

Secure Ad-hoc Pairing with Biometrics: SAfE

Received: date / Accepted: date

Abstract The *pairing problem* is to enable two devices, which share no prior context with each other, to agree upon a security association that they can use to protect their subsequent communication. Secure pairing should offer guarantees of the association partner's identity and it should be resistant to eavesdropping or to a man-in-the-middle attack. We propose a user friendly solution to this problem. Keys extracted from images of the participants are used for authentication. Details of the SAfE pairing system are presented along with a discussion of the security features and a usability analysis.

Keywords ad-hoc authentication · biometrics · fuzzy extractors

1 Introduction

Saxena, *et al.* [14] define the *pairing problem* as to enable two devices which share no prior context with each other, to agree upon a security association that they can use to protect their subsequent communication. This means that a common key must be established. Solutions to this problem have applications in the area of spontaneous interaction between mobile devices. There are two types of challenges in achieving such a secure pairing:

Technical challenges Due to the spontaneous nature of the interaction we cannot rely on any previously shared secret information. A solution to this problem is solved in cryptography by a public key infrastructure (PKI). If Bob wants to send Alice an encrypted message he will use Alice's public key to encrypt the message and send the encrypted text to her. Only the private key corresponding to the public key used can decrypt this message. However, how can Bob be sure that the public key

he has used does not belong to Eve? A commonly accepted solution is the use of certificates validated by a trusted third party which guarantees the authenticity of a public key. However, we cannot assume that wherever Alice and Bob meet there will be a network connection available to connect to a trusted third party, which also knows both Alice and Bob. Another approach to secure pairing, mentioned for the first time by Balfanz, *et al.* [1], is to use a location limited side channel to authenticate the key established on the main channel of communication. A location limited channel (for instance an infrared connection) makes it physically difficult for others to eavesdrop, or to interfere. Devices exchange a limited amount of information over the side channel, which will then allow them to complete the authenticated key exchange protocol over the main channel.

We propose a protocol that can transfer the trust relation between people to a trust relation between devices. Let us elaborate on this idea. 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 pairing 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's device. Kindberg, *et al.* [9] 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.

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 user friendliness 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 at worst similar to that of using 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, forgotten and is always available. However, there is one technical problem that needs to be solved: no two biometric measurements, even coming from the same user and using the same measurement setup are identical. This is due to noise, which is typically Gaussian. In cryptography, good quality cryptographic secret keys are generated from a uniform distribution. Because of the distribution mismatch, biometric data is not directly suitable for use as cryptographic keys. Fuzzy extractors have been introduced by Dodis, *et al.* [6] as a general tool for extracting cryptographic keys from noisy data such as biometrics. Thus fuzzy extractors can be used to extract binary, reproducible key material from biometric data.

Contribution. We present a practical solution to the secure pairing problem where biometrics is used as a side channel in pairing the devices. This 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 intuitive. 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, equipped with a camera and a short range radio (for example a mobile phone or a PDA). Each device is capable of recognizing its owner by face recognition [7]. However, in our protocol 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. The act of taking a picture corresponds to sending a message on the location limited channel. In our solution, all Alice has to do to set up a secure communication with Bob is to take a picture of him.

2 Preliminaries

Biometric devices use pattern recognition of individual data found on the body to differentiate individuals. Once the biometric system receives a signal from its sensor it will extract a feature vector. The feature vector contains the essential characteristics to differentiate between individuals. There are two stages in the lifetime of a biometric system. The first one is the enrolment phase when

the biometric system learns the identity of its users by collecting several feature vectors and estimating a mean biometric template 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. 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 denote by X a random user enrolled in the biometric system. By t we denote the template of user X and x represents a random instance of X . We construct two hypotheses: $[H_0]$ x is sampled from X ; and $[H_1]$ x is not sampled from X ; 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 *false acceptance rate* represents the probability that H_0 will be accepted when in fact H_1 is true. The *false rejection rate* represents the probability that the outcome of the matching engine is H_1 but H_0 is true.

3 Related Work

To describe our protocol we adopt the pairing model proposed by Balfanz, *et al.* [1], which has two stages: (1) *pre-authentication* when the two devices exchange secret information using some particular physical contact, in our case the camera, and (2) *authentication* when the two devices identify each other based on the information exchanged in the pre-authentication stage. In most solutions the pre-authentication channel is used mainly to authenticate public-keys. The hash of the public key is either vocalized [8] or photographed [12]. Others use Diffie-Hellman like key agreement scheme where short sequences transmitted on the private channel authenticate the key establishment exchange on the main channel [18].

Many types of side channels were proposed in the literature each with their own properties. Saxena, *et al.* [14] propose physical contact (i.e. cables) between devices. This type of channel has the property that the user can control precisely which devices are communicating, however it can become too bulky to carry around all the necessary interfaces. The authors then extend this approach to location limited channels, for which they use short range wireless infrared communication.

McCune, *et al.* [12] propose to use a visual side channel and make a photograph of the hash code of a public key. In the same line of work, Googrich, *et al.* [8] propose a human assisted authentication audio channel as a side channel. They use a text to speech engine for vocalizing a sentence derived from the hash of a device's public key. In previous work [3], we use grip pattern biometrics for the secure transfer of biometric templates. Mayrhofer, *et al.* [11] propose accelerometer based authentication for devices without interfaces.

Our protocol is a key-agreement protocol in the sense that both parties equally contribute to the session key. The protocol achieves authentication of both parties because the partial keys are extracted from the biometric identification data of the individual.

4 Extracting cryptographic material from biometric data

Biometric data is unsuitable to use directly as cryptographic key material due to noise. Biometrics work by recognizing a user by his biometric template t , which is a vector consisting of many components (depending on the exact biometric). When a users biometric is measured, generally there will be some noise; we will denote the measurement of the biometric with x . To combat this noise problem, Dodis, *et al.* [6] propose to use fuzzy extractors. Also, fuzzy extractors solve the template storage issue - it is enough to store a (potentially public) sketch to recognize a person, instead of the sensitive template itself.

A fuzzy extractor consists of two algorithms: Encoding and Decoding. Encoding is used during enrolment (Figure 1 left) of a user X . As input it takes a noise-free version 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 [16] or it can be generated independently [10]. During authentication (Figure 1 right), the decoding algorithm takes as input a noisy measurement x of the users biometric together with the public sketch w , and outputs the binary string m if the measurement is close enough to the original biometric. Note that generally both these algorithms operate componentwise on the feature vector. In other words, if the feature vector t has N components t_1, \dots, t_N , the noisy measurement (eg. a photograph of the user for face biometrics) will be processed to a feature vector $\{x_1, \dots, x_N\}$. Each of these components will have its own string m_i (generally consisting of 0-3 bits), and its own public sketch w_i . 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. For example, consider the reliable components scheme of Tuyls, *et al.* [16] with security parameter s . This scheme assumes that an estimate of the global mean μ_g is known (for each feature vector component). Enrollment is performed by taking s measurements of the users biometric. If each of those measurements is larger than the mean, we get $m = 1$. Otherwise, if all measurements are smaller than the mean, we have $m = 0$. 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.

Three parameters are important for the performance of fuzzy extractors. Firstly, the reliability represents the

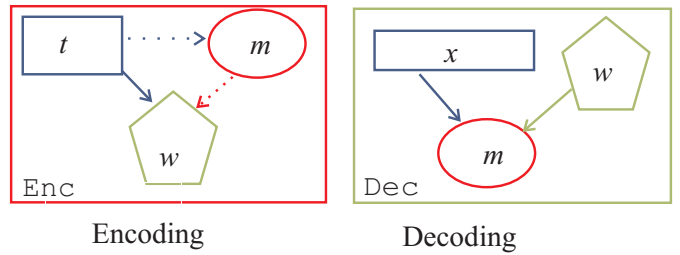


Fig. 1 Cryptographic material from biometric data

probability of an identification error; it is computed as $1 - FRR$. Secondly, the embedding rate is the number of bits that is embedded in each component of the feature vector. The higher the embedding rate, the longer the resulting key. In previous work [2], we show that the FAR determines an upper bound on the number of truly random bits that can be extracted from a noisy sequence. Thirdly, the leakage quantizes the amount of secret information leaked by publishing the public string. It is characterized by Linnartz, *et al.* [10] as the Shannon mutual information between the secret information m and the public sketch w . Having low leakage means that (almost) no information is disclosed on the user data or secret key.

Face biometrics typically have about 280 features [15]. In order to achieve a key length of 128 bits, this means that we need to embed about $\frac{1}{2}$ bit per feature vector component. While the reliable component scheme described above achieves a high reliability, in this case it may result in too short keys. Whether or not this is enough will have to be decided on a case-by-case basis. 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.* [5]. 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.

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 analyze 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.

Protocol preliminaries. The SAFE protocol has three steps: 1. *Enrolment* is performed once in the lifetime of the protocol. This step is performed by each of the participants

independently, for example at home, and it is performed once. Each participant X takes multiple (low-noise) measurements of his own biometric, and uses these to calculate his biometric template vector t_X . Next, each participant picks a random string m_X , and uses the Encode functionality of the fuzzy extractor to calculate the matching public sketch w_X . After enrolment we have achieved that: (1) the identity of a user can be verified by her own device, and (2) a device is prepared to be paired up with another device on which the SAfE protocol has been implemented.

2. *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 detail of this step.

3. *Secure communication* when the paired users send messages, documents etc. encrypted with the key they derived by the SAfE protocol.

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 help Alice discover m'_A . We expect that due to noise or poor quality of the biometric sensor $m_A \neq m'_A$: However, we also expect that 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 recognize 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.

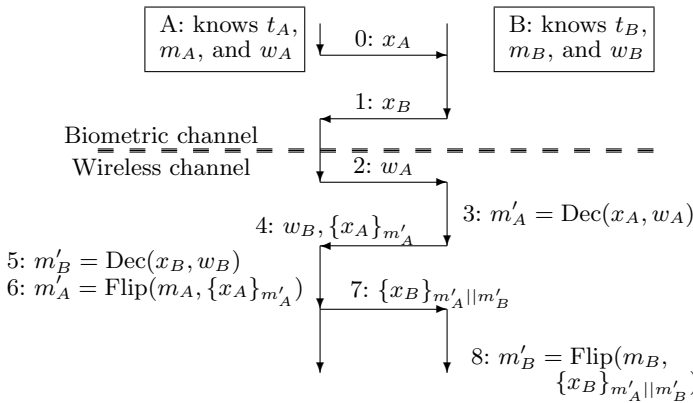


Fig. 2 Message flow for the SAfE protocol showing the steps taken by Alice to the left and Bobs actions to the right. The steps in the middle represent the message exchange.

The Pairing Protocol Before the protocol starts each of the devices knows the data of its owner, i.e. the template, the key and the public sketch. Thus, Alice knows initially t_A , m_A , and w_A and Bob knows t_B , m_B , and w_B . The message flow of the SAfE protocol is presented in Figure 2. Without loss of generality we assume that Bob starts the protocol. We explain each of the 9 steps: 0: Bob takes a picture of Alice's face. This is shown as a transfer of the measurement x_A from Alice to Bob on the biometric channel.

1: Similarly Alice takes a picture of Bob's, 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 .

Optionally, if a higher security level is desired, Alice and Bob can make sure they arrive at the same key by having both devices calculate a checksum on $x_A \| x_B$, displaying the number to the users so that they can check that the numbers agree. There may be some hesitation about storing the actual template t_A on the device. This is not strictly necessary, but gives more certainty in the key search phase as described below. Also, when the device poses the template t_A , it is possible to calculate a fresh m_A for every instance of the protocol.

We note that (a) during the protocol both the devices of Alice and Bob have to perform the same amount of computation, which makes the protocol fair, and (b) when a user transmits his public sketch w , some information is leaked about his biometric template t and his bitstring m .

Key search algorithm In classical symmetric cryptography to decrypt a message encrypted with a key m one must possess m . 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 [19] 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 $\sum_{i=0}^{\tau} \binom{l}{i}$ combinations before key search failure.

Alice starts the key search by assuming there are no errors in m'_A , and uses m_A to 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. The recovery of m'_A is a related-key at-

tack [13]. When the value of m'_A is discovered, Alice can decrypt the message encrypted with m'_A and recognize x_A by comparing it to t_A . The comparison is 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 recognize whether the decrypted result x is a measurement of Alice's biometric, by checking if $Dec(x, w_A)$ 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 actually correct errors in the (noisy) measurement, not for recognition, we expect this solution to be somewhat less secure. Bob performs the same search as Alice, but using m_B and m'_B .

6 Security Analysis

We want to prevent two attacks. One is an intruder eavesdropping on the communication line and second is a man-in-the-middle attack. We assume the adversary, Eve to be a Dolev-Yao intruder which means that she has complete control of the main wireless communication channel. She can listen or modify messages on the communication channel between the devices. However, Eve cannot send or tamper with messages on the secure biometric side channel. We will assume that Eve has the ability to compare two biometric measurements, and that she can determine whether or not these come from the same person.

Eavesdropping 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 records a measurement of Alice's biometric from a previous round of the protocol. From the public string Eve constructs m''_A .

We denote by $W(x \rightarrow y)$ the average number of trials that Eve has to do to guess y when she knows x . We analyze 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 is $\mathbf{wt}(e) \leq \tau$, and where e is the noise and τ is an appropriate threshold. As the secret key length is l , there are $\binom{l}{i}$ different error patterns if the actual number of errors is i , thus on average Alice will have to guess:

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

Table 1 Guesswork required for Eve to compute the session key

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

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 $\mathbf{wt}(e' + e) \leq 2\tau$, Eve has workload:

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

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:

$$W(0 \rightarrow m'_A) \approx \frac{2^l + 1}{2}.$$

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 summarizes 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 2), 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 summarized in table 1. In the worst-case scenario, if Eve has had interactions with both Alice and Bob, this means that Eve only has to do twice as much work as 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.

MiM attack. A man-in-the-middle attack is an attack where Eve 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. Eve must be able to observe and intercept messages exchanged between the two victims. To prevent such an attack keys need to be authenticated. Key authentication is achieved by the optional verification function in the last step of the protocol. During verification a check sum is computed on the pair (x_A, x_B) . Alice authenticates

x_A , by successful decryption of message 4 while Bob authenticates x_B by successful decryption of message 7. If the check sums computed by both device match upon visual comparison it means key $m'_A||m'_B$ is authentic.

7 Usability Analysis

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.* [17] for secure pairing methods. Our results are presented for a comparable target population.

Test design and procedure Each subject was given a brief introduction to the spontaneous interaction 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 minutes per subject with at least 15 minutes of free discussions. During both pairing protocols subjects were using the same ETEN M600+ PDA.

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 9 computer hours per day. All participants have a mobile phone, a PDA or a laptop.

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 3 the score was tight with slightly more people preferring PIN pairing.

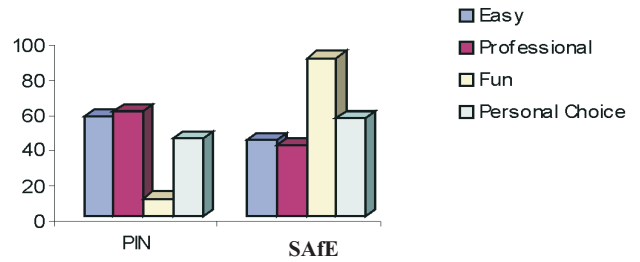


Fig. 3 Summary of participants opinion (in percent)

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.

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 3 the percentage of only PIN or only SAfE choices are shown). Due to the game 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. They suggested that it would make them feel better if they could have a privacy guarantee such as 'picture deleted after pairing complete'. To our satisfaction 87% of the users want to have security while communicating wirelessly. A technical report version of this paper [4] provides all the details of the experiments.

Table 2 Participant profile.

Gender	Age	Education
Male: 60% Female: 40%	18–24: 10%	High school: 6.66%
	25–29: 56.6%	Bachelor: 16.66%
	30–34: 20%	Masters: 46.66%
	35–39: 6.66%	Doctorate: 30.66%
	40+: 6.66%	

8 Conclusion

Secure spontaneous device pairing is a challenging problem from both the technical and 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.

We propose to use biometrics as a secure side channel. Our SAFe protocol offers physical validation while at the same time the protocol is easy to use. We analyze the resilience to eavesdropping and a Man-in-the-Middle (MiM) attack (if the optional verify step is added) in the general setting of a Dolev-Yao intruder. We show that our protocol is not vulnerable to a MiM attack and we analyze eavesdropping in four different scenarios. We show that in the best case scenario for the eavesdropper her workload grows exponentially faster than the workload of Alice and Bob (the participants to the protocol).

The usability analysis shows that our subjects find the SAFe method 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: (a) when the participants wish to communicate without drawing attention (such as in a restaurant or at a business meeting) (b) 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.

We are working on a complete prototype that runs SAFe on two mobile devices. We are particularly interested in testing the influence of different environments on the key search failure since environmental effects such as changing the light conditions can seriously affect the face recognition performance.

9 Acknowledgments

We would like to thank our shepherd Rene Mayrhofer and Kaisa Nyberg for helpful comments and the two anonymous reviewers for suggestions on section 7.

References

1. D. Balfanz, D. K. Smetters, P. Stewart, 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.
2. I. Buhan, J. Doumen, P. Hartel, and R. Veldhuis. Fuzzy extractors for continuous distributions. In R. Deng and P. Samarati, editors, *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, Singapore, pages 353–355, New York, March 2007. ACM.
3. I. Buhan, J. Doumen, P. Hartel, and R. Veldhuis. Feeling is believing: a secure template exchange protocol. *To appear in ICB 2007, South Korea*, 2007.
4. I. R. Buhan, J. M. Doumen, P. H. Hartel, and R. N. J. Veldhuis. Secure ad-hoc pairing with biometrics: Safe. Technical Report TR-CTIT-06-72, Enschede, July 2007. <http://eprints.eemcs.utwente.nl/10783>.
5. Yao-Jen Chang, Wende Zhang, and Tsuhan Chen. Biometrics-based cryptographic key generation. In *ICME*, pages 2203–2206. IEEE, 2004.
6. Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *LNCS*, pages 523–540. Springer, 2004.
7. A.M. Bazen G.M. Beumer, Q. Tao and R.N.J. Veldhuis. Comparing landmarking methods for face recognition. *Proc. ProRISC 2005 16th Annual Workshop, Veldhoven, The Netherlands*, pages pp. 594–597, Nov 2005.
8. M. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud and clear: Human-verifiable authentication based on audio. In *ICDCS*, page 10, 2006.
9. T. Kindberg and K. Zhang. Secure spontaneous device association. *5th International Conference on Ubiquitous Computing*, pages 124–131, 2003.
10. J.P. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In Josef Kittler and Mark S. Nixon, editors, *AVBPA*, volume 2688 of *LNCS*, pages 393–402. Springer, 2003.
11. Rene Mayrhofer and Hans Gellersen. Shake well before use: Authentication based on accelerometer data. In *pervasive*, pages 144–161, 2007.
12. J. McCune, A. Perrig, and M. Reiter. Seeing-is-believing: using camera phones for human-verifiable authentication. *Security and Privacy, 2005 IEEE Symposium on*, pages 110–124, 2005.
13. A.J. Menezes. *Handbook of Applied Cryptography*. CRC Press, 1997.
14. N. Saxena, J.E. Ekberg, K. Kostianen, and N. Asokan. Secure Device Pairing based on a Visual Channel (Short Paper). *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, pages 306–313, 2006.
15. Q. Tao and R. Veldhuis. Verifying a user in a personal face space. In *9th Int. Conf. Control, Automation, Robotics, and Vision*, Singapore, 2006.
16. P. Tuyls, A. Akkermans, T. Kevenaer, G. Schrijen, A. Bazen, and R. Veldhuis. Practical biometric authentication with template protection. In Takeo Kanade, Anil K. Jain, and Nalini K. Ratha, editors, *AVBPA*, volume 3546 of *LNCS*, pages 436–446. Springer, 2005.
17. E. Uzun, K. Karvonen, and N. Asokan. Usability Analysis of Secure Pairing Methods. Technical report, NRC-TR-2007-002, Nokia Research Center, 2007.
18. S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. *LNCS*, 3621:309 – 326, Nov 2005.
19. T. D. Wu. The secure remote password protocol. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 1998, San Diego, California, USA*. The Internet Society, 1998.