AN ATTACK AGAINST IMAGE-BASED SELECTIVE BITPLANE ENCRYPTION

Dominik Engel and Andreas Uhl

Department of Computer Sciences
Salzburg University
Jakob-Haringer-Str. 2, A–5020 Salzburg, Austria
Email: dengel@cosy.sbg.ac.at; uhl@cosy.sbg.ac.at

ABSTRACT

We analyze a recently published lightweight encryption scheme for fingerprint images and discuss several shortcomings. A low-cost attack on this scheme is proposed, which allows access to the full plaintext for most given ciphertexts. We give some recommendations for improvements of the encryption scheme, but conclude that the analyzed scheme remains insecure.

Index Terms— lightweight encryption, selective bitplane encryption, least significant bitplane, attack, fingerprints

1. INTRODUCTION

The increasing use of biometric systems raises the question of how to store and handle biometric data in a secure way. In this respect, sample data, i.e. data acquired by the sensor, is more sensitive than template data. Whereas template data contains a description of the biometric features that pertain only to the used system, sample data can potentially be used on other systems as well. As generally the revocation of biometric data is extremely problematic, a compromise of this data has severe security implications. Therefore, the secure handling and storage of biometric sample data may become imperative due to the security and privacy concerns of the users.

Controlling the computational demand is important, especially in distributed scenarios with weak and low-power sensor devices. Classical encryption techniques can be too demanding to be employed, therefore a careful but significant reduction of encryption complexity is required for this type of applications. The limited computational resources in embedded processors are addressed in recent work by Moon et al. [1], where an approach involving selective encryption of fingerprint images employing XOR on a bitplane basis is suggested. In this work, we analyze this approach in detail and demonstrate several shortcomings and a computationally efficient attack.

2. IMAGE-BASED SELECTIVE BITPLANE ENCRYPTION

In recent work [1], a lightweight fingerprint image encryption technique has been proposed, which has been denoted as "image-based selective bitplane encryption protocol". The approach, which is essentially a Vigenère cipher, constructs a keystream from the least significant bit (LSB) plane of the input image. This keystream is XORed with the binary representation of the plaintext data and then encrypted with AES. The aim of the approach is a reduction in computational complexity to facilitate real-time processing on low-end processors.

This work has been partially supported by the Austrian Science Fund (FWF) under Project No. P19159-N13 and by the European Commission through the IST Programme under contract IST-2002-507932 ECRYPT.

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, it is repeated for the remaining binary data of I 7 times using the identical key K_0. Finally, K_0 is encrypted using AES and transmitted to the receiver together with the encrypted data c_i, $0 \leq i \leq s - 1$.

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 [1]).

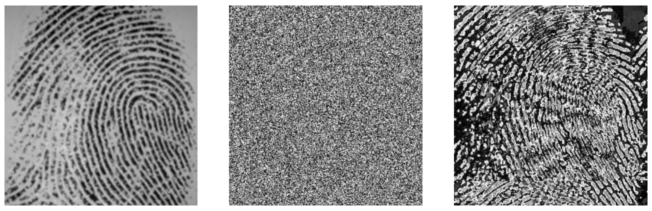
3. VULNERABILITIES

3.1. Key-length

Encryption with simple XOR is only secure if the keystream is truly random and of the same length as the plaintext, i.e. if it is a one-time pad (e.g. [2]). Both conditions are violated for the proposed approach. The key-length, which is an eighth of the message length in the proposed approach, 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 [2]. Figure 1 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.

The images we use for our tests are taken from the databases of the fingerprint verification contest 2004 (FVC2004)[1]. Databases DB1 and DB2 contain images obtained with two different optical sensors. The images in DB3 have been acquired with a thermal sensor. DB4 contains synthetic fingerprint images.

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

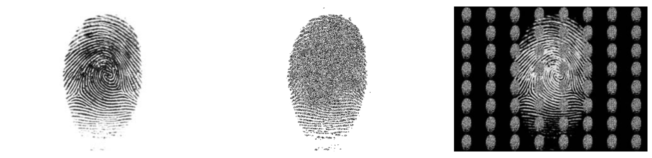


(a) Original image (b) Ciphertext C (c) $C \oplus (C >> \mathit{key\_length})$

Fig. 1. Illustration of shifting the ciphertext by key-length (part of DB4_1_8).

3.2. Randomness

Another problem lies with the assumption that for fingerprint sensors the LSB-plane is sufficiently random and not correlated with the other bitplanes. The authors of [1] argue that 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" and that the LSB plane "looks similar to a random number field". Figure 2 shows that this is not the case. The image 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 2(c).



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

Fig. 2. Random LSB-plane: a counterexample (DB1_1_3).

3.3. 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 6.

3.4. Data expansion

The ciphertext is 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 avoided, which will be discussed in Section 6 as well.

4. 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

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

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. $k_j \hat{\oplus} k_i$, 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}, $K_{i+1} \subseteq K_i$:

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

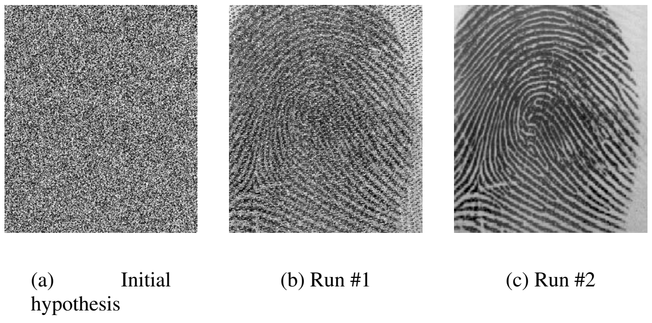
As long as one or more bits from K_{i+1} are mapped to the same bits in K_i, K_{i+1} is a proper subset of K_i: $K_{i+1} \subset K_i$, i.e. we reduce the number of referenced keybits. It can easily be seen that after a number of iterations N no more reduction is possible:

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

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. Generally, K_N will be too large for a brute force search. Note that if s is coprime to 8, then $K_N = K_0$, i.e. no reduction is possible. 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. E.g., for a key of length 73, k_{30} is mapped to k_{28} which is mapped to k_{12} which is mapped to k_{30} 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.



(a) Initial hypothesis (b) Run #1 (c) Run #2

Fig. 3. Variance attack on DB2_3_3.

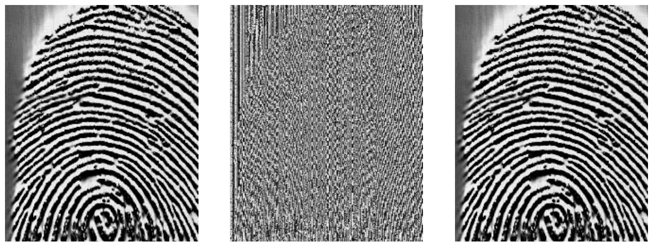
If the correct setting for one bit in the chain is known, then the complete chain can be reconstructed. So we can choose a single element from each chain as a representative. Each of these representative elements influences a multitude of bit positions in the plaintext.

As an example, we investigate fingerprint images used by [1]: 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.

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 needs not be overly sophisticated. For natural images, the sample variance can be used as a simple measure, for fingerprint images we introduce a more suitable measure below. 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. 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 [3].

This process is illustrated for a version of the DB2_3_3 fingerprint image of size 248×292 in Figure 3. Each image represents a whole run over the 175 chain bits. After run number 2 the variance does not change anymore 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 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 4(a). In such cases, the minimum variance fails as a condition for the correct plaintext image, as shown in Figure 4(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 4(c) shows the attack result for Figure 4(a).



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

Fig. 4. Variance versus neighborhood attack (part of DB3_84_2).

5. EVALUATION

If the proposed attack is successful for a given ciphertext, the decrypted image will generally be bitwise identical to the original image. The 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. We verified this by successfully attacking images for which the LSB was replaced by a pseudo-random number field. Furthermore, the attack is not restricted to fingerprint images, but also works for natural images. The attack can be adapted to work with any type of plaintext, if a suitable measure can be found to be used in the iteration for this type of plaintext.

There are some key-lengths that produce a large number of short chains, each of which only influences relatively few positions in the image. In these cases, the measures for the plaintext are too crude to produce a successful attack. To quantify the success rate of the proposed attack, we investigate the ratio of the number of chains to the number of bits in the key for image sizes ranging from 64^2 to 512^2. Figure 5 shows the ratio of key-lengths on the ordinate that achieve a certain ratio of number of chains to total key bits, given on the abscissa.

We found that the attack reliably produces the original plaintext for ratios that are below approximately 0.02. It can be seen that for 95.5% of the key-lengths the ratio lies below 0.01. 98.2% of the key-lengths lie below 0.02. That means that for 98.2% of the possible image sizes between 64^2 to 512^2 the proposed attack will work reliably for natural and fingerprint images. For the rest, the reliability of the attack decreases with the number of representative bits. The time the attack needs increases with the number of bits. A more sophisticated measure could help to improve reliability in the range above 2%, at the expense of higher computational demands.

Note that the proposed attack cannot be transferred to Vigenère encryption in general (i.e. encryption with keystreams shorter than the plaintext where the keystream is not XORed with itself). Even for very short keys with a couple of hundred bits, for which each of the bits in the key influences many bits in the image, the iterative attack is unsuccessful. Figure 6 illustrates this for the Lena image: a

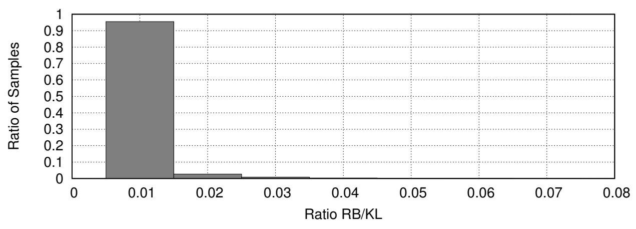
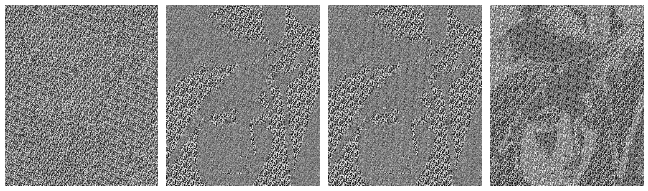


Fig. 5. Ratio of Samples by Ratio of (Representative Bits)/(Total Keybits)



(a) Ciphertext (b) Variance (c) Neighborhood (d) Blocked Variance

Fig. 6. Unsuccessful attacks for general Vigenère encryption

700-bit pseudo-random one-time pad is used for encryption (a), neither variance (b) nor local neighborhood (c) can retrieve the plaintext. A blocked version of the variance leads to slightly better results (d). The proposed attack fails, because it utilizes the fact that the reduction of the key results in an irregular influence of the representative bits on the bits in the image, which facilitates the use of very simple plaintext measures. For general Vigenère encryption, a more suitable and possibly more complex measure has to be used. Substitution ciphers with short keys have been shown to be easy to crack [2], especially for plaintexts with low entropy like natural language texts and images.

The computational demands of the attack depend on the size of the image and the number of representative bits. Table 1 shows some timing results, which 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 for an attack on DB3_84_2 was unsuccessful (†), all other attacks produced the original plaintext image. Some of the images were cropped to a certain size (marked by $\star$) to reflect different key-lengths (and to remove the scanner background in some cases). 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 needed.

6. IMPROVEMENTS

6.1. 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
DB1_1_3	640 × 480	8	13.4 s	7.5 s
DB4_1_8	233 × 384	8	4.4 s	4.8 s
DB3_84_2	300 × 480	32	†	14.8 s
DB3_84_2*	205 × 251	527	†	96.9 s
DB2_3_3	328 × 364	210	67 s	107 s
DB2_3_3*	248 × 292	175	52 s	54 s
DB2_3_3*	187 × 279	1556	433.2 s	529.3 s
DB2_3_3*	256 × 256	1	1.2 s	

Table 1. Timing results.

6.2. Randomness

In order to produce a keystream that exhibits more properties of a random number field, we suggest to extract the LSB-plane first (or any other bitplane), subject it to AES encryption, and finally 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 [4]. This procedure also invalidates the proposed attack. It has to be noted, however, that with this approach, the encrypted LSB-plane has to be regarded as proper key material and has to be transferred over a secure channel.

6.3. 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 like RC-4 [2].

7. CONCLUSION

We have analyzed a published encryption scheme for fingerprint images. With the computationally undemanding attack we presented, access to the full plaintext can be obtained for most given ciphertexts. We have shown that although some improvements to the original scheme are possible, it remains insecure.

Simplistic schemes used to secure fingerprint image data may be a severe threat to the security of the biometric data. It has to be pointed out that fundamental knowledge in the cryptographic area has to be obeyed as well, when designing lightweight encryption schemes.

8. REFERENCES

[1] Daesung Moon, Yongwha Chung, Sung Bum Pan, Kiyoung Moon, and Kyo Il Chung, "An efficient selective encryption of fingerprint images for embedded processors," *ETRI Journal*, vol. 28, no. 4, pp. 444–452, Aug. 2006.

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

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

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