Further Improvements of Fast Encryption Algorithm for Multimedia

Depeng Li and Srinivas Sampalli
(Corresponding author: Srinivas Sampalli)
Faculty of Computer Science, Dalhousie University
6050 University Avenue, Halifax, Nova Scotia B3H 1W5, Canada (Email: srini@cs.dal.ca)

(Received May 19, 2006; revised and accepted July 31, 2006 & Nov. 8, 2006)

Abstract

In order to efficiently encrypt multimedia streams delivered in real-time environments, a Fast Encryption Algorithm for Multimedia (FEA-M) was proposed [8, 9]. Cryptanalyses of this technique [1, 3, 4, 5] have identified its weaknesses and an improved variant has been suggested in [5]. In this paper, we identify further weaknesses in the original FEA-M and also in the improved variant. Our solution provides message integrity, guarantees zero packet loss and protects against specific known plaintext attacks.

Keywords: Cryptanalysis, multimedia encryption, reliable transportation

1 Introduction

Securing real-time multimedia data is a challenging task since the size of data is usually very large and the data needs to be processed in a short time interval. Standard cryptographic algorithms will usually result in a large overhead, rendering them inefficient.

Yi, Tan, Siew, and Syed [8, 9] have proposed a novel algorithm called FEA-M (Fast Encryption Algorithm for Multimedia) which only requires 1.5 XOR operations to encrypt one bit of plaintext. This is significantly less compared to other encryptions such as Rijndael, Crypton, Twofish, RC6, MARS, Cast256 and Serpent [8]. FEA-M is based on the Boolean matrix theory which involves matrix addition and multiplication over the finite field GF (2) = {0, 1}. FEA-M's security is based on the complexity to solve non-linear equation groups and variable linear equation groups. To protect the key material against both passive and active attacks, an ID-based key agreement is utilized to secure FEA-M's key exchanges [9].

Mihaljevic and Kohno [3, 4] analyze FEA-M's security and find it is not secure enough when the first plaintext blocks are all 0s. Furthermore, Mihaljevic indicates that FEA-M cannot work if one ciphertext package is lost during transmission. He proposes an improvement to counter this vulnerability [5].

Li and Lo [1] indicate that there are still some security problems for the improved variant [5] for FEA-M. In case that the involved random process is tampered with (e.g. the pseudo-random process is uniquely controlled by an external illegal party), the secret key of the cryptosystem could be compromised by implementation-dependent differential attacks. Furthermore, they propose an efficient differential attack which can reveal the secret key benefiting from only two pairs of chosen plaintext blocks.

In this paper, we identify further weaknesses in the original FEA-M and also in the improved variant, specifically, 1) vulnerability of Mihaljevic's proposal [5] to block replay attacks and 2) security degradation due to the use of fixed pad. We also propose corresponding improvements to overcome these defects. This paper is organized as follows. Section 2 provides a brief description of FEA-M. Section 3 discusses Mihaljevic and Kohno's analyses and suggested improvements to the algorithm. Section 4 identifies further weaknesses in the original FEA-M and in its improved variant. Section 5 describes our proposal to overcome these weaknesses. Section 6 draws concluding remarks.

2 Description of FEA-M

FEA-M uses an ID-based Diffie-Hellman key agreement protocol to generate a common secret key, k, an integer, between the sender and the receiver [9]. Based on the value of k, FEA-M generates a common key matrix K and a common initial matrix V_0 which are binary matrices of order n. We refer the reader to [8, 9] for the details of the algorithm to generate K and V_0.

The plaintext message is divided into a series of blocks, $P_1, P_2 \cdots P_r$, with the same length, n^2, where n is 64 and r is an integer [8]. If the length of the last block is less than n^2, it is padded with 0s to make its length n^2. Each plaintext matrix, $P_i (1 \leq i \leq r)$, is encrypted into a ciphertext matrix C_i and each corresponding ciphertext matrix C_i is decrypted into a plaintext matrix P_i according to for-

mulas below:

$$\begin{aligned} C_i &= K \cdot (P_i + C_{i-1}) \cdot K^i + P_{i-1} \\ P_i &= K^{-1} \cdot (C_i + P_{i-1}) \cdot K^{-i} + C_{i-1} \\ P_0 &= C_0 = V_0. \end{aligned}$$

3 Previous Analyses and Improvements of FEA-M

The vulnerability of FEA-M has been identified and improvements have been proposed. Mihaljevic and Kohno point out in [3, 4] that the real uncertainty about the secret key of FEA-M is undesirably smaller than expected since the effective secret key size, under realistic known and chosen plaintext attacks, is much smaller than the nominal one. It occurs while the first set of blocks is all 0s. They conclude that when the key is a 64*64 matrix, the nominal secret key size is 4096 bits but the effective secret key size is only 134 bits.

Furthermore, Mihaljevic [5] indicates that, if one ciphertext block is lost during transmission, subsequent ciphertext blocks cannot be decrypted since they depend on former ciphertext blocks. To overcome this weakness, he proposes a new encryption algorithm, which is described by the formulas below:

$$C_i = K \cdot (P_i + K \cdot V \cdot K^i) \cdot K^{i+n} + K \cdot V \cdot K^i \quad (1)$$

$$P_i = K^{-1} \cdot (C_i + K \cdot V \cdot K^i) \cdot K^{-(i+n)} + K \cdot V \cdot K^i \quad (2)$$

If C_i is a lost block, no further impact on subsequent blocks occurs.

4 Further Weaknesses

In this section, we identify further weaknesses in the original FEA-M and in its improved variant.

4.1 Weakness of the Improvement Proposed in [5]

FEA-M provides the connection between the neighbor plaintext blocks. If the attackers replay the earlier packets, the receiver can notice the faked message. Therefore, it shows a feature to persist the packet replay attacks. However, we observe that, although it tolerates the packet loss problems, Mihaljevic's improvement [5], which is described by Formulas (1) and (2), is vulnerable to the packet replay attack. For example, the attacker can obtain earlier i^{th} ciphertext block, C'_i. Then, s/he captures the current i^{th} ciphertext block, C_i and replaces C_i with C'_i. In case that K and V_0 are not changed, the receiver cannot be aware that the cipher text is the earlier plaintext. According to Formulas (1), (2) and (3), what the receiver gets is P'_i rather than P_i if C'_i replaces C_i.

$$P'_i = K^{-1} \cdot (C'_i + K \cdot V \cdot K^i) \cdot K^{-(i+n)} + K \cdot V \cdot K^i \quad (3)$$

The reason this kind of attack works is because the improvement in [5] treats each block of the plaintext independently.

4.2 Security Degradation due to the Use of Fixed Pad

In the original FEA-M, 0s are appended in the last plaintext block so that its length will be exactly n^2. The obvious disadvantage of this method is that it introduces insecure information redundancy. If the plaintext in the last block is all 0s, after appending the all 0 pad, it will result in an all 0 last bock. As analyzed in [4], the secret key length of FEA-M should be significantly smaller than expected. If the nominal one is 4096 bits, the real secret key size, under realistic known and chosen plaintext attacks, is just 134 bits. Furthermore, the multiplication by bit 0 will result in all zeros. So the intermediate value will leak part of the key matrix K and part of the initial matrix V_0 to the attacker.

5 Impovements to FEA-M

In this section, we propose the following techniques to overcome the weaknesses mentioned above.

5.1 Randomly Generated Bit Streams to Replace the all 0s Pad

A randomly generated pad can overcome the insecurity introduced by all-zero padding. In our proposal, the Blum-Blum-Shub pseudorandom bit generator [2], which is independent of the external party, is utilized to produce the random bit stream. For detail, please refer to [2].

Before the last block of plaintext is encrypted, a randomly generated bit stream, D_0, will be appended. The sum of the length of the last plaintext block and that of the padding will be exactly $n^2 - 8$. The last 8 bits are used to record the number of padding bytes. Then, based on the last two bytes, the receiver knows which part of the last decrypted block is the pad. Subsequent pads can be updated following the formula below:

$$D_{i+1} = Hash(D_i),$$

where i is an integer and $Hash$ is a one way hash function such as SHA-256 [2].

5.2 Compress Plaintext to Avoid Mihaljevic's Assumption [4]

FEA-M's security degradation due to plaintext blocks being all 0s [3, 4] can be solved with the compression algorithm. In general, before encryption by FEA-M, the multimedia plaintext is compressed. Figure 1 describes how the multimedia data is compressed, encrypted and transmitted across the insecure channel from party A to party B.

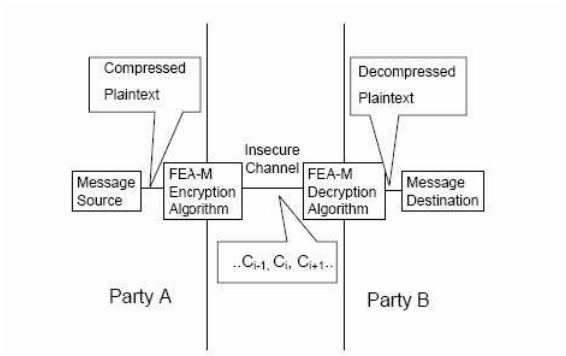

Figure 1: Multimedia communication model

The well-known Routine Length Encode (RLE) technique [2] which reduces the redundant messages is deployed in a number of popular compression algorithms. In this paper, we utilize the RLE technique to overcome the weakness of blocks being all 0s. The details of RLE are listed below:

Algorithm 1: Multimedia Data Compression - RLE

INPUT: Plaintext byte stream B_1, $B_2 \cdots B_n$,
OUTPUT: Compressed Byte Stream $(*N_i, B_i)$ or (B_i) where $*$ indicates how many times the subsequent data byte repeats itself where $N_i > 1$

```
1:  B = 1st Byte; count = 0;
2:  FOR (not end of the byte stream)
3:      While (current byte vaule = B)
4:          {count++; read next byte}
5:      If (count > 1)
6:          Put (*count; B);
7:      Else
8:          Put (B);
9:      B = current byte value; count = 0;
10  END FOR
```

The computational complexity of this algorithm is $O(n)$.

Then, here is an example. Suppose that, in the first set of plaintext blocks, there are 0s in a row with its length n^2. After compression, we no longer have an all-zero block:

$$\overbrace{\ldots\ldots\underbrace{0,0,\ldots,0}_{n^2}}^{Before} \Longrightarrow^{Compress} \overbrace{\ldots\ldots n^2,0}^{After}$$

5.3 Reliable Transportation to Handle Packet Loss

FEA-M is vulnerable to packet losses. Furthermore, the improved variant in [5] is vulnerable to the block replay attack. In this sub-section, we propose two methods to overcome these weaknesses respectively.

1) Reliable FEA-M (rFEA-M):
Packet loss is not a problem for applications using reliable transport protocols (e.g. TCP). However, in case that applications deploying FEA-M algorithm do not utilize such protocols, techniques that are robust against packet loss have to be used. For instance, multimedia applications such as medical imaging systems which cannot tolerate source data packet losses demand this requirement. Therefore, we propose a robust FEA-M (rFEA-M) method to overcome the flaw of FEA-M. We use FEA-M to encrypt/decrypt data packets. To make FEA-M robust against message losses, Rabin's Information Dispersal Algorithm (IDA) [6, 7] is utilized which encodes every cipher text block introducing some amount of redundancy. In the following, Algorithm 2 is proposed to implement rFEA-M:

Algorithm 2: rFEA-M

The Sender Party A:

INPUT: Plaintext blocks P_1, $P_2 \cdots P_r$,
OUTPUT: Ciphertext blocks C_1, $C_2 \cdots C_r$, which are encoded by IDA-Encode.

```
1:  Notation. ||: concatentation;
              K: key matrix;
              Pi: a block of plaintext
2:  for 1 ≤ i ≤ r; i++;
3:      Ci = FEA-M-encrypt (Pi);
4:      A → B: IDA-Encode (Ci).
5:  end for
```

The Receiver Party B:

INPUT: Ciphertext blocks, C_1, $C_2, \cdots, C_r$ which are encoded by IDA-Encode;
OUTPUT: Plaintext blocks P_1, $P_2 \cdots P_r$,;

```
1:  for 1 ≤ i ≤ r; i++;
2:      C'i = IDA-Decode (Ci);
3:      Pi = FEA-M-decrypt(C'i);
4:  end for
```

Algorithm 2 provides no-packet-lost service and makes FEA-M to be implemented for network settings with packet loss. However, it requires more computational cost due to the implementation of IDA to process every cipher text block. The computation complexity of IDA is $O(n^2)$ where n is the data length of plaintext.

2) Correction for improved variant [5] for FEA-M:
To provide data source authentication against packet replay attacks for improved variant [5], we utilize the secret-key-based message authentication code (e.g. MD5-MAC [2]) to process every cipher text block. Then, we use IDA to encode/decode all MD5-MAC result to guarantee that they are all received by the receiver party. This method is suitable for applications such as Internet TV or Internet Radio which can tolerate source data packet losses.

In details, improved variant [5] proposed by Mihaljevic is used to encrypt/decrypt the data packets. Furthermore, to resist against the block replay attack, we use MD5-MAC to process every source data block. Then, the result of MD5-MAC is encoded

with the IDA to guarantee MD5-MAC information is not lost during packet transport. The receiver party uses IDA algorithm to reconstruct all MD5-MAC's results it received. Finally, the receiver can verify the integrity of every cipher block. In the following, Algorithm 3 is proposed to implement this method:

Algorithm 3: Correction for improved variant [5]

The Sender Party A:

INPUT: Plaintext blocks: P_1,$P_2 \cdots P_r$, where $r \geq 1$ and it is defined by applications.

OUTPUT: Ciphertext blocks, C_1, $C_2, \cdots C_r$ and H which is encoded by IDA which are encoded by IDA-Encode.

1: Notation. $||$: concatentation; K: key matrix; P_i: a block of plaintext
2: H: concatentaed result of MD5-MAC; H is *empty*;
3: ***for*** $1 \leq i \leq r$; i++;
4: $H_i = MD5 - MAC(P_i, K_{upper128bits})$;
5: $H = H_i||H$;
6: C_i = Improved variant in [5]-encrypt (P_i);
7: $A \rightarrow B : C_i$;
8: ***end for***
9: $A \rightarrow B$: IDA-Encode (H);

The Receiver Party B:

INPUT: Ciphertext blocks, C_1, $C_2, \cdots, C_r$, and H which are encoded by IDA;

OUTPUT: Plaintext blocks P_1, $P_2 \cdots P_r$,;

1: ***for*** $1 \leq i \leq r$; i++;
2: P_i = Improved variant in [5]-decrypt(C_i');
3: ***end for***
4: H' = IDA-Decode (H);
5: ***for*** $1 \leq i \leq r$; i++;
6: ***if*** (P_i is received) and ($MD5 - MAC(P_i, K_{upper128bits}) \neq H'[I]$);
8: ***endif***
9: ***endfor***

Algorithm 3 provides data source authentication service to persist against packet replay attacks for improved variant [5]. However, it requires more computational cost due to the implementation of IDA and MD5-MAC. The computation complexity of them is $O(n^2)$ where n is the data length of MD5-MAC results and n is smaller than the plaintext size.

3) IDA:
We propose Algorithm 4 which describes the implementation of the IDA.

Algorithm 4 focuses on the implementation of the IDA which presents reliable transmission for data packets by introducing some amount of information redundancy. IDA splits the source data, for example, C_j, into n pieces, which are, then, encoded by the IDA algorithm. At the receiver end, the IDA can reconstruct C_j after receiving any m pieces where $m < n$. However, guaranteeing zero packet loss comes at the cost of increased communication overhead. For example, for r blocks, assume every block is 4096 bits. So, n is 64, m can be 50. For Algorithm 2, $4096 * r * n/m$ bits' data are sent over the network and at least $4096 * r$ bits' data are received. For Algorithm 3, in addition to source data, $64 * r * n/m$ bits' hash are sent over the network and at least $64 * r$ bits' hash are received.

According to Algorithm 4, we find the computation complexity of IDA is $O(n^2)$.

6 Conclusion

After examining the FEA-M algorithm and its improvement, we have identified some of its weaknesses, namely, 1) vulnerability of Mihaljevic's proposal [5] to block replay attacks and 2) security degradation of the original FEA-M due to the fixed pad. Our solution presents message integrity for Mihaljevic's proposal and packet loss resistance for FEA-M. Furthermore, it is robust against specific known plaintext attacks.

References

[1] S. Li and K. T. Lo, "Security problems with improper implementations of improved FEA-M," *Journal of Systems and Software*, vol. 80, no. 5, pp. 791-794, May 2007.

[2] A. Menezes, P. v. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[3] M. J. Mihaljevic and R. Kohno, "Cryptanalysis of fast encryption algorithm for multimedia FEA-M", *IEEE Communications Letters*, vol. 6, no. 9, pp. 382-384, Sep. 2002.

[4] M. J. Mihaljevic and R. Kohno, "On wireless communications privacy and security evaluation of encryption techniques," in *Proceedings of the IEEE wireless Communication and Networking Conference (WCNC'02)*, pp. 865-868, Mar. 2002.

[5] M. J. Mihaljevic, "On vulnerabilities and improvements of fast encryption algorithm for multimedia FEA-M," *IEEE Transactions on Consumer Electronics*, vol. 49, pp. 1199-1207, Nov. 2003.

[6] J. M. Park, E. K. P. Park, and H. J. Siegel, "Efficient multicast stream authentication using erasure codes," *ACM Transactions on Information and System Security*, vol. 6, no. 2, pp. 258-285, May 2003.

[7] M. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *ACM Transactions on Information and System Security*, vol. 36, no. 2, pp. 335-348, 1989.

Algorithm 4: IDA

The Sender Party A: IDA-Encode

INPUT: a block of data C_j

OUTPUT: encoded vectors T_1, $T_2 \cdots T_n$

(1) Split C_j into N/m pieces where $N = n/8$:

$C_j = (c_1, \cdots, c_m), (c_{m+1}, \cdots, c_{2m}), \cdots, (c_{N-m-1}, \cdots, c_N)$where,$c_i$:byte

$R_i = (c_{(i-1)m+1}, \cdots, c_{im})$,where, $i < N/m$

(2) Process C_j: following the specification of IDA [7], choose n vectors, $A_i = (a_{i1}, \cdots, a_{im})$, $1 \leq i \leq n$,let every subset of m different vectors are linearly independent. Then, process C_j:

$$T_i = A_i \cdot (R_1 R_2 \cdots R_{N/m}) = (a_{i1} \cdots a_{im}) \cdot \begin{pmatrix} c_1, c_{m+1}, \cdots, c_{N-m+1} \\ \cdot \\ \cdot \\ \cdot \\ c_m, c_{2m}, \cdots, c_N \end{pmatrix} \text{ where } 1 \leq i \leq n \quad (4)$$

(3) Send T_1, $T_2 \cdots T_n$ to the receiver.

The Receiver Party B: IDA-Decode

INPUT: encoded vectors T_1, $T_2 \cdots T_m$

OUTPUT: a block of data C_j;

(1) Assume that the receiver receives T_1, $T_2 \cdots T_m$

$T_1 = A_1 \cdot R_1, A_1 R_2 \cdots A_1 \cdot R_{N/m}$

$T_2 = A_2 \cdot R_1, A_2 R_2 \cdots A_2 \cdot R_{N/m}$

$\cdots$;

$T_m = A_m \cdot R_1, A_m R_2 \cdots A_m \cdot R_{N/m}$

(2) Prepare for the calculation of R_1

Based on $T_1 \cdots T_m$, and Formula (4), we can get:

$$A'g \begin{pmatrix} c_1 \\ \cdot \\ \cdot \\ \cdot \\ c_m \end{pmatrix} = \begin{pmatrix} A_1 g R_1 \\ \cdot \\ \cdot \\ \cdot \\ A_m g R_1 \end{pmatrix} \text{ where\_} A' = \begin{pmatrix} a_{11} \cdots a_{1m} \\ \cdots \\ \cdots \\ \cdots \\ a_{m1} \cdots a_{mm} \end{pmatrix}$$

(3) Since A' is invertible, we can calculate R_1:

$$R_1 = \begin{pmatrix} c_1 \\ \cdot \\ \cdot \\ c_m \end{pmatrix} = \begin{pmatrix} a_{11} \cdots a_{1m} \\ \cdots \\ \cdots \\ a_{m1} \cdots a_{mm} \end{pmatrix}^{-1} \begin{pmatrix} A_1 g R_1 \\ \cdot \\ \cdot \\ A_m g R_1 \end{pmatrix}$$

(4) Repeat Step 3, we can calculate $R_2 \cdots R_{N/m}$.

(5) Reconstruct C_j:

$C_j = R_1 || R_2 K || R_{N/m}$ where $||$ denotes concatenation.

[8] X. Yi, C. H. Tan, C. K. Siew, and M. R. Syed, "Fast encryption for multimedia," *IEEE Transactions on Consumer Electronics*, vol. 47, pp. 101-107, Feb. 2001.

[9] X. Yi, C. H. Tan, C. K. Siew, and M. R. Syed, "ID-based key agreement for multimedia encryption," *IEEE Transactions on Consumer Electronics*, vol. 48, pp. 298-303, May 2002.

Depeng Li received the Bachelor and Master degrees in Computer Science from Shandong University, Jinan, P. R. China. He is currently working toward the Ph. D. degree in Computer Science at Dalhousie University, Halifax, Canada. His research interests are in the areas of network security, peer-to-peer group communication and performance evaluation.

Srinivas Sampalli is a Professor and 3M Teaching Fellow in the Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada. His research interests are in the areas of security and quality of service in wireless and wireline networks. Specifically, he has been involved in research projects on protocol vulnerabilities, security best practices, risk mitigation and analysis, and the design of secure networks. He is currently the principal investigator for the Wireless Security project sponsored by Industry Canada. Dr. Sampalli has received many teaching awards including the 3M Teaching Fellowship, Canada's prestigious national teaching award.