Security Problems with Improper Implementations of Improved FEA-M

Shujun Li * and Kwok-Tung Lo

Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong SAR, China

Abstract

This paper reports security problems with improper implementations of an improved version of FEA-M (fast encryption algorithm for multimedia). It is found that an implementation-dependent differential chosen-plaintext attack or its chosen-ciphertext counterpart can reveal the secret key of the cryptosystem, if the involved (pseudo-)random process can be tampered (for example, through a public time service). The implementation-dependent differential attack is very efficient in complexity and needs only $O(n^2)$ chosen plaintext or ciphertext bits. In addition, this paper also points out a minor security problem with the selection of the session key. In real implementations of the cryptosystem, these security problems should be carefully avoided, or the cryptosystem has to be further enhanced to work under such weak implementations.

Key words: multimedia encryption, FEA-M, insecure implementation, differential attack, chosen-plaintext attack, chosen-ciphertext attack, pseudo-random process

1 Introduction

Multimedia data play important roles in today's digital world. In many multimedia applications, such as pay-TV services, commercial video conferences and medical imaging systems, fast and secure encryption methods are required to protect the multimedia contents against malicious attackers. In recent years, many different multimedia encryption schemes have been proposed to fulfill such an increasing demand (Uhl and Pommer, 2005; Furht et al., 2004; Li

⋆ This paper has been published in *Journal of Systems & Software*, vol. 80, no. 5, pp. 791-794, 2007, doi: 10.1016/j.jss.2006.05.002.

* The corresponding author, personal web site: `http://www.hooklee.com`.

et al., 2004). In (Yi et al., 2001), a new fast encryption algorithm for multimedia (FEA-M) was proposed, which bases the security on the complexity of solving nonlinear Boolean equations. Later FEA-M was employed to construct a key agreement protocol by the same authors in (Yi et al., 2002). Since then, some attacks of FEA-M have been reported (Mihaljević and Kohno, 2002; Mihaljević, 2003; Wu et al., 2003; Youssef and Tavares, 2003), most of which can break the key with a smaller complexity than the simple brute force attack (Mihaljević and Kohno, 2002; Mihaljević, 2003; Wu et al., 2003), and one of which can completely break the whole cryptosystem with only one known and two chosen plaintext blocks (Youssef and Tavares, 2003).

To enhance the security and to avoid some other defects, an improved version of FEA-M was proposed in (Mihaljević, 2003). This paper reports some security problems with improper implementations of the cryptosystem. We point out that the secret key of the cryptosystem can be revealed by an implementation-dependent differential attack if the involved (pseudo-)random process can be tampered. One of such situations is when the pseudo-random process is uniquely controlled by an external source (such as a public time service), though it appears that such an implementation would not compromise the security of the cryptosystem itself. The proposed differential attack is very efficient, since only two pairs of chosen plaintext blocks are needed to completely reveal the key. As a result, in a real implementation of the cryptosystem, it should be ensured that the embedded pseudo-random process cannot be controlled by illegal users. Or, the improved FEA-M has to be further enhanced to resist this implementation-dependent attack. In addition, a minor problem with the selection of the session key is also discussed in this paper.

2 Improved FEA-M

The original FEA-M (Yi et al., 2001) is a block cipher with both plaintext and ciphertext feedback. It encrypts the plaintext in the form of $n \times n$ Boolean matrices, by an $n \times n$ Boolean key matrix. The elements of the matrices are either 0 or 1 and all matrix operations are made over $GF(2)$, i.e., modulo 2. As a result, the ciphertext is also in the form of $n \times n$ Boolean matrices.

Previous works have shown that the original FEA-M has the following defects: 1) the key can be easily broken by an adaptive chosen-plaintext attack proposed in (Youssef and Tavares, 2003); 2) an efficient known-plaintext attack can break it with a complexity smaller than the brute force attack (Mihaljević and Kohno, 2002; Mihaljević, 2003; Wu et al., 2003); 3) it is sensitive to packet loss (Mihaljević, 2003) and channel errors due to the use of plaintext feedback.

To overcome the above-mentioned security defects, Mihaljević proposed an improved FEA-M in 2003. The improved scheme contains two stages: key distribution and working stage. The first stage generates two $n \times n$ secret Boolean matrices, a session key $\boldsymbol{K}$ and an initial matrix $\boldsymbol{V}$, generally from a master key $\boldsymbol{K}_0$, which is also an $n \times n$ Boolean matrix and known by both the sender and the receiver. The key distribution protocol is actually the one used in (Yi et al., 2002) and can be described as follows.

- The sender selects $\boldsymbol{K}$ and $\boldsymbol{V}$ via a (pseudo-)random process, and computes

$$\boldsymbol{K}^* = \boldsymbol{K}_0 \boldsymbol{K}^{-1} \boldsymbol{K}_0, \tag{1}$$
$$\boldsymbol{V}^* = \boldsymbol{K}_0 \boldsymbol{V} \boldsymbol{K}_0, \tag{2}$$

  then sends $(\boldsymbol{K}^*, \boldsymbol{V}^*)$ to the receiver.
- The receiver recovers $\boldsymbol{K}^{-1}$ and $\boldsymbol{V}$ by computing

$$\boldsymbol{K}^{-1} = \boldsymbol{K}_0^{-1} \boldsymbol{K}^* \boldsymbol{K}_0^{-1}, \tag{3}$$
$$\boldsymbol{V} = \boldsymbol{K}_0^{-1} \boldsymbol{V}^* \boldsymbol{K}_0^{-1}. \tag{4}$$

After the key distribution stage, the sender and the receiver sides can start the encryption/decryption procedure with the session key $\boldsymbol{K}$ and the initial matrix $\boldsymbol{V}$. Denoting the i-th $n \times n$ plain-matrix by $\boldsymbol{P}_i$ and the i-th $n \times n$ cipher-matrix by $\boldsymbol{C}_i$, the encryption procedure is as follows:

$$\boldsymbol{C}_i = \boldsymbol{K} \left(\boldsymbol{P}_i + \boldsymbol{K} \boldsymbol{V} \boldsymbol{K}^i \right) \boldsymbol{K}^{n+i} + \boldsymbol{K} \boldsymbol{V} \boldsymbol{K}^i, \tag{5}$$

and the decryption procedure is

$$\boldsymbol{P}_i = \boldsymbol{K}^{-1} \left(\boldsymbol{C}_i + \boldsymbol{K} \boldsymbol{V} \boldsymbol{K}^i \right) \boldsymbol{K}^{-(n+i)} + \boldsymbol{K} \boldsymbol{V} \boldsymbol{K}^i. \tag{6}$$

The above procedure repeats for each plain/cipher-matrix until the plaintext/ciphertext exhausts.

3 Implementation-Dependent Differential Attack

In this section, we describe an implementation-dependent differential attack of the improved FEA-M. This attack works under the conditions that one can tamper the involved (pseudo-)random process of the improved FEA-M to use the same $\boldsymbol{K}$ and $\boldsymbol{V}$ in two separate encryption sessions.

Given two plain-matrices, $\boldsymbol{P}_i^{(1)}$ and $\boldsymbol{P}_i^{(2)}$, and their corresponding cipher-matrices, $\boldsymbol{C}_i^{(1)}$ and $\boldsymbol{C}_i^{(2)}$, we can get Eq. (7).

$$\begin{aligned}\boldsymbol{C}_i^{(1)} + \boldsymbol{C}_i^{(2)} &= \left(\boldsymbol{K}\left(\boldsymbol{P}_i^{(1)} + \boldsymbol{K}\boldsymbol{V}\boldsymbol{K}^i\right)\boldsymbol{K}^{n+i} + \boldsymbol{K}\boldsymbol{V}\boldsymbol{K}^i\right) \\ &\quad + \left(\boldsymbol{K}\left(\boldsymbol{P}_i^{(2)} + \boldsymbol{K}\boldsymbol{V}\boldsymbol{K}^i\right)\boldsymbol{K}^{n+i} + \boldsymbol{K}\boldsymbol{V}\boldsymbol{K}^i\right) \\ &= \boldsymbol{K}\left(\boldsymbol{P}_i^{(1)} + \boldsymbol{K}\boldsymbol{V}\boldsymbol{K}^i\right)\boldsymbol{K}^{n+i} + \boldsymbol{K}\left(\boldsymbol{P}_i^{(2)} + \boldsymbol{K}\boldsymbol{V}\boldsymbol{K}^i\right)\boldsymbol{K}^{n+i} \\ &= \boldsymbol{K}\left(\boldsymbol{P}_i^{(1)} + \boldsymbol{P}_i^{(2)}\right)\boldsymbol{K}^{n+i}\end{aligned} \tag{7}$$

Apparently, Eq. (7) means a simple relation between $\Delta\boldsymbol{C}_i = \boldsymbol{C}_i^{(1)} + \boldsymbol{C}_i^{(2)} = \boldsymbol{C}_i^{(1)} - \boldsymbol{C}_i^{(2)}$ and $\Delta\boldsymbol{P}_i = \boldsymbol{P}_i^{(1)} + \boldsymbol{P}_i^{(2)} = \boldsymbol{P}_i^{(1)} - \boldsymbol{P}_i^{(2)}$, i.e., the plaintext and the ciphertext differentials (sums):

$$\Delta\boldsymbol{C}_i = \boldsymbol{K}\left(\Delta\boldsymbol{P}_i\right)\boldsymbol{K}^{n+i}. \tag{8}$$

As a result, for two consecutive plaintext-matrices, if we choose $\Delta\boldsymbol{P}_{i+1} = \Delta\boldsymbol{P}_i$, we can immediately deduce:

$$\begin{aligned}\Delta\boldsymbol{C}_{i+1} &= \boldsymbol{K}\left(\Delta\boldsymbol{P}_{i+1}\right)\boldsymbol{K}^{n+i+1} \\ &= \boldsymbol{K}\left(\Delta\boldsymbol{P}_i\right)\boldsymbol{K}^{n+i}\cdot\boldsymbol{K} \\ &= \Delta\boldsymbol{C}_i\boldsymbol{K}.\end{aligned} \tag{9}$$

Thus, if $\Delta\boldsymbol{C}_i$ is invertible, the session key can be derived easily as follows:

$$\boldsymbol{K} = \left(\Delta\boldsymbol{C}_i\right)^{-1}\Delta\boldsymbol{C}_{i+1}. \tag{10}$$

To make $\Delta\boldsymbol{C}_i$ invertible, one should choose $\Delta\boldsymbol{P}_i$ to be an invertible matrix over $GF(2)$, where note that $\boldsymbol{K}$ is always invertible following the design of the cryptosystem.

After $\boldsymbol{K}$ is broken, one can substitute it into Eq. (5) to get a linear equation with n^2 unknown variables, i.e., the n^2 elements of the initial matrix $\boldsymbol{V}$:

$$\boldsymbol{V}\boldsymbol{K}^{n+i} + \boldsymbol{K}^{-1}\boldsymbol{V} = \boldsymbol{K}^{-2}\left(\boldsymbol{C}_i - \boldsymbol{K}\boldsymbol{P}_i\boldsymbol{K}^{n+i}\right)\boldsymbol{K}^{-i}. \tag{11}$$

By solving this linear equation, it is easy to recover $\boldsymbol{V}$. Actually, we can further reduce the linear equation to directly deduce $\boldsymbol{V}$. Choosing two continuous plaintext matrices $\boldsymbol{P}_i$, $\boldsymbol{P}_j$ and adding the two linear systems, one has

$$\begin{aligned}\boldsymbol{V}\boldsymbol{K}^{n+i}\left(\boldsymbol{I} + \boldsymbol{K}^{j-i}\right) &= \boldsymbol{K}^{-2}\left(\boldsymbol{C}_i - \boldsymbol{K}\boldsymbol{P}_i\boldsymbol{K}^{n+i}\right)\boldsymbol{K}^{-i} \\ &\quad + \boldsymbol{K}^{-2}\left(\boldsymbol{C}_j - \boldsymbol{K}\boldsymbol{P}_j\boldsymbol{K}^{n+j}\right)\boldsymbol{K}^{-j}.\end{aligned} \tag{12}$$

When $\boldsymbol{I} + \boldsymbol{K}^{j-i}$ is invertible, $\boldsymbol{V}$ can be immediately solved by multiplying the right side by $\left(\boldsymbol{I} + \boldsymbol{K}^{j-i}\right)^{-1}\boldsymbol{K}^{-(n+i)}$ at the end. Note that $\boldsymbol{K}^{n+i} + \boldsymbol{K}^{n+j}$

may never be invertible over $GF(2)$ (for example, when $\boldsymbol{K} = \boldsymbol{I}$), though the probability is relatively small when n is relatively high. Once such an event occurs, one can turn to solve Eq. (11). If $\boldsymbol{V}$ can still not be solved from Eq. (11), one has to carry out the attack with some other different values of $\boldsymbol{K}$ until $\boldsymbol{V}$ can be uniquely solved.

Once $\boldsymbol{K}$ and $\boldsymbol{V}$ are both known, one can use the method proposed in Sec. III of (Youssef and Tavares, 2003) to recover the master key $\boldsymbol{K}_0$.

To carry out a successful attack, in most cases, the attacker only needs to choose two plaintexts with four chosen plaintext matrices, $\boldsymbol{P}_i^{(1)}$, $\boldsymbol{P}_{i+1}^{(1)}$, $\boldsymbol{P}_i^{(2)}$ and $\boldsymbol{P}_{i+1}^{(2)}$, which satisfy $\boldsymbol{P}_{i+1}^{(1)} - \boldsymbol{P}_{i+1}^{(2)} = \boldsymbol{P}_i^{(1)} - \boldsymbol{P}_i^{(2)} = \Delta\boldsymbol{P}$ and $\Delta\boldsymbol{P}$ is an invertible matrix. Considering each matrix is a $n \times n$ Boolean matrix, $4n^2$ chosen plain-bits are required in total. When $n = 64$, as suggested in (Yi et al., 2001, 2002), only 2048 plain-bytes are needed. In addition, the complexity of the proposed attack is very small, actually it is of the same order as the one proposed in (Youssef and Tavares, 2003). In the case that $\boldsymbol{V}$ can not be solved with four chosen plaintext matrices, more plaintext matrices have to be chosen, but the number of chosen plaintext bits is still of the same order – $O(n^2)$.

Next, let us see in which improper implementations an attacker can manage to tamper the involved (pseudo-)random process to activate the above differential attack. Apparently, the above attack requires two encryption sessions with the same session key $\boldsymbol{K}$ and the same initial matrix $\boldsymbol{V}$, one for encrypting the first plaintext $\left\{\cdots, \boldsymbol{P}_i^{(1)}, \boldsymbol{P}_{i+1}^{(1)}\right\}$ and the other for encrypting the second plaintext $\left\{\cdots, \boldsymbol{P}_i^{(2)}, \boldsymbol{P}_{i+1}^{(2)}\right\}$. However, in each encryption session, $\boldsymbol{K}$ and $\boldsymbol{V}$ have to be reset at the sender side via a (pseudo-)random process and distributed to the receiver side via the key distribution protocol. As a result, generally two different sessions use different $\boldsymbol{K}$ and $\boldsymbol{V}$. However, in real world the encryption scheme may be improperly implemented such that the attacker can tamper the (pseudo-)random process. As a typical example, let us assume that the process is uniquely determined by the system clock [1]. In chosen-plaintext attacks, the attacker has a temporary access to the encryption machine, so he can intentionally alter the system clock to control the (pseudo-)random process before running each session to get the same $\boldsymbol{K}$ and $\boldsymbol{V}$ for two separate sessions. In addition, if the improved FEA-M is implemented in such an insecure way that the second stage can restart without running the key distribution stage, the attack becomes straightforward.

[1] In (Yi et al., 2001, 2002; Mihaljević, 2003), it is not mentioned how to realize the random process. One of the simplest (though maybe less frequently-used) method to realize a pseudo-random process is to initialize the seed of the pseudo-random number generator using the current time stamp. A list of some other more complicated ways can be found in Section "The Collection of Data Used to Create a Seed for Random Number" of (Microsoft Corporation, 2005).

At last, it deserves mentioned that the above differential chosen-plaintext attack can be easily to generalize to a differential chosen-ciphertext attack, provided that the (pseudo-)random process at the decryption machine can be tampered. Rewrite Eq. (8) into the following form:

$$\Delta \boldsymbol{P}_i = \boldsymbol{K}^{-1} \left(\Delta \boldsymbol{C}_i \right) \boldsymbol{K}^{-(n+i)}. \tag{13}$$

Then, by choosing $\Delta \boldsymbol{C}_{i+1} = \Delta \boldsymbol{C}_i$, one has

$$\begin{aligned} \Delta \boldsymbol{P}_{i+1} &= \boldsymbol{K}^{-1} \left(\Delta \boldsymbol{C}_{i+1} \right) \boldsymbol{K}^{-(n+i+1)} \\ &= \boldsymbol{K}^{-1} \left(\Delta \boldsymbol{C}_i \right) \boldsymbol{K}^{-(n+i)-1} \\ &= \Delta \boldsymbol{P}_i \boldsymbol{K}^{-1}. \end{aligned} \tag{14}$$

Other steps are identical with the above differential chosen-plaintext attack.

4 A Minor Problem with Selection of Session Key

It is noticed that $\boldsymbol{K}$ cannot be selected at random from all invertible matrices over $GF(2)$. Since all $n \times n$ invertible matrices form a general linear group $GL(n, 2)$, whose order is $O = \prod_{i=0}^{n-1} (2^n - 2^i)$ (Wikipedia, 2005). So, denoting the order of $\boldsymbol{K}$ over $GL(n, 2)$ by $o(\boldsymbol{K})$, it is true that $o(\boldsymbol{K}) \mid O$, i.e., $\boldsymbol{K}^{o(\boldsymbol{K})} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity Boolean matrix (Gilbert and Gilbert, 2005). It is obvious that $o(\boldsymbol{K})$ actually corresponds to the periodicity of the encryption/decryption function with respect to the plaintext/ciphertext index i. Generally speaking, the periodicity should not be too small to maintain an acceptable security level. As an extreme example, when $\boldsymbol{K} = \boldsymbol{I}$, $o(\boldsymbol{K}) = 1$ and the encryption procedure becomes $\boldsymbol{C}_i = \boldsymbol{P}_i$ (the cipher vanishes). Thus, $\boldsymbol{K}$ should be selected randomly from all invertible Boolean matrices with sufficiently large orders, which means a significant reduction of the session key space.

5 Conclusions

This paper reports an implementation-dependent differential attack of an improved fast encryption algorithm for multimedia (FEA-M) proposed in (Mihaljević, 2003). The attack works under the condition where the involved (pseudo-)random process can be tampered by the attacker. In this case, the attack can reveal the key with four or more chosen plaintext/ciphertext matrices, i.e., $4n^2$ chosen plain/ciphertext bits, in two or more separate encryption sessions.

The result shows that a secure cryptosystem may become totally insecure with seemingly-harmless implementation details in real world (Schneier, 2000). In addition, a minor problem with the selection of the session key is also discussed in this paper.

6 Acknowledgements

This research was supported by The Hong Kong Polytechnic University's Post-doctoral Fellowships Scheme under grant no. G-YX63. The authors thank the anonymous reviewers for their valuable comments to enhance the quality of this paper.

References

Furht, B., Socek, D., Eskicioglu, A. M., December 2004. Fundamentals of multimedia encryption techniques. In: Furht, B., Kirovski, D. (Eds.), Multimedia Security Handbook. CRC Press, LLC, Ch. 3, pp. 93–131.

Gilbert, J., Gilbert, L., 2005. Elements of Modern Algebra, 6th Edition. Thomson Brook/Cole, Pacific Grove, California, USA.

Li, S., Chen, G., Zheng, X., 2004. Chaos-based encryption for digital images and videos. In: Furht, B., Kirovski, D. (Eds.), Multimedia Security Handbook. CRC Press, LLC, Ch. 4, pp. 133–167.

Microsoft Corporation, 2005. Microsoft enhanced cryptographic provider – FIPS 140-1 documentation: Security policy. Available online at `http://csrc.nist.gov/cryptval/140-1/140sp/140sp238.pdf`.

Mihaljević, M. J., 2003. On vulnerabilities and improvements of fast encryption algorithm for multimedia FEA-M. IEEE Trans. Consumer Electronics 49 (4), 1199–1207.

Mihaljević, M. J., Kohno, R., 2002. Cryptanalysis of fast encryption algorithm for multimedia FEA-M. IEEE Trans. Communications Letters 6 (9), 382–384.

Schneier, B., 2000. Secrets and Lies: Digital Security in a Networked World. John Wiley & Sons, Inc., New York.

Uhl, A., Pommer, A., 2005. Image and Video Encryption: From Digital Rights Management to Secured Personal Communication. Vol. 15 of Advances in Information Security. Springer Science + Business Media, Inc., Boston, USA.

Wikipedia, 2005. General linear group. Available online at `http://en.wikipedia.org/wiki/General_linear_group`.

Wu, H., Bao, F., Deng, R. H., 2003. An efficient known plaintext attack on FEA-M. In: Qing, S., Gollmann, D., Zhou, J. (Eds.), Information and

Communications Security: 5th International Conference, ICICS 2003, Huhehaote, China, October 10-13, 2003, Proceedings. Vol. 2836 of Lecture Notes in Computer Science. Springer-Verlag, Berlin Heidelberg, pp. 84–87.

Yi, X., Tan, C. H., Siew, C. K., Syed, M. R., 2001. Fast encryption for multimedia. IEEE Trans. Consumer Electronics 47 (1), 101–107.

Yi, X., Tan, C. H., Siew, C. K., Syed, M. R., 2002. ID-based key agreement for multimedia encryption. IEEE Trans. Consumer Electronics 48 (2), 298–302.

Youssef, A. M., Tavares, S. E., 2003. Comments on the security of fast encryption algorithm for multimedia (FEA-M). IEEE Trans. Consumer Electronics 49 (1), 168–170.