Random Number Generation Based on Fingerprints

Shkodran Gerguri, Václav Matyáš, Zdeněk Říha, and Luděk Smolík

Faculty of Informatics, Masaryk University, Botanická 68a
60200 Brno, Czech Republic

Abstract. Current research often focuses on the design of new methods that extend the well-established role of biometrics in authentication and identification into key management and cryptography. Methods have been proposed that involve biometric-governed access to cryptographic keys, as well as methods that utilize biometric samples to derive keys for specialized encryption algorithms. The aim of this paper is to present a possible method for random number generation from repeated measurements of fingerprints as an alternative approach to the proposed applications of biometrics in cryptography, and to analyze some of its properties. Such method could provide a valuable source of ranomness in mobile devices equipped with a fingerprint scanner.

1 Introduction

Biometric systems have become a well-established technology in modern authentication and identification solutions. The ready availability of biometric scanners, and their widespread integration into increasingly larger numbers of consumer products such as low-cost laptops and mobile phones have presented a viable alternative to traditional systems based on passwords. With the technology available, the research focus has been on improving the matching algorithms that govern the identity decision process, with the aim of decreasing the False Accept and False Reject Rates (i.e., the percentage of users incorrectly authenticated as some valid user, or rejected as an invalid one, respectively). More recently, researchers also began exploring the possibility of further applications of biometrics beyond simple authentication and identification mechanisms. The focus has been on reliable methods of key management using biometrics, in an effort to replace traditional methods based on passwords.

We believe biometrics have another potential application. Inherent to each biometric measurement is a variability, which is the result of different measurement conditions and ways in which the user presents his or her features to the scanner. This variability effectively represents randomness, which, if extracted, could then be used as a seed for pseudorandom number generators, or as a random number itself.

Surprisingly, there has been little research done on the suitability of biometrics as a direct source of randomness. Sczepanski et al. have devised a general

P. Samarati et al. (Eds.): WISTP 2010, LNCS 6033, pp. 170–182, 2010.

method [10] for random number generation from a large dataset of decimal values, and evaluated the performance of galvanic skin response and neurophysical brain signals. To the best of our knowledge, this has been the only such method proposed so far.

The aim of this paper is to present a possible approach towards random number generation using biometrics, with fingerprints as the biometrics of choice. Specifically, the main goal of this method is to provide on-demand session key generation capabilities, typically for symmetric key generation, and to authenticate the user in the process – we do not aim for continuous, high bitrate as from some other random number generators. We then analyze several entropy-contributing factors of the method and provide the results of our analysis.

The focus here is on experimental analysis; mathematical proofs of security are beyond the scope of the paper. The main goal of this paper is not an introduction of a (new) fundamental mathematical theory for entropy extraction from biometric data, but rather an initial investigation of user accessible randomness source, with an experimental study within common environments like mobile phones. We have chosen the biometric data as a new and promising information source and we have developed a simple algorithm for extraction of the inherent randomness introduced by user interaction with the mobile device. We are aware that our treatment of the subject is far from perfect and we see our analysis as one of the initial steps for a probably new kind of usage of biometric information. It should be noted that this paper is an extension of [3], where the method and its first analysis have been presented.

The rest of the paper is structured as follows: Section 2 provides a short overview of some of the recent research in biometric key management and biometric encryption. Section 3 introduces the proposed method and provides a brief overview of the method proposed by Sczepanski et al., with an emphasis on the differences between the two methods. Section 4 analyzes the entropy contributing factors in terms of entropy, and provides the results. Finally, we conclude in Section 5.

2 Existing Work

Several biometric key management methods have been proposed so far. These can be generally divided into two categories: *key locking* and *key generation.*

Key locking methods combine information extracted from a biometric sample with the secret key to obtain a hardened template from which it is impossible to determine either the key or the biometric sample used. The hardened template is then stored on a storage medium that needs not necessarily be secure; to "unlock" the key from the template, the user authenticates himself or herself to the system, and the extracted features are combined with the template to release the original key. Sometimes, such methods are called *biometric cryptography.*

Perhaps the most profound key locking scheme presented is that of [5]. There, Juels and Watenberg propose a new cryptographic primitive, the so-called *fuzzy commitment*, as an extension of bit commitment to binary strings, and provide a set of model schemes that employ the new primitive, including one for biometric key locking. A more recent approach towards reliable key locking mechanism was presented by Hao et al. in [4]. The authors proposed method uses the IrisCode, a 2048-bit representation of an iris image, to securely lock a secret key on a smart card. The method is especially noteworthy because of its simplistic design, good FAR and FRR (0.47% and 0%, respectively), and its easy extension to a three-factor authentication, where a password is required in addition to the user's IrisCode and the smart card on which the locked key is stored. A similar method to that of Hao et al. was proposed by Li et al. in [7]. The method employs a set of transforms to obtain an iris feature vector, which is then combined with the secret key. The obtained iris feature vector, however, is different from the IrisCode used in [4], and binary addition is used instead of the exclusive OR operation.

Key generation methods do not store any information on an external storage; instead, the key material is generated from the freshly obtained biometric sample "on the fly". Instead of locking the key with a biometric sample, the system is trained to reproduce the same key every time a legitimate user presents his or her features to the system. This usually means that no storage is required, as there is no additional information to be stored.

Feng and Wah have proposed a method for key generation using online handwritten signatures in [2]. The method checks the static features of a handwritten signature (such as shape) to filter out trivial impostors, and quantized dynamic features (such as pen pressure and pendown time) to generate a matching private DSA key to a stored public key that has been enrolled during the training phase. Ramírez-Ruiz et al. proposed a different approach, based on FingerCodes, a method of fingerprint representation based on texture, in [9]. The method utilizes Support Vector Machines (SVMs) as a key generation mechanism – in the training phase, the SVMs are trained on a dataset of user's fingerprints to produce the desired bits of the secret key. The key is then generated when the user presents his or her finger to the system. Castañón et al. proposed a conceptually similar method based on iris image in [8]. The method employs Generalized Regression Neural Networks to generate corresponding bits of the secret key using a set of feature vectors extracted from the user's iris image.

3 The Proposed Method

Current biometric key management schemes compensate for the inherent variability in one of the following ways: either through the use of a suitable error-correcting code, proximity mapping with a threshold, or through the use of classifiers such as SVM or neural networks. These are either applied to the secret key material to be stored, or to the obtained biometric information itself to link slightly differing biometric samples to the same user. Since the variability effectively represents randomness, however, biometrics present opportunities for use in random number generation mechanisms as a source of randomness, if such variability can be extracted in a meaningful way.

This concept has been first explored by Sczepanski et al. in [10]. Their method is based on the observation that measurements of physical phenomena yield values that fluctuate randomly in their rightmost decimal digits. The values are partitioned into numbered intervals and then used to generate bits based on interval membership, one bit per value. The authors have tested the method on neurophysical brain signals and galvanic skin response, and statistical tests show that the resulting binary sequences have good randomness properties. The generic construction of the method also makes it possible for the method to be used with any biometrics where the results of the measurements can be quantized to decimal values in a meaningful manner. However, the method requires large datasets of values to generate longer bistrings, and its security is dependent on the security of the measurement process of the biometrics used.

Out of all existing biometric technologies, fingerprint scanning is by far the most widespread due to its ease of use, low price and high accuracy. This has lead to the proliferation of mobile devices equipped with fingerprint readers, making the fingerprint scanning the ideal candidate for such application. Therefore, we designed our method primarily with fingerprints as the biometrics in mind, but the method is generic enough in its construction to be usable with any other biometrics available. Our focus was on low computational complexity to make the method suitable for mobile devices equipped with a fingerprint reader.

The basic rationale behind the method is to extract various changes and differences that occur between two consecutive measurements, as in reality no two fingerprint images, even coming from the same finger, are ever the same. Differences can be introduced in the change of shape of fingerprint area, or its position inside the image, as a result of user interaction with the scanning device. The resulting value is then mapped to a different value with uniform distribution to allow for direct use as a seed for pseudorandom number generators.

The basic proposed method goes as follows:

1. Obtain a set of fingerprint images $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$; $n \geq 2$.
2. Apply the exclusive OR (XOR) operation on the set of obtained fingerprints to compute the binary difference vector $diff = \bigoplus \mathcal{F} = f_1 \oplus f_2 \oplus \ldots \oplus f_n$.
3. Apply a suitable hash function on the difference vector to obtain a binary vector with uniform distribution: $uni = h(diff)$.

Unlike key management schemes, which have to consider the structure of the biometrics to choose the best representation of the processed features, our method works on the raw data of a fingerprint image, i.e. it treats the fingerprint as a binary sequence. This is because the variability in the measurement process will be captured in the fingerprint image, and even small changes are likely to affect a high number of bits. Ultimately, working with binary sequences eliminates the process of complex feature extraction and subsequent binarization.

The number of fingerprints to obtain is a system parameter, and can be set arbitrarily; however, we have yet to establish the effect that a larger number of fingerprints has on the resulting binary vector. The XOR operation serves as the extractor of variability – the resulting difference binary vector encodes change between the consecutive fingerprints on respective bit positions. The difference vector will likely not come from a uniformly distributed set of values, which is a requirement for cryptographic applications. Therefore, the method utilizes a hash function[1] as a primitive randomness extractor to map the difference vector, which comes from a nonuniformly distributed set, to a value from a uniformly distributed set. The resulting binary vector is then ready for use.

Alternatively, if the amount of entropy contained in a single fingerprint image is proven to be high, the method can be modified to use parts of an image to generate multiple random binary sequences. In such a case, only a specific subsequence of the binary representation of the fingerprint images, containing the required minimum amount of entropy, would then be XORed and hashed to produce a single random binary sequence.

One of the main advantages of the proposed number generation procedure are the low computational requirements – only the XOR logical operation and a suitable hash function are required for the method implementation. The XOR operation is straightforward to implement in hardware, and the hash function can be implemented in either hardware or software at choice. This provides for a very flexible application, as the proposed method can be fully implemented in computationally restricted devices with no access to proper random number generators.

4 Analysis

To establish the theoretical entropy and security levels of the proposed method, it is crucial to identify factors that contribute to the variability in the measurement process. These can, in theory, be environmental, behavioral, or coming from the device itself. Environmental factors are specific to the time and place where the measurement process takes place, and include conditions like moisture levels, lighting conditions, or temperature levels. Behavioral factors refer to the user's habits of presenting his or her features to the reader. Finally, the scanning device itself produces electronic noise[2] that introduces perturbations into the obtained samples.

For the purpose of our analysis, we focused on the effects of behavioral factors that affect the fingerprint scanning process, as well as the amount of electronic noise the reader is capable of producing. Concerning the user-contributed variability, we identified the effects of behavioral factors on fingerprint images to be:

- fingerprint position,
- fingerprint rotation,
- fingerprint area size.

Since we treat the fingerprint sample as an image, the behavioral factors will affect its appearance – different placement of the finger on the scanner during two

[1] The hash function may be replaced by any suitable randomness extractor.

[2] We use the term "electronic noise"' to refer to all contributing electrical noise sources in the reader components.

consecutive measurements will result in two different images, with the position and rotation of the fingerprint itself inside the images being different. Habits in presenting the fingerprint to the scanner as well as applying varying amounts of pressure will also expose different areas of the finger to the fingerprint reader, and the resulting fingerprints will vary in both shape and space they occupy inside the image. Additionally, higher pressure levels will deform the skin in contact with the reader, and the ridges and valleys may become stronger or weaker as a result.

Therefore, we have taken the difference between two fingerprint images as a result of these factors and calculated the expected amount of variability they provide in terms of uncertainty measure, the information entropy. We also calculated the corresponding minimum entropies to lower-bound the amount of uncertainty these factors can provide.

4.1 Behavioral Factors

The dataset upon which the calculations were conducted consisted of 218 8-bit grayscale fingerprint images of the same finger, scanned consecutively. These were acquired using the Sagem MSO 300 scanner, with the resolution of 416×416 pixels. In an effort to put a lower bound on the amount of computation an attacker has to perform, we presented the finger to the reader in such a manner so as to minimize the difference between the individual measurements.

To simulate real-world operation of the proposed method with a system parameter $n = 2$, we calculated every possible pair of fingerprints from the dataset, yielding a total of 23653 pairs. These were then used to calculate the difference in position, angle, and the number of pixels inside the fingerprint convex hull, and in turn the entropies and min-entropies of that difference (with the exception of fingerprint rotation, where we had to remove one fingerprint due to missing reference point in the image; in that case, the total number of pairs was 23436). Additionally, we examined the difference in gray levels of the fingerprints, and also calculated the amount of entropy the fingerprint reader noise can provide.

In the case of fingerprint shift, we used both automated and semi-automated methods to calculate the difference. The automated approach consisted of systematic alignment of one fingerprint on top of another, until the best pixel match occurs. The difference between the coordinates of the starting and end position of the alignment was then taken as a *shift vector*. The shift vectors were calculated for every pair of fingerprint images, and their frequency was recorded; these were then used to calculate the shift vector probabilities and the corresponding expected and minimum entropy. In addition to that, we also calculated the entropy for vectors grouped by length into intervals of length 5, to simulate partial knowledge about the shift in position of the fingerprint.

For the semi-automated approach, we manually measured the coordinates of a preselected reference point inside the fingerprint area. The difference between the coordinates, calculated for every possible pair of fingerprints, was then taken as the shift vector for the pair, with the rest of the process being analogous to

Table 1. Entropy of fingerprint shift, calculated using both the automated (A) and semi-automated (SA) method. Results are given in bits.

Fingeprint Shift Entropy				
	Entropy	Min-entropy	Error	Min-error
Vectors (A)	10.88	6.18	0.06	0.06
Vectors (SA)	13.34	10.93	0.07	0.26
Vector Lengths (A)	2.97	2.07	0.01	0.01
Vector Lengths (SA)	4.77	3.78	0.03	0.02

the fully automated one. Table 1 lists the entropy levels calculated using both the automated and semi-automated method.

The theoretical maximum entropy for a system of 23653 shift vectors is approximately 14.53 bits – this corresponds to each shift vector occurring exactly once across all pairs of fingerprints. As can be seen from the results, fingerprint rotation exhibits good variability; the system of shift vectors, calculated using the semi-automated approach, exhibited almost maximum entropy. Grouping together shift vectors within a given length interval reduces the entropy accordingly, since doing so correctly represents partial knowledge about the shift between the two fingerprints, and thus less uncertainty.

The lower entropy calculated using the automated approach is due to the fact that manual measurements can more accurately describe a shift between two positions, since the coordinates in both fingerprint images always refer to the same point. When the fingerprints are being aligned automatically, however, the decision whether a match occurred is governed by the size of the area in which the two fingerprints overlap. As the two fingerprints are likely to have different shapes and may be rotated by different angles, the calculated alignment may result in regions from one fingerprint image being aligned with different regions in the other fingerprint image, making the calculated shift vector incorrect.

Fingerprint rotation was calculated in both angular degrees and pixels. Two reference points per fingerprint image were chosen, and the difference in their coordinates was then taken as the *inner vector* of the fingerprint. The inner vectors were then used to calculate the rotation in angles, rounded up to degrees. Pixel rotation was calculated from the *rotation vector*, obtained as the difference between two inner vectors. The corresponding vector lengths (rounded up to pixels) were taken as the amount of pixels a fingerprint has been rotated, compared to the second fingerprint in the pair. Table 2 lists the calculated entropy levels for both angles and rotation vectors.

Most of the rotation between two fingerprints occurred within a few degrees, showing very little variability. This is likely a direct consequence of the design of fingerprint readers – the actual scanning area is not much larger than a single finger and usually delimited by rectangular border that does not allow for much rotation in the first place. As a result, the calculated entropy is low, and fingerprint rotation appears to have little effect on the amount of uncertainty a pair of fingerprints can provide.

Table 2. Entroy fingerprint rotation. Results are given in bits.

Fingerprint Rotation Entropy				
	Entropy	Min-entropy	Error	Min-error
Degrees	3.65	2.77	0.02	0.02
Pixels	4.60	3.60	0.03	0.02

We concluded our analysis of user-induced variability by examining the effects of pressure. First, we verified that pressure translates into the size of the area the fingerprint occupies inside the captured image. To test our hypothesis, we collected five sets of 200 fingerprints each, with all fingerprints belonging to the same set being captured while the user applied a specific amount of force. Standard electronic kitchen scale, with the reader on top, was used to measure and control the amount of pressure applied during individual measurements.

Next, we calculated the number of pixels inside the convex hull of the fingerprints, and used these values along with the corresponding pressure level to test the hypothesis of the two being dependent. Statistical tests did rejected our hypothesis on a significance level $\alpha = 0.05$ that the size of the fingerprint area is independent on the amount of force applied on the scanning device, with a strong linear correlation between the two. With the hypothesis verified, we proceeded to calculate the difference in pixel size of the convex hulls of the fingerprint area. This was done by calculating the pixel size of convex hull of every fingerprint from the original dataset, consisting of 218 fingerprint images, and then calculating the difference in pixel sizes for each of the 23653 different pairs.

Finally, we examined the variability in the gray levels of the fingerprint. As the actual fingerprints vary both in shape and size, it is impossible to make a pixel-by-pixel comparison of gray levels, and thus we opted for average gray level, calculated over all pixels inside the fingerprint area, to abstract from the difference in the fingerprints other than the gray level itself. The calculated gray levels were then paired up accordingly and their difference calculated for every pair. Table 3 gives the results for the entropy of both the difference in pixel size of the fingerprint area and the average gray levels.

Similar to fingerprint rotation, difference in pixel size exhibits good variability, with the difference ranging up to tens of thousands of pixels. This number is likely to increase with the resolution of the scanner used to acquire the images, as the fingerprint area will comprise of more pixels due to the increase in resolution.

Table 3. Entropy of fingerprint pizel size difference and average gray level difference. Results are given in bits.

Pixel Size and Average Gray Level Entropy				
	Entropy	Min-entropy	Error	Min-error
Pixel size	13.46	11.36	0.07	0.28
Average gray level	6.57	5.03	0.04	0.04

Average gray levels and their difference, on the other hand, exhibited much lower variability, which resulted in lower entropy. It is important to recognize, however, that the average values only approximate a normalized value for every pixel in the fingerprint area. Real pixels are likely to vary in gray levels depending on their location, and both ridges and valleys in the fingerprint area will contain pixels with gray levels that will be skewed from the calculated average.

4.2 Factor Independence

With the entropy of the effects of individual factors established, we investigated the relationships between them. In particular, we were interested in discovering whether fingerprint position, rotation, pixel size and average gray levels are independent. The concrete relationship has practical implications for the calculated entropy levels, as the entropy of independent factors can be added together.

The factors were measured on the original dataset of 218 fingerprints. Fingerprint position was measured in terms of reference point coordinates that were used to calculate the shift vectors for fingerprint pairs. Since a coordinate is effectively a vector of length 2, we split the coordinates into two values, describing the position on X and Y axis, respectively. Fingerprint rotation was determined using the inner vector of the fingerprint, relative to a vector of the same length describing the Y axis. Both pixel size and average gray level were measured exactly the same as described in the previous subsection.

The obtained values were subsequently used as random variables, paired up, and tested for independence on a significance level of $\alpha = 0.05$. For most of the pairs, the independence hypothesis was rejected at the given significance level. The only pairs which the hypothesis was not rejected for were X and Y coordinates; X coordinates and pixel size; and pixel size and fingerprint rotation. Last, we determined the correlation coefficients for the dependent factors; these are listed in table 4. As can be seen, most of the pairs are weakly correlated, with only pixel size and average gray level being strongly correlated; the only other pair showing stronger correlation is that of the X coordinate and the rotation.

4.3 Reader Noise

While the effects of fingerprint shift and the varying size of the fingerprint area result in the most visible changes in the fingerprint image, the electronic noise

Table 4. Correlation coefficients for the individual factors

	Correlation coefficients				
	X	Y	Rotation	Pixel Size	Average GL
X	1.000	-	-0.481	-	0.155
Y	-	1.000	0.129	-0.364	0.347
Rotation	-0.481	0.129	1.000	-	-0.194
Pixel Size	-	-0.364	-	1.000	-0.761
Average GL	0.155	0.347	-0.194	-0.761	1.000

produced by the fingerprint scanner during the scanning process has a potential to provide the most uncertainty through random subtle changes to every pixel in the image. Therefore, our next step was to evaluate the amount of entropy the electronic noise, produced by the scanning device, can contribute to the overall amount of entropy in a fingerprint image.

We used the Crossmatch Verifier 300 LC fingerprint reader during this experiment, as, unlike the Sagem MSO 300, it allows for continuous, empty scanning to be captured on video. Readers should therefore be aware of the fact that the images produced for the purpose of electronic noise analysis come from a different reader than those used in the experiments with behavioral factors. As the original 218 fingerprint images have a lower resolution, it is also reasonable to expect the amount of entropy contained in the electronic noise of these would be lower.

For the purpose of entropy estimation, we assume pixel independence of all positions inside the captured image. In practice, this may not be the case and neighbouring pixel positions are likely going to be dependent. However, precise relationship between individual positions is difficult to establish and are outside the scope of this paper. The calculated entropy and min-entropy should therefore be viewed as a theoretical upper bound the electronic noise can provide in such an image.

To measure the variability in electronic noise, we recorded a one minute session of the scanning process with no finger attached to the scanner. The produced video had the 640×480 pixel resolution, and a framerate of 5 frames per second. We then split the recorded video into individual frames. With a frame rate of 5 fps, we were able to obtain 300 images from the one minute video of the "empty scanning" session. Next, we recorded the gray levels and their corresponding frequencies separately for every pixel position across all 300 images. The obtained frequencies were used to calculate gray level probabilities for the given pixel position, and in turn the entropy and min-entropy for the pixel position itself. Finally, the calculated entropies were added up together to form the overall entropy and min-entropy of electronic noise in an image produced by the scanner.

More formally, the process goes as follows:

1. For a given position (x,y) in the fingerprint image, obtain the gray level of the pixel on that position.
2. Repeat step 1 for all 300 fingerprints, recording the frequencies of the obtained values.
3. Calculate the probability of pixel X = (x,y) taking on gray level l as $P(X = l) = \frac{f(l)}{N}$, where $f(l)$ denotes the frequency of l for pixel X, and N is the number of fingerprints.
4. Calculate the entropy for a given pixel position as $H(X) = \Sigma_x - P(X = x) \cdot \log_2 x$.

Additionally, we calculated the conditional entropy of electronic noise, given the knowledge of average gray levels in the neighbouring pixels. First, we recorded gray levels for both the given pixel position as well as all its neighbouring pixels.

An average of gray levels of neighbouring pixels was then calculated, rounded up to integers; the obtained average was then recorded both separately and jointly with the gray level for the given pixel position. This process was repeated for all the 300 fingerprints, and the recorded frequencies were used to calculate the joint probability of a pixel and the average of its neighbours taking on particular values, as well as the probability of the average of neighbours taking on particular values, in the same manner as above. Finally, the obtained probabilities were used to calculate the conditional probability of the pixel position. A more formal description is given below:

1. Obtain the gray level of the pixel on position (x,y).
2. Next, obtain the gray levels $l_1, l_2, \ldots, l_n$ of pixels on adjacent positions, and calculate their average $l_{aver} = \frac{\Sigma_i l_i}{n}$. Round the average to whole integers.
3. Repeat step 1 and 2 for all 300 fingerprints, recording the frequencies of the obtained values.
4. Calculate the probability of the neighbours taking on an average gray level of l_{aver} as $P(Y = l_{aver}) = \frac{f_n(l_{aver})}{N}$, where $f_n(l_{aver})$ denotes the frequency of l_{aver} for the neighbouring pixels, and N is the number of fingerprints.
5. Calculate the joint probability of pixel X = (x,y) taking on gray level l and its neighbours taking on value l_{aver} as $P(X = l, Y = l_{aver}) = \frac{f(l, l_{aver})}{N}$, where $f(l, l_{aver})$ denotes the frequency of l and l_{aver} occurring concurrently for pixel X and its neighbours, respectively.
6. Calculate the conditional probability of pixel (x,y) taking on gray level l under the condition of its neighbours taking on gray level l_{aver} as $P(X = l|Y = l_{aver}) = \frac{P(X=l, Y=l_{aver})}{P(Y=l_{aver})}$.
7. Calculate the conditional entropy of pixel (x,y): $H(X|Y) = \Sigma_{x,y} P(X = x, Y = y) \cdot \log_2(X = x|Y = y)$.

The results are listed in table 5. While the figures may appear very large at first, the higher resolution of the image means that each pixel position contains approximately 2 bits of entropy on average. Furthermore, the calculated amount of entropy is only correct under the assumption of pixel independence; if the pixels are dependent, the overall entropy of the image will be lower. Nevertheless, scanner noise appears to have a solid potential to contribute substantial amount of uncertainty to the produced binary sequences. This corresponds to our findings for sources of randomness in mobile phones [6,1].

One might ask why we do not just use the internal electronic noise in the scanner itself as a good randomness source. We have to stress that we have consciously introduced the user and his controlled action, e.g., the fingerprint scan, in order to eliminate the unbiased interaction between the user and the "machine". It is not surprising that the electronic noise of the scanner produces much higher entropy (table 5) than the well shaped ridges and valleys pattern on the fingerprints does. But such scanner noise is measured without the direct users control and with a limited user influence. Our main idea of source of randomness is hidden strictly in the user introduced interaction with the scanner, contributing a small but still considerable amount on usable entropy. The second

Table 5. Entropy of electronic noise in an image produced by a fingerprint scanner. Results are given in bits.

Scanner Noise Entropy				
	Entropy	Min-entropy	Error	Min-error
Normal	$6.4 \cdot 10^5$	$4.4 \cdot 10^5$	$2.4 \cdot 10^4$	$2.9 \cdot 10^4$
Conditional	$4.4 \cdot 10^5$	-	$1.7 \cdot 10^4$	-

important improvement for future applications would be the interconnection between the randomness generation and a synchronous biometric authentication of the user.

5 Conclusion

As could be seen, analysis of the proposed method is not a straightforward task. The number of entropy-contributing factors is high, and it is difficult to precisely establish the effect they have on the produced binary sequences. Furthermore, behavioral factors are largerly dependent on each other, and so the calculated entropies cannot simply be added together. On the other hand, the electronic noise the scanning device produces seems to contain a lot of uncertainty, but as we are unclear on the relationship between neighbouring pixels in electronic noise, the calculated entropy levels are only correct under the assumption of pixel independence.

Overall, the results indicate the proposed method is promising, but to allow for practical application, precise entropy levels of the produced binary sequences have to be established. Still, our method presents a possible approach towards a new direction in the applications of biometrics in security for mobile devices, which could become a viable alternative to traditional random number generators in situations where such mechanisms are not be available.

Acknowledgments. We acknowledge the support of the research project PICOS (Privacy and Identity Management for Community Services), No. 215056. We would like to thank Dr. Ulman and Dr. Kozubek, from the Centre for Biomedical Image Analysis at Masaryk University, for initial consultation on the possibilities of automated image matching, and for providing us with a sample program for the calculation of image alignment for two scanner images. We also thank the anonymous reviewers of our initial submission for their comments and feedback.

References

1. Bouda, J., Krhovjak, J., Matyas, V., Svenda, P.: Towards true random number generation in mobile environments. In: Jøsang, A., Maseng, T., Knapskog, S.J. (eds.) NordSec 2009. LNCS, vol. 5838, pp. 179–189. Springer, Heidelberg (2009)
2. Feng, H., Wah, C.C.: Private key generation from on-line handwritten signatures. Information Management & Computer Security, 159–164 (2002)

3. Gerguri, S., Matyáš, V., Říha, Z., Smolík, L.: Generating random sequences from fingerprints. In: MEMICS 2008, pp. 51–59 (2008)
4. Hao, F., Anderson, R., Daugman, J.: Combining crypto with biometrics effectively. IEEE Transactions on Computers 55(9), 1081–1088 (2006)
5. Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: Proceedings of the Sixth ACM Conference on Computer and Communications Security (1999)
6. Krhovják, J., Švenda, P., Matyáš, V.: The source of randomness in mobile devices. In: Proceeding of the 12th Nordic Workshop on Secure IT Systems, pp. 73–84. Reykjavik University, Kringlan 1, 103 Reykjavik (2007)
7. Li, X., Wu, X., Qi, N., Wang, K.: A novel cryptographic algorithm based on iris feature. In: International Conference on Computational Intelligence and Security, vol. 2, pp. 463–466 (2008)
8. Casta nón, L.E.G., Reigosa, M.P., Nolazco-Flores, J.A.: Biometric-iris random key generator using generalized regression neural networks. In: Ali, M., Dapoigny, R. (eds.) IEA/AIE 2006. LNCS (LNAI), vol. 4031, pp. 530–539. Springer, Heidelberg (2006)
9. Ramírez-Ruis, J.A., Pfeiffer, C.F., Nolazco-Flores, J.A.: Cryptographic keys generation using fingercodes. In: Sichman, J.S., Coelho, H., Rezende, S.O. (eds.) IBERAMIA 2006 and SBIA 2006. LNCS (LNAI), vol. 4140, pp. 178–187. Springer, Heidelberg (2006)
10. Szczepanski, J., Wajnryb, E., Amigó, J.M., Sanchez-Vives, M.V., Slater, M.: Biometric random number generators. Computers & Security 23 (February 2004)