quantize a feature value properly, but on the other hand, since more features are considered "strong" and used in key generation, the chance of one of the feature values falling into the wrong quantization bucket increases. Finally, larger δ_i results in a lower total number of buckets for a specific feature, which can reduce the number of guessing attempts for an adversary who attempts to directly guess the quantized feature values.

4. SECURITY GUARANTEES

We now analyze the proposed scheme against security requirements stated in Section 2.

4.1 Key Randomness and Irreversibility

As mentioned earlier, we use Randomized Biometric Templates (RBT) which ensures key randomness and irreversibility by construction, under the assumption that biometric uncertainty holds. We now outline our main arguments why key randomness and irreversibility hold and refer to [1] for further details.

Even assuming that the biometric template T_U, the token z and the key K are known, an adversary can not infer any information about biometric input a user provided to Enroll during enrollment. This is due to the fact that without knowing the correct PIN π that decrypts the template, an adversary can not use the quantization offsets α_i in T_U, as he does not know to which features they correspond[1], and the offsets reveal no information about quantized feature values x_i. Hence, in order to learn any information about the biometric, an adversary can only guess π and the biometric input β simultaneously, by running KeyGen on the guessed values π' and β'. It can verify the guesses by checking whether either the resulting key K' corresponds to K or the computed token z' matches z. Thus, we conclude that if the combined entropy of the biometric and the PIN is sufficiently high, the key generation scheme is irreversible for a computationally bounded adversary.

Similarly, we argue that the key generation scheme yields keys indistinguishable from random. Input to the cryptographic hash function used to compute $K = \mathsf{Hash}^{\text{key}}(\dots)$ consists of π, which is unknown to the adversary, and quantized feature values, which can only be inferred by enumerating all possible combinations of π and β, as described above. Therefore, a computationally bounded adversary can not distinguish between K and random, even if T_U and z are known.

However, recall that key randomness and irreversibility hinge on biometric uncertainty, i.e., difficulty of predicting biometric input. We focus on it in the following section.

4.2 Biometric Uncertainty

The most important requirement for a biometric key generation scheme is that the used biometric exhibits high unpredictability. In order to assess the unpredictability provided solely by the biometric input, one must assume that all other parameters of the key generation scheme, i.e., the user's template (including the token) and PIN, are known. This residual unpredictability can be estimated by computing measures such as Shannon entropy or min-entropy. However,

[1]This also assumes that the set of "strong" features Ψ for user U is a random subset of all features.

estimating conditional entropy in a high-dimensional space is usually infeasible due to lack of sufficient data. Moreover, entropy estimation might not capture the variety of different approaches an adversary could pursue to predict the biometric input, such as impersonation attempts or guessing attacks, since entropy measures are summary statistics and do not reflect concrete strategies.

It is therefore important to consider different types of adversaries when quantizing unpredictability of a biometric. We distinguish between *imposters* and *algorithmic adversaries*. To show that body impedance can withstand both of these adversaries, we verify whether the biometric is resistant against impersonation and automated searches, i.e., guessing biometric input in a structured way.

Resistance against Imposters.

In order to test how well body impedance withstands impersonation attempts, we measure FAR – the probability that the key generation scheme outputs the correct biometric key based on forged biometric material. The lower the FAR, the more secure the key generation scheme. We report FAR for different combinations in the parameter space (N, k) in Section 5.4. For certain configurations, the average FAR is as low as 1.9% when measured over the entire test subject population. For comparison, biometric key generation based on voice characteristics achieves FAR between 0.0% and 10.0% (see Section 6).

Guessing Raw Biometric Data.

We assert that, for an adversary in possession of a user's decrypted biometric template, the corresponding token, and the PIN, it is always more beneficial to guess the intermediate set of quantized feature values, than to guess raw biometric measurement data. This is due to the feature extraction process, which strongly reduces the dimensionality of biometric measurements by several magnitudes and arrives at a smaller set of intermediate features which are used in key generation. Taking into account that the final biometric key is a result of a cryptographic hash function, the adversary would also prefer guessing intermediate feature values over trying to guess the key directly. We therefore focus our analysis of guessing attacks on an adversary that tries to predict the inputs to the hash function, i.e., quantized feature values x_i of RBT.

Guessing Feature Values.

In the remainder of this section, we estimate the effort of the adversary in guessing a set of quantized feature values x_i, that, combined with the legitimate user's template and PIN (which are at the attacker's disposal), result in generating the correct biometric key. Similar analysis can be made for a scenario where the adversary has the user's biometric template, but not the PIN. In that case, by construction of RBT, the adversary must simultaneously guess the correct PIN and feature values. This increases guessing complexity by a factor proportional to the number of PIN guessing attempts.

To estimate adversarial effort in guessing correct feature values, we adopt and improve the probabilistic search approach described by Ballard et al. [2]. Their approach assumes that an estimate of the strength of biometric keys can be given as $\log_2(\,\mathrm{RunTime}(\mathcal{A})\,)$ where $\mathcal{A}$ is an algorithm that repeatedly guesses the biometric key until it succeeds, with run-time proportional to the number of guesses. The algorithm has access to the user's template (including token), the PIN π used to encrypt it, and a population statistic P_{ϕ_i} for each feature ϕ_i measured over all the feature values for several other users (not including the user under attack).

The algorithm starts by sorting the features according to their information content and then traverses them in that order by recursively guessing values for each feature ϕ_i using the population statistic P_{ϕ_i}. It follows this *depth-first search* pattern until the search space is fully exhausted or one of the guessed feature value combinations yield the correct key.

Considering that the original algorithm was developed to estimate key strength of arbitrary key generation schemes, it does not take into account the fact that in RBT, the "strong" features are stored in user's template before other features. Exploiting this insight, an adversary that is attacking RBT can take a *breadth-first search* approach to deploy a more efficient key guessing strategy, which results in a successful guess in considerably less tries. We describe such improved key guessing strategy in the remainder of this section and then use it in Section 5 to compute an estimate of the security guarantees provided by the biometric keys.

4.3 Improved Key Guessing Strategy

We improve the approach from [2] and tailor it specifically to RBTs by ensuring that the adversary never attempts to guess the values of those features which are not considered "strong" for a specific user and hence not included in the resulting key. Our approach thus gives a more conservative estimate of security of the evaluated key generation scheme. In Section 5.5, we experimentally show that the original approach significantly overestimates the number of guessing attempts. Our guessing strategy reduces – on average – the number of guesses by a factor of 2^{20}, for almost half of the keys in the dataset.

We now describe the improved key guessing strategy:

Guessing the number of "strong" features ($|\Psi|$).

Even if the adversary has a user's template and the corresponding PIN π, he can only obtain partial knowledge about the user's set of "strong" features Ψ. This is due to how the Enroll algorithm of RBT generates the template T_U. Enroll shuffles the order of the features before they are stored in T_U, keeping apart "strong" features and features used for padding. It then encrypts the index i and quantization offset α_i of every feature:

$$T_U = [\ \{(\mathsf{Enc}_1^\pi(i), \mathsf{Enc}_2^\pi(\alpha_i)) : \phi_i \in \Psi\},\\ \{(\mathsf{Enc}_1^\pi(i), \mathsf{Enc}_2^\pi(\alpha_i)) : \phi_i \notin \Psi\}\ ]$$

Recall from Section 3 that z serves as a stopping criterion for KeyGen, as it allows evaluating if the key generated from some subset of features is correct, or more features should be included. The exact number $|\Psi|$ of "strong" features included in a user's template is therefore unknown, even if the adversary can decrypt T_U. Still, the earlier a feature ϕ_i appears in the user's template, the higher the likelihood that $\phi_i \in \Psi$. A natural choice for an adversary is therefore to guess features in the order of their appearance in T_U.

Instead of always attempting to guess values for all features (as in [2]), in our algorithm $\mathcal{A}$, the adversary initially assumes that $|\Psi| = 1$ and evaluates all possible values for only the first feature. If no key is successfully generated, $\mathcal{A}$ iteratively increments its current estimate of the size of Ψ, i.e., $|\Psi| = 2, 3, \ldots$, and thereby includes more and more features in its

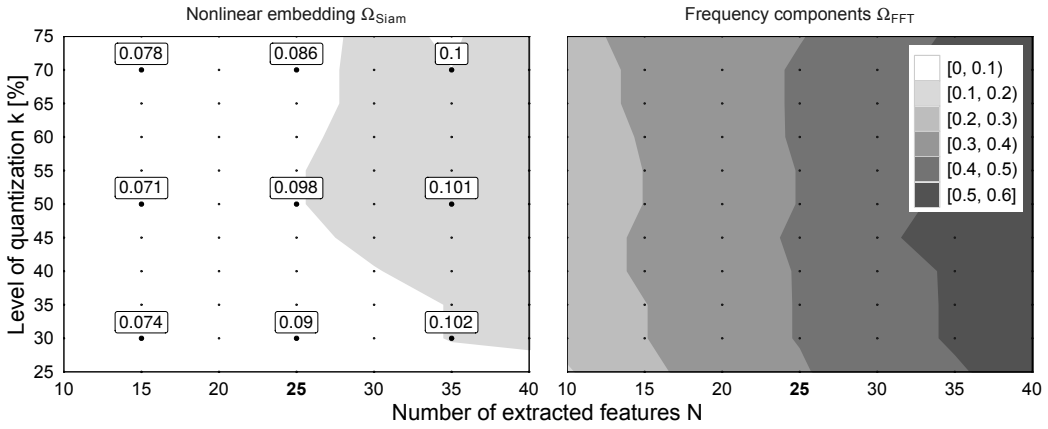

Figure 6: False reject rate (FRR) based on the number of extracted features N and the level of quantization k. Contour lines are linear interpolations on the grid of parameters marked with black dots. The white area represents combinations of parameters N and k for which the average FRR for generated keys is lower than 0.1.

search. This process continues until the adversary assumes the correct number of "strong" features and, while evaluating all possible combinations of feature values for those "strong" features, finally generates the correct key.

Guessing feature values ϕ_i.

When the adversary is evaluating all possible keys which consist of the first $|\Psi'|$ features, we assume that he guesses the value of the first $|\Psi'|$ quantized features one by one, in a depth-first-search manner. When guessing the value for a specific feature ϕ_i, we assume that the adversary uses the population statistics P_{ϕ_i} acquired from other users' biometric features to guess the values in the order of their decreasing probabilities, thus increasing the probability of an early success. To take this into account, we denote with $\mathrm{G}(\phi_\mathrm{i}, \mathrm{P}_{\phi_\mathrm{i}})$ the number of different values that $\mathcal{A}$ will try when guessing ϕ_i, before it guesses the correct value.

The runtime estimate of $\mathcal{A}$ can now be expressed as the summation of W_1 failed attempts when $|\Psi'| < |\Psi|$ and W_2 failed attempts made with the correct estimate of $|\Psi'| = |\Psi|$, plus 1 for the last attempt that yields the correct key:

$$\mathrm{RunTime}(\mathcal{A}) \propto W_1 + W_2 + 1$$

Since generated keys and templates are known in our experimental setting, if $\mathcal{A}$ follows the described search pattern, we can evaluate the number of unsuccessful guesses W_1 and W_2 (and therefore the runtime of $\mathcal{A}$) on the used dataset:

W_1: Number of wrong guesses when $|\Psi'| < |\Psi|$.

When assuming a too small number of "strong" features, $\mathcal{A}$ has to enumerate all possible quantized feature values for each feature ϕ_j (denoted by $|\Gamma_j|$), which gives a total of

$$W_1 = \sum_{i=1}^{|\Psi|-1} \left(\prod_{j=1}^{i} |\Gamma_j| \right)$$

wrong guesses, for all possible keys generated from the first $1, 2, \ldots (|\Psi| - 1)$ features.

W_2: Number of wrong guesses when $|\Psi'| = |\Psi|$.

Once $\mathcal{A}$ assumes the correct number of "strong" features, the key will likely be found before all $\prod_{j=1}^{|\Psi|} |\Gamma_j|$ possibilities are explored as $\mathcal{A}$ can use the population statistics to improve its chances of guessing the correct feature values. However, presuming $\mathcal{A}$ guesses a wrong value for feature ϕ_i, it will unsuccessfully enumerate all $\prod_{j=i+1}^{|\Psi|} |\Gamma_j|$ combinations spanned by the remaining features. Since $\mathcal{A}$ will make $\mathrm{G}(\phi_\mathrm{i}, \mathrm{P}_{\phi_\mathrm{i}})$ unsuccessful guesses before arriving at the correct value for ϕ_i and $\mathcal{A}$ can assume an incorrect value for any "strong" feature (i.e., $i = 1, \ldots, |\Psi|$), we get at a total of

$$W_2 = \sum_{i=1}^{|\Psi|} \left(\mathrm{G}(\phi_\mathrm{i}, \mathrm{P}_{\phi_\mathrm{i}}) \cdot \prod_{j=i+1}^{|\Psi|} |\Gamma_j| \right)$$

unsuccessful key guessing attempts.

As we show and discuss in the experimental evaluation in Section 5.5, results indicate that 70% of generated keys require over 2^{30} guessing attempts in the described scenario. Furthermore, half of the keys are even stronger, since $\mathcal{A}$ makes at least 2^{50} guesses before finding the correct key.

5. EVALUATION OF GENERATED KEYS

We now experimentally evaluate performance of the proposed scheme and compute error rates for key regeneration. We then examine the strength of biometric keys derived from the experimentally acquired dataset by simulating attacks discussed in Section 4.

5.1 Experimental Dataset

We derive biometric keys and compute RBT-s based on a dataset of pulse-response measurements (see Section 3.1) comprising a test subject population of 16 people. The dataset was acquired in five seatings where participants' pulse-response was measured five times in a row, summing up to a total of 25 samples per participant. The median timespan between consecutive seatings was 8 days and there was a minimum dwell time of at least one day between seatings. In order to evaluate the key generation scheme under conditions as diverse as possible, we did not impose any requirements or restrictions on the participants besides ensuring that their hands were not overly sweaty. Acquiring data over multiple

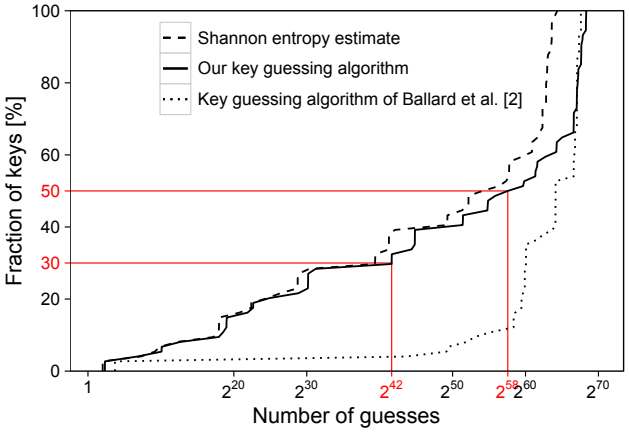

Figure 7: Empirical cumulative distribution of the keys found after a given number of guessing attempts estimated using two different algorithms. As our more conservative algorithm shows, for 70% of the generated keys, the adversary has to try guessing a minimum of 2^{42} combinations. For 50% of the keys, the required effort increases to 2^{55} guesses.

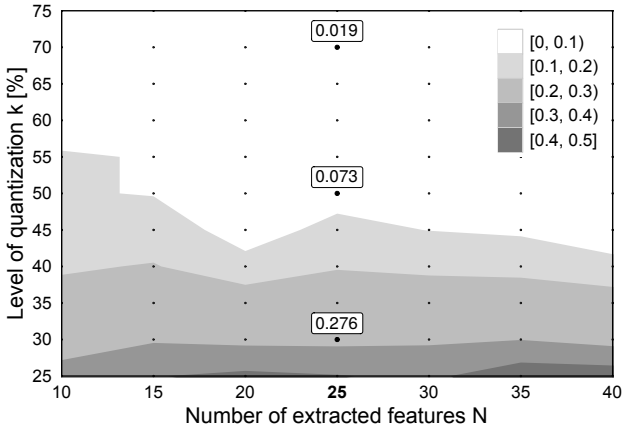

Figure 8: False accept rate (FAR). Darker colors indicate higher FAR and therefore higher probability of success for an imposter attack. The white area represents combinations of parameters N and k for which the average FAR is below 0.1.

sessions thus allowed us to capture various other potential factors, such as varying body water percentage or body temperature.

5.2 Experiment Parameters

The proposed key generation scheme has three steps: (1) feature extraction, (2) feature selection (as part of enrollment), and (3) key generation. As described in Section 3.2, behavior of feature extraction is characterized by a N – the total number of features extracted from biometric measurements. Feature selection, on the other hand, depends on k – the feature quantization level, defined in Section 3.3. In the remainder of this section, we show the performance of our scheme by presenting results for a range of parameter pairs: k between 25 and 75 and N between 10 and 40.

5.3 False Reject Rate

Correctness of a biometric key generation scheme is primarily determined by measuring FRR – the probability that a user fails to recreate the same biometric key using new measurements.

The left plot of Figure 6 shows FRR as contour lines in the 2-dimensional plane for various combinations of N and k. Using the nonlinear feature extraction method Ω_{Siam}, our key generation scheme achieves FRR between 7.4% and 9.8% for a broad range of parameters. These results are better or comparable to FRR values for quantization-based key generation schemes based on other biometric modalities, such as iris, face and voice [4, 30]. Being the first of its kind, this evaluation also serves as a reference for the future comparison of FRR values of key generation schemes based on body impedance.

Values are computed by running a repeated κ-fold cross validation on the whole dataset with $\kappa = 5$ and 5 repetitions to compute average FRR per user. Given a total of ζ user samples, we enroll a user by creating the key and the template on $\zeta \cdot (\kappa - 1)/\kappa$ samples. The remaining ζ/κ samples are used to test the rate of key regeneration based on the previously computed template. We measure the fraction of incorrectly reconstructed keys for all users to arrive at the average FRR.

Comparison with Baseline Feature Extraction.

As a comparison to our feature extraction method, the right side of Figure 6 shows FRR achieved with the baseline feature extraction method which relies on frequency components of the response signal (Ω_{FFT}).

Figure 6 shows that independent of quantization level and number of features, Ω_{Siam} achieves significantly lower FRR than the baseline feature extraction method Ω_{FFT}. We thus conclude that the nonlinear embedding performs significantly better in describing idiosyncratic features. For Ω_{FFT}, FRR almost entirely depends on the number of features while the level of quantization only plays a marginal role, suggesting that most features are less user-specific then those extracted with the Siamese architecture.

Presented false reject rates render performance of the baseline feature extraction method impractical. Therefore, we do not experimentally evaluate the security of that scheme.

5.4 Imposter Attacks

We now assess resistance of the proposed key generation scheme to impostor attacks. This is done by measuring the false accept rate (FAR). In Figure 8, contour lines indicate average FAR, measured on biometric keys generated for different pairs: (N, k). Our key generation method is robust and achieves strong results for a range of N and k (white area), wherein the probability for a successful attack is lower than 0.1.

For each combination (N, k), FAR-s are computed using 5-fold cross validation to generate legitimate user's template from different subsets of data, and then trying all other users' biometric measurements as inputs, in an attempt to derive the legitimate user's key.

Figure 8 also shows that higher quantization levels reduce the likelihood of impersonation attacks. As discussed in Section 3.3, this is due to higher number of features assigned to each enrolled user, which reduces the probability of the adversary generating the correct key.

5.5 Key Guessing Complexity

We use the experimental dataset to evaluate runtime complexity of an attack by an adversary that has the legitimate user's biometric template and PIN. As described in Section 4.2, the adversary tries to generate the legitimate user's key by probabilistic search over all possible feature values using our improved algorithm $\mathcal{A}$.

In this evaluation, we focus on the particular configuration of $N = 25$ and $k = 70\%$. On condition that FRR-s remain practical, we set $N = 25$ to maximize the number of extracted features (see Figure 6). We set $k = 70\%$ in order to minimize the probability of an impostor attack while maintaining a sufficiently high quantization threshold for stringent feature selection (see Figure 8).

Key strength estimates for this particular combination are presented in Figure 7. As in Section 5.4, results are obtained with 5-fold cross validation. We compute population statistics P_{ϕ_i} using biometric data of all users, except the one being evaluated.

Results of the guessing complexity estimation show that majority of keys are strong against an adversary using the approach of algorithm $\mathcal{A}$. Over 70% of generated keys require the adversary to make at least 2^{42} guesses. Also, more than half of generated keys require over 2^{55} attempts.

About 10% of users' keys are predictable due to the feature selection procedure choosing only a small number of "strong" features. In a production system, selection of such a small set of "strong" features would be detected and would result in enrollment failure. In such cases, the user would be required to re-enroll. We included these users and their corresponding keys for the sake of completeness.

Comparison of Key Guessing Algorithms.

Figure 7 compares our improved key guessing algorithm $\mathcal{A}$ with the original guessing algorithm in [2], which $\mathcal{A}$ is built up on. As an additional reference, we show an estimate of Shannon entropy[2] of generated keys from a worst-case perspective by assuming the set of "strong" features to be known for every generated key. By design of RBT, a guessing adversary does not learn the number of "strong" features until the key is found, even if the template is decrypted with the correct PIN; see Section 3.3. Therefore, this entropy calculation underestimates the strength of the keys. In addition, it is not directly comparable to the runtime of a guessing attack, as entropy measures unpredictability in bits and does not describe a concrete strategy to guess keys. However, it serves as an indication of tightness for the estimates derived by key guessing algorithms.

As Figure 7 shows $\mathcal{A}$ performs significantly better than the algorithm of Ballard et al. [2], when guessing generated keys from body impedance. This is mainly due to $\mathcal{A}$ being specifically tailored to guessing keys generated by RBT. Although entropy and the number of guesses are not directly comparable [24], the runtime of $\mathcal{A}$ reflects the estimated entropy of the biometric keys much more precisely than the algorithm of Ballard et al.

[2]We assume feature independence and estimate the probability densities based on all biometric measurements, except those from the user being evaluated.

6. RELATED WORK

Body Impedance.

Analysis techniques based on body impedance—such as Electrical Impedance Tomography [32] and Bioelectrical Impedance Analysis [20]—are well-known technologies in the medical field. However, the use of impedance as a biometric has been explored only recently.

Two initial papers on the topic [7, 8] propose a bracelet which uses electrical characteristics of wrist tissue to authenticate and identify users. Similarly, Rasmussen et al. [23] propose *pulse-response*, a biometric which authenticates users based on the modification of an electrical pulse as it travels from one user's palm to the other. Both methods achieve promising equal error rates even for measurements spanning several weeks, and show feasibility of further research on impedance-based biometrics.

Building upon wrist impedance results, Holz et al. [17] recently extended the concept to support user authentication on any device with a capacitive touchscreen. The bracelet identifies the user based on the characteristics of the wrist and encodes this information into a modulated periodic electrical signal. The signal travels through the user's hand and is detected by the capacitive touchscreen sensors upon every touch, which allows the underlying system to perform continuous and seamless user authentication.

Despite the growing body of related research, we believe that our work represents the first study of generating keys from body impedance. To the best of our knowledge, the only other work which discusses body impedance in the context of generating biometric keys is the study by Gupta et al. [14], which uses skin conductance as an additional measure to fight coercion attacks during key generation from voice biometrics.

Comparison to Key Generation from Voice.

Since previous research has not explored body impedance for generating biometric keys, we compare our results with other biometrics used in key generation. In terms of modality, we find body impedance most comparable to voice biometrics since both rely on a (one-dimensional) signal with a time component, albeit of different scales. Our experimental evaluation suggests that it is possible to generate keys from body impedance with a FRR of 8.6% and a FAR of 1.9%. Furthermore, entropy estimation from a worst-case point of reference (PIN and template known to the adversary) shows that keys from body impedance yield around 46 bits of entropy, on average.

One of the earliest studies that proposes a technique to reliably generate a key from a user's voice is [25], which achieves both FRR and FAR of near 2.0%. Key strength is estimated to be at around 60 bits using guessing entropy. The work in [35] presents cancellable speech template protection in speaker verification systems. FRR and FAR range from 0.0% to 3.0%, depending on the dimension of the extracted features. While these are remarkably low error rates, no entropy or key strength estimation is reported. Although FAR usually correlates with the amount of information extracted from the biometric data, key strength can not be directly deduced from error rates, and a separate analysis is needed.

Probably most related to our work is [4] where the authors also make use of Randomized Biometric Templates to generate cryptographic keys from voice, reporting FRR of 7% and FAR of 10% for two utterances and a quantization threshold $k = 50\%$. They ran the original (and less tailored) guessing

algorithm proposed in [2] to assess strength of generated keys: At least 2^{30} guesses are needed for 36% of the population and over 2^{40} guesses are needed for 7% of the population.

Finally, a recent result on biometric templates for speech [19] reports equal FRR and FAR rates between 8% and 10% when assuming a similar scenario to our analysis, i.e., adversary knows the PIN or password. They estimate the entropy of the scheme to be 76 bits.

While most of these reported FRR and FAR for voice and speech patterns are comparable to our work, it is important to note the methodologies to estimate entropy and key strengths differ widely and are in most cases not directly comparable. We report a very conservative estimate based on an improved guessing algorithm that also provides a concrete strategy an adversary could follow to guess keys.

7. CONCLUSION

This paper explored the use of body impedance for deriving secret keys with the aim of minimizing leakage of sensitive biometric data. Our construction is based on Randomized Biometric Templates [1] and on features extracted using convolutional networks in a Siamese architecture. According to our experiments, the proposed feature extraction method achieves an average false reject rate of 8.6% and a false accept rate of 1.9% for impersonation attempts.

In order to assess strength of generated keys, we improve the approach proposed in [2] to estimate the required guessing effort. This is far from trivial as an adversary might exploit public and leaked user-specific (auxiliary) information. By taking this into account, our guessing strategy results in a more conservative estimate than commonly used methods, such as summary statistics based on entropy. Using our conservative approach, the presented evaluation shows that strength of generated biometric keys is on a par with keys obtained with similar schemes applied to voice biometrics and speech patterns. We estimate the majority of experimentally generated keys to provide between 30 and 60 bits of entropy.

In conclusion, this paper is first to show that biometrics derived from body impedance can be successfully used as a source to generate secret keys and biometric templates that do not leak sensitive information. More importantly, these results support proposals to deploy body impedance in a wide range of security contexts in the future: from door handles that identify tenants, to customer authentication on ATMs, or continuous driver authentication via conductive steering wheels.

8. REFERENCES

[1] L. Ballard, S. Kamara, F. Monrose, and M. K. Reiter. Towards practical biometric key generation with randomized biometric templates. *Proceedings of the 15th ACM conference on Computer and communications security*, page 235, 2008.

[2] L. Ballard, S. Kamara, and M. K. Reiter. The Practical Subtleties of Biometric Key Generation. *USENIX Security Symposium*, pages 61–74, 2008.

[3] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a "siamese" time delay neural network. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 737–744. Morgan-Kaufmann, 1994.

[4] B. Carrara and C. Adams. You are the key: Generating cryptographic keys from voice biometrics. *8th International Conference on Privacy, Security and Trust*, pages 213–222, 2010.

[5] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 539–546, 2005.

[6] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning.

[7] C. Cornelius, R. Peterson, J. Skinner, R. Halter, and D. Kotz. A wearable system that knows who wears it. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, pages 55–67, New York, NY, USA, 2014. ACM.

[8] C. Cornelius, J. Sorber, R. A. Peterson, J. Skinner, R. J. Halter, and D. Kotz. Who wears me? bioimpedance as a passive biometric. In C. A. Gunter and Z. N. J. Peterson, editors, *Proceedings of the 3rd USENIX Workshop on Health Security and Privacy*, Berkeley, CA, USA, 2012. USENIX Association.

[9] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM Journal on Computing*, pages 97–139, 2006.

[10] H. Feng and C. Choong Wah. Private key generation from on-line handwritten signatures. *Information Management & Computer Security*, 10(4):159–164, 2002.

[11] J. Galbally, A. Ross, M. Gomez-Barrero, J. Fierrez, and J. Ortega-Garcia. Iris image reconstruction from binary templates: An efficient probabilistic approach based on genetic algorithms. *Computer Vision and Image Understanding*, 117(10):1512–1525, 2013.

[12] S. Gibbs. HTC stored user fingerprints as image file in unencrypted folder. http://www.theguardian.com/technology/2015/aug/10/htc-fingerprints-world-readable-unencrypted-folder, 2015. Accessed: 2016-21-01.

[13] A. Greenberg. OPM Now Admits 5.6m Feds' Fingerprints Were Stolen By Hackers. http://www.wired.com/2015/09/opm-now-admits-5-6m-feds-fingerprints-stolen-hackers/, 2015. Accessed: 2016-20-01.

[14] P. Gupta and D. Gao. Fighting coercion attacks in key generation using skin conductance. In *Proceedings of the 19th USENIX Conference on Security*, USENIX Security'10, pages 30–30, Berkeley, CA, USA, 2010. USENIX Association.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.

[16] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint*, 2012.

[17] C. Holz and M. Knaust. Biometric Touch Sensing. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 303–312, 2015.

[18] International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). ISO/IEC 24745. BSI Standards Publication, June 2011.

[19] K. Inthavisas and D. Lopresti. Secure speech biometric templates for user authentication. *IET Biometrics*, 1(1):46, 2012.

[20] R. D. Janeiro, C. E. Neves, and M. N. Souza. A method for bio-electrical impedance analysis based on a step-voltage response. *Physiological Measurement*, 21(3):395–408, 2000.

[21] Q. Li, M. Guo, and E.-C. Chang. Fuzzy extractors for asymmetric biometric representations. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–6. IEEE, 2008.

[22] Q. Li, Y. Sutcu, and N. Memon. Secure sketch for biometric templates. In *Advances in Cryptology–ASIACRYPT 2006*, pages 99–113. Springer, 2006.

[23] I. Martinovic, K. B. Rasmussen, M. Roeschlin, and G. Tsudik. Authentication using pulse-response biometrics. In *Proceedings of the 21st Annual Network and Distributed System Security Symposium*, NDSS 2014. Internet Society, 2014.

[24] J. Massey. Guessing and entropy. In *Information Theory, 1994. Proceedings., IEEE International Symposium on*, page 204, 1994.

[25] F. Monrose, M. K. Reiter, Q. Li, and S. Wetzel. Cryptographic key generation from voice. In *Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on*, pages 202–213. IEEE, 2001.

[26] K. Nandakumar and A. Jain. Biometric template protection: Bridging the performance gap between theory and practice. *Signal Processing Magazine, IEEE*, 32(5):88–100, Sept 2015.

[27] K. Nandakumar, A. K. Jain, and A. Nagar. Biometric template

security. *Eurasip Journal on Advances in Signal Processing*, 2008, 2008.

[28] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):561–572, 2007.

[29] C. Rathgeb and A. Uhl. Privacy preserving key generation for iris biometrics. In *Communications and Multimedia Security*, volume 6109, pages 191–200. Springer, 2010.

[30] C. Rathgeb and A. Uhl. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, 2011(1):3, 2011.

[31] A. Ross, J. Shah, and A. K. Jain. From template to image: Reconstructing fingerprints from minutiae points. *IEEE transactions on pattern analysis and machine intelligence*, 29(4):544–560, 2007.

[32] G. J. Saulnier, R. S. Blue, J. C. Newell, D. Isaacson, and P. M. Edic. Electrical impedance tomography. *IEEE Signal Processing Magazine*, 18(6):31–43, 2001.

[33] Y. Sutcu, Q. Li, and N. Memon. Protecting biometric templates with sketch: Theory and practice. *Information Forensics and Security, IEEE Transactions on*, 2(3):503–512, 2007.

[34] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[35] A. B. J. Teoh and L.-Y. Chong. Secure speech template protection in speaker verification system. *Speech communication*, 52(2):150–163, 2010.

[36] C. Vielhauer, R. Steinmetz, and A. Mayerhöfer. Biometric hash based on statistical features of online signatures. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 1, pages 123–126. IEEE, 2002.

[37] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328, 2014.

APPENDIX

A. STRUCTURE OF THE CONVOLUTIONAL NETWORK

We specify and train the multilayer convolutional network for feature extraction (Ω_{Siam}) in Torch [6], an open source machine learning framework.

As shown in Listing 1, the network consists of six sequential layers, followed by two fully connected linear layers. As input, the network takes an impedance measurement, encoded as a one-dimensional vector with 4000 elements. Each layer of the network consists of three sublayers: one-dimensional convolution, rectified linear unit (ReLU), and maximum value subsampling. The convolution layers act as linear filters with learnable parameters which are optimized during training to detect patterns of increasing complexity. ReLU is a transfer function which adds non-linearity, while max-pooling layers allow temporal generalization of the detected patterns. In each layer, the dimensionality of processed data is reduces, and multiple sequential layers allow the network to learn hierarchical representations of input patterns. Finally, the number of extracted features (N) is controlled by specifying the sizes of the fully connected linear layers (`nrOutputFeatures`) at the output stage of the network.

Once the network is trained, it serves as a non-linear mapping function from a raw biometric measurement to a set of N features that we use as input to the key generation scheme.

```
model = nn.Sequential()

model:add( nn.TemporalConvolution(1, 5, 7, 2) )
model:add( nn.ReLU() )
model:add( nn.TemporalMaxPooling(4, 4) )

model:add( nn.TemporalConvolution(5, 10, 7, 1) )
model:add( nn.ReLU() )
model:add( nn.TemporalMaxPooling(3, 3) )

model:add( nn.TemporalConvolution(10, 20, 5, 1) )
model:add( nn.ReLU() )
model:add( nn.TemporalMaxPooling(3, 3) )

model:add( nn.TemporalConvolution(20, 40, 3, 1) )
model:add( nn.ReLU() )
model:add( nn.TemporalMaxPooling(2, 2) )

model:add( nn.TemporalConvolution(40, 80, 3, 1) )
model:add( nn.ReLU() )
model:add( nn.TemporalMaxPooling(2, 2) )

model:add( nn.TemporalConvolution(80, 120, 3, 1) )
model:add( nn.ReLU() )
model:add( nn.TemporalMaxPooling(2, 2) )

model:add( nn.Linear(480, 200) )
model:add( nn.Tanh() )
model:add( nn.Linear(200, nrOutputFeatures) )
```

Listing 1: Specification of Ω_{Siam} in Torch