An Analysis of Lightweight Encryption Schemes for Fingerprint Images

Dominik Engel, *Student Member, IEEE,* Elias Pschernig, and Andreas Uhl

Abstract— **Two lightweight encryption schemes for fingerprint images based on a bitplane representation of the data are assessed. We demonstrate a low complexity attack against a scheme recently proposed in literature which exploits one of several weaknesses found. A second scheme is evaluated with respect to two fingerprint recognition systems and recommendations for its safe use are given.**

EDICS Category: CRY-CRYP, BIO-FING, BIO-ATTA

I. Introduction

With the increasing usage of biometric systems the topic of storing and handling sample data (i.e., the acquired sensor data) in an optimal way has to be addressed. Recorded and stored sample data is obviously more sensitive with respect to privacy issues as compared to template data, therefore a proper protection of these data is mandatory. One strategy in this context is to separate the person's identity from the data files. Moreover, the encryption of the template data may become imperative under certain circumstances due to the security and privacy concerns of the users.

Among other possibilities, encryption technology may be used in two stages of the processing chain in classical biometric recognition:

1) **Storage of reference data**: In most template databases (where the reference data of the enrolled individuals is stored) only the extracted features required for the matching step are stored as opposed to retaining the originally acquired sensor data. However, in case the features should be replaced for some reason (e.g., when a superior or license-free matching technique involving a different feature set becomes available), having stored only extracted features implies the requirement for all legitimate users to re-enroll, which can be expensive and is highly undesired since user-acceptance of the entire biometric system will suffer. Storing the original sensor data in addition to the features required for the current matching technique solves this problem. Of course, these data need to be stored in an encrypted form.
2) **Transmission after sensor data acquisition**: In distributed biometric systems, the data acquisition stage is often dislocated from the feature extraction and matching stage (this is true for the enrollment phase as well as for authentication). In such environments the sensor data have to be transferred via a network link to the respective location, in some cases even over wireless channels. Obviously, sample data has to be protected during transmission – security may be provided either by the transmission protocol (e.g., an encrypted network link like IPSec) or by direct encryption of the image data.

Usually, classical cryptographic techniques are suggested to be used for biometric sample data [7]. A small number of specific techniques has been developed for fingerprint sample images. A Fourier-type transform-based private encryption scheme for fingerprints is proposed in [15]. The concept of "cancelable biometrics" is proposed in [12], where the acquired biometric signal (i.e., the sample) is distorted with an intentional repeatable transform before the extraction of the template. In case of a compromise, the transformation can simply be changed. A very similar approach are the so-called "biometric cryptosystems" proposed in [16] and [12]: A secret transformation is applied to the biometrics templates to render them useless for intruders. Matching can be performed in the encrypted domain.

Controlling the computational demand is important, especially in distributed scenarios with weak and low-power sensor devices. Classical encryption techniques like AES can be too demanding to be employed, therefore a careful but significant reduction of encryption complexity for this type of applications has been suggested in the literature. One approach is to use energy efficient stream ciphers that can easily fit small environments, see, e.g., the ECRYPT eSTREAM project.[1] The limited computational resources in embedded processors are addressed in recent work by Moon et al. [10], where an approach involving selective encryption of fingerprint images employing XOR on a bitplane basis is suggested.

In this work we thoroughly analyze two algorithms discussed in the latter manuscript. After an investigation of the properties of fingerprint image bitplanes acquired with different sensors, we demonstrate several shortcomings and a computationally efficient attack against the proposed scheme. The second approach discussed in this manuscript is also evaluated and we give recommendations for its secure employment.

Manuscript received January 12, 2007; revised January 29, 2008. This work has been partially supported by the European Commission through the IST Programme under contract IST-2002-507932 ECRYPT. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Bart Breneel.

The authors are with the Computer Sciences Department, Salzburg University, Jakob-Haringer-Str. 2, A–5020 Salzburg, Austria (e-mail: dengel@cosy.sbg.ac.at; epschern@cosy.sbg.ac.at; uhl@cosy.sbg.ac.at).

[1] `http://www.ecrypt.eu.org/stream/`

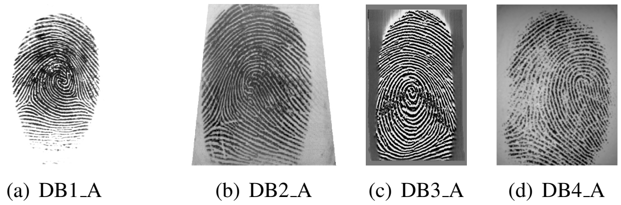

(a) DB1_A (b) DB2_A (c) DB3_A (d) DB4_A

Fig. 1. Example fingerprint images from the FVC2004 database.

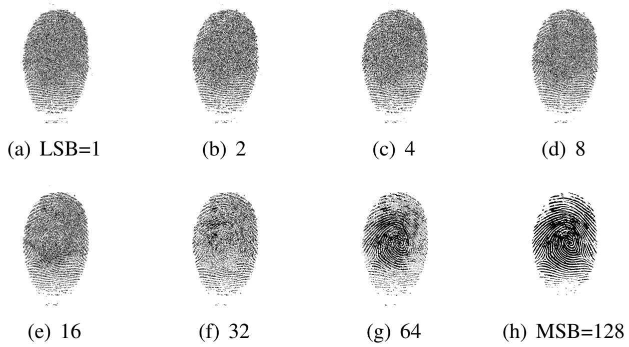

(a) LSB=1 (b) 2 (c) 4 (d) 8
(e) 16 (f) 32 (g) 64 (h) MSB=128

Fig. 2. Bitplanes of example fingerprint image of DB1_A.

II. Analysis of Lightweight Fingerprint Encryption Techniques

Since both lightweight fingerprint image encryption schemes analyzed in this work rely on the bitplane representation of the corresponding grayscale fingerprint images, we initially investigate the properties of the corresponding bitplanes.

A. Fingerprint Images' Bitplane Properties

For our investigations (in this section and in Section II-C), we use the four sets of fingerprints from the fingerprint verification contest 2004 (FVC2004).[2] Databases 1 and 2 contain images of two different optical sensors (DB1, DB2), database 3 originates from a thermal sweeping sensor (DB3), and database 4 consists of synthetically generated prints (DB4). Figure 1 displays an example fingerprint from each database. Figures 2 to 5 show the eight bitplanes of the fingerprints given in Figure 1.

Figure 2 shows that ridge information is present in all bitplanes of images acquired with the optical sensor used for DB1. While ridge structure is most clear in the four most significant bitplanes (MSB-planes), it is even visible in the least significant bitplane (LSB-plane). Note that therefore the statement "the LSB-plane is not correlated with other bitplanes if the images are acquired by various sensors such as a digital camera, scanner and other devices" [2] does obviously not hold in general for fingerprint images. Furthermore, "the LSB looks similar to a random number field" [10] is also not correct in our context.

Figure 3 on the other hand exhibits visual properties found in most digital images. The overall structure of the image (i.e., its ridges in our case) is visible in the four or five most significant bitplanes, especially the LSB-plane in fact looks like random noise. Recall that, like the images in DB1, the images in DB2 originate from an optical sensor, in spite of their contrasting properties.

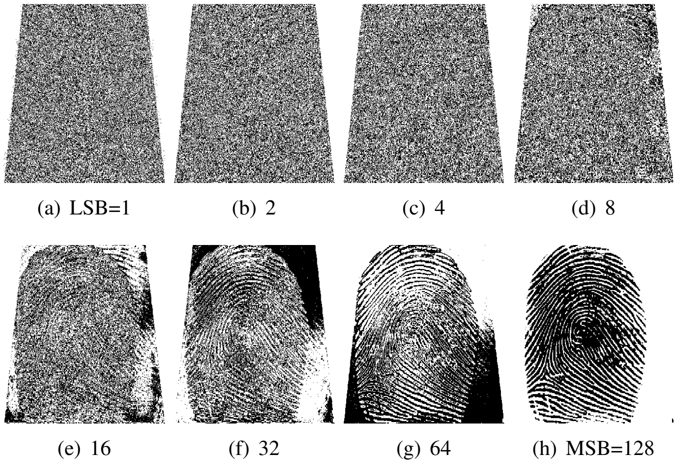

(a) LSB=1 (b) 2 (c) 4 (d) 8
(e) 16 (f) 32 (g) 64 (h) MSB=128

Fig. 3. Bitplanes of example fingerprint image of DB2_A.

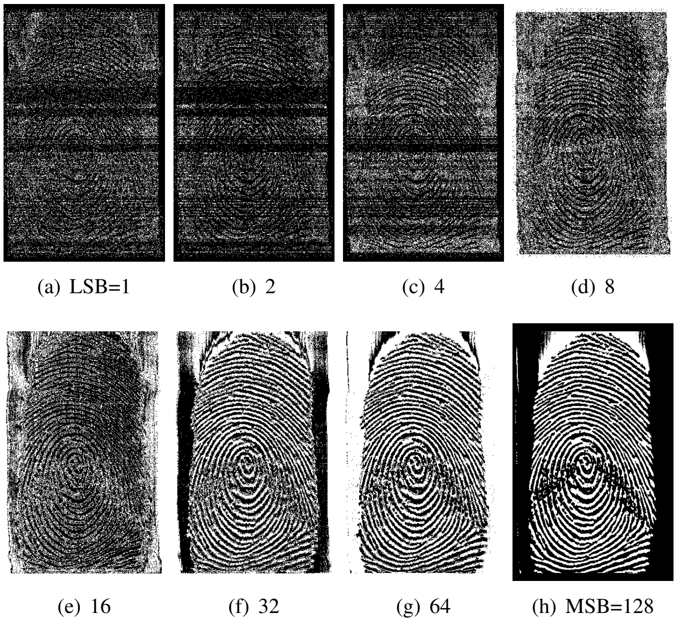

(a) LSB=1 (b) 2 (c) 4 (d) 8
(e) 16 (f) 32 (g) 64 (h) MSB=128

Fig. 4. Bitplanes of example fingerprint image of DB3_A.

The bitplanes of an image acquired by a thermal sweeping sensor are shown in Figure 4. While the three most significant bitplanes are almost identical to the original fingerprint, important ridge structures are recognizable even in the LSB-plane. The principal findings and corresponding conclusions with respect to images in DB1 are therefore confirmed. Finally, the synthetic fingerprint in Figure 5 exhibits a decreasing amount of ridge information for a decreasing significance of the bitplanes, the two least significant planes including the LSB-plane again look similar to random noise.

Figure 6 shows larger versions of the LSB-planes of two fingerprint images from DB1 and DB3, confirming the above

[2] http://biometrics.cse.msu.edu/fvc04db/index.html

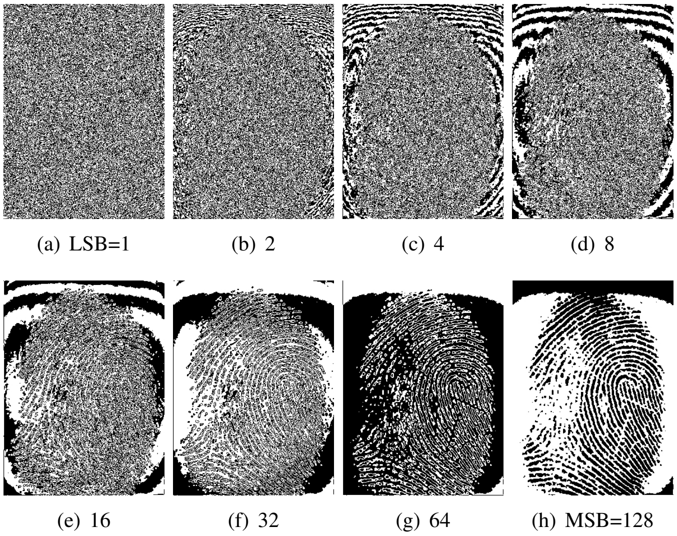

(a) LSB=1
(b) 2
(c) 4
(d) 8
(e) 16
(f) 32
(g) 64
(h) MSB=128

Fig. 5. Bitplanes of example fingerprint image of DB4_A.

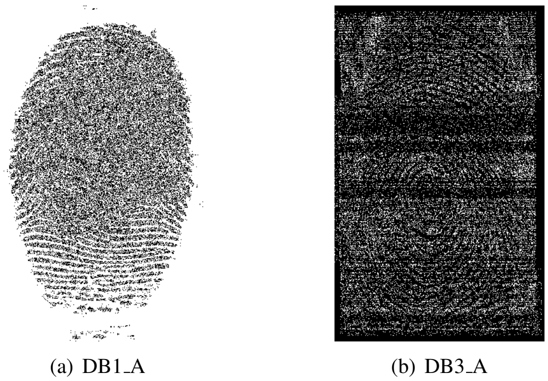

(a) DB1_A
(b) DB3_A

Fig. 6. LSB-planes of fingerprint images from the FVC2004 database.

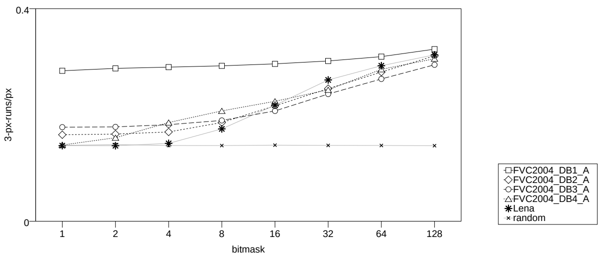


Fig. 7. Number of 3-pixel runs of single bitplanes.

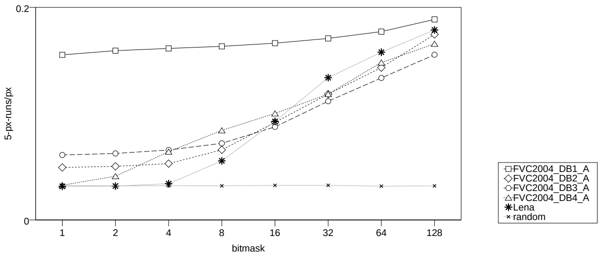


Fig. 8. Number of 5-pixel runs of single bitplanes.

visual impression. This raises the question how to quantify these visual properties.

In order to find a measure for quantifying discernibility from noise, we use two approaches: the number of consecutive runs and how well the bitplanes can be compressed by an arithmetic coder. The number of consecutive runs (in raster scan order) of a given length of bits with the same value is counted in the lines of the bitplane, and normalized by the number of total bits. From each set, 100 fingers with 8 prints each are used, the plots show averaged run counts. The images from DB1 contain large empty areas (see Figure 2), so they generally have many more runs. For the sake of a comparison to a "classical" digital image we also present the values of bitplanes of the Lena image. As a reference, we have also added the results for an image that only contains random noise to the plots. The image was generated by using the Mersenne Twister PRNG [8] to produce a random byte for each pixel value.

Figures 7 and 8 show typical results as obtained by those measurements. Bitmask 1 refers to the LSB-plane and bitmask 128 correspondingly refers to the MSB-plane. For the three least significant bitplanes of the Lena image the number of runs is equally low and attains the lowest value found. Only the LSB-plane of images in DB4 shows a comparably low value. The number of runs in LSB-planes increases for images in DB2, DB3, and DB1, in this order. Note that this corresponds well with the visual impression that especially the LSB-plane of DB1 and DB3 contains ridge structures whereas the LSB-plane of DB2 and DB4 looks more like noise. The higher number of runs in the more significant planes of the Lena image is due to the ridges in the fingerprint images which prevent an equally high number of runs due to the absence of large regions with uniform gray value (except for images in DB1 of course).

The most striking feature of the curves are the slopes – whereas images with noisy LSB-plane (Lena, DB2, DB4) exhibit a rather steep slope, images with ridge information also contained in less significant bitplanes (DB1, DB3) result in a flatter slope. This behavior corresponds to the fact that the number of runs is not decreased as much as expected when decreasing the significance of the bitplanes. Summarizing, we have found qualitative (i.e., visual) as well as quantitative arguments that for some types of fingerprint images (depending on the acquiring sensor) the properties of the bitplanes and especially those of the LSB-planes are different as compared to "classical" digital images. Specifically, the LSB-plane does not behave like random noise and contains structural ridge information under such circumstances.

Another way to assess how close a bitplane is to a random field is to encode it with an arithmetic coder. The idea is that if the bitplane can be compressed well, it cannot be very close to a random field. We use the arithmetic coder proposed by [9] in a mode where it accepts a sequence of bits without a specific background model. Figure 9 shows the compression ratio of each bitplane for the fingerprint images in the four databases and, as a reference again, for Lena and the randomly generated image. It can be seen that the arithmetic coder is successful at compressing all bitplanes of the images in DB1, including the

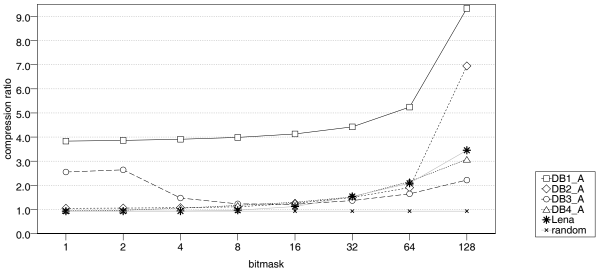


Fig. 9. Compression performance of arithmetic coding by bitplane.

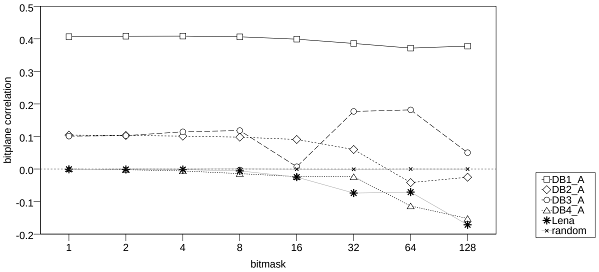


Fig. 10. Correlation of each bitplane with the other bitplanes

LSB-plane. An interesting phenomenon can be observed for the images in DB3: the thermal sensor yields images for which the medium bitplanes exhibit more noise than the LSB-plane. The compression ratios for the LSB-planes of the images in DB2 and DB4 and also Lena are nearly as low as the results for the randomly generated image. For these images the lower bitplanes are closer to random than for the other images.

Another point apart from the randomness of the LSB-plane is the correlation of each bitplane with the other bitplanes. To assess this correlation, we compute the sample correlation of the bitplane under consideration with each other bitplane, and then average the correlation values. The results for each bitplane are shown in Figure 10.

For the images of DB1, which have a uniform background, the correlation is naturally high. The phenomenon of more noise in the medium bitplanes in the case of DB3 can again be observed in the lower correlation of these bitplanes, especially for the fifth bitplane. For DB2 a little correlation can be observed at lower bitplanes. The lower bitplanes of DB4 and Lena exhibit no correlation to the other bitplanes, just like the randomly generated image.

It should be noted that better sensors, due to inherent (thermal) noise, may produce images with more random LSB-planes as compared to weaker sensors. Furthermore, if post-processing is applied, it may influence the randomness of the LSB-plane in one or the other direction. What our observations show is that the assumption that all scanners always produce images for which the LSB-plane is (close to) a random field that shows no correlation to the other bitplanes is not true and is, in fact, a dangerous assumption with regard to security.

B. Vigenére LSB Encryption

In recent work [10], a lightweight fingerprint image encryption technique has been proposed, which has been denoted as "image-based selective bitplane encryption protocol". While the classical notion of selective encryption restricts the application of cryptographic techniques to a zone of influence (ZoI) only, this is not the case in the investigated approach. In fact, the proposed approach is a Vigenére-cipher. We will therefore denote this algorithm as "Vigenére LSB encryption", which is much more appropriate as will get clear soon.

1) The Proposed Algorithm: Let I be the original 8 bpp fingerprint image with a width of w pixels and a height of h pixels. s denotes the size of the image in bits, $s = h \cdot w \cdot 8$. Consider now the binary representation of the image I being given as

$$I = \{b_0, b_1, \cdots, b_6, b_7, b_8, \cdots, b_{s-1}\}$$

where $b_{m \cdot 8}$, $0 \leq m \leq h \cdot w - 1$ is the MSB of the binary representation of pixel $m+1$, whereas $b_{m \cdot 8-1}$, $1 \leq m \leq h \cdot w$ is the LSB of pixel m. We extract a set of key bits

$$K_0 = \{k_0, \cdots, k_{h \cdot w-1}\}$$

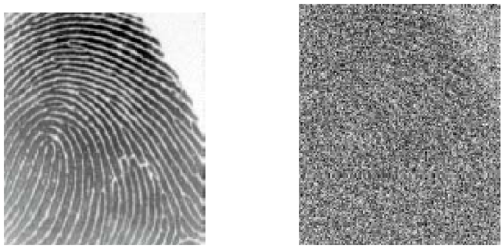
where the m-th keybit k_m of key K_0 is constructed by taking the LSB of each pixel m, i.e., $k_m = b_{m \cdot 8-1}$, $1 \leq m \leq h \cdot w$. Subsequently, to obtain the encrypted data c_i, we apply an exclusive-or operation (XOR) between I and K_0:

$$c_i = b_i \oplus k_i \ , \ 0 \leq i \leq s-1 \ .$$

Since this operation only processes $1/8$ of the binary representation of I, for the remaining binary data of I the operation is repeated 7 times, using the identical key K_0. Finally, K_0 is encrypted using AES [1] and transmitted to the receiver together with the encrypted data c_i, $0 \leq i \leq s-1$.

To summarize, the proposed algorithm is a Vigenére encryption of the binary representation of the image data where the key-pad used for the XOR operation consists of the LSBs of the image's pixels, which is AES encrypted. Compared to a full (i.e., 100%) AES encryption, the approach reduces the AES encryption effort to 12.5% and introduces only little additional overhead (XOR and extraction of the binary image data – compare Table 3 in [10]).

2) Problems: Figure 11 displays plaintext and ciphertext example images as given in the original publication [10]. It is somewhat surprising to observe dominant ridge structures in the ciphertext, although concealed by noise (which could be reduced by classical denoising techniques). This information leakage raises the suspicion that there might be security problems in the proposed scheme.



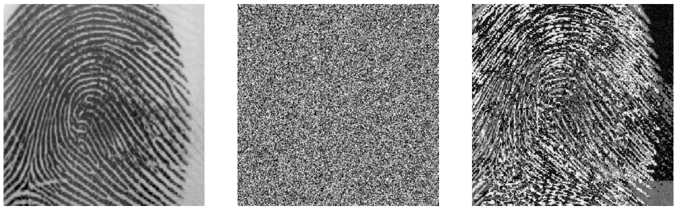
(a) Original Image (b) Ciphertext

Fig. 11. Example images from [10], Figs. 11(g) and (i).

In fact, the lightweight method for encryption of fingerprint images proposed by [10] has a number of problems which are outlined in the following.

Vigenére encryption is only secure if the key has the same length as the message and if it is truly random [14], i.e., if it is a one-time pad encryption. Both conditions are violated for the proposed approach.

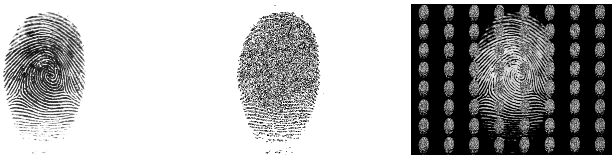
a) Key-length: The key-length, which is an eighth of the message length, gives an attacker the possibility to shift the ciphertext by the size of the key and XOR it with itself. This operation removes the key and leaves the attacker with the plaintext XORed with a version of itself that has been shifted by the key-length [14]. Figure 12 illustrates this for a fingerprint image. As can be seen, the image obtained by this operation yields a lot more information of the original fingerprint than the ciphertext.



(a) Original Image (b) Ciphertext C (c) $C \oplus (C \;>>$key-length)

Fig. 12. Illustration of shifting the ciphertext by key-length (part of DB2_2_3).

b) Randomness: Section II-A and Figure 13 show that the assumption made in [10], that for fingerprint sensors the LSB is sufficiently random and not correlated with the other bitplanes, does not hold. In fact, for some sensors there is significant ridge information present in the LSB-plane and considerable correlation among bitplanes can be observed. The image shown in Figure 13 is from database DB1 and has been acquired with an optical fingerprint sensor. As can be seen, the LSB does not generally behave like a random number field for all fingerprint sensors. The ciphertext – if the term is indeed appropriate in this case – for this fingerprint image is shown in Figure 13(c).



(a) Original Image (b) LSB-plane (c) Ciphertext

Fig. 13. Random LSB-plane: a counterexample (DB1_1_3).

c) Key XORed with itself: Even if the LSB-plane produced by the used sensor is assumed to be sufficiently random, the scheme is not secure. The authors propose to XOR *all* bits of the plaintext with the LSB-plane. That means that the LSB-plane is XORed with itself at some positions. We show below that this is a fundamental problem. This critical mistake in the design of the encryption scheme could have easily been avoided, as we will discuss in Section II-B.4.

d) Data expansion: The ciphertext has 112.5% of the size of the plaintext. Since the transmission is intended for weak network links, this property is highly undesired. Similar to the issue of XORing the key with itself, also data expansion could have been easily avoided, which will be discussed in Section II-B.4 as well.

3) Attack: We attack the scheme at its most vulnerable point: the key being XORed with itself. Let C be the bits of the ciphertext that are obtained by encrypting I with key K_0. During encryption, each of the keybits (being the LSB of plaintext pixels) is XORed with an element of K_0. We subsume these elements as K_1. Note that $K_1 \subseteq K_0$. We introduce the operation $\hat{\oplus}$ with the meaning of $m \hat{\oplus} n$ as "the bit at position m gets encrypted by the bit at position n". We can conceive the operation as a mapping from K_0 to K_1, where

$$K_1 = \{k_i \in K_0 \mid \exists k_j \in K_0 : k_j \hat{\oplus} k_i \in C\}.$$

If one or more of the keybits in K_0 are mapped to the same position, then $K_1 \subset K_0$, i.e., the mapping reduces the number of key positions. As the ciphertext is known, K_0 can be reconstructed from K_1, if the correct settings for the keybits in K_1 can be determined.

We can further investigate the mappings of the keybits in K_1. All of the elements of K_1 are mapped to an element of K_1. This can easily be shown: let k_j be an element of K_1, then also $k_j \in K_0$, because $K_1 \subseteq K_0$. If we now assume that k_j is mapped to $k_i \in K_0 \backslash K_1$, i.e., $k_j \hat{\oplus} k_i \in C$, then by the definition of K_1 and because $k_j \in K_0$ and $k_i \in K_0$, it follows that $k_i \in K_1$, which contradicts the assumption. Therefore the set K_1 can be mapped to a set K_2 with $K_2 \subseteq K_1$.

This process can be applied repeatedly. We can map the keybits in K_i to a set K_{i+1}, $K_{i+1} \subseteq K_i$:

$$K_{i+1} = \{k_i \in K_i \mid \exists k_j \in K_i : k_j \hat{\oplus} k_i \in C\}.$$

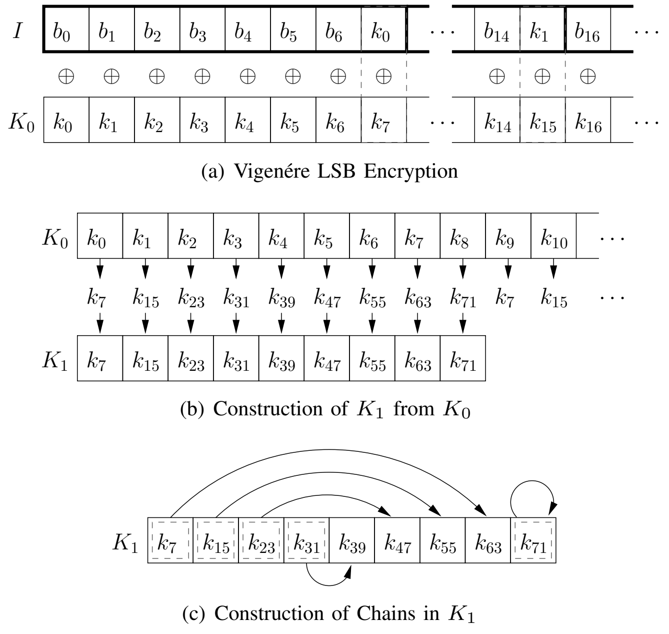
As long as one or more bits from K_i are mapped to the same bits in K_{i+1}, K_{i+1} is a proper subset of K_i: $K_{i+1} \subset K_i$, i.e., we reduce the number of referenced keybits. It can be easily seen that after a number of iterations N, no more reductions are possible:

$$\exists N \geq 0 : K_{i+1} = K_i \text{ for } i \geq N.$$

If the correct settings for the bits in K_N are known, then K_{N-1} can be reconstructed. As generally the correct settings of K_{i+1} can be used to reconstruct K_i, the correct settings of the bits in K_N are sufficient to get the settings for all bits in the key.

It can be shown that for key-lengths of a power of 2, $|K_N| = 1$, i.e., the whole key depends on the setting of a single bit. In this case, the plaintext can be easily reconstructed by testing the two possible settings of this bit and then reconstructing the key. After decryption, one setting will yield the original plaintext, the other setting will yield the original plaintext with its pixels inverted.

For images of other sizes, more positions will remain in the set K_N. Note that if the greatest common divisor of s (the size of the image in bits) and 8 is 1, then $K_N = K_0$. Generally, K_N


I
b0 b1 b2 b3 b4 b5 b6 k0 ··· b14 k1 b16 ···
⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕
K0 k0 k1 k2 k3 k4 k5 k6 k7 ··· k14 k15 k16 ···
K0 k0 k1 k2 k3 k4 k5 k6 k7 k8 k9 k10 ···
k7 k15 k23 k31 k39 k47 k55 k63 k71 k7 k15 ···
K1 k7 k15 k23 k31 k39 k47 k55 k63 k71
K1 k7 k15 k23 k31 k39 k47 k55 k63 k71

(a) Vigenére LSB Encryption

(b) Construction of K_1 from K_0

(c) Construction of Chains in K_1

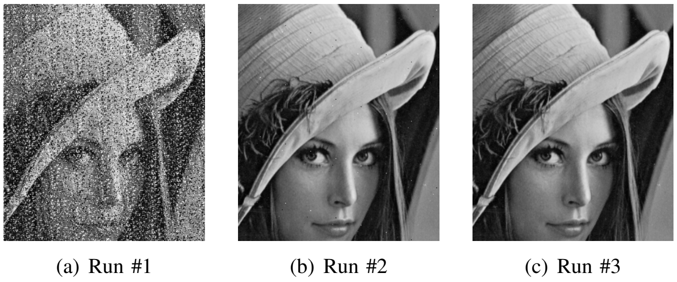
Fig. 14. Reduction of keybits for $n = 72$, $N = 1$.

will be too large for a brute force search. However, the existing mappings of the bits in K_N can be used to further reduce the number of relevant keybits. The elements of K_N are either mapped to themselves or to another element of K_N. During the encryption process, only a limited number of elements are actually mapped to themselves. For the rest of the elements we can define circular chains of keybits. For the keybits in each of these sequences a mapping exists between each element and its successor, with the successor of the last element being the first element. For the example in Figure 14(c), a chain is formed by k_7 and k_{63}, because k_7 is mapped to k_{63} which is mapped to k_7 again. The elements of K_N which are mapped onto themselves, form a chain of length 1. The number of chains that exist in K_N and their lengths depend on the length of the key.

If the correct setting for one bit in the chain is known, then the complete chain can be reconstructed. So we can choose a single element from each chain as a representative. Each of these representative elements influences a multitude of bit positions in the plaintext.

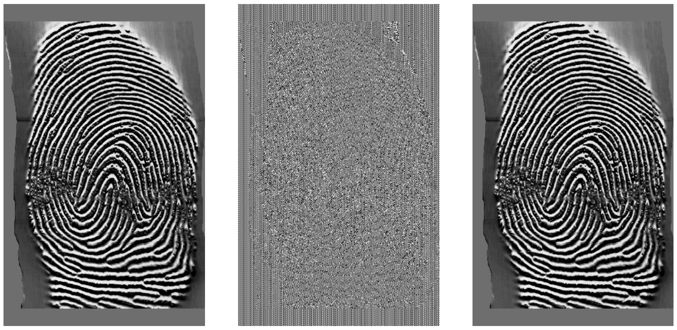
The procedure of reducing the keybits is illustrated in Figure 14 for an input image of 72 pixels. During encryption each element of K_0 is XORed with an element of K_0 (a). The referenced elements are collected in K_1 (b). For $n = 72$, no further reduction is possible after this step, so $N = 1$. In the final step, chains of mappings are found in K_1 (c). The number of representative bits for the 72-bit key is reduced to 5 bits.

As an example, we investigate fingerprint images used by [10]: for two different sensors, they obtain images of 320×440 and 248×292, respectively. For the first sensor this leads to a set $K_{N=3}$ with 275 elements. These elements can be grouped into 16 chains of varying length. During encryption, each chain influences between 4096 and 81920 bit positions in the plaintext. For the second sensor the reduced set of keybits $K_{N=2}$ has 2263 elements which can be organized into 175 chains. Each chain influences between 256 and 3840 bits in the plaintext.



(a) Run #1 (b) Run #2 (c) Run #3

Fig. 15. Variance attack on Lena.



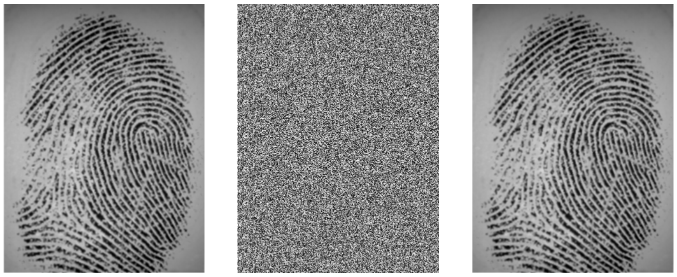
(a) Original image (b) Result of variance attack (c) Result of neighborhood attack

Fig. 16. Variance versus neighborhood attack (DB3_84_2).

For a brute-force search, the number of chains is still too large. But we can formulate some conditions that should hold for the plaintext. Because the representative bits influence so many positions in the image, the condition need not be overly sophisticated. For natural images, the sample variance can be used as a simple measure. A hypothesis for the value of the representative bit of each chain is formed and iteratively tested. We start by setting each representative bit to zero. Then the chains are reconstructed to form the set K_N. A hypothetical key is created by reconstructing K_{N-1} through K_0 from K_N. K_0 is used to reconstruct an image. In the next step one of the representative bits is flipped from 0 to 1. Again an image is reconstructed, and its sample variance is compared to the previous run. If the sample variance has decreased, then the bit is left at 1 otherwise it is flipped back to 0. This process is repeated over the whole set of representative bits, until the variance no longer changes. For images with a sufficient degree of smoothness the result will be the original image (or an inverted version of it, depending on the initial setting of the representative bits). This iterative refinement of a hypothetical key is similar to the method for cryptanalysis of substitution ciphers proposed by [6].

The process is illustrated for a version of the Lena test image of size 248×292 in Figure 15. Each image represents a whole run over the 175 chain bits. After run number 3 the variance does not change any more and the image is found.

The variance for testing the hypothesis does not only work for most natural images but also for many of the tested



(a) Original image (b) Ciphertext (c) Result of attack

Fig. 17. Example: attack on DB4_1_8.

fingerprint images. However, some of the fingerprint images exhibit strong oscillatory patterns. An example image, which was captured by a thermal sweeping sensor, is shown in Figure 16(a). In such cases, the minimum variance fails as a condition for the correct plaintext image, as shown in Figure 16(b). Therefore we use a more local measure that reflects the properties of fingerprint images in a better way: For each pixel in the image decrypted with the hypothetical key, we measure the difference of this pixel to all pixels surrounding it. The sum of these differences should be minimized. We found that considering the eight immediately surrounding pixels is sufficient. With the neighborhood measure we can decrypt both, fingerprint and natural images. Figure 16(c) shows the attack result for one of the fingerprint images, for which the sample variance failed. Figure 17 shows the attack on a (synthetic) fingerprint image for which both variance and neighborhood attacks work.

Note that the proposed attack does not depend on the randomness of the LSB-plane: even for a truly random LSB-plane the encrypted image can be easily decrypted without knowing the key. If the proposed attack is successful, then the image decrypted with the proposed attack is bitwise identical to the original image. Depending on the used measure, if there exists an image that is not the original plaintext (but possibly similar to it) and which has a lower measure than the real plaintext, then the iteration will terminate, and the result of the attack in this case would not be the perfect plaintext.

The computational demands of the attack depend on the number of representative bits (which in turn depends on the size of the image). Table I shows the time the attack needed for the images discussed above. The results were obtained with a Java implementation running on an AMD Athlon 1.6 GHz with 2 GB of RAM. It can be seen that the costs for the attack are low. Using the sample variance in an attack on DB3_84_2 was unsuccessful (*), all other attacks produced the original plaintext image. The last image represents a special case for which the keybits can be reduced to a single representative bit. In this case, the attack is extremely fast and no measure is necessary.

4) Improvements:

a) Key XORed with itself: In this respect, the scheme can be designed in a more secure way: only XOR the bitplanes apart from the LSB-plane with the LSB key, i.e., bitplanes 7 through 1, but leave the LSB untouched. The original scheme proposes to encrypt the LSB-plane with AES anyway. The encrypted version can be inserted into the ciphertext at the LSB positions. Apart from enhancing security by avoiding the key being XORed with itself, this modification brings another advantage: unlike in the original scheme, the LSB-plane information is not transmitted twice, therefore also solving the *data expansion* problem.

Image	Size	R. Bits	Variance	Neighborhood
Lena (mod.)	248 × 292	175	60.28 s	65.76 s
DB1_1_3	640 × 480	8	13.44 s	7.54 s
DB3_84_2	300 × 480	32	*25.01 s	14.78 s
DB4_1_8	233 × 384	8	4.42 s	4.8 s
DB2_3_3 (mod.)	256 × 256	1	1.2 s	

TABLE I
TIMING RESULTS.

b) Randomness: In order to produce a keystream that exhibits more properties of a random number field, we suggest to extract the LSB-plane (or any other bitplane) first, encrypt it with AES, and use the resulting data as the keystream for the XOR operation. Of course, AES ciphertext is not truly random, but at least it passes several strong statistical tests for randomness [5]. It has to be noted, however, that with this approach, the encrypted bitplane has to be regarded as proper key material and has to be transferred over a secure channel.

c) Key-length: The length of the key remains restricted to $1/7$ of the data size even if implementing the improvements as suggested so far. This is a major obstacle. A possible solution would be to additionally introduce 6 different permutations of the key data at the cost of additional key material. It is doubtful (and of course depends on the type of permutations applied) if such a scheme would still be more efficient and equally secure as compared to full encryption with a fast stream cipher (e.g. [13], [3], [4]).

Note that these suggestions can be used to improve the scheme, but they do not make the scheme secure from a cryptographical point of view. Even with the improvements the scheme remains insecure and should not be used in practical applications.

C. Selective Bitplane Encryption

A second lightweight encryption scheme for fingerprint images is also briefly discussed in the manuscript by Moon et al. [10]. This approach uses the concept of selective bitplane encryption as proposed for image protection [11] and try-and-buy scenarios [2] previously. The basic idea is to encrypt a subset of the bitplanes only, starting with the MSB and proceeding towards the less significant bitplanes as a higher degree of security is required. This approach with encrypting two bitplanes is rated as being insecure by visual inspection of an attacked fingerprint image (Figure 6 in [10]).

Here, we try to give the analysis a more sound basis by assessing the actual matching performance of a fingerprint recognition system applied to selectively encrypted images. From each set of the FVC2004 data, 100 fingers with 8 prints each are used. The bitplanes are extracted from every fingerprint image, and converted to grayscale by shifting bright pixels to a medium gray in order to lower contrast. This strategy improves matching behavior for VeriFinger. In case the MSB of all pixels is 0, pixel values are multiplied by two.

The converted grayscale images are matched against the 8 original images. This results in 64 matching scores for each finger and bitplane. The average of 100 fingers or 6400 scores is then plotted for each bitplane.

Matching is done using the NFIS2 minutiae matching software from NIST.[3] The `mindtct` software extracts minutiae information from the original fingerprints and from the prints consisting only of single bitplanes (or with single bitplanes masked out). The program `bozorth3` is then used to generate matching scores between minutiae sets of different finger prints. In the following plots, we use the term "bitmask" to denote which parts of the binary representation of a pixel (an image) is used in the matching process. The bitmask is the unique decimal value which is obtained by setting omitted bit positions to 0 and used bit positions to 1. For example, bitmask $128 = 10000000$ denotes an image where only the MSB bits are kept for all pixels, $63 = 00111111$ is an image which consists of all but the MSB-plane and the next significant bitplane.

Figure 18 shows the average matching score of the single 8 bitplanes. For the three most significant bitplanes (masks 32, 64, 128) of the third database DB3 (the thermal sweeping sensor), the scores are even greater than 40 on average, which the NFIS documentation gives as the score above which prints usually can be considered matching. For the two optical sensors of DB1 and DB2 and the generated finger prints (DB4), the most significant bitplane shows a good matching score as well. Note that this exactly corresponds to the visual impression in Section II-A where the three most significant bitplanes have been considered as being almost identical to the original. In the images acquired by this sensor, the ridge information is almost given as black & white, so the significant bitplanes contain roughly the same information as the images themselves. The less significant bitplanes of the images in DB1 and DB3 have been shown to contain ridge information in Section II-A – this is also reflected in the matching scores which are clearly higher as compared to the values for DB2 and DB4; in perfect accordance to the visual impression, the LSB of DB1 images scores highest whereas DB3 takes the lead for bitmasks 4 to 128 (i.e., MSB).

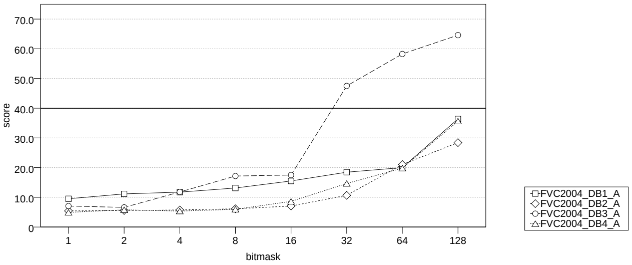


Fig. 18. Average minutiae matching scores of single bitplanes against 8 impressions from the databases.

Figure 19 shows the scores for the same images if instead of extracting single bitplanes from the original, single bitplanes are missing. Note that this experiment simulates a selective encryption of a single bitplane with perfect replacement attack, see [10] and [11]. With the most significant bitplane missing (mask 127), the scores are 41.4 for DB1, 22.1 for DB2, 61.7 for DB3 and 12.9 for DB4. The scores from DB1 and from DB3 are always above the 40 mark which indicates that this strategy is highly insecure if applied for encrypting fingerprint images. The synthetic prints from DB4 are on average below 40 only when the MSB is missing. Note that the results of DB1 are not monotonically increasing as would have been expected when the less significant bitplanes are omitted, but the values are in fact decreasing up to the 5th bitplane (contrasting to the other DBs). Especially the 5th most significant bitplane has the most severe impact if missing which questions the general strategy to restrict encryption to the MSB and subsequent planes. With only the three least significant planes missing (masks 251, 253, 254) the matching score is only slightly less than that of the unmodified prints for all four databases, with scores around 100. This result suggests that a selective encryption of 4 or 5 bitplanes could lead to a rather secure strategy since the remaining less significant bitplanes do hardly contribute to the matching score at all.

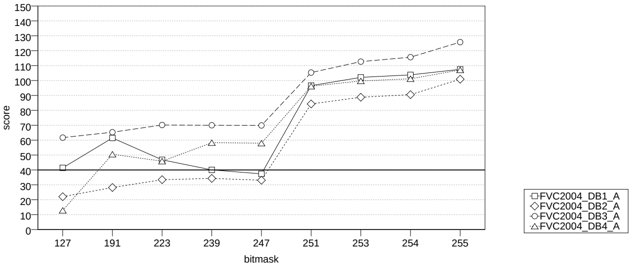


Fig. 19. Average matching scores with single bitplanes missing.

In Figure 20 we display matching results in case more bitplanes are "encrypted" (i.e., more bitplanes are missing). Bitmasks 63 to 7 simulate scenarios where the MSB-plane plus additional subsequent bitplanes are encrypted (and attacked). Results indicate that encryption of the MSB-plane plus at least two bitplanes lead to results sufficiently below the "acceptance score" of 40. This is true in any case for DB2 and D4, DB1 and DB3 give higher values but those DBs seem to give higher matching values also in other scenarios (compare Figure 18).

In Figure 19 for DB1, bitmask 247 showed the lowest score value, which seems to indicate that the 5th significant bitplane might have the strongest impact if missing for these data. For this reason, bitmasks 71 to 231 simulate scenarios where this bitplane and additional bitplanes with varying significance are encrypted (and attacked). Interestingly, while the expected behavior for DB1 (i.e., low score values) can be observed in Figure 20 for bitmask 71, it does not show up for the bitmasks 135, 199, and 231. Instead, the results for DB1 for these bitmasks show another unexpected behavior – bitmasks 135 and 199 give a higher score than mask 231 although more bitplanes (i.e., more information) are missing. The reason is that missing binary positions in the pixels representation may lead to artifacts negatively influencing matching behavior. For

[3] `http://fingerprint.nist.gov/NFIS/`

bitmask 71 one encrypts the MSB-plane in addition to the 5^{th} bitplane (among others) which gives the lowest score for all bitmasks not containing the 5^{th} bitplane, including those for DB1. This shows that for all data considered selective bitplane encryption should involve protection of the MSB-plane.

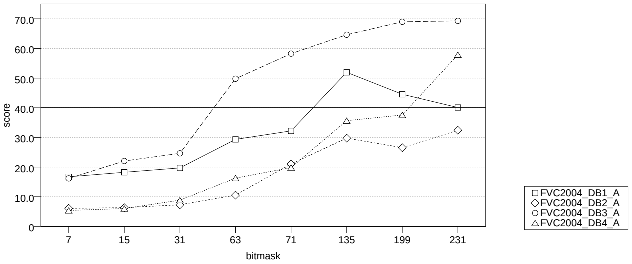


Fig. 20. Average matching scores with various bitplanes missing.

In the following, we relate our findings to a different fingerprint recognition system and a different set of images. Figures 21 to 23 show a comparison of the NFIS matcher with VeriFinger from Neurotechnologija,[4] using the average of 3 selected fingerprints (these images come with the distribution of the demonstration software of VeriFinger which relies on minutiae matching without employing core and delta points and claims tolerance to translation and rotation). The VeriFinger data are obtained from manually matching the 24 bitmasked images for each fingerprint with the demo version of the application. The same bitmasks of the same fingerprints are matched with the NFIS matcher, the score shown is the relative matching score of a bitmask to the score obtained by matching the print to itself.

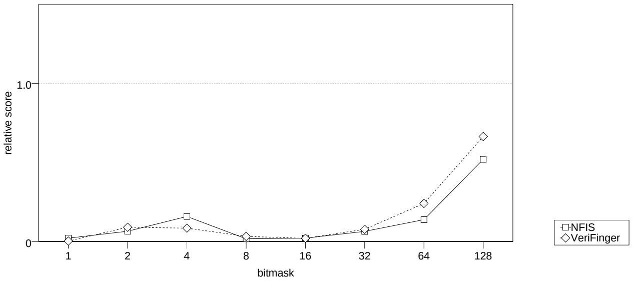


Fig. 21. Comparison of NFIS and VeriFinger (single bitplanes).

While observing a similar behavior for the matching scores of single bitplanes (Figure 21), the simulation of selective bitplane encryption reveals that VeriFinger is much less sensitive to the absence of single bitplanes. Therefore, the effect of selective bitplane encryption is highly dependent of the employed fingerprint recognition system. Additionally, the shape of the "scores-curve" is again not monotonically increasing for a decreased significance of the omitted bitplane and is quite different from all four curves obtained from the FVC2004 data. As a consequence, selective bitplane encryption needs also to be tuned to the type of fingerprint sensor employed.

However, when considering the results when protecting the MSB-plane and subsequent significant bitplanes (see bitmasks 7 to 63 and 71 in Figure 23) it turns out that this strategy is also sensible for the VeriFinger scheme and not only for the NIST software. It should be noted that the encryption complexity is reduced only to 37.5% if the MSB-plane and two additional bitplanes are encrypted, and it remains questionable if there indeed exist application scenarios where this makes sense.

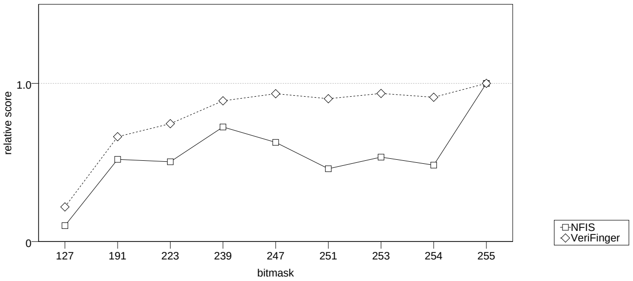


Fig. 22. Comparison of NFIS and VeriFinger (single bitplanes missing).

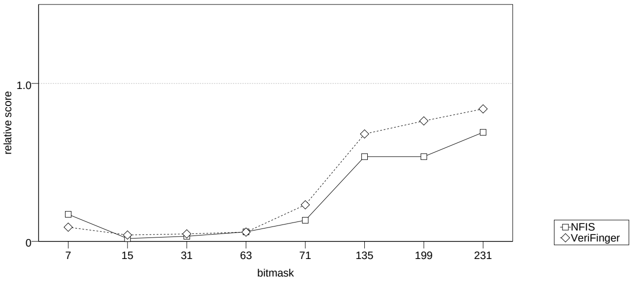


Fig. 23. Comparison of NFIS and VeriFinger (various bitplanes missing).

III. Conclusion

We have analyzed two lightweight encryption schemes for fingerprints relying on the bitplane representation of the images. The first approach, employing an XOR-encryption between the image data and the fingerprints LSB data (used as key in repeated manner) with subsequent AES encryption of the key data, is shown to suffer from several shortcomings. We demonstrate a computationally low-cost attack exploiting the fact that key data is XORed with itself which renders this encryption strategy useless. In this context we propose a few improvements of the scheme. The security of the second approach, based on selective encryption of fingerprint bitplanes, is shown to be highly dependent on the actual fingerprint recognition scheme used and has to be optimized with respect to the used fingerprint sensor to acquire the image data. A strategy which protects the MSB-plane and at least two additional bitplanes has been shown to be secure in terms of low matching scores for two fingerprint recognition systems.

We have shown that simple schemes used to secure fingerprint image data may be a severe threat to the security of that kind of biometric authentication schemes. It has to be pointed out that fundamental knowledge in the cryptographic area has to be obeyed as well, when designing lightweight encryption schemes and that it may be necessary to optimize such schemes with respect to the actual environment where they are employed.

[4] http://neurotechnologija.com

References

[1] J. Daemen and V. Rijmen, *The Design of Rijndael: AES — The Advanced Encryption Standard*. Springer Verlag, 2002.

[2] M. V. Droogenbroeck and R. Benedett, "Techniques for a selective encryption of uncompressed and compressed images," in *Proceedings of ACIVS (Advanced Concepts for Intelligent Vision Systems)*, Ghent University, Belgium, Sept. 2002, pp. 90–97.

[3] P. Ekdahl and T. Johansson, *A New Version of the Stream Cipher SNOW*, ser. LNCS. Berlin, Heidelberg, New York, Tokyo: Springer-Verlag, 2003, vol. 2595, pp. 47–61.

[4] S. Halevi, D. Coppersmith, and C. Jutla, *Scream: A Software-Efficient Stream Cipher*, ser. LNCS. Berlin, Heidelberg, New York, Tokyo: Springer-Verlag, 2002, vol. 2365, pp. 195–209.

[5] P. Hellekalek and S. Wegenkittl, "Empirical evidence concerning AES," *ACM Trans. Model. Comput. Simul.*, vol. 13, no. 4, pp. 288–302, 2003.

[6] T. Jakobsen, "A fast method for the cryptanalysis of substitution ciphers," *Cryptologia*, vol. 19, no. 3, pp. 265–274, 1995.

[7] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. Springer Verlag, 2003.

[8] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, 1998.

[9] A. Moffat, R. M. Neal, and I. H. Witten, "Arithmetic coding revisited," *ACM Trans. Inf. Syst.*, vol. 16, no. 3, pp. 256–294, 1998.

[10] D. Moon, Y. Chung, S. B. Pan, K. Moon, and K. I. Chung, "An efficient selective encryption of fingerprint images for embedded processors," *ETRI Journal*, vol. 28, no. 4, pp. 444–452, Aug. 2006.

[11] M. Podesser, H.-P. Schmidt, and A. Uhl, "Selective bitplane encryption for secure transmission of image data in mobile environments," in *CD-ROM Proceedings of the 5th IEEE Nordic Signal Processing Symposium (NORSIG 2002)*. Tromso-Trondheim, Norway: IEEE Norway Section, Oct. 2002.

[12] N. Ratha, J. Connell, and R. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Systems Journal*, vol. 40, no. 3, pp. 614–634, 2001.

[13] G. Rose and P. Hawkes, *Turing: a fast stream cipher*, ser. LNCS. Berlin, Heidelberg, New York, Tokyo: Springer-Verlag, 2003, vol. 2887, pp. 290–306.

[14] B. Schneier, *Applied cryptography (2nd edition): protocols, algorithms and source code in C*. Wiley Publishers, 1996.

[15] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. Kumar, "Biometric encryption using image processing," in *Optical Security and Counterfeit Deterrence Techniques II*, ser. Proceedings of SPIE, vol. 3314, 1998, pp. 178–188.

[16] U. Uludag, S. Pankanti, and S. Prabhakar, "Biometric cryptosystems: issues and challenges," *Proceedings of the IEEE*, vol. 92, no. 6, pp. 948–960, 2004.

Dominik Engel studied Computer Science at Salzburg University (Austria), where he focussed on image processing and graduated in 2002. As a research assistant at the department of Computer Sciences of Salzburg University, he is currently finishing his PhD in the area of multimedia security. He is also a lecturer at the Salzburg University of Applied Sciences.

Elias Pschernig holds a Bachelor degree in Computer Science from Salzburg University (Austria). His Master thesis is in the field of multimedia signal processing and security.

Andreas Uhl is an associate professor at the Computer Sciences department of Salzburg University (Austria) where he leads the Multimedia Signal Processing and Security Lab. He is also lecturer at the Carinthia Tech Institute and the Salzburg University of Applied Sciences. His research interests include image and video processing, wavelets, multimedia security, biometrics, parallel algorithms, and numbertheoretical numerics.