Secure pairing with biometrics

Ileana Buhan*

Philips Research,
Eindhoven
E-mail: Ileana.buhan@philips.com
*Corresponding author

Bas Boom

University of Twente,
Enschede, The Netherlands
E-mail: B.J.Boom@utwente.nl

Jeroen Doumen

Irdeto, Eindhoven
E-mail: jdoumen@irdeto.com

Pieter H. Hartel and Raymond N.J. Veldhuis

University of Twente,
Enschede, The Netherlands
E-mail: pieter.hartel@utwente.nl
E-mail: R.N.J.Veldhuis@utwente.nl

Abstract: *Secure pairing* enables two devices that share no prior context with each other to agree upon a security association, which they can use to protect their subsequent communication. Secure pairing offers guarantees of the association partner identity and it should be resistant to eaves dropping and to a man-in the middle attack. We propose the SAfE pairing system, a user friendly solution to this problem. Details are presented along with a discussion of the security features, experimental validation with two types of biometric data (face recognition and hand grip pressure pattern) and a usability analysis for face recognition biometric pairing.

Keywords: ad-hoc authentication; biometrics; fuzzy extractors; secure pairing.

Reference to this paper should be made as follows: Buhan, I., Boom, B., Doumen, J., Hartel, P.H. and Veldhuis, R.N.J. (2009) 'Secure pairing with biometrics', *Int. J. Security and Networks*, Vol.

Biographical notes: Ileana Buhan is currently a Research Scientist at Philips Research, Eindhoven. She received her PhD from the University of Twente, the Netherlands in 2008. Since 2004 she has been doing research in the area of security with noisy data and secure spontaneous interaction. In 2008 she received the EBF European Biometric Research Industry Award for her work on combining secure spontaneous interaction with biometrics.

Bas Boom is a PhD student at the Signals and Systems group of the University of Twente, in the field of biometrics. He received his Master's Degree in Computer Science from Free University of Amsterdam. His research topics are face detection, face registration and face recognition in video surveillance environments.

Jeroen Doumen is currently an Assistant Professor at the University of Twente. He received his PhD in 2003 from the Eindhoven University of Technology. His current research interests are on the border between cryptology and coding theory, in particular operations on encrypted data and handling fuzzyness in cryptographic protocols.

Pieter H. Hartel received his PhD in Computer science from the University of Amsterdam, The Netherlands, in 1989. He was with CERN, Geneva, Switzerland and the Universities of Nijmegen, Amsterdam, and Southampton (UK). He is currently a full Professor of Computer Science at the University of Twente, Enschede, The Netherlands. His research area is distributed and embedded systems security.

Raymond N.J. Veldhuis received his Engineer Degree in Electrical Engineering in 1981 from the University of Twente, The Netherlands. In 1988 he received the PhD degree from Nijmegen University. From 1982 until 1992 he worked as a researcher at Philips Research Laboratories in Eindhoven in various areas of digital signal processing, such as audio and video signal restoration and audio source coding. From 1992 until 2001 he worked at the IPO (Institute of Perception Research) Eindhoven in speech signal processing and speech synthesis. He is now an Associate Professor at Twente University, working in the fields of signal processing, biometrics, pattern recognition and template protection.

1 Introduction

Mobile devices are designed to interact anytime, anywhere. In many scenarios however is it desirable to associate devices in a secure way. For example when using a mobile phone to pay for tickets or when sharing private contact information via the wireless link in an unsecured environment. This problem is known in the literature as secure device association (Kindberg and Zhang, 2003). Solutions have to be specifically designed such that secure association can be realised between previously unassociated devices. Security means that the solution must offer guarantees of the association partner identity and must be resistant to eavesdropping and to a man-in the middle attack. The ideal solution must provide a balance between security and ease of use.

1.1 Scenario

When two users, Alice and Bob, meet at a conference and decide to exchange business cards or other documents, they talk for a while until they trust each another sufficiently to exchange information. However, they do not wish other participants to eavesdrop on their communication or to tamper with their documents. At this stage the only secure association that they have is their trust in each other. To set up a secure association between their devices a protocol is needed that can transfer this trust to their devices. It is not enough for Alice's device to guarantee a secure pairing with device: 128.196.1.3. Alice needs to know that there is a secure association with Bob. Kindberg and Zhang (2003) use the term physical validation for this type of trust transfer. Physical validation can be seen as the physical counterpart of cryptographic authentication of identity. The strength of the physical validation depends on the length of the key established after pairing. Our solution is a protocol that can transfer the trust relation between people to a trust relation between devices using biometrics as the main tool, offering strong physical validation.

1.2 User friendliness

The most important reason why security often fails is the lack of user friendliness. To establish a secure communication, Alice and Bob have to agree on a key. From a usability point of view we want Alice and Bob to have minimal interaction with their devices, and the technical difficulty of the required task should be no worse than to dial a number on a mobile phone. Also we do not like the idea of Alice and Bob having to remember a password or a pin code for establishing the communication key. A user friendly solution is readily provided by appropriate use of biometrics, since a fingerprint or the image of a face has the advantage that it cannot be lost or forgotten and is thus always available.

1.3 Contribution

We present a practical solution to the secure device association problem where biometrics are used to establish a common key between the pairing devices. Our approach has at least two major advantages. Firstly, it offers the possibility to transfer trust from humans to machines without any available security infrastructure. Biometric authentication offers physical validation, thus guaranteeing the identity of a device owner. Secondly, the process is short and we believe user friendly. We propose a protocol in which the keys extracted from biometric data are combined to form a session key. The idea is both simple and effective. Suppose that two users wish to set up a secure communication channel. Both own a biometrically enabled handheld device (for example with face recognition biometrics). Both devices are equipped with a biometric sensor (a camera for face recognition) and a short range radio. Each device is capable of recognising its owner for example by face recognition (Beumer et al., 2005). Then the users take each others picture. Each device now contains a genuine template of its owner and a measurement that approximates the template of the other user. The idea is that each device calculates a common key from the owner template and the guest measurement. In our solution, all Alice has to do to set up a secure communication with Bob is to take a picture of him and let Bob take a picture of her. The protocol is even more general: it can be applied on any type of biometric channel. Our protocol is innovative compared to a key exchange protocol in the sense that legitimate users have to 'find' the communication key by performing a related key search attack. The reasons are twofold. Firstly, fuzzy extractors can create a repeatable sequence out of biometric and our key search mechanism helps lower the error rates of the fuzzy extractor in a practical situation. Secondly, the key search mechanism uses the unpredictable randomness between two measurements as a random salt for the session key thus strengthening the key.

1.4 Road map

We start with a description of related work in Section 2 to put this paper into perspective. Section 3 gives general background information regarding biometrics and we describe the notation used in the rest of the paper. Extracting keys from biometric data is an entire research field on its own; we dedicate Section 4 to summarise the main results from this topic. In this section, we describe how a reliable, uniformly random sequence can be extracted from noisy data such as biometrics highlighting the tradeoffs that have to be made and we give two examples that can be used in a practical setting. Section 5 is dedicated to the pairing protocol. In Section 6, we look at security properties achievable against two powerful adversaries Eve and Charlie. Eve is an eavesdropper. She can record messages sent between Alice and Bob and try to find the key used to secure their messages. The other adversary, Charlie cannot search for the key but he has complete control over the communication environment so that he can listen, or modify any message. These two adversaries correspond to two different but complementary views on security: computational security and formal security. In Section 7, we validate our protocol by experiments on real life biometric data. We look at two different flavours of biometric recognition: hand grip pressure pattern recognition and face recognition. Results obtained from these experiments are promising. Results of a usability study regarding the secure device association using face recognition are presented in Section 8. Finally conclusions are presented in Section 9.

2 Related work

Saxena et al. (2006) define the *pairing problem* as enabling two devices that share no prior context, to agree upon a security association that they can use to protect their subsequent communication. Pairing is intensively studied in the area of pervasive and mobile computing.

Most protocols for secure spontaneous interaction rely on two channels to perform the pairing process. The first, in-band channel, has high bandwidth but no security properties while the second, out-of-band, channel has limited bandwidth while offering additional security properties. There are two approaches in performing secure device association. The first approach uses the out-of-band channel to verify keys exchanged on the in-band channel with human assistance. We call this approach out-of-band verification. The second approach uses the out-of-band channel to send a secret but small message from which the common communication key is then derived and then the key is verified on the in-band channel. We call this approach in-band verification.

Different flavours of out-of-band channels have been proposed that depend on the available hardware equipment, achievable bandwidth, offered security properties and requirements for user interaction with the devices. We summarise the history and evolution of the most well known out-of-band channels.

Stajano and Anderson (1999) brought the secure device pairing problem to the attention of the research community. They propose to use physical interfaces and cables as the out-of-band channel. The physical channel has a high bandwidth and offers confidentiality, authenticity and integrity. It is however impractical since all possible physical interfaces have to be carried around at all times.

Balfanz et al. (2002) propose to use a physically constrained channel (e.g., infrared) to establish a secure association between devices in close proximity. They advanced the state of the art by eliminating the need to carry around all the bulky interfaces. However, the disadvantage of this approach is the infrared channel which is slow, and which requires line-of-sight. Bluetooth users can pair devices by introducing the same PIN, usually a 4 digit number in the paired devices.

Shaked and Wool (2005) show how a passive attacker can find the PIN used during pairing. The randomness and length of the PIN number influences the speed with which an attacker can perform this attack (a 4 digit PIN is cracked in less then 0.3 s). To make things worse (Uzun et al., 2007) note in a usability study performed on different strategies for pairing that the choices of PIN numbers are not really random. We make the same observation in Section 8.

McCune et al. (2005) propose to use the visual channel as an out of band channel. In their protocol, called Seeing is Believing (SiB), devices send their public key on the in-band channel while displaying the hash of the public key as a bar code. If the devices have no display, a sticker is suggested for displaying the hash of the public key. If mutual authentication is required both devices should have a camera to photograph the bar codes. SiB does not rely on the human ability to recognise the bar keys. Saxena et al. (2006) propose a variation of the SiB protocol which achieves secure pairing if one device is equipped with a light detector. Goodrich et al. (2006) propose a human assisted authentication audio channel as the out-of-band channel. A text to speech engine is used for vocalising a sentence derived from the hash of a device's public key.

For small, mobile devices Mayrhofer and Gellersen (2007) propose accelerometer based authentication. Devices that need to be securely associated are shaken together and cryptographic keys are generated from data recorded by the two accelerometers. This approach is different from previous solutions in two ways. The first difference is that accelerometer data is used to produce cryptographic keys and the second difference is that the out-of-band channel is used to share the data from which keys are generated and not to authenticate public keys. They report a key length obtained from accelerometer data between 7–14 bits for every second of shaking. By shaking longer the entropy may be increased.

We take a similar approach in the sense that cryptographic key are transferred on the out-of-band channel. We propose to use biometrics as an out-of-band channel. The main advantage of biometrics over accelerometer data is the higher bandwidth that can be achieved, this can establish a key length of up to 60 bits

(when we use face recognition biometrics) or 80 bits (when using hand grip pressure pattern biometrics).

3 Preliminaries

Biometric devices use pattern recognition of individual data found on the body to differentiate individuals. There are two stages in the lifetime of a biometric system. The first stage is the enrollment phase when the biometric system learns the identity of its users by collecting several feature vectors under good conditions and estimating a mean biometric template and a variance for each particular user. The second stage is authentication when a measurement of a user biometric is taken, a feature vector is extracted and compared to the stored template.

In this paper, we refer to two different biometric systems. The first one uses face recognition. Face recognition analyses the characteristics of a person's face image taken with a digital video camera. It measures the overall facial structure, including distances between eyes, nose, mouth, and jaw edges. The second biometric system is a hand grip pressure pattern where the image of the pressure pattern exerted while holding an object can be used to authenticate or identify a person.

We assume biometric measurements of user to have a multivariate Gaussian statistical model. For face biometrics the number of elements of a feature vector (N in our notation) can range between 30 features to about 280 features (Beumer et al., 2005) while for hand grip pressure pattern N is equal to 40 features (Veldhuis et al., 2004).

According to the statistical model a user is specified by a mean vector $t = (t_1, t_2, \ldots, t_N)$, termed in the rest of the paper as the template and a standard deviation vector $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_N)$. By $x = (x_1, x_2, \ldots, x_N)$ we denote a noisy measurement. Due to differences in environmental condition and user behaviour (e.g., changes in the pose for face recognition or the presence of a ring for the hand grip pressure pattern) we expect that each x_i can be perturbed by a small amount of noise respective to t_i. The amount of noise depends on the value of the standard deviation σ_i. If σ_i is small then we expect the difference between x_i and t_i to be small on the other hand if the value of σ_i is large then we expect the difference between x_i and t_i to be large as well.

The error rates of a biometric system are determined by the accuracy with which the matching engine can determine the similarity between a measured sample x and the expected value of the template t. We construct two hypotheses: $[H_0]$ *x and t are sampled from the same probability distribution*; and $[H_1]$ *x and t are not sampled from the same probability distribution*; The matching engine has to decide which of the two hypotheses H_0 or H_1 is true. To express the accuracy of a biometric system the terms False Acceptance Rate, FAR and False Rejection Rate, FRR are used. The FAR represents the probability that H_0 will be accepted when in fact H_1 is true. The FRR represents the probability that the outcome of the matching engine is H_1 when H_0 is true.

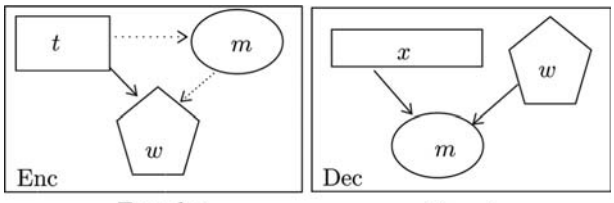
4 Cryptographic keys from biometrics

Our protocol requires the construction of keys from biometric data. In raw form, biometric data is unsuitable to be used as cryptographic key material for two reasons. The first is its representation, usually the real domain while cryptographic keys are represented in the binary domain. The second reason is noise. Two consecutive biometric samples of the same individual will differ by a small, but unpredictable amount of noise while a cryptographic key should be exactly reproducible.

Dodis et al. (2004) propose a general construction termed fuzzy extractor that allows cryptographic keys to be generated from noisy, non-uniform biometric data. In principle a fuzzy extractor does two things: providing error correction to compensate for the unpredictable noise in the biometric and smoothening the non-uniform representation of biometric data.

There are two main components in a fuzzy extractor scheme: the encoder and the decoder. The encoder function is used during enrollment (Figure 1 left) of a user X. As input it takes a low noise template t (for instance obtained by taking multiple low-noise measurements and averaging) of the biometric feature vector and a binary string m (which will be used as a cryptographic key later on), to compute the public sketch w. The binary string m can be extracted from the biometric data itself (Tuyls et al., 2005) or it can be generated independently (Linnartz and Tuyls, 2003). During authentication (Figure 1 right), the decoder function takes as input a noisy measurement x of the users biometric (e.g., a photograph of the user for face biometrics) together with the public sketch w, and outputs the binary string m if the measurement is close enough to the original biometric. The exact reproduction of the binary string m is required to authenticate user X.

Figure 1 A fuzzy extractor is a two step construction. The first step is the encoder which is executed once when the device learns the identity of its owner. The second step is the decoder which is executed each time a secure pairing is performed



Encoder Decoder

There is an important difference between creating the binary key m from biometric data vs. creating the binary key independently of the biometric data. In our construction, we prefer the second option because if the binary key is somehow compromised it is difficult to change the key, because this would mean changing the biometrics, i.e., changing ones face of fingerprint.

Both these algorithms operate component wise on the feature vector. In other words, the noisy measurement will be processed to a feature vector $(x_1, \ldots, x_N)$. From each

x_i and w_i the decoder outputs a binary string m_i (generally consisting of 0-3 bits). In particular, this means that even if some failures occur when processing the complete feature vector, the resulting bit string will still be close to the correct one. Later we show how this property can be used to improve the overall performance of a fuzzy extractor construction.

Three parameters are important for a fuzzy extractor construction. The *robustness* represents the amount of noise tolerated between two measurements x and x' such that m is correctly computed by the decoder. The *key length* represents the length of m in bits and the *entropy loss* (Dodis et al., 2004) measures the advantage that w gives to an adversary in guessing m. We require a fuzzy extractor to have long keys, high robustness and high security (i.e., low entropy loss). However, these are conflicting requirements. Usually the more secure (long key or small entropy loss) the less robust (high values for the error rates FAR and FRR) the fuzzy extractor becomes.

The key length depends on the number of features available. The number of features is a function of the users enrolled in the system and the quality of the measurements. Typically if there are N users in the system the maximum number of features that can be extracted is $N - 1$. However, if the collected data has poor quality the number of used features can be lower compared to the theoretical limit.

In the following, we give two examples of fuzzy extractor schemes to illustrate how one can balance the robustness and key length in a practical setting.

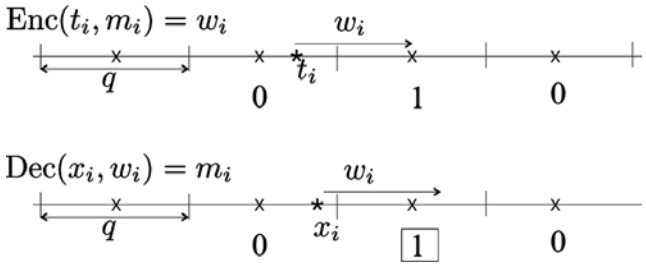
As the first example, let us consider the reliable components scheme of Tuyls et al. (2005) with security parameter s. This scheme assumes that a global estimate of the mean t is known. Enrollment is performed by taking s measurements of the users biometric. If the component i of each of those measurements is always bigger than a chosen threshold μ_i, we set $m_i = 1$. Otherwise, if all measurements are smaller then μ_i, we set $m_i = 0$. In all other cases, the component is not used. The public sketch w is set to 0 or 1 according to whether the feature is used or not.

While the reliable component scheme described above achieves a high robustness, it may result in keys that are too short. Whether or not this method is satisfactory will have to be decided according to the intended use scenario. If a longer key is required, one should look at other fuzzy extractors that embed one (or even more) bits per component of the feature vector, like the schemes proposed by Chang et al. (2004). However, a higher embedding rate does not come for free; it raises the FRR, or the longer key may not even have more entropy than the short one, meaning that it actually does not offer more security despite its greater length (Buhan et al., 2007a).

As second example, we give the fuzzy extractor scheme proposed by Linnartz and Tuyls (2003) known in the literature as the shielding scheme. This construction was one of the first fuzzy extractor constructions that works on continuously distributed data as required for biometric data. They propose to divide the probability density function of each feature component in odd-even bands of equal length q and label the odd-even bands with 1 and the even-odd bands with 0. The embedding of binary data is done by shifting the template distribution mean t_i to the center of the closest even-odd q interval if $m_i = 0$, or to the center of an odd-even q interval if $m_i = 1$. The public sketch w_i is the difference between the location of the mean t_i and the center of the chosen q interval, see Figure 2. During authentication the measurement x_i is shifted by the value of the public sketch w_i and the label of the corresponding interval is output. We describe this construction further in Section 5.3. In the shielding scheme construction the key length is fixed beforehand. More precisely it is equal to the number of features of the biometric template. A trade-off can be made between robustness and entropy-loss by varying the quantisation step (Linnartz and Tuyls, 2003).

The main difference between the reliable component scheme of Tuyls et al. (2005) and the shielding scheme of Linnartz and Tuyls (2003) is the way the cryptographic key is generated. In the first case, the biometric key is extracted from the biometric data, whereas in the second case the cryptographic key is generated independently. The biometric data is used to unlock the value of the pre-generated cryptographic key. Thus if the scheme is compromised a new key can be generated for the same biometric. That is our reason for choosing the shielding scheme in this work.

Figure 2 The (Linnartz and Tuyls, 2003) fuzzy extractor for continuously distributed data. For embedding a bit $m_i = 1$ the encoder function outputs the public sketch w_i which is the difference between the template t_i and the closest middle of a 1 interval; The decoder function adds the measured x_i to the public sketch w_i and outputs the label of the result, in this case 1



As a conclusion, the properties of the biometric data and the selection of the encoding and decoding functions determine the quality (in terms of randomness) of the cryptographic material that can be extracted from it. In the following, we explain the authentication protocol and we analyse the impact of the key quality on the security of the protocol.

5 SAfE protocol

The SAfE protocol establishes a shared secret key between devices whose owners happen to meet and who have no

prior security association. There are three phases in the lifetime of our protocol. The first (past), is the enrolment which can be regarded as a necessary precondition. The second (present), is the SAfE protocol which is the action taken by Alice and Bob to achieve their goal which is secure communication (future), third and final phase. We detail these phases below.

- *Enrollment*, is performed once in the lifetime of the protocol. This step is performed by both Alice and Bob, the participants, independently, for example at home, and it is performed once. Each participant takes multiple (low-noise) measurements of his own biometric, and uses these to calculate his biometric template vector t. Next, each participant picks a random string m, and uses the encoder function of the fuzzy extractor to calculate the matching public sketch w. To differentiate between the participants we use t_A, m_A, w_A for the template, key and public sketch of Alice and t_B, m_B, w_B respectively for Bob.

  After enrollment we have achieved that:

  - the identity of a user can be verified by her own device
  - a device is prepared to be paired up with another device on which the SAfE protocol has been implemented.

- *Pairing*, where the SAfE protocol is used to create a secure channel, a secret key is computed by the decode function of the fuzzy extractor. The protocol description below provides all the details of this step.
- *Secure communication*, when the paired users send messages, documents etc. encrypted with the key derived by the SAfE protocol.

5.1 SAfE details

The SAfE protocol uses two communication channels for key establishment as in the pairing model proposed by Balfanz et al. (2002). One, the in-band channel, is used for authentication. This channel has a high bandwidth but offers no security guarantees. While the second is the out-of-band channel used for pre-authentication. This channel has low bandwidth but offers security guarantees like authentication, integrity or confidentiality. In the SAfE protocol we use the out-of-band channel to exchange a limited amount of information. Later, we use this information to establish the common key by exchanging messages on the in-band channel.

Out-of-band channel

In the SAfE protocol we use biometrics as the out-of-band channel. The first reason for our choice is that biometrics is a source of high entropy data which means high bandwidth compared to other out-of-band channels (e.g., infrared). The type and quality of the biometric modality used (fingerprint, face, iris, palm print) determines the value of the bandwidth capacity for the out-of-band channel. We analyse, in Section 7, the performance of two different biometric modalities: face and grip pressure pattern. The second reason for biometrics as an out-of band channel is that it is easy to send messages on this channel since the main characteristic of biometrics is user friendliness (see Section 8 for the results of usability analysis when face recognition biometric, are used as the out-of-band channel).

For the SAfE protocol, the particular type of biometrics used for sending messages is not important. However, it is interesting to note that the security properties of the out-of-band channel depend on the properties of the biometric used. By default, biometric authentication offers authenticity and integrity. It offers authenticity because we know the source of the message and integrity since the message collected by Alice on the out-of-band channel cannot be changed by a third party. For some biometrics, like hand grip pressure pattern, retina or ear recognition we may even assume channel confidentiality because it is difficult for an adversary to collect a sample of the biometric without the user noticing. We discuss the implications of the properties of the out-of-band channel on the security guarantees of the SAfE protocol in Section 6.

In-band channel

The in-band channel is a broadcast channel (e.g., WLAN) thus all messages sent on this channel are public and can be manipulated.

Message flow

The message flow of the SAfE protocol is presented in Figure 3. Without loss of generality, we may assume that Alice starts the protocol. We explain each of the steps:

0 Bob measures Alice's biometric. This is shown as a transfer of the measurement x_A from Alice to Bob on the biometric channel.

1 Similarly Alice takes a measurement of Bob's biometrics, yielding x_B.

2 Alice broadcasts her public sketch w_A on the wireless channel.

3 Bob feeds the public sketch w_A and the measurement x_A of Alice to the decode function of the fuzzy extractor to compute a key m'_A.

4 Bob broadcasts $w_B, \{x_A\}_{m'_A}$, i.e., the tuple consisting of w_B and the encryption of x_A using key m'_A.

5 Alice uses w_B received in plain in Step 4 and x_B received in Step 1 to compute m'_B with the decoding function of the fuzzy extractor.

6 The second part $\{x_A\}_{m'_A}$ of the message is used to compensate for eventual errors in decoding m_A.

We expect that due to noise or poor quality of the biometric sensor $m_A \neq m'_A$: However, due to their construction m_A and m'_A are close in terms of the Hamming distance so that Alice can perform an efficient key search algorithm to obtain m'_A from m_A. The key search algorithm systematically flips bits in m_A until $\{x_A\}_{m'_A}$ can be decrypted successfully (see the key search algorithm below for details). Since Alice can recognise a measurement of her own biometric, she can check the decryption results.

7 Alice broadcasts $\{x_B\}_{m'_A||m'_B}$.

8 Bob also performs a key search, flipping bits in the concatenation of m'_A and m_B until x_B can be decrypted successfully.

The action on the out-of-band channel 'Bob takes a measurement from Alice' can be translated to: "Bob takes a picture of Alice" when face recognition biometric is used. In this case x_A represents the picture of Alice while x_B represents the picture of Bob (see Figure 4). The same action translates to "Bob hands his mobile device to Alice who holds it firmly" in the case of hand grip pressure pattern generating x_A a grip pressure pattern (see Figure 5).

Figure 3 Message flow for the SAfE protocol showing the steps taken by Alice to the left (5,6) and Bobs actions to the right (3,8) to pair their mobile devices. The steps in the middle represent the message exchange on the out-of-band channel (0,1) and the in-band-channel (2,4,7)

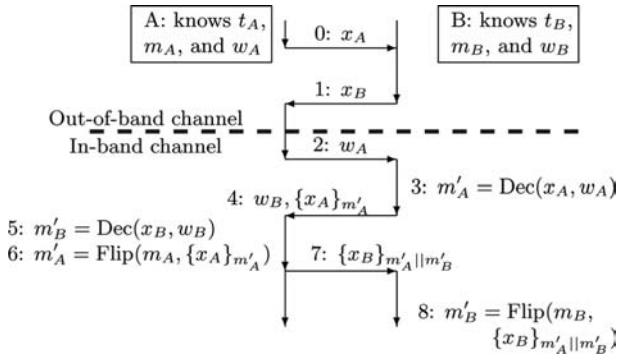
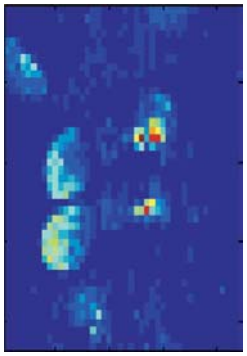


Figure 4 Data transferred on the out-of-band channel for face recognition biometrics (see online version for colours)

Figure 5 Data transferred on the out-of-band channel for hand grip pressure pattern recognition biometric (see online version for colours)



5.2 *Key search algorithm*

In classical symmetric cryptography to decrypt a message encrypted with a key m one must posses m exactly. In particular, with a key m' that differs only in one bit from m, decryption will fail. The SAfE protocol uses this apparent disadvantage of symmetric key cryptography as an advantage: m' is used to form the session key. The noise of the measurements is used as random salt (Wu, 1998) for the session key. The key search algorithm makes it possible to recover m'. Before the algorithm starts we decide on how many trials we make to discover the key. If we set the error threshold to τ bits the algorithm will try out at most $\sum_{i=0}^{\tau} \binom{N}{i}$ combinations before key search failure is declared. Then the protocol has to be restarted or the user gives up.

Alice starts the key search by assuming there are no errors in m'_A, and uses m_A to try and decrypt the encrypted message received in step 4. If decryption fails Alice assumes that there is a one bit difference between m_A and m'_A and so on until she has tried all combinations, i.e., two bits, three bits etc. Finally, when Alice reaches the limit on the number of trials she assumes that the key is coming from an intruder and aborts the protocol. The recovery of m'_A is a related-key attack (Menezes, 1997). When the value of m'_A is discovered, Alice can decrypt the message encrypted with m'_A and recognise x_A by comparing it to t_A. The comparison can be performed by a classifier based matching algorithm designed for this particular biometrics.

A slightly less secure way is to use the decode functionality of the fuzzy extractor to recognise whether the decrypted result x is a measurement of Alice's biometric, by checking if $\mathrm{Dec}(x, w_A)$ is equal to m'_A. The advantage of this method is that the device does not need to store the sensitive template t_A, but only the (fixed) m_A and w_A. Since a fuzzy extractor is designed to correct errors in the (noisy) measurement, not for recognition, we expect this solution to be less secure since m_A is fixed for multiple protocol rounds. Bob performs the same search as Alice, but using m_B and m'_B.

We note that during the protocol both the devices of Alice and Bob have to perform the same amount of computation, which makes the protocol fair.

5.3 Smart key search

When the key space is large the approach described above can become prohibitively expensive and unusable in practical situations. To increase the search speed with which Alice finds m'_A from m_A we propose a method that computes weighting coefficients on each of the key bits. The weight associated with a particular bit represents the probability of error for that bit. The vector of N weighting coefficients for a particular user is the *error profile*. The error profile gives, in fact the order in which bits are flipped. For example assume that 1 bit is changed in m'_A. Without the error profile all N bits are equally likely to flip thus on average Alice will have to perform $\frac{N}{2}$ flips. On the other hand the error profile gives her the position of the most likely bit, giving an advantage.

There is another important reason for using error profile enhanced key search. Due to the nature of the protocol, Alice only has to find variations of her own key m_A and not keys coming from other parties. In particular, this means that we can reduce the FRR without significantly increasing the FAR. We will see in Section 7 how effective this approach can be.

The error profile computation is related to the specifics of the encoder and decoder function implementation. In the evaluation of our protocol we use the fuzzy extractor proposed by Linnartz and Tuyls (2003) as described in Section 4. To calculate the error profile, we give the mathematical description of the encoder and decoder function below. The public sketch is computed by the encoder function as:

$$w_i = \mathrm{Enc}(x_i, m_i) = \begin{cases} \left(2n + \frac{1}{2}\right)q - t_i & \text{when } m_i = 1 \\ \left(2n - \frac{1}{2}\right)q - t_i & \text{when } m_i = 0. \end{cases}$$

Where $n \in \mathbb{Z}$ and is chosen such that: $-q < w_i < q$.

The decoder is defined as:

$$m_i = \mathrm{Dec}(x_i, w_i) = \begin{cases} 1 & \text{when } 2nq \le x_i + w_i < (2n+1)q \\ 0 & \text{when } (2n-1)q \le x_i + w_i < 2nq. \end{cases}$$

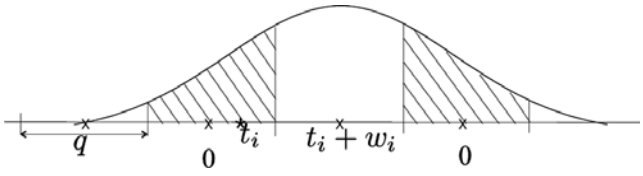
5.3.1 Error profile

Having described the fuzzy extractor above we remind the reader that extractors are not perfect, particularly because during key extraction whenever the distance between the measured x_i and the expected t_i is larger than $\frac{q}{2}$ an error appears. The probability of error is then the probability of a measurement falling outside the chosen odd-even (labelled 1) or even-odd (labelled 0) interval of length q. Figure 6 shows a feature with a normal distribution $N(t_i, \sigma_i)$ when the chosen interval is a 1. During encoding the public sketch w_i shifts the mean of the distribution to the closest 1 interval. The probability of error is then close to the probability of a measurement x_i shifted with the same w_i (the decoding operation) falling in the neighbouring 0 intervals, represented in Figure 6 by the hatched area. The error probability for this feature is computed using the function following:

$$E_i(\sigma_i, q) = \sigma_i\, 2\sqrt{2} \sum_{j=0}^{\infty} \int_{\frac{(1+4j)}{2\sqrt{2}}\frac{q}{\sigma_i}}^{\frac{(3+4j)}{2\sqrt{2}}\frac{q}{\sigma_i}} e^{-x^2}\,\mathrm{d}x$$

$$> \sigma_i\, 2\sqrt{2} \int_{\frac{q}{\sigma_i\, 2\sqrt{2}}}^{\frac{3q}{\sigma_i\, 2\sqrt{2}}} e^{-x^2}\,\mathrm{d}x.$$

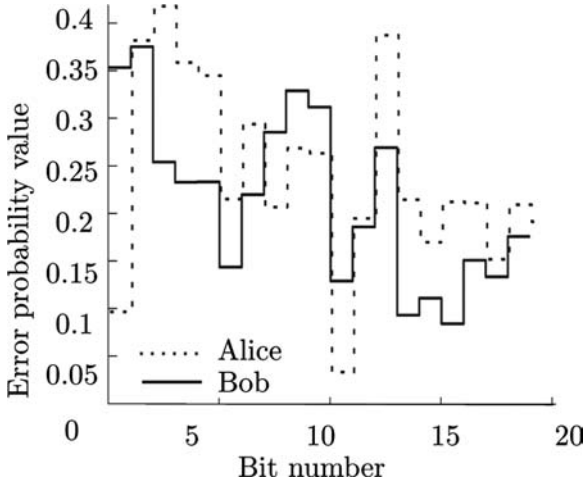
Figure 6 Error computation for a feature element with normal distribution $N(t_i, \sigma_i)$, with quantisation step q



Here, the integral represents the probability associated to one of the 0 labelled intervals of length q (one of the crosshatched intervals) and the summation is done over all the 0 intervals. If q is large enough we can approximate the error as being mostly determined by the 2 neighbouring 0 intervals. Regardless of the chosen 0 or 1 labelled interval the error probability is computed exactly the same.

The error profile is the error probability of all N features of the template t. In Figure 7, we show the error profile for the first 20 features computed on hand grip pressure pattern biometric data for two users named Alice and Bob. We can see that different users have different error profiles.

Figure 7 Error profiles computed for Alice and Bob



5.3.2 Key search with error profile

When the template t and measurement x belong to the same user we expect a small number of errors to appear during decoding. This means that even if m_A and m'_A are different, the difference should not be more then a few bits which can be further corrected using the error profile $e_A = (E_1(\sigma_1, q), \ldots, E_N(\sigma_N, q))$.

Now, the initial Flip function

$$m'_A = \text{Flip}(m_A, \{x_A\}_{m'_A})$$

can be refined as:

$$m'_A = \text{SmartFlip}(m_A, \{x_A\}_{m'_A}, e_A).$$

We start the key search by assuming that there are no errors in m'_A, and we use m_A to decrypt the message $\{x_A\}_{m'_A}$. If decryption fails we assume there is a one bit error. We start flipping one bit of the key according to the position indicated by the largest component of e_A. If the operation is not successful we assume that two bits are wrong and we try combinations of highest two components 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 and the protocol is aborted.

For example, the probability of two components to flip simultaneously can be higher than the highest probability of one component. We leave this as future work.

6 Security analysis

There are two distinct, rigorous views of cryptography that have been developed over the years. One is a formal approach where cryptographic operations are seen as black box functions represented by symbolic expressions and their security properties are modelled formally. The other is based on a detailed computational model where cryptographic operations are seen as strings of bits and their security properties are defined in terms of probability and computational complexity of successful attacks. In the following, we look at both aspects of security to analyse the vulnerability of the protocol to two very different adversaries.

The first adversary, named Charlie is a Dolev-Yao (Dolev and Yao, 1983) intruder who has complete control over the in-band communication channel. He can listen to, or modify messages on this channel. However, Charlie does not have the computational capabilities of Eve. The actions of Charlie on the out-of-band channel depend on the properties of this channel.

The second adversary, named Eve, is a passive adversary, i.e., eavesdropper. She can listen to the communication on the in-band channel and can perform a key search operation similar to Alice and Bob to find the communication key. If the out-of-band channel is not confidential she has access to a noisy version of the information sent on this channel. By modelling this adversary we try to answer the following question: "If both Alice and Bob have to guess the session key, how much more difficult is it for Eve to do the same?". We use the computational model to verify the vulnerability to an eavesdropper such as Eve.

From security point of view we realise that an adversary with the abilities of both Charlie and Eve is a potential threat and we should test the resilience of our protocol to such an adversary. Unfortunately as far as we know there is no formal approach that can handle such an attacker.

We use the formal approach to verify the vulnerability of the protocol to a man-in-the-middle attack. This is an attack where Charlie is able to read, insert and modify at will, messages between Alice and Bob without either party knowing that the link between them has been compromised.

To prevent such an attack keys need to be authenticated.

6.1 Formal verification (Charlie)

We have formally verified that SAfE satisfies mutual authentication and secrecy of messages exchanged after key establishment. The tool used for this purpose is the constraint based security protocol verifier CoProVe (Corin and Etalle, 2002). An earlier version of the protocol was verified and found buggy, the published version of the protocol above 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:

- *No biometric errors.* 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 rounds. When the initiator cannot work out the key the protocol is aborted. In this case we assume that Charlie does not get useful information from the aborted protocol messages.
- *Modelling the out-of-band channel.* We have two types of out-of-band channels:
  - when hand grip pressure pattern biometric is used Charlie cannot listen, modify or send messages thus the out-of-band channel is authentic and confidential
  - when face recognition is used Charlie cannot influence the picture Alice takes of Bob which makes the channel authentic.

  However, Charlie could himself take a picture of Bob. The picture Charlie takes of Bob will be slightly different from the picture Alice takes of Bob. Because systems without an equational theory such as CoProVe, do not have the notion of similarity we verify the protocol with the out-of-band channel in the case of handgrip we leave this as future work. We assume that when the protocol starts Alice knows x_B the biometric of Bob and Bob has x_A the measurement of Alice biometric while Charlie knows neither.

We have verified the model in Figure 3 with the assumptions above. 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.

Resilience to a man-in-the-middle attack depends on the assumptions made. Verification with CoProVe shows that the efforts of Charlie remain unrewarded when he does not have information about the biometric measurements x_A and x_B.

On the other hand, if we assume that Charlie knows the biometric measurements of Alice and Bob, x_A and x_B the protocol is broken. However, in real life situation this assumption is too strong since it is not possible to predict the noise in a biometric measurement and Charlie has no direct access to the measurements that Alice and Bob make. It is possible for Charlie to get a part of x_A and x_B. In the next paragraph, we look at the security guarantees one can hope to achieve when the adversary knows some information about x_A and x_B but not all info.

6.2 Computational analysis (Eve)

When the adversary has some useful initial knowledge as in the out-of-band channel case (b) we look at a different adversary, Eve. To derive keys from fuzzy data we use a related-key attack in steps 6 and 8 of the protocol, to recover the session key. This approach raises two questions: "If both Alice and Bob have to guess the session key, how much more difficult is it for Eve (the intruder) to do the same?", and "What kind of guarantees is this protocol offering?" To answer these questions we study the following scenarios:

AE(0) No previous contact between Alice and Eve.

AE(1) Eve has a measurement of Alice's biometric. From the public string Eve constructs m''_A.

We denote by $W(x \to y)$ the average number of trials that Eve has to do to guess y when she knows x.

We analyse Eve's workload to guess m'_A in the two scenarios above. Alice (and the same holds for Bob) who knows m_A and who has to guess $m'_A = m_A + e$ where the Hamming weight of the noise e is $\mathtt{wt}(e) \leq \tau$, and where τ is an appropriate threshold. As the secret key length is N, there are $\binom{N}{i}$ different error patterns if the actual number of errors is i, thus on average Alice will have to guess (without knowing her error profile):

$$W(m_A \to m'_A) \approx \frac{1}{2} \sum_{i=0}^{\tau} \binom{N}{i}.$$

In scenario AE(1), Eve knows m''_A and has to guess m'_A where $m''_A = m_A + e'$, thus $m''_A = m'_A - e' + e$. Since $\mathtt{wt}(e' - e) \leq 2\tau$, Eve has workload:

$$W(m''_A \to m'_A) \approx \frac{1}{2} \sum_{i=0}^{2\tau} \binom{N}{i}.$$

In scenario AE(0) Eve has no information on Alice thus she has to brute force all possibilities. Thus the number of trials is approximately:

$$W(0 \to m'_A) \approx 2^{N-1}.$$

The scenarios for Bob are analogous:

BE(0) No previous contact between Bob and Eve.

BE(1) Eve records a measurement of Bob.

Eve's workload for guessing m'_B is equal to guessing m'_A in the analogous scenario. To be able to listen on the communication channel Eve has to guess $m'_a || m'_b$ in all scenarios. Table 1 summarises her workload. In each row, we have the information that Eve knows about Bob and in the column the information that Eve knows about Alice. Due to the message flow in the protocol (see Figure 3), Eve might have an advantage if she has information about Alice. Eve can intercept message 4: $w_B, \{x_A\}_{m'_A}$ and recover m'_A if the biometrics allows for taking a decision on whether two measurements come from the same individual. This explains the plus sign between the work of guessing m'_A and the work of guessing m'_B in the columns where Eve has some knowledge about Alice. The amount of work that is required from Eve in the scenarios above is summarised in Table 1. In the worst-case scenario, if Eve has had interactions with both Alice and Bob, this means that Eve has to do a quadratic amount of work compared to either of the participants. In all other cases, there is at least one key that has to be recovered from scratch, making the attack infeasible.

Table 1 Guesswork required for Eve to compute the session key

	AE(0)	*AE*(1)
BE(0)	$W(0 \to m'_A) \cdot W(0 \to m'_B)$	$W(m''_A \to m'_A) + W(0 \to m'_B)$
BE(1)	$W(0 \to m'_A) \cdot W(m''_B \to m'_B)$	$W(m''_A \to m'_A) + W(m''_B \to m'_B)$

We summarise why it is more difficult for Eve to guess the communication key compared to Alice and Bob:

- It is easier to start to guess $m' = m + e$ when m is available, as is the case with the legitimate participants Alice and Bob compared to guessing m' when $m'' = m' + e$ is available as is the case for Eve.
- A very good quality camera for Eve will not improve her workload compared to a legitimate participant. Always Alice has as salt $m'_A = m_A + e_A$ while Eve

will have $m''_A = m_A + e_E = m'_A - e_A + e_E$. With a good camera the best Eve can do is control e_E.

- Alice and Bob work in parallel to find the session key each computing their share while the best Eve can do is find the key sequentially, first find m'_A then find m'_B.
- Alice and Bob have an error profile that Eve does not have.

As a conclusion, SAfE protocol can be assumed to be secure with respect to an eavesdropper for a short lived association as in the case with secure device association.

7 Validation with real life data

We present experiments with two different sets of biometric data: hand grip pressure pattern data and face recognition data for validating the performance of the protocol. The goal of these experiments is to determine whether it is possible for Alice and Bob to determine their own key using the SmartFlip function knowing that biometric recognition is not perfect. We note that simulation results presented in this section were obtained in Matlab on real life data.

7.1 Face recognition biometrics

To verify the potential of constructing cryptographic keys from face recognition data in the ad-hoc settings of our protocol we need a database with face recognition recorded with a mobile device. Since, as far as we know such database is not publicly available we recorded our own database. This database contains 31 individuals. For each individual we recorded four video files using the same mobile device (ETEN M600+, which has a two mega-pixels camera). The four files were recorded in two sessions on two different days, each day we recorded two movies. On the first day, each movie was approximately 10 s. On the second day, we recorded shorter movies of approximatively 5 s. Location of subjects (background), pose and light were different in the two sessions.

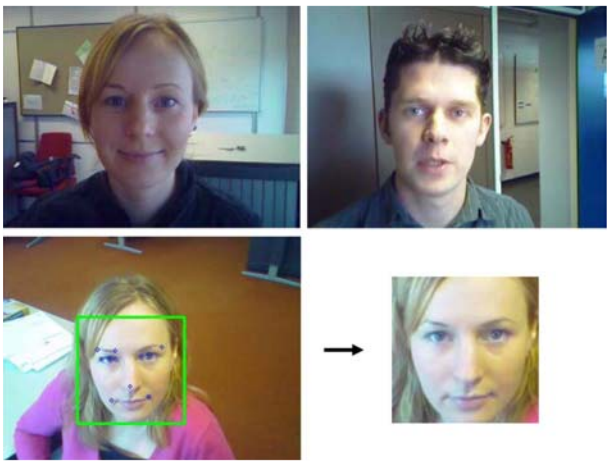
For face recognition, we use the method described in Boom et al. (2006) with manually labelled landmarks on the resolution of 128×128 pixels.

We first trained a generic face model using the Face Recognition Grand Challenge (FRGC) version 1 database. The FRGC v.1 database contains 275 individuals with face images taken both under controlled conditions and uncontrolled conditions. The difference between controlled and uncontrolled conditions can be see in Figure 8 where the same person is captured in controlled conditions (right) and uncontrolled conditions (left).

In the movies we recorded we extract frames which contain the face of the individuals. Movies recorded in the first session resulted into 5994 images that were used during enrollment. Movies recorded in the second session resulted into 2959 images that were used during testing. Images from our mobile database are shown in Figure 9 where the images on the top were recorded in the second session and thus were used for testing and the bottom images were recorded in the first session and were used for testing. In each of these images, we automatically located the faces using the face detection method of Viola and Jones (2001) which finds facial landmarks like eyes, nose and mouth. These landmarks are used to align the faces (see the bottom images of Figure 9). We only used the first 100 correctly found faces for the recognition in both sessions. For each image the region of interest is selected, the background is removed (see Figure 9 bottom left) and the region of interest is normalised to zero mean and unit variance. The difference between the face in the image and the generic face model generated from the FRGC database is computed. As a result each biometric sample can be represented as N (in our case equal to 30) independent feature vectors. On this database, the face recognition is more difficult due to larger deviations in the pose of individuals, illumination and the low quality of the movies. The equal error rate (the point on the (FRR, FAR) curve where the FRR and FAR are equal) using the face recognition algorithm without correction is 15.7%.

Figure 8 Sample face images from FRGC database (see online version for colours)

Figure 9 Sample images from the mobile database (see online version for colours)



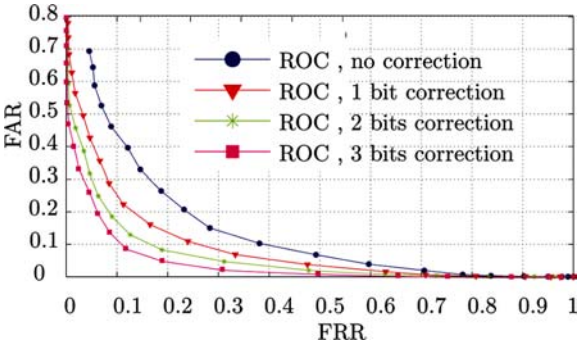
At this stage we apply the shielding scheme fuzzy extractor proposed by Linnartz and Tuyls (2003) to extract cryptographic keys from face data. We use the data collected in the first session to estimate an average face template for each of the 31 users. We generate a random key of 30 bits length for each user. We use the encode function to generate the helper data and the error profile as described in Section 5.3.

To estimate the FRR we do the following: for each user we use the biometric measurements from the second session and the helper data of each user as input to the decode function. The result of this operation is a binary key. We compare this result to the original key generated during enrollment. If they do not match exactly it means that we a have a false rejection. The FRR represents the percentage of the false rejections from the total number of trials.

To estimate the FAR we first choose the target to attack (one particular user). We apply the decode function to all the biometric measurements of the other users and the helper data of the target. The resulting key is compared with the the key of the target if they match we have a false acceptance. The FAR represents the percentage of false acceptance from the total number of trials where all users in the database were target.

By varying the quantisation step q in the encoder function we can tune the FAR and the FRR. The curve obtained by varying the FRR and FAR is called the ROC curve. Figure 10 shows the ROC curves obtained with and without correction. Of interest is the Equal Error Rate (EER) which allows one to evaluate the performance of the fuzzy extractor on the target data as well as the effect of the SmartFlip function. We notice that without any corrections the EER is around 29% with 1 bit correction the EER drops to approximately 19% after further correcting 2 bits the EER is approximately equal to the one obtained by the biometric based classifier 15% and with 3 bits correction we obtain an EER of approximately 12%.

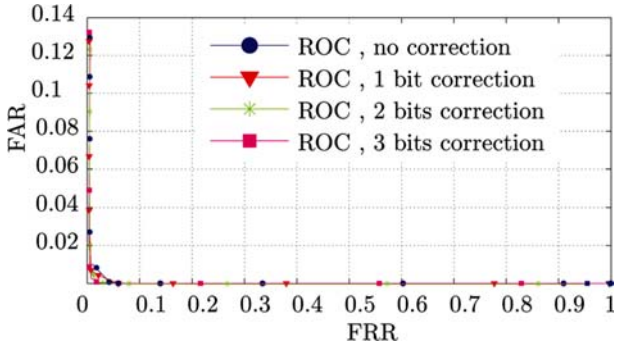
Figure 10 Results on face data, uncontrolled set (see online version for colours)



7.2 Hand grip pressure pattern biometric

The evaluation is performed on real life grip pattern biometric data collected from 41 participants, in one session. A detailed description of this biometric can be found in Veldhuis et al. (2004). Each of the 41 participants contributed 25 different measurements. Approximately 75% of these samples (18), are used for training the algorithm and 25% (7) are used for testing. Firstly, we reduce the dimensionality of the data to the maximum of 40 independent features. For training and testing we use the same data that is used for verification by the classifier based recognition algorithm. Secondly, we construct cryptographic keys using the fuzzy extractor as described above only this time the length of the key is 40 bits. Figure 11 presents the ROC obtained from the collected data. Without corrections the EER on the target data set is around 5%. After, 1 bit correction the EER drops significantly to 3.5% further after correcting 2 bits the EER goes down to 2.7% while correcting 3 bits further lowers the EER to approximately 2%. The EER values are better in the case of hand grip pressure biometric compared to the face data. One of the reasons is that hand data was collected in one session thus the variations between the training data used for enrollment and the testing data is not too large allowing for better authentication performance.

Figure 11 Results on hand data, controlled set (see online version for colours)



7.3 Workload evaluation on test data

We analyse in this paragraph how difficult it is for Eve to guess the communication key when keys are extracted from our mobile data set in four different scenarios as described in Section 6. In this evaluation, the most difficult thing is to give a realistic estimation of the noise. By noise we understand a binary pattern which represents bits that are different between two binary strings or keys. The noise expected for Alice we denote with e and the noise expected for Eve when she takes a picture of Alice with e'. However, when Eve is guessing the communication key she has to guess $e - e'$, see Section 6 for details. Our task is to evaluate from the experimental data the Hamming weights for e and $e - e'$. We make a few observations. As has been showed in Section 6, Eve cannot lower her workload below that of Alice by using a good quality camera. Since Eve does not have the noise free key (m_A is never revealed during the protocol) her expected workload is larger then the workload of Alice. The noise between any two independent biometric measurements is also independent. The noise

expected for Eve or Alice depends on the errors the algorithm can tolerate. Thus, for each point on the ROC curve in Figure 10 the amount of noise will vary.

For a realistic estimation of the noise we adopt the following solution. On the available data set we compute the average number of bits that are different between the keys of all users for each point on the ROC curve. The average values are seen as the noise of the legitimate participants thus represent the Hamming weight of e.

The question now is: if we know e what is a realistic assumption for $e - e'$? We look at two cases:

- worse case scenario (for us) where Eve obtains exactly the same biometric measurements as the Alice and Bob, written formally as $e = e - e'$
- an average case scenario where the e and e' are not identical but they overlap.

The overlap is estimated analytically as the percentage of the total length of the key that the Hamming weight of e represents. Figure 12 shows the Hamming weight of e vs. the Hamming weight of $e - e'$ for different quantisation steps. When the quantisation step is relatively small (few errors are tolerated) the expected noise (the number of bits that are different) is relatively high for both Alice and Eve. The more the quantisation step increases the more errors can be tolerated, the noise decreases and there is less work for Alice but also for Eve.

Figure 12 Expected noise for Alice (dark-blue) and Eve (light-orange) for different quantisation steps (different points on the ROC curve). Eve and Alice use the same type of camera (see online version for colours)

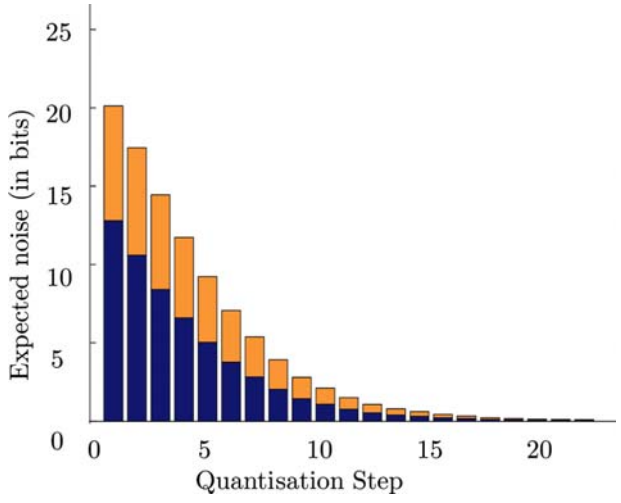
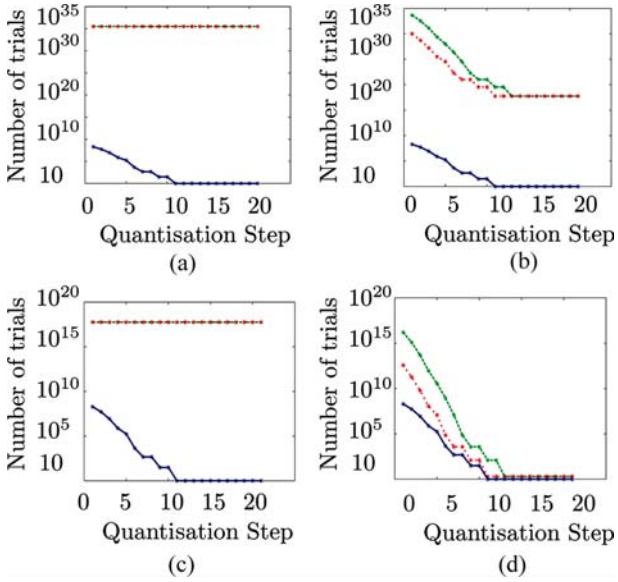


Figure 13 shows the number of trials that Eve has to perform vs. the workload of Alice in the four scenarios described in Section 6. When Eve has no information of Alice and Bob her workload is constant regardless the size of the quantisation interval. In this scenario she will have to make on average 10^{36} trials before she finds the correct key, Figure 13(a).

We look at the quantisation step where the EER is reported, in our case the EER is obtained when the quantisation step is ten. In this point the workload of Eve in the scenario were she has no information of Alice but she has the picture of Bob is approximately 10^{18} trials in the worst case scenario and 10^{20} in the average case, Figure 13(b). When Eve has the picture of Alice but no information of Bob, due to the asymmetry of the protocol she has to perform approximately 10^{17} trials, Figure 13(c). When Eve has both the picture of Alice and Bob she has to make in the worst case the same number of trials (in order of 10th) as Alice and 10^4 in the average case, Figure 13(d). In this case workload of Eve is unacceptably low. A solution is to use another quantisation step. For example when using quantisation step number 3 Alice has to perform on average 10^7 trials while Eve has to make between 10^{10} (worst case) and 10^{14} (average case) trials.

Figure 13 Workload of Eve in worst case scenario (dotted) and average case scenario (dashed) vs. the workload of Alice without using and error profile enhanced search (solid) when (a) Eve has no information about Alice or Bob; (b) Eve has no information over Alice and has the picture of Bob; (c) Eve has the picture of Alice and no information over Bob (d) Eve has the pictures of both Alice and Bob (see online version for colours)



Assume that Alice and Eve can perform one trial operation at the same speed. Assume further that it takes Alice 10 s to perform 10^7 trials (each trial implies setting a new key, a decryption operation and a comparison to decide whether the result is correct). In these settings, it takes Eve in the worst case approximately 2.7 h to find the communication key and three years in the average case.

7.4 Validation experiments conclusion

We offer four conclusions from the evaluation on the two sets of biometric data. The first conclusion is that error rates and thus performance of our protocol depend mostly on the quality of the collected biometric data, regardless of the biometric type of data. The second

conclusion is that the influence of the correction algorithm is significant, however the EER of the fuzzy extractor will be around the EER of the biometric based matcher. Increasing the number of bits that are corrected does not increase linearly the performance of the fuzzy extractor, the most significant improvement is obtained after the first bit of correction after which the improvement decreases. The third conclusion is that the correction mechanism is stable, meaning that the effect of correction is independent of the type of biometric. The fourth conclusion is that it is possible to tune the workload of Eve compared to that of Alice such that security level is acceptable, even when Eve has the picture of both Alice and Bob.

8 Usability analysis

Security only works if people use it therefore we conducted a comparative usability analysis between a PIN based pairing method and SAfE pairing. As a guideline we used the usability study by Uzun et al. (2007) for secure pairing methods. Our results are presented for a comparable target population.

8.1 Test design and procedure

Each subject was given a brief introduction to the secure device association scenario where people need to exchange sensitive information without having any prior security association. The researcher explained that the subject has to try two different pairing methods; one is the standard Bluetooth PIN based pairing method and the other is our SAfE protocol. The subjects were asked to complete a background questionnaire first, so that we could learn about the subject demographics and mobile device usage history. Next, the subject was asked to try both pairing methods in a random order. For the SAfE protocol we wrote a program that implements only the user interaction part of the SAfE protocol. For the PIN based pairing we used the standard Bluetooth pairing method as provided in our device. Each subject was asked to choose a 4 digit PIN number and to enter it. For the SAfE protocol the subject was asked to take a picture of the researcher. All other actions with the PDAs were performed by the researcher. It was explained that only the steps required to perform the pairing are the subject of our experiment. After completing both pairing protocols subjects were asked to fill in the post-test questionnaire. The testing was done in a room with no disturbance and the testing time was around 20 min per subject with at least 15 min of free discussions. During both pairing protocols subjects were using the same ETEN M600 + PDA.

8.2 Participant profile

Our usability experiment had 30 participants from a university environment representing 13 different countries. The demographics such as gender, age and education for our subjects are presented in Table 2. Most of our subjects have a computer science background. The average computer usage history was around 15 years with an average of nine computer hours per day. All participants have a mobile phone, a PDA or a laptop.

Table 2 Participant profile

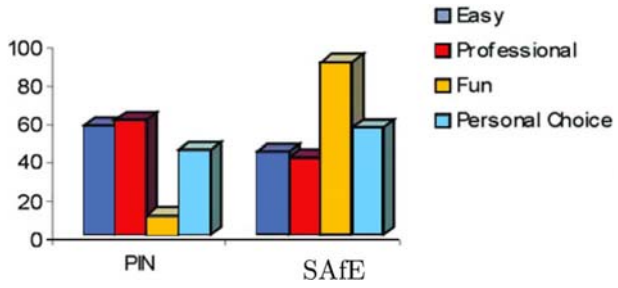
Gender	*Age*	*Education*
Male: 60% Female: 40%	18–24: 10% 25–29: 56% 30–34: 20% 35–39: 7% 40+: 7%	High school: 7% Bachelor: 17% Masters: 46% Doctorate: 30%

8.3 Analysis and discussions

The conclusions drawn from the experiment can be considered only as indicative due to the small number of participants and the (university) biased profile of our subjects.

The main purpose of our experiment was to discover whether users would find it easier to use SAfE protocol compared to a standard 4 digit PIN based pairing. As shown in Figure 14 the score was tight with slightly more people preferring PIN pairing.

Figure 14 Summary of participants opinion (in percent) (see online version for colours)



The explanation for the overall preference for the PIN based method is that subjects are familiar with PIN based security (ATMs, Bluetooth) and typing numbers is natural to subjects with a computing background. Some subjects used the adjective 'easy' to describe the SAfE method. Others found it easy to understand how PIN based pairing method works but they used the word 'magic' to describe the SAfE protocol. We did not try the experiment with a longer PIN and it is worth noting that approximately 80% of our participants choose the same PIN number (1234).

Most of our subjects, 90%, found it fun to perform the pairing using a camera and 73% would like to have both pairing methods on their mobile device (in Figure 14 the percentage of only PIN or only SAfE choices are shown). Due to the 'fun' effect of taking pictures the adjective 'professional' was used more to describe PIN than SAfE.

A separate topic in the questionnaire concerned the privacy effect of giving away a photo to the researcher.

To our surprise 56% of the subjects were not bothered to have their picture taken by a relative stranger. For those 44% who are bothered nothing changes if they have the photograph of the researcher. It was suggested that a privacy guarantee such as "picture deleted after pairing complete" would improve things significantly. To our satisfaction 87% of the users want to have security while communicating wirelessly. Summarising, the usability experiment provides an indication that taking pictures provides a possible route towards creating security associations because it is fun. Whether people believe that taking pictures is professional enough to provide good security is an open question. Bluetooth based pairing method is poor since most subjects use the same pin. A technical report version of this paper Buhan et al. (2007b) provides all the details of the experiments.

9 Conclusions

Secure device association is a challenging problem from both the technical and the user interface point of view. Firstly, users need to exploit a common secret source of randomness from which to extract a shared secret key. Secondly, it should be possible to link the device we connect to with the person who owns it. Thirdly, the process should be simple such that for any person with non technical background the protocol is easy to use.

In this paper, we propose the SAfE protocol which uses biometrics as the out-of-band channel. We analyse our protocol from three different perspectives. Firstly, we analyse the security of the protocol against two types of adversaries Eve which has computational capabilities and Charlie a Dolev-Yao attacker. We show that our protocol is not vulnerable to a man-in-the-middle attack and we analyse eavesdropping in four different scenarios both from theoretical and practical point of view. We show that in the average case when Eve has the biometric measurements of both Alice and Bob her workload is significant. When face data from our database is used to create the communication key and both Alice and Eve execute at the same speed 1 trial operation while Alice finishes in 10 s it would take Eve three years to do the same. Of course Eve can use more powerful computers or execute operations in parallel. Since our protocol is intended for ad-hoc situations were confidential but not critical information is exchanged as long as it would take Eve more then seven days to find the communication key we consider our protocol secure. The workload of Eve, thus the security of the protocol can be increased but it would also increase the error rates. A convenient balance can be found on a case by case basis. It would have been extremely interesting to test the resilience of the protocol against an attacker that has both the abilities of both Eve and Charlie. Unfortunately we are not aware of any formal approach that can handle such an attacker.

Secondly, we evaluate the performance of the protocol with two types of real life biometric data: face recognition and hand grip pressure pattern. Binary keys are generated independently of the biometric data for each protocol round and combined with biometric information. This is a necessary approach since one has only one face, 10 fingerprints, etc. For face recognition we collected face data with a camera of a mobile device, in two different days in uncontrolled environment (light, face expression) as it would be the case in the real world. We obtained on this data set an EER of approximately 12% after applying a correction function that we designed. On the hand grip pressure pattern biometric we obtained a better EER that is below 1%. The main reason is quality of the data, all hand grip data were recorded in one session from trained individuals. As we noted before the quality of biometric data is the main factor that can lower the error rates. A carefully designed data acquiring interface is needed for good performance.

Thirdly, we look at our protocol from the users perspective. Our usability analysis shows that our subjects find the SAfE protocol fun to use, and that they would like to have the SAfE pairing available on their mobile devices. However, there are some situations where SAfE is not appropriate:

- when the participants wish to communicate without drawing attention (such as in a restaurant or at a business meeting)
- when the protocol fails (for example under bad lighting conditions).

Therefore a back-up solution for SAfE is needed that is smoothly integrated with the system. The user would then have the choice of a more user friendly biometric based pairing method and a more robust alternative method.

Acknowledgements

We would like to thank our shepherd Rene Mayrhofer and Kaisa Nyberg for helpful comments and the two anonymous reviewers for suggestions in Section 8. We thank Xiaoxin Shang for providing the hand grip biometric data, Qian Tao for providing help with face recognition data for mobile devices and Vidhan Srivastava for porting the face recognition algorithm on a mobile device.

References

Balfanz, D., Smetters, D.K., Stewart, P. and Wong, H.C. (2002) 'Talking to strangers: authentication in ad-hoc wireless networks', *Network and Distributed Systems Security Symposium (NDSS)*, The Internet Society, Reston, Virginia, San Diego, California.

Beumer, G.M., Tao, Q. and Veldhuis, R.N.J. (2005) 'Comparing landmarking methods for face recognition', *Proc. ProRISC 2005 16th Annual Workshop*, Veldhoven, The Netherlands, November, pp.594–597.

Boom, B.J, Beumer, G.M., Spreeuwers, L.J. and Veldhuis, R.N.J. (2006) 'The effect of image resolution on the performance of a face recognition system', *ICARCV '06, 9th International Conference on Control, Automation, Robotics and Vision*, pp.1–6.

Buhan, I., Doumen, J., Hartel, P. and Veldhuis, R. (2007a) 'Fuzzy extractors for continuous distributions', in Deng, R. and Samarati, P. (Eds.): *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, Singapore, ACM, New York, pp.353–355.

Buhan, I., Doumen, J., Hartel, P. and Veldhuis, R. (2007b) *Secure Ad-Hoc Pairing With Biometrics: Safe*, Technical Report TR-CTIT-06-72, Enschede, http://eprints.eemcs.utwente.nl/10783.

Chang, Y., Zhang, W. and Chen, T. (2004) 'Biometrics-based cryptographic key generation', *International Conference on Multimedia and Expo (ICME)*, IEEE, pp.2203–2206.

Corin, R. and Etalle, S. (2002) 'An improved constraint-based system for the verification of security protocols', in Hermenegildo, M.V. and Puebla, G. (Eds.): *SAS*, Vol. 2477 of LNCS, Springer, pp.326–341.

Dodis, Y., Reyzin, L. and Smith, A. (2004) 'Fuzzy extractors: how to generate strong keys from biometrics and other noisy data', in Cachin, C. and Camenisch, J. (Eds.): *Advances in Cryptology – Eurocrypt 2004, International Conference on the Theory and Applications of Cryptographic Techniques*, Interlaken, Switzerland, 2–6 May, Proceedings, Volume 3027 of LNCS, Springer, pp.523–540.

Dolev, D. and Yao, A. (1983) 'On the security of public key protocols', *Information Theory, IEEE Transactions on*, Vol. 29, pp.198–208.

Goodrich, M., Sirivianos, M., Solis, J., Tsudik, G. and Uzun, E. (2006) 'Loud and clear: human-verifiable authentication based on audio', *26th IEEE International Conference on Distributed Computing Systems (ICDCS 2006)*, IEEE Computer Society, 4–7 July 2006, Lisboa, Portugal, p.10.

Kindberg, T. and Zhang, K. (2003) 'Secure spontaneous device association', in Dey, A.K., Schmidt, A. and Mc-Carthy, J.F. (Eds.): *Ubicomp*, Volume 2864 of LNCS, Springer, pp.124–131.

Linnartz, J. and Tuyls, P. (2003) 'New shielding functions to enhance privacy and prevent misuse of biometric templates', in Kittler, J. and Nixon, M.S. (Eds.): *AVBPA*, Volume 2688 of LNCS, Springer, pp.393–402.

Mayrhofer, R. and Gellersen, H. (2007) 'Shake well before use: authentication based on accelerometer data', in LaMarca, A., Langheinrich, M. and Truong, K.N. (Eds.): *Pervasive*, Volume 4480 of LNCS, Springer, pp.144–161.

McCune, J., Perrig, A. and Reiter, M. (2005) 'Seeing-is-believing: using camera phones for human-verifiable authentication', *IEEE Symposium on Security and Privacy*, IEEE Computer Society, pp.110–124.

Menezes, A. (1997) *Handbook of Applied Cryptography*, CRC Press, Boca Raton.

Saxena, N., Ekberg, J., Kostiainen, K. and Asokan, N. (2006) 'Secure device pairing based on a visual channel (short paper)', *S&P*, IEEE Computer Society, pp.306–313.

Shaked, Y. and Wool, A. (2005) 'Cracking the bluetooth pin', in Shin, K.G., Kotz, D. and Noble, B.D. (Eds.): *MobiSys*, ACM, pp.39–50.

Stajano, F. and Anderson, R. (1999) 'The resurrecting duckling: security issues for ad-hoc wireless networks', in Christianson, B., Crispo, B., Malcolm, J.A. and Roe, M. (Eds.): *Security Protocols Workshop*, Volume 1796 of LNCS, Springer, pp.172–194.

Tuyls, P., Akkermans, A., Kevenaar, T., Schrijen, G., Bazen, A. and Veldhuis, R. (2005) 'Practical biometric authentication with template protection', in Kanade, T., Jain, A.K. and Ratha, N.K. (Eds.): AVBPA, Volume 3546 of LNCS, Springer, pp.436–446.

Uzun, E., Karvonen, K. and Asokan, N. (2007) *Usability Analysis of Secure Pairing Methods*, Technical Report, NRCTR-2007-002, Nokia Research Center.

Veldhuis, R., Bazen, A., Kauffman, J. and Hartel, P. (2004) 'Biometric verification based on grip-pattern recognition', *Security, Steganography, and Watermarking of Multimedia Contents VI*, January 18–22, 2004, Proceedings, Volume 5306 of Proceedings of SPIE, San Jose, California USA, SPIE, pp.634–641.

Viola, P.A. and Jones, M.J. (2001) 'Rapid object detection using a boosted cascade of simple features', *CVPR*, IEEE Computer Society, Vol. 1, pp.511–518.

Wu, T.D. (1998) 'The secure remote password protocol', *Proceedings of the Network and Distributed System Security Symposium, NDSS 1998*, The Internet Society, San Diego, California, USA, paper 3.