10-2003

An Efficient Known Plaintext Attack on FEA-M

Hongjun WU
Institute for Infocomm Research, Singapore

Feng BAO
Institute for Infocomm Research, Singapore

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Citation

WU, Hongjun; BAO, Feng; and DENG, Robert H.. An Efficient Known Plaintext Attack on FEA-M. (2003). *Information and Communications Security: 5th International Conference, ICICS 2003, Huhehaote, China, October 10-13: Proceedings*. 2836, 84-87.
Available at: https://ink.library.smu.edu.sg/sis_research/1078

An Efficient Known Plaintext Attack on FEA-M

Hongjun Wu, Feng Bao, and Robert H. Deng

Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
{hongjun,baofeng,deng}@i2r.a-star.edu.sg

Abstract. Yi et al. have proposed a cipher called the fast encryption algorithm for multimedia (FEA-M). Recently Mihaljević and Kohno pointed out that FEA-M is insecure. However, their attacks are not efficient: their chosen plaintext attack and known plaintext attack require 2^{37}-bit chosen plaintext and 2^{60}-bit known plaintext, respectively. In this paper we give an efficient known plaintext attack against FEA-M. Our attack requires only 2^{28}-bit known plaintext and about 2^{33} XOR operations.

1 Introduction

Yi et al. have proposed a fast encryption algorithm for multimedia (FEA-M) [4]. FEA-M is a cipher based on the Boolean matrix operations. Mihaljević and Kohno broke FEA-M with two attacks [2]. Their chosen plaintext attack requires about 2^{25} chosen messages with the first 4096 bits being 0. Their known plaintext attack requires about 2^{60}-bit known plaintext. Both attacks are not efficient due to the large amount of chosen/known plaintext required.

In this paper, we give a very efficient known plaintext attack against FEA-M. Under our attack, the key is recovered with 2^{28}-bit known plaintext. And only about 2^{33} XOR operations are needed in the attack. Our attack shows that FEA-M is extremely insecure.

This paper is organized as follows. Section 2 introduces the cipher FEA-M. Our efficient known plaintext attack is given in Section 3. Section 4 concludes this paper.

2 Description of FEA-M

The secret key of FEA-M is a 64×64 invertible binary matrix denoted as $\bar{K}$. For each message being encrypted, a session key pair (K,V) is generated, where K and V are 64×64 binary matrices and K is invertible. This pair is encrypted with the use of $\bar{K}$ as

$$K' = \bar{K} \cdot K^{-1} \cdot \bar{K} \tag{1}$$

$$V' = \bar{K} \cdot V \cdot \bar{K} \tag{2}$$

where '$\cdot$' denotes the matrix multiplication over $GF(2)$ and K^{-1} denotes the inverse of K over $GF(2)$. To encrypt a message, the message is divided into 64×64 binary matrices P_1, P_2, $\cdots$, P_r, $\cdots$. Each plaintext block P_i is encrypted into ciphertext C_i as

$$C_1 = K \cdot (P_1 + V) \cdot K + V \tag{3}$$

$$C_i = K \cdot (P_i + C_{i-1}) \cdot K^i + P_{i-1} \qquad \text{for } i \geq 2 \tag{4}$$

where the '+' denotes the matrix addition over $GF(2)$. The ciphertext together with (K',V') are sent to the receiver. The message could be recovered with $\bar{K}$.

3 The Efficient Known Plaintext Attack

In this section, we will introduce our efficient known plaintext attack against FEA-M. The attack is applied to recover the session key pair (K, V) in Subsection 3.1. The master secret key $\bar{K}$ is recovered in Subsection 3.2.

3.1 Recovering the Session Key Pair (K, V)

We assume that $(P_{i-1} + C_i)$ is invertible for $i \geq 2$. The impact of the non-invertible $(P_{i-1} + C_i)$ on the attack is discussed at the end of this subsection. From (4), we obtain that

$$I = (P_{i-1} + C_i)^{-1} \cdot K \cdot (P_i + C_{i-1}) \cdot K^i \qquad \text{for } i \geq 2 \tag{5}$$

where I is the identity matrix. Let $A_i = P_{i-1} + C_i$ and $B_i = P_i + C_{i-1}$, we rewrite (5) as

$$I = A_i^{-1} \cdot K \cdot B_i \cdot K^i \qquad \text{for } i \geq 2 \tag{6}$$

Combine any two consecutive equations in (6), we obtain

$$A_{i+1} \cdot A_i^{-1} \cdot K \cdot B_i = K \cdot B_{i+1} \cdot K \qquad \text{for } i \geq 2 \tag{7}$$

Solve the following linear equations for the binary unknown variables x_i ($2 \leq i \leq 4098$),

$$\sum_{i=2}^{4098} x_i \cdot B_{i+1} = 0 \tag{8}$$

To solve (8), we write (8) as $M \cdot X = 0$, where M is a 4096×4097 binary matrix with each element $M^{(i,j)} = B_{j+2}^{(\lfloor \frac{i}{64} \rfloor, i \bmod 64)}$ for $1 \leq i \leq 4096$ and $1 \leq j \leq 4097$ (the floor function $\lfloor \frac{i}{64} \rfloor$ denotes the integer part of $\frac{i}{64}$), and X is a binary vector with 4097 elements with each element $X_i = x_i$. A non-zero vector X satisfying $M \cdot X = 0$ always exists since the rank of M is at most 4096, which is less than the number of variables.

From a non-zero solution X, we define a set S as

$$S = \{i | x_i = 1\}$$

From (7) and the definition of S, we obtain the following relation

$$\sum_{i \in S} A_{i+1} \cdot A_i^{-1} \cdot K \cdot B_i = 0 \tag{9}$$

(9) can be written as $T \cdot Y = 0$, where T is a 4096×4096 binary matrix, and Y is a binary vector with 4096 elements, each element $Y_i = K^{(\lfloor \frac{i}{64} \rfloor, i \bmod 64)}$. It is known that the rank of a randomly generated $m \times n$ binary matrix is r $(1 \le r \le \min(m,n))$ with probabiltiy

$$P_r = 2^{r(m+n-r)-nm} \prod_{i=0}^{r-1} \frac{(1-2^{i-m})(1-2^{i-n})}{1-2^{i-r}}$$

For an $n \times n$ binary matrix ($n \ge 64$), the rank is n, $n-1$, $n-2$, $n-3$ and $n-4$ with probability 0.2888, 0.5776, 0.1284, 0.0052 and 4.7×10^{-5}, respectively. The probability that the rank being less than $n-4$ is negligible. Since a non-zero Y (the session key K) is a solution to (9), the rank of T is less than 4096. The rank of T is less than 4092 with negligible probability, so there are only a few non-zero solutions to (9). We can filter the wrong K by substituting those solutions into any equation in (4). Once we know the value of K, V can be obtained by solving (3).

Note that (7) holds only if A_i and A_{i+1} are invertible. A randomly generated 64×64 binary matrix is invertible with probability 0.2888. The probability that both A_i and A_{i+1} are invertible is about 0.083. We thus need about 2^{16} blocks of known plaintext in the attack, that is equivalent to 2^{28}-bit known plaintext.

3.2 Recovering the Secret Key $\bar{K}$

We proceed to recover the master secret key $\bar{K}$ from the session key pair (K, V). Let $Z = \bar{K}^{-1}$. From (1) and (2), we obtain

$$Z \cdot K' = K^{-1} \cdot \bar{K}, \qquad Z \cdot V' = V \cdot \bar{K} \tag{10}$$

V' is invertible with probability 0.2888. If V' is invertible, (10) can be simplified further by eliminating Z. Otherwise we solve (10) directly. The pair (K', V') is known to the attacker since it is sent together with the ciphertext. $(Z, \bar{K})$ can be retrieved by solving at most 8192 linear equations in (10). In case that too many solutions exist, one more pair (K, V) is needed to refine the results.

3.3 Complexity of the Attack

The expensive operations in the attack are related to 1) computing the inverses of 2^{16} 64×64 binary matrices to find out 4097 invertible (A_i, A_{i+1}) pairs, 2) computing the matrix T, and 3) solving four groups of binary linear equations (8), (9), (3) and (10). We use the standard Gaussian elimination in the attack and assume that the attack is implemented on the 32-bit microprocessor. Computing the inverse of a 64×64 binary matrix requires about 2^{13} XOR operations. We need about 2^{31} XOR operations to form the matrix T. Solving each of (8), (9),

(3) requires $2^{29.4}$ XOR operations. Solving (10) requires $2^{32.4}$ XOR operations. The amount of XOR operations required in the attack is about $2^{16} \times 2^{13} + 2^{31} + 3 \times 2^{29.4} + 2^{32.4} \approx 2^{33.28}$. In the attack we use the standard Gaussian elimination instead of Strasen's algorithm [3] and Coppersmith and Winograd's algorithm [1]. The reason is that the dimension of the matrices being involved in the attack is small (at most 8192) and the Gaussian elimination performs well already.

The complete attack requires 2^{28} bits known plaintext and about 2^{33} XOR operations. Our attack is more efficient than that in [2]. The reason is that we developed efficient technique to eliminate the quadratic terms in (1), (2) and (7), while the standard linearization technique (replacing each quadratic term with a new variable) is used in [2].

4 Conclusions

In this paper, we proposed a known plaintext attack against FEA-M. It is much more efficient than the attacks reported early. Our attack shows that FEA-M is extremely weak and should not be used.

Acknowledgements. We would like to thank the anonymous reviewers of ICICS for the helpful comments.

References

1. D. Coppersmith, and S. Winograd, "On the Asymptotic Complexity of Matrix Multiplication", *SIAM Journal on Computing*, Vol. 11 (1982), pp. 472–492.
2. M.J. Mihaljević, and R. Kohno, "Cryptanalysis of Fast Encryption Algorithm for Multimedia FEA-M", *IEEE Communications Letters*, Vol. 6, No. 9, pp. 382–385, September 2002.
3. V. Strassen, "Gaussian Elimination is not Optimal", *Numerical Mathematics*, Vol. 13 (1969), pp. 354–356.
4. X. Yi, C.H. Tan, C.K. Siew, and M.R. Syed, "Fast Encryption for Multimedia", *IEEE Transactions on Consumer Electronics*, Vol. 47, No. 1, pp. 101–107, February 2001.