A Related-Key Attack on TREYFER

Aleksandar Kircanski and Amr M. Youssef
Computer Security Laboratory
Concordia Institute for Information Systems Engineering
Concordia University,
Montreal, Quebec, Canada, H3G 2W1

Abstract

TREYFER is a lightweight cipher designed for resource constrained environments. In this paper, we present a related-key attack that directly recovers the secret key of TREYFER using about 2^{11} chosen plaintext encryptions. Our attack is based on a set of deterministic algebraic relationships between TREYFER ciphertexts corresponding to related plaintexts encrypted under circularly byte shifted versions of the same key. The attack complexity is independent of the number of rounds of the cipher.

1 Introduction

Despite its current implementation advances, the Advanced Encryption Standard (AES) [1] may not always be the optimal choice for applications with tight resource constraints such as radio frequency identification (RFID) tags and tiny sensor networks. In fact, with the widespread applications of these resource-constrained devices, the analysis and design of lightweight encryption algorithms have started to gain a new momentum (e.g [2], [3].)

TREYFER [4] is a 64 bit block cipher (and also a MAC) proposed by Gideon Yuval, from Microsoft, at FSE'97. According to Gideon Yuval, TREYFER is targeting an environment for which even TEA [5] and SAFER [6] are "*gross overdesign* [4]". The simple and compact design of TREYFER makes it an attractive choice for resource constrained environments such as smart cards, RFIDs, and sensor networks. For example, TREYFER requires only 29 bytes of executable code on the 8051 micro-controller.

The best known attack against TREYFER, presented by Alex Biryukov and David Wagner [7], is a slide attack that requires 2^{32} known plaintexts, 2^{44} time for analysis and 2^{32} memory.

In this paper, we derive a set of deterministic algebraic relationships between the ciphertexts corresponding to related plaintexts encrypted with TREYFER under circularly byte shifted versions of the same key. Based on these relationships, we present a chosen related-key attack [8]-[14] that directly recovers the secret key of TREYFER using about 2^{11} chosen plaintext encryption operations. The attack complexity is independent of the number of rounds of the cipher.

The rest of the paper is organized as follows. In section 2 we briefly review the description of TREYFER. The algebraic relationship between ciphertexts corresponding to related plaintexts encrypted under circularly (byte-wise) shifted versions of the secret key is derived in section 3. Using this relationship, our attack is described in section 4.

2 Description of TREYFER

TREYFER can be seen as an iterative block cipher with the round function shown in Figure 1 where $<<<$ denotes a circular left shift by 1 bit, $+$ denotes addition mod 256 and SKi, $i = 0 \ldots 7$, denotes the key addition and s-box lookup operations

978-0-7695-3329-2/08 $25.00 © 2008 IEEE
DOI 10.1109/SECURWARE.2008.51

$S[(x + K_i) \bmod 256]$ function. All operations are byte-oriented, and there is a single 8×8-bit s-box.

In [4], the s-box is left undefined; it is suggested that the implementation can simply use whatever data is available in memory. In each round, each byte has added to it the s-box value of the sum of a key byte and the previous data byte, then it is rotated left one bit. The design attempts to compensate for the simplicity of this round transformation by using a large number of rounds: 32.

The pseudo code implementation of TREYFER is as follows:

$for(r = 0; r < NumRounds; r++)\{$
$text[8] = text[0];$
$for(i = 0; i < 8; i++)$
$text[i+1] =$
$(text[i+1] + S[(key[i] + text[i])\%256]) <<< 1;$
$text[0] = text[8];$
$\}$

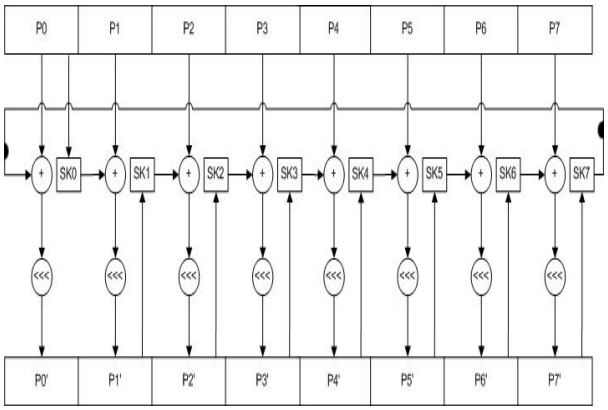

Figure 1: Round function of TREYFER, where $SKi(x) = sbox((x + K_i)\ mod\ 256)$

In order to obtain a more compact and faster cipher under the assumed hardware constraints, the designer of TREYFER opted not to have any complex key scheduling. In particular, TREYFER simply uses its user supplied key, K, byte by byte in exactly the same way at each round.

The following notation will be used throughout the paper.

- f denotes round function mapping of TREYFER (see Figure 1).
- $P = P_0P_1 \cdots P_7$ denotes the 8 byte plaintext input
- $K = K_0K_1 \cdots K_7$ denotes the 8 byte key
- $C = C_0C_1 \cdots C_7$ denotes the 8 byte ciphertext

3 Main Observations

In this section, we derive an algebraic relationship between the ciphertexts corresponding to related plaintexts encrypted under circularly (byte-wise) shifted versions of the secret key.

Lemma 1 *Let*

$$P_0^{'}P_1^{'} \cdots P_7^{'} = f(P_0 \cdots P_7, K_0 \cdots K_7)$$

and

$$P_0^{''}P_1^{''} \cdots P_7^{''} = f^2(P_0 \cdots P_7, K_0 \cdots K_7).$$

Then we have

$$f(P_1^{'}P_2 \cdots P_7P_0, K_1 \cdots K_7K_0) = P_1^{''}P_2^{'} \cdots P_7^{'}P_0^{'}$$

Proof: Let $p_i, i = 0 \ldots 7$ denote i-th byte of $f(P_1^{'}P_2 \cdots P_7P_0, K_1 \cdots K_7K_0)$. Then, by TREYFER definition, we have:

$$\begin{aligned}
p_1 &= (P_2 + S(P_1^{'}, K_1)) <<< 1 = P_2^{'} \\
p_2 &= (P_3 + S(P_2^{'}, K_2)) <<< 1 = P_3^{'} \\
&\vdots \qquad\qquad\qquad \vdots \\
p_7 &= (P_0 + S(P_7^{'}, K_7)) <<< 1 = P_0^{'} \\
p_0 &= (P_1^{'} + S(P_0^{'}, K_0)) <<< 1 = P_1^{''}
\end{aligned}$$

The lemma holds by noting that $f(P_1^{'}P_2 \cdots P_7P_0, K_1 \cdots K_7K_0) = p_0 \cdots p_7$.

This observation can easily be extended to hold for composition of multiple rounds. By $P \overset{K}{\longmapsto} C$ we will denote the TRYEFER encryption of the plaintext P with a key, K.

Lemma 2 *Let*

$$P_0 \cdots P_7 \stackrel{\mathrm{K}}{\longmapsto} C_0 \cdots C_7,$$

and

$$C_0^{'} \cdots C_7^{'} = f(C_0 \cdots C_7, K_0 \cdots K_7).$$

Then

$$P_1^{'} P_2 \cdots P_7 P_0 \stackrel{\mathrm{rot(K,1)}}{\longmapsto} C_1^{'} C_2 \cdots C_7 C_0$$

Proof: Follows from previous Lemma and the fact that TREYFER function is equal to f^n, $n = 32$.

That way, for each TREYFER pair (P, C), we can derive another "similar" plaintext-ciphertext pair, encrypted by key circularly shifted to the left by one byte. Furthermore, if previous Lemma is applied multiple times, we can get 7 such pairs, as given by the following Theorem.

Theorem 1 *Let* $\mathrm{rot}(K, i)$ *denote the left circular shift of* K *by* i *bytes, then, for any*

$$P_0 \cdots P_7 \stackrel{K}{\longmapsto} C_0 \cdots C_7 \tag{1}$$

we have

$$P_1^{'} P_2 P_3 P_4 P_5 P_6 P_7 P_0 \stackrel{\mathrm{rot}(K,1)}{\longmapsto} C_1^{'} C_2 C_3 C_4 C_5 C_6 C_7 C_0 \tag{2}$$

$$P_2^{'} P_3 P_4 P_5 P_6 P_7 P_0 P_1^{'} \stackrel{\mathrm{rot}(K,2)}{\longmapsto} C_2^{'} C_3 C_4 C_5 C_6 C_7 C_0 C_1^{'} \tag{3}$$

$$\vdots \qquad\qquad \vdots$$

$$P_7^{'} P_0 P_1^{'} P_2^{'} P_3^{'} P_4^{'} P_5^{'} P_6^{'} \stackrel{\mathrm{rot}(K,7)}{\longmapsto} C_7^{'} C_0 C_1^{'} C_2^{'} C_3^{'} C_4^{'} C_5^{'} C_6^{'} \tag{8}$$

It should be noted that the above presented property of TREYFER does not depend on any particular choice of the s-box.

4 The Attack

Related-key cryptanalysis assumes that the attacker learns the encryption of certain plaintexts not only under the original unknown key, K, but also under some related keys (e.g., $K' = g(K)$). In a chosen-related-key attack, the attacker specifies how the key is to be changed. It should be noted that the attacker knows or chooses the relationship between keys, i.e., $g(\cdot)$, but not the actual key values.

Based on the relations derived in the above section, we describe a chosen-related-key attack against TREYFER.

Given the plaintext-ciphertext pair (P, C), $P \stackrel{K}{\longmapsto} C$ where $K = K_0 \cdots K_7$, the proposed attack proceeds to recover K_0 as follows:

For($X = 0; X < 256; X++$) {

- Encrypt the plaintext $XP_2 \cdots P_7 P_0$ under the key $\mathrm{rot(K, 1)} = \mathrm{K_1} \cdots \mathrm{K_7 K_0}$
- For each ciphertext in the form $YC_2 \cdots C_7 C_0$, determine K_0 that satisfies

$$X = P_1^{'} = (P_1 + S[K_0 + P_0]) <<< 1,$$
$$Y = C_1^{'} = (C_1 + S[K_0 + C_0]) <<< 1.$$

}

The process above finds $P_1^{'}$ and $C_1^{'}$, second bytes of $f(P_0 \cdots P_7, K_0 \cdots K_7)$, and $f(C_0 \cdots C_7, K_0 \cdots K_7)$, respectively.

Theoretically it is possible that there may exist an $X \neq P_1^{'}$ such that $XP_2 \cdots P_7 P_0 \stackrel{\mathrm{rot(K)}}{\longmapsto} YC_2 \cdots C_7 C_0$, leading to false information on K_0. However, the probability that this will happen is practically negligible ($\approx 3 \times 10^{-15}$).

If the s-box is bijective, then only one of the equations above can be used to uniquely determine $K_0 = S^{-1}[(P_1' >>> 1) - P_1] - P_0$. In the case of non bijective s-boxes, the above two steps can be repeated until K_0 is uniquely determined.

K_1 to K_6 can be sequentially recovered by performing the above steps using related plaintexts and keys, according to relations (3)-(8). Similarly, the last byte of the key, K_7, can be recovered using relation (1) or simply by exhaustive search.

Thus, the attack requires about $8 \times 256 = 2^{11}$ chosen plaintext-ciphertext pairs, each 256 of them are encrypted under a key that is circularly bytes shifted version of the original secret key.

Remark 1 *The above attack can be thought of as a slide attack in which the sliding pairs are produced at the s-box level and not at the level of the whole round function.*

5 Conclusion

The simple, and compact design of TREYFER makes it a very attractive cipher for resource constrained devices. On the other hand, the key scheduling of the cipher seems to be oversimplified which has yielded to several weaknesses. While it might be argued that it is unlikely that an attacker can persuade a human operator to encrypt plaintexts under the 8 related keys, modern cryptography is implemented using complex protocols, and in some cases a related-key attack can be made feasible. In these scenarios, our attack can recover the secret keys in few milliseconds. It is interesting to investigate if there are other permutations of plaintexts and keys that derive similar ciphertexts. It is also interesting to provide further analysis of TREYFER which employs an enhanced key scheduling algorithm (e.g., by adding some round dependent constants).

References

[1] National Institute of Standards and Technology, *FIPS-197: Advanced Encryption Standard*, November 2001.

[2] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin and C. Vikkelsoe, *PRESENT: An Ultra-Lightweight Block Cipher*, CHES 2007, LNCS4727, pp.450- 466, Springer, 2007.

[3] M. Wang, *Differential Cryptanalysis of PRESENT*, In proc. of Africacrypt 2008, to appear, Also available at Cryptology ePrint Archive: Report 2007/408.

[4] G. Yuval, *Reinventing the Travois: Encryption/MAC in 30 ROM Bytes*, Proc. of the 4th International Workshop on Fast Software Encryption (FSE '97), pp. 205-209, Springer-Verlag, 1997.

[5] David J. Wheeler and Roger M. Needham, *TEA, a tiny encryption algorithm*, Proc. of the 2nd workshop on Fast Software Encryption (FSE'94), pp. 363-366, Leuven, Belgium, 1994, Springer-Verlag.

[6] James L. Massey, *SAFER K-64: A Byte-Oriented Block-Ciphering Algorithm*, Proc. of the first workshop on Fast Software Encryption (FSE'93), pp. 1-17, Springer-Verlag.

[7] Alex Biryukov and David Wagner, *Slide Attacks*, Proc, of the 6th International Workshop on Fast Software Encryption (FSE '99), pp. 245-259, Rome: Springer-Verlag.

[8] E. Biham, *New Types of Cryptanalytic Attacks Using Related Keys*, Advances in Cryptology, EUROCRYPT '93, Springer-Verlag, 1994, pp. 398-409.

[9] J. Kelsey, B. Schneier and D. Wagner, *Related-key cryptanalysis of 3-WAY, Biham-DES,CAST, DES-X, NewDES, RC2, and TEA*, Proc. of Information and Communications Security, LNCS 1334, 1997, pp. 233-246.

[10] W. Zhang, L. Zhang, W. Wu and D. Feng, *Related-Key Differential-Linear Attacks on Reduced AES-192*, Proc. of INDOCRYPT 2007, LNCS 4859, pp. 73-85.

[11] E. Biham, O. Dunkelman and N. Keller, *Related-Key Impossible Differential Attacks on 8-Round AES-192*, Topics in Cryptolog: CT-RSA 2006, LNCS 3860, 2006, pp. 21-33.

[12] G. Sekar, S. Paul and B. Preneel, *Related-Key Attacks on the Py-Family of Ciphers and an Approach to Repair the Weaknesses*, Proc. of INDOCRYPT 2007, LNCS 4859, 2007, pp. 58-72.

[13] S. Lucks, *Ciphers Secure against Related-Key Attacks*, Proc. of Fast Software Encryption, LNCS 3017, 2004, pp. 359-370.

[14] E. Biham, O. Dunkelman and N. Keller, *A Related-Key Rectangle Attack on the Full KA-SUMI*, Proc. of ASIACRYPT 2005, LNCS 3788, 2005, pp. 443-461.