

# Feeling is Believing: a location limited channel based on grip pattern biometrics and cryptanalysis

Ileana Buhan, Jeroen Doumen, Pieter Hartel, Raymond Veldhuis,  
EEMCS Faculty, University of Twente,  
{Ileana.Buhan, Jeroen.Doumen, Pieter.Hartel, Raymond.Veldhuis}@utwente.nl

## Abstract

We use grip pattern based biometrics as a location limited channel to achieve pre-authentication in a protocol that sets up a secure channel between two handheld devices. The protocol efficiently calculates a shared secret key from biometric data using quantization and cryptanalysis. The protocol is used in an application where grip pattern based biometrics is used to control access to police hand guns.

## 1 Introduction

A location limited channel (LLC) is a communication channel that, because of specific properties of its physical realization, offers some degree of confidentiality, integrity, and/or authenticity to the messages being sent on the channel. For example two people having a face to face conversation in a noisy public place know that eavesdroppers need sophisticated equipment to follow their conversation (some confidentiality), and that a third party trying to perpetrate a man in the middle attack will be conspicuous (good integrity and authenticity). Many types of location limited channel have been proposed, for example using physical contact [9], infra red [3], visual channels based on cameras and screens [7], audio channels based on speakers and microphones [8], wireless channels based on short range radio [2], and finally "human" channels based on users entering pin codes [12].

We offer three contributions:

*A location limited channel based on biometrics.* The first contribution is a new type of LLC, which uses biometrics. We explain the idea with grip pattern based bio-

metrics, but this can be generalized to other types of biometrics. The idea is both simple and effective. Suppose that two users wish to set up a secure communication channel. Both own a handheld device, equipped with a grip pattern sensor and a short range radio, that holds the owners stored biometric grip pattern template [11]. The users swap the devices, so that each device can measure the grip pattern of the other user. Then the devices are returned to their owners. Each device now contains a genuine template of its owner and a measurement that approximates the template of the other user, which we will call the guest. The idea is that each device calculates a common key from the owner template and/or the guest measurement. The act of the guest putting her hand on the other device corresponds to sending a message on the LLC. Therefore we have termed our protocol Feeling is Believing (FiB).

*A security protocol based on cryptanalysis.* The second contribution is in the realization of the FiB protocol where we use cryptanalysis to recover key material from a grip pattern measurement. Assume that each device holds an owner template and the guest measurement. Now each device should be able to derive a common key in essentially two steps: (1) by a quantization algorithm [6] each device calculates key material from the guest measurement, which, because of measurement errors and noise might slightly differ in a number of bits from the same calculation made by the other device on the owner template. (2) The statistics calculated on the owner template during enrolment are used to determine which bits are most likely to be wrong. The cryptanalysis then flips the bit corresponding to the feature with the highest probability of being wrong first, then the second highest etc. Eventually

the key derived from the owner template on one device will be the same as the key derived from the measured template on the other device.

The efficiency of the protocol depends on the number of steps to be made by the cryptanalysis. We apply the correction mechanism on a key extraction strategy previously proposed by Linnartz and Tuyls [6]. In this way, we lift their template protection scheme to a mechanism for generating shared keys from biometric data. We show by an experiment that using a correction mechanism when constructing the key can significantly improve the overall results.

The security of the protocol depends on a number of factors. Firstly, only the communicating devices are able to perform the cryptanalysis efficiently because they can start on key material that is almost correct. An eavesdropper has to start from random key material, and should thus take longer to derive the correct key. Secondly, because the two users are actively engaged in swapping devices they will notice when someone else tries to grab or replace one of their devices. In this case the users will abort the attempt to set up the secure channel.

*Application to smart guns.* The third contribution is the application of the protocol to smart guns with grip pattern biometrics. These weapons are intended for use by the police, to reduce the risk of police officers being shot by their own weapon in case of a take away situation [10]. In this application a police handgun authenticates the owner by grip pattern biometrics integrated with the grip of the gun. The research challenge is that the False Rejection Rate (FRR) must be less than  $10^{-4}$ , which is the accepted rate of misfire for a police weapon. The FiB protocol solves the following problem. Police officers mostly work in teams of two. Each officer must be able to fire the other officer's weapon. Normally, teams are scheduled in advance so that appropriate templates can be loaded into the weapons at the police station. However, in emergencies this is not possible. In this case police officers have to team up unprepared and swap templates in the field. Police officers may work with colleagues from other departments, even from neighboring countries, so we may not assume a common key, or even a public key infrastructure. In this case the FiB protocol is used to calculate a shared key, which can then be used to swap templates securely. In the context of smart gun, our protocol has to achieve two security goals:

1. An intruder cannot prevent one of the team members from firing either gun.
2. An intruder cannot load his template into either gun.

We survey related work in Section 2. We provide an overview of the FiB protocol in Section 3. Section 4 describes the key extraction algorithm and the personalized correction mechanism used. Section 5 presents results of experiments on key extraction with correction on two sets of data collected from 41 police officers. Future work and conclusions are presented in Section 6.

## 2 Related work

Many types of location limited channel have been proposed in the literature, each with its own properties. Balfanz *et al.* [2] propose physical contact between devices. This type of channel has the property that the user can precisely control which devices are communicating with each other but it can become too bulky to carry around all the interfaces that connect for this purpose. Balfanz *et al.* then extend this approach to using infrared communication. McCune *et al.* [7] use a visual channel to make photographs of the hash codes of public keys. This offers significant user friendliness. In the same line of work, Googrich *et al.* [8] propose a human assisted authentication audio channel as a secure side channel. They use a text to speech engine for vocalizing a sentence derived from the hash of a device public key.

LLC is used mostly to authenticate public-keys. The hash of the public key is either vocalized [8] or photographed [7]. Others use a Diffie-Hellman [13] like key agreement scheme where short sequences transmitted on the secure channel authenticate the key sent on the main channel.

## 3 FiB Protocol Description

Before we delve into the description of the protocol, which represents the core of our solution, we describe the context (Figure 1). Enrollment of users A and B takes place before the protocol starts. During enrollment low noise measurements of the users are taken. Then a bit string representing key, helper data and error profiles are

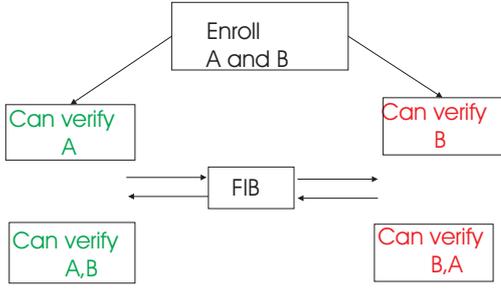


Figure 1: Context and goals for FiB protocol

computed for each user. All this information is loaded into the device. After the enrollment we have achieved that: (1) the identity of the user can be verified by his own device, and (2) a device is prepared to run the FiB protocol which also allows the *other device* to verify the identity of the user.

The FiB protocol uses an LLC to create a shared secret key. This key is then used to create a secure communication channel between two devices that do not share any other secret and where a public key infrastructure is unavailable.

The process of key generation and reconstruction is described in the next section. Here we describe the FiB protocol.

*Protocol initialization:* We assume that one device starts the protocol. The device then advertises that it is ready to exchange templates by the broadcast of some initial data.

*Key extraction:* We use a *fuzzy extractor* to extract the bio-key [5] from the biometric data. Dodis *et al.* [15] define a fuzzy extractor as a function that can extract a uniformly random bit string  $K$  from its input  $T$  in a noise tolerant way. Noise tolerance means that if the input changes to some  $F$  that is close to  $T$  according to some appropriate metric, the bit string  $K$  can be reproduced exactly. To assist in reproducing  $K$  from  $F$  the fuzzy extractor publishes a non-secret bit string  $H$ , the so-called *helper data*. Unlike traditional keys,  $K$  does not have to be stored because it can be recovered from the combination  $F$  and  $H$ . Section 4.1 presents this step in detail.

*Template exchange:* The actual template exchange takes place. This will preserve the recognition algorithm performance and serves as validation of the device owner.

*Verification:* This final step will confirm that the transfer was completed successfully. All that a user has to do is to submit his biometric data to the sensor and see whether he is accepted or not.

$D_x$	identifies Device $x$
$T_x \in \mathbb{R}^n$	the enrolled Template of user $x$
$H_x \in \mathbb{R}^n$	Helper data of user $x$
$F_x \in \mathbb{R}^n$	Feature vector currently measured for user $x$
$K_x \in \{0, 1\}^n$	Key extracted from $T_x$ and $H_x$
$K'_x \in \{0, 1\}^n$	Key extracted from $F_x$ and $H_x$
$E_x \in \langle \mathbb{N}, \mathbb{R} \rangle^n$	Error profile for a user

Table 1: Notations used in the FiB protocol description.  $n$  is the dimension of the various vectors.

### 3.1 FiB Protocol Formal Description

The notation used below is summarized in Table 1. The principals are two devices  $D_a$  and  $D_b$ . Before the protocol starts each of the devices knows the data of its owner, i.e. the template, the key, the helper data constructed from the previous template, the key and the error profile. Hence initially  $D_a$  knows  $\{T_a, H_a, K_a, E_a\}$  and  $D_b$  knows  $\{T_b, H_b, K_b, E_b\}$ . Without loss of generality we assume that  $D_a$  starts the protocol.

1.  $D_a$  : Measure feature vector of a guest:  $F_b$ .
2.  $D_a \rightarrow D_b$  :  $H_a$ 
  - 2.1  $D_b$  : Measure feature vector of a guest:  $F_a$ .
  - 2.2  $D_b$  :  $K'_a = \text{Extract}(F_a, H_a)$ .
  - 2.3  $D_b$  : Measure feature vector of owner:  $F'_b$ .
  - 2.4  $D_b$  : Verify that  $T_b$  matches  $F'_b$ .
3.  $D_b \rightarrow D_a$  :  $H_b, \{T_b\}_{K'_a}$ 
  - 3.1  $D_a$  :  $K'_a = \text{Correct}(K_a, \{T_b\}_{K'_a}, E_a, F_b)$ .
  - 3.2  $D_a$  :  $K_b = \text{Extract}(T_b, H_b)$ .
4.  $D_a \rightarrow D_b$  :  $\{F_b, T_a\}_{K_b || K'_a}$ 
  - 4.1  $D_b$  : Verify that  $T_a$  matches  $F_a$ .

4.2  $D_b$  : Verify that  $T_b$  matches  $F_b$ .

During steps: 2, 3 and 4 information is sent over the main communication channel. Steps: 2.2, 2.3, 2.4, 3.1, 3.2, and 4.1 are computed locally by the devices. LLC operations are steps 1 and 2.1.

**Step 1.**  $D_a$  assumes the initiator role.

**Step 2.** When an unknown grip pattern is detected,  $D_a$  will broadcast  $H_a$ , its helper data.

*Step 2.1* Device  $D_b$  detects an unknown grip pattern.  $D_b$  has also received  $H_a$ . At this stage  $D_b$  assumes the responder role.

*Step 2.2*  $D_b$  applies function  $Extract(F_a, H_a)$ , which is described in detail in Section 4.1. The result of this operation is key  $K'_a$ .

*Step 2.3* A new measurement of the owner of  $D_b$  is taken. Only if the owner is recognized it will send  $T_b$  encrypted with the key extracted in the previous step.

*Step 2.4* The device verifies that  $T_b$  and  $F'_b$  originate from the same user. Verification is based on the classifier based verification algorithm [11], implemented in the device.

**Step 3.** As a responder  $D_b$  sends his helper data  $H_b$  and  $\{T_b\}_{K'_a}$ .

*Step 3.1* The key  $K'_a$  extracted from  $F_a$  and  $H_a$ , might differ slightly from  $K_a$ . However, we expect that the keys are close in terms of their Hamming distance because of the way they are generated, see Section 4. Thus  $Correct(K_a, \{T_b\}_{K'_a}, E_a, F_b)$  will flip carefully chosen bits in  $K'_a$  until it can successfully decrypt  $T_b$ .  $T_b$  is not known to  $D_a$ , but it can be recognized because  $D_a$  knows  $F_b$  and  $T_b$  is similar to  $F_b$ .

*Step 3.2* Device  $D_a$  will extract key  $K_b$  from  $T_b$  and the received  $H_b$ .

**Step 4.**  $D_a$  sends  $\{F_b, T_a\}_{K_b || K'_a}$ .  $D_b$  knows both  $K'_a$  and  $K_b$ .  $F_b$  is sent so that the responder can verify whether his owner has submitted his biometric data to the other device.

*Step 4.1* Before accepting  $T_a$ ,  $D_b$  verifies the combination  $T_a, F_a$ . If the verification is successful the common established key is  $K_b || K'_a$ .  $||$  is a generic operation between two keys, for example concatenation. We choose  $K'_a$  as part of the session key to provide the responder the proof that the device knows  $T_a$  and that it has performed a measurement of  $F_b$ . It also provides freshness for the responder.

*Step 4.2* The responder has a proof that the initiator device measured an impression of his biometric.

Function *Correct* is our key correction method using a pre-computed personalized error profile computed during enrollment. It is described in detail in section 4.2.1.

### 3.2 Formal verification of the FiB protocol with CoProVe

We have formally verified that the FiB protocol satisfies secrecy of the templates and mutual authentication. The tool used for this purpose is the constraint based security protocol verifier CoProVe by Corin and Etalle [1].

An earlier version of the protocol was verified and found buggy, the published protocol fixes the flaw found.

A (security) protocol is normally verified using a model of the protocol, to avoid getting bogged down in irrelevant detail. The quality of the model then determines the accuracy of the verification results. The basic difference between a protocol and a model lies in the assumptions made when modelling the protocol. We believe that the following assumptions are realistic:

1. *No biometrics* We assume that the correction mechanism always works perfectly and thus the initiator knows the key used by the sender. Thus, we look only at complete protocol round. When the initiator cannot work out the key the protocol is aborted. We assume, in this case, that the intruder learns nothing.
2. *No validation* This is because systems without an equational theory such as CoProVe cannot compare two terms. Steps 2.4, 4.1, 4.2 are left out. We have verified a simplified version of the protocol, offering the intruder the best opportunity to break the protocol.

We argue that the above abstractions do not affect the secrecy and the authentication property.

Verification with CoProVe explores a scenario in which one of the parties involved in the protocol plays the role of the initiator (i.e. the party starting the protocol) and the other plays the role of the responder. A third party, the intruder learns all message exchanged by the initiator and the responder. The intruder can devise new messages and send them to honest participants as well as replay or delete messages. Should the intruder learn a secret key and a message encrypted with that key, then the intruder also knows the message. This is the classical Dolev-Yao intruder [4].

We have explored two scenarios that we believe to be realistic and representative for real attacks.

*Secrecy of the templates.* In the first scenario two honest participants Alice and Bob are plagued by a powerful intruder who is assumed to know  $T_b$  (because the intruder might have communicated with Bob in a previous session). In the scenario the intruder tries to find  $T_a$ . The secrecy of  $T_a$  is established by CoProVe under the assumption that the intruder does not know  $F_a$ . This is a realistic assumption because  $F_a$  is never disclosed by Alice. Verification thus shows that the intruder cannot learn the templates nor the key extracted from the templates.

In the second scenario one participant, Alice, has to deal with the intruder, who does not have useful initial knowledge. Verification shows that  $T_a$  remains secret when Alice is the initiator of the protocol. This means that an intruder cannot trick Alice into disclosing her template data if she does not provide the intruder a sample of her biometric data.

When Alice is the responder and the intruder has full access to the gun (i.e. the intruder can submit his own biometric data),  $T_a$  will not be disclosed. This is because before the device sends anything on the wireless link it will check whether his owner is there (steps 2.3 and 2.4).

*Authentication.* In both scenarios we have also verified authentication of responder to the initiator and authentication of initiator to responder.

## 4 Biometric Key

From a biometric template measurement we generate a bio-key in two steps. The first step, *distinguishable fea-*

*ture generation*, depends on the biometric data. The second step, *generation of a repeatable sequence*, is independent of the biometric data.

The goal of the distinguishable feature generation step is to find a transformation such that each transformed feature can be used to separate an authentic user from an impostor.

A detailed description of this step for grip pattern data is described by Veldhuis *et al.* [11]. We will not present this step here in detail. It is sufficient to say that from a  $44 \times 44$  data matrix obtained from the grip pattern sensor, a characteristic vector of 40 values is computed using Principal Component Analysis (PCA). This vector represents the *biometric identity* of an individual.

Here we focus on the second step, the repeatable sequence generation. The key extraction mechanism describes the process of quantizing the 40 numbers that represent the bio-key of the user.

Due to several factors like variations of a user emotional status, aging of the sensor, noise, measurement errors etc., two measurements coming from the same user are similar but never the same. Our goal is to map two sequences that are "close enough" to the same binary string.

### 4.1 Function Extract

For repeatable sequence generation we use the quantized index modulation method proposed by Linnartz and Tuyls [6] for the purpose template protection schemes. We summarize their approach to make the paper self contained. The fuzzy extractor consists of two functions:

**Generate** takes as input a template  $T$  and a secret key  $K$  and calculates as output the public helper data  $H$ . The calculation  $\text{Generate}(t_i, k_i) = h_i$  proceeds component wise as follows:

$$h_i = \begin{cases} (2p + \frac{1}{2}) \cdot q - t_i, & k_i = 1 \\ (2p - \frac{1}{2}) \cdot q - t_i, & k_i = 0. \end{cases} \quad (1)$$

Here  $q$  is the quantization range and  $p$  is chosen such that  $-q \leq h_i \leq q$  for all  $i = 1 \dots n$  and  $p \in \mathbb{Z}$ . The Generate function is performed during enrolment.

**Extract** recovers the secret key  $K$  from a measured feature vector  $F$  and the public helper data  $H$ . The calculation  $\text{Extract}(f_i, h_i) = k_i$  proceeds component

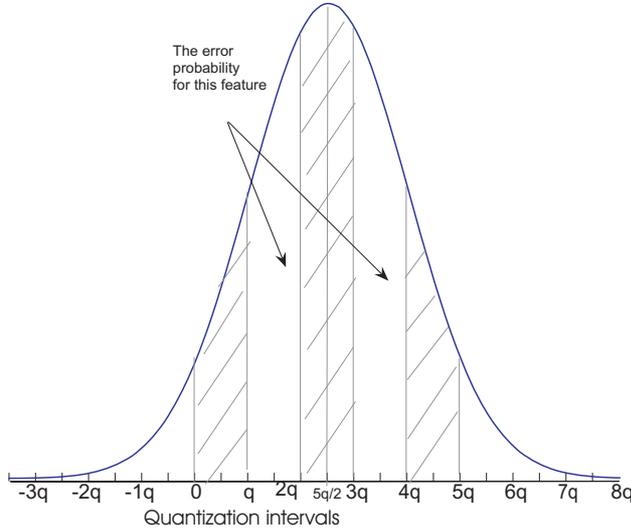


Figure 2: Pdf for one feature.

wise as follows:

$$k_i = \begin{cases} 1, & 2pq \leq f_i + h_i < (2p + 1)q \\ 0, & (2p - 1)q \leq f_i + h_i < 2pq. \end{cases} \quad (2)$$

Having described the fuzzy extractor of Linnartz and Tuyls we remind the reader that extractors are not perfect, particularly because during key extraction errors may appear due to noise. However, we expect a small number of errors between the random sequence  $K$  and the result of the Extract function when  $T$  and  $F$  belong to the same user. To correct such errors is the topic of the next section.

## 4.2 Function Correct

The Correct function is used to correct the errors that might appear after applying the function Extract.

The Correct function requires as input an error profile that represents the position of the bits with the highest probability of being wrong. An error profile is user specific; it is calculated during enrolment and it is kept secret just like the user's key.

For example if a user has error profile:  $E_a = [(3, 0.45), (10, 0.32), (21, 0.27), (13, 0.1), (4, 0.02)..]$ ,

the third key bit  $k_3$  has the highest probability of being wrong,  $k_{10}$  is the second best guess of a key bit being wrong, etc.

The Correct function itself uses a semi-known plain text attack to recover the key. The Correct function is performed by the initiator of the FiB protocol if  $K_a$  cannot decrypt  $\{T_b\}_{K'_a}$ . The semi-known plain text is  $T_b$ . It is semi-known in the sense that in case of a successful decryption  $T_b$  will match  $F_b$ .

### 4.2.1 Error profile computation

Linnartz and Tuyls propose a multiple quantization level system with odd-even bands. The embedding of binary data is done by shifting the template distribution to the center of the closest even-odd  $q$  interval if the value of the key bit is a 1, or to the center of an odd-even  $q$  interval if the value of the key bit is a 0. When we want to extract a particular key bit we sample this distribution and set the key bit accordingly. Whenever the measured value has an error greater than  $\pm \frac{q}{2}$  we get an error in the key computation.

Each key extracted from the a grip pattern has a length of 40 bits. Each bit of the key is extracted independently of the other bits from the corresponding feature. Each feature is independent from the others and we assume as by Veldhuis *et al.* [11] that each feature has a normal distribution. For a particular user, the standard deviations of the measurements are characteristic.

The error profile is a vector of length 40. The  $i$ -th value of the error profile represents the probability of the  $i$ -th key bit to be computed wrongly.

In Figure 2 we see the effect of quantization with step  $q$  on a feature that has a Gaussian distribution. The mean of the distribution is shifted to an even-odd interval because the value of the embedded key was a 1. Each time the input for Extract is in the shaded area a key bit with the value 1 is output by function Extract. If the input falls in the area indicated by the arrows, the output of the Extract function is a 0. This is an error. Since  $q$  is fixed, we can approximate the error by the area indicated by the arrows in Figure 2.

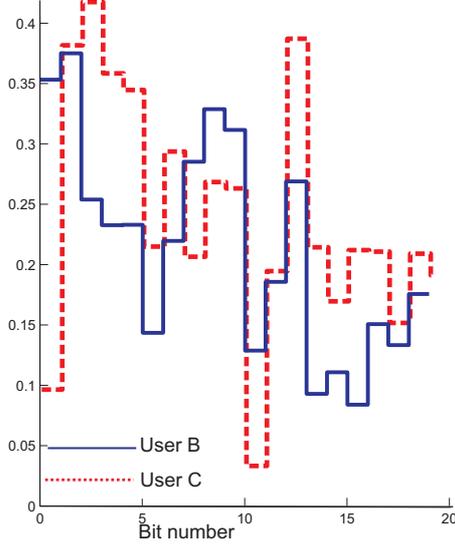


Figure 3: Error profiles computed for two different users.

$$\begin{aligned}
error(\sigma, q) &= 2 \sum_{i=0}^{\infty} \int_{\mu+(1+4i)\frac{q}{2}}^{\mu+(3+4i)\frac{q}{2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \\
&= \sigma 2\sqrt{2} \sum_{i=0}^{\infty} \int_{\frac{(1+4i)q}{2\sqrt{2}\sigma}}^{\frac{(3+4i)q}{2\sqrt{2}\sigma}} e^{-x^2} dx \\
&\approx \sigma 2\sqrt{2} \int_{\frac{q}{\sigma 2\sqrt{2}}}^{\frac{3q}{\sigma 2\sqrt{2}}} e^{-x^2} dx
\end{aligned}$$

In Figure 2,  $\mu$  is equal to  $5 \cdot \frac{q}{2}$ . The error depends on the standard deviation of the feature for which it is computed. This permits us to construct personalized error profiles. Figure 3 shows the error profiles, and the associated probabilities of errors occurring for two different users. We can see that for the first bit of the key the probability of error for user B is small; approximately 0.15 while user C has a higher error probability of 0.45 for the same bit.

The computed errors for each bit depend on the quantization step  $q$ . We can see in Figure 2 that the larger  $q$  the higher the noise tolerance. The higher the  $q$  the smaller

the error area.

Figure 4 shows the error profiles for 41 users (see section 5) computed for  $q = 1$ . Figure 5 shows that the error profiles for the same users with  $q = 3$  are less precise. A decisive influence on the exactness of the error profiling is the precision with which the standard deviation for each feature is computed.

#### 4.2.2 Key search algorithm

In classical symmetric cryptography to decrypt a message encrypted with a key  $K$  one must possess  $K$ . In particular, with a key  $K'$  that differs only in one bit from  $K$ , decryption will fail. The FiB protocol uses this apparent disadvantage of symmetric key cryptography as an advantage:  $K'$  is used to form the session key. The noise of the measurements is used as random salt [14] for the session key. The key search algorithm makes it possible to recover  $K'$ .

Before the algorithm starts we decide on how many trials we make to discover the key. If we set the error threshold to  $k$  bits the algorithm will try out  $\sum_{i=0}^k C_{40}^i$  combinations before it gives up.

We start the key search by assuming there are no errors in  $K'$ , and we use  $K'$  to decrypt. If decryption fails we assume that we have a one bit error. We start flipping one bit of the key according to the position indicated by the error profile, until we have exhausted the error profile. Then we assume that two bits are wrong and we try all combinations of two bits from the error profile. Finally if we reach the limit on the number of trials we assume that the key is coming from an intruder.

*Example:* We have  $K = [101100101]$  and  $\{T_b\}_{K'}$  available.  $K' = [001101101]$ . We decide before the algorithm starts that the maximum accepted error is 2 bits.

**number of bits in error is 0** we discover that  $[101100101]$  does not decrypt  $\{T_b\}_{K'}$ .

**number of bits in error is 1** Bits are corrected in the order given by the error profile of the user. For this example the positions within the error profile are (1,6,2,3,5,4,8,9,7).

(1,,,,,,).  $K = [001100101]$  decryption unsuccessful.

(.,6,,,,,,).  $K = [101100001]$  decryption unsuccessful.

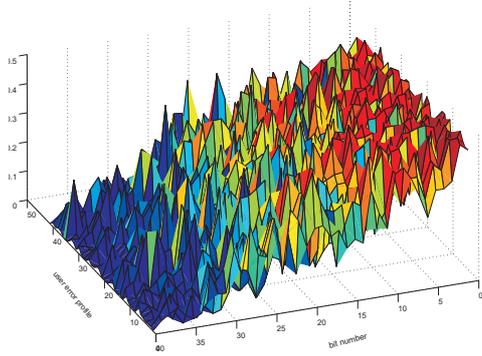


Figure 4: Error profile for  $q_i = 1, i = 1, 40$ .

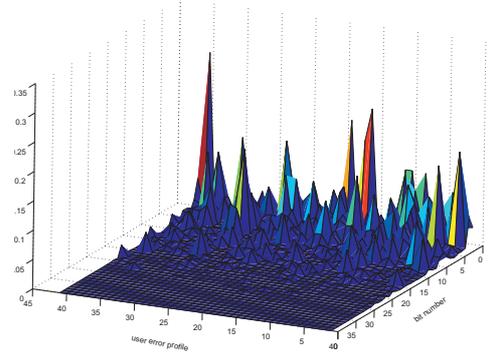


Figure 5: Error profile for  $q_i = 3, i = 1, 40$ .

(...,2,.....).  $K = [111100101]$  decryption unsuccessful. And so on until the last bit.

**number of bits in error is 2** Two bits are corrected at a time.

(1,6,.....).  $K = [001101101]$  decryption successful

The recovery of  $K'$  is a semi-known plain text attack. When the correct value of  $K'$  is discovered the initiator will recognize the message encrypted with  $K'$ . This is possible since the encrypted message is a biometric template. The initiator of the protocol possesses a fresh measurement of this template and hence is able to recognize a correct match. The verification is performed by a classifier based matching algorithm designed for this particular biometrics.

## 5 Evaluation

The evaluation is performed on grip pattern biometric data collected from 41 police officers.

Each officer contributes 25 different samples to each set. 75% of these samples are used for training the algorithm and 25% are used for testing. For training and testing we use the same data that is used for verification by the classifier based recognition algorithm.

Figures 6 presents the results obtained from the collected data. We offer two conclusions from this evaluation.

The first conclusion is, as expected, the larger the quantization step the lower the FRR but the higher the FAR. We tested 8 different values for  $q$ , ranging from 1 to 5 in increments of 0.5. We did not try larger values for  $q$  because the FAR then becomes unacceptably large.

The second conclusion is that the influence of the correction algorithm is significant. For example for  $q = 3$ , without correction the FRR=13.9300% and the FAR=0%. When we correct 1 bit the FRR goes down to 5.923%, while the FAR retains the same value 0%. After correcting 2 bits the FRR goes down to 2.431% while the FAR remains equal to 0%. Correcting 3 bits further reduces the FRR to 1.7422% while the FAR increases only slightly to 0.0784%. Table 2 presents the values obtained for the FAR and FRR for different settings of  $q$  and different numbers of corrected bits.

## 6 Conclusions and future work

The contributions of this paper are threefold. Firstly, we propose a new location limited channel (LLC) using biometrics. Our LLC works as follows: a user first grips one device that measures her hand grip pattern and then grips a second device with this measurement capability. This effectively transfers her grip pattern from one device to another in a location limited way.

The second contribution is that we use fuzzy extractors with cryptanalysis to derive a common, secret encryption key from the measured grip patterns. Fuzzy extractors are not perfect, and therefore require error correction, for

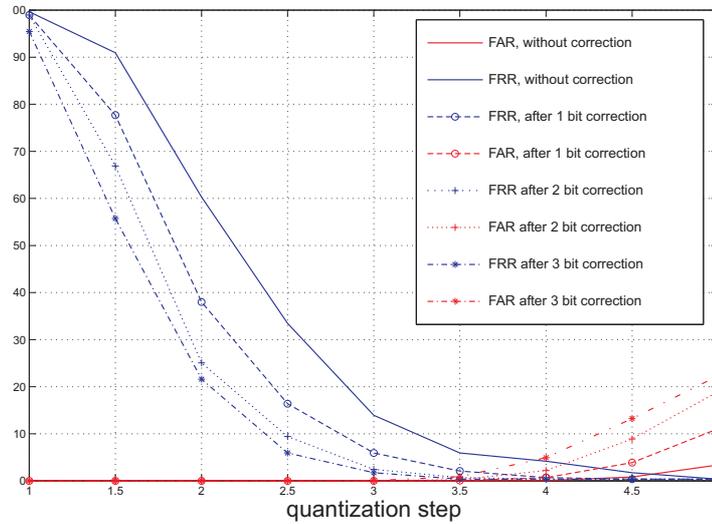


Figure 6: Results on grip pattern data.

		q=2	q=3	q=4	q=5
no correction	FRR	60.27	13.93	4.18	0.34
	FAR	0	0	0.08	3.38
1 bit corrected	FRR	37.98	5.92	0.69	0.35
	FAR	0	0	0.67	11.09
2 bits correction	FRR	25.09	2.43	0.69	0.35
	FAR	0	0.08	2.18	18.97
3 bits correction	FRR	21.60	1.74	0.35	0.35
	FAR	0	0.08	3.23	22.33

Table 2: Selected results

which we use cryptanalysis. The idea is that keys extracted from the same user would differ only in a few bits, whereas keys extracted from different users would differ in more than just a few bits. We exploit the fact that each time a user measures her grip pattern, the generated key is slightly different. This ensures that the secret keys are automatically salted, thus providing freshness in the cryptographic protocol.

The third contribution is that our correction algorithm is efficient because it uses a personalized error profile. This algorithm is used to narrow down the search in case that key bits are mistaken.

We present an evaluation of the performance in terms of FAR and FRR and we prove that the correction algorithm significantly improves the overall results.

As future work we will focus on improving the search algorithm and apply the FiB protocol to other types of biometrics.

## References

- [1] R. Corin and S. Etalle. An improved constraint-based system for the verification of security protocols. In *9th Int. Static Analysis Symp. (SAS), Madrid, Spain*, volume LNCS 2477, pages 326–341, Berlin, September 2002. Springer-Verlag.
- [2] P. Stewart D. Balfanz, D. K. Smetters and H. C. Wong. Talking to strangers: Authentication in Ad-Hoc wireless networks. In *Network and Distributed Systems Security Symposium (NDSS)*, San Diego, California, Feb 2002. The Internet Society, Reston, Virginia.

- [3] R. E. Grinter D. Balfanz, G. Durfee and D. K. Smetters. In search of usable security: Five lessons from the field. *IEEE Journal on Security and Privacy*, 2(5):19–24, Sep 2004.
- [4] D. Dolev and A. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29:198–208, 1983.
- [5] Q. Li F. Monrose, M. K. Reiter and S. Wetzel. Cryptographic key generation from voice. *IEEE Symposium on Security and Privacy*, pages 200–213, 2001.
- [6] J-P. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. *Proc. AVBPA 2003, 4. International Conference on Audio- and Video-Based Biometric Person Authentication*, 4:393–402, Sep 2003.
- [7] A. Perrig M. McCune and M. K. Peiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. in *Symposium on Security and Privacy. Oakland, USA: IEEE*, May 2005.
- [8] J. Solis G. Tsudik M.T. Goodrich, M. Sirivianos and Ersin Uzun. Loud and clear: Human verifiable authentication based on audio. (*Accepted for publication in ICDCS 2006*), Jul 2006.
- [9] K. Kostiaainen N. Saxena, J-E. Ekberg and N. Asokan. Secure device pairing based on a visual channel. (*Accepted for publication in:2006 IEEE Symposium on Security and Privacy* ), May 2006.
- [10] NJIT. Personalized weapons technology project, progress report. Technical report, New Jersey Institute of Technology, April, 2001.
- [11] J. A. Kauffman R. N. Veldhuis, A. M. Bazen and P. Hartel. Biometric verification based on grip-pattern recognition. *Proceedings- SPIE The International Society for Optical Engineering*, 5306:634–641, 2004.
- [12] S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. *Lecture Notes in Computer Science*, 3621:309 – 326, Nov 2005.
- [13] F. Wong and F. Stajano. Multi-channel protocols. *Proceedings of 13th International Workshop on Security Protocols*, Cambridge England, LNCS, Springer-Verlag:to appear, 2005.
- [14] T. Wu. The secure remote password protocol. Proceedings of the Internet Society Symposium on Network and Distributed System Security:97–111, 1998.
- [15] J. Katz R. Ostrovsky X. Boyen Y. Dodis and A. Smith. Secure remote authentication using biometric data. In *Advances in Cryptology—EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 147–163. Berlin: Springer-Verlag, 2005.