

A Reflection Attack on Blowfish

Tom Gonzalez

Abstract—Since its introduction in 1993, the Blowfish algorithm has come to be regarded as a strong algorithm. However, some attacks are possible for certain poor choices of keys. A certain class of keys called reflectively weak keys are examined, and a method for detecting these weak keys is studied. If a weak key is detected, then it is possible to recover the key.

Index Terms—Blowfish, cryptanalysis, reflection attack, fixed point, key dependent S-Box, self similarity analysis, weak key.



1 INTRODUCTION

Blowfish is a symmetric key cryptosystem which is based on a Feistel network. The algorithm was developed by Bruce Schneier in 1993. Blowfish is a block cipher with a block size of 64 bits, and the full version uses 16 rounds to complete the encryption of a block. Schneier had an admirable motivation to develop Blowfish, to provide the world with a secure, unpatented, and freely-available encryption algorithm by the turn of the century [1]. At the time of publication, Schneier made no claims about the security of his algorithm, but over the course of time the algorithm has been carefully studied and is now considered to be a strong algorithm [2].

When Schneier introduced the Blowfish algorithm, he outlined several requirements and design goals for the algorithm that implements Blowfish. One particular design goal was to minimize the number of weak keys. Schneier stated that he wanted no weak keys, if possible, and if not possible, the probability that a weak key is chosen at random should be small. Schneier did not claim that Blowfish had no weak keys, but he said that if there were weak keys, the probability that one was chosen at random should be small. Kara and Manap have

shown that weak keys do exist for the Blowfish algorithm [4].

The Blowfish algorithm is used extensively in many commercial products. On his personal website, Schneier lists over 200 products that use the algorithm [2]. In his original paper on Blowfish, Schneier stated that any weak keys should be described and classified so that they could be excluded during the key generation process. The fact that the algorithm is still being used extensively makes the study of its weak keys important. The few known results for weak keys are discussed in the second section. The basic Blowfish algorithm and Kara and Manap's reformulation of the algorithm are described in the third section, and Kara and Manap's attack is outlined in the fourth section.

2 RELATED WORK

Schneier himself notes that very little has been published regarding cryptanalysis of Blowfish [2]. Serge Vaudenay was the first to show that weak keys for Blowfish are possible, but Vaudenay's result only holds for a scaled-down version of Blowfish [3]. Vincent Rijmen studied Blowfish using differential cryptanalysis, but Rijmen's results are only valid for Blowfish using no more than four rounds. Kara and Manap have shown that there are weak keys for the full 16 round versions of the Blowfish algorithm, and use of a weak key in the encryption process allows an attacker to recover the key with some knowledge of encryption

• T. Gonzalez is a student with the TSYS School of Computer Science, Columbus State University, Columbus, GA, 31907.

process [4]. Kara refers to this type of an attack as a reflection attack [5] and it exploits the fact that each round of the Feistel network has the same structure.

3 THE BLOWFISH ALGORITHM

First, a general description of the Blowfish algorithm is given. The focus will be on the standard version of Blowfish with 16 rounds. The block size of the Blowfish algorithm is 64 bits. The P array, P_1, P_2, \dots, P_{18} , is an array of 32 bit subkeys. The initialization of the P array is done by using the hexadecimal digits of pi, the key, and the XOR function. There are four 8×32 S -boxes that are used within the function F . Below is a step-by-step description of the algorithm, starting with the plaintext (x_1, y_1) . The 64 bit block of plaintext is split into a left half, x_1 , and a right half y_1 . The symbol \oplus denotes the XOR operation.

$$\begin{aligned} (x_2, y_2) &= (F(P_1 \oplus x_1) \oplus y_1, P_1 \oplus x_1) \\ (x_3, y_3) &= (F(P_2 \oplus x_2) \oplus y_2, P_2 \oplus x_2) \\ (x_4, y_4) &= (F(P_3 \oplus x_3) \oplus y_3, P_3 \oplus x_3) \\ &\dots \\ (x_{17}, y_{17}) &= (F(P_{16} \oplus x_{16}) \oplus y_{16}, P_{16} \oplus x_{16}) \\ (x_{18}, y_{18}) &= (P_{18} \oplus y_{17}, P_{17} \oplus x_{17}) \end{aligned}$$

In their study of weak keys, Kara and Manap [4] reformulated the standard Blowfish algorithm. Kara and Manap's reformulation is described below. Starting with plaintext (x, y) ,

$$\begin{aligned} (x_1, y_1) &= (P_1 \oplus x, P_2 \oplus y) \\ (x_2, y_2) &= (F(x_1) \oplus y_1, x_1) \\ (x_3, y_3) &= (F(y_2) \oplus x_2, y_2 \oplus P_4) \\ (x_4, y_4) &= (F(y_3 \oplus P_3) \oplus x_3, y_3 \oplus P_3 \oplus P_5) \\ (x_5, y_5) &= (x_4 \oplus F(y_4), y_4) \\ (x_6, y_6) &= (x_5 \oplus F(y_5), y_5) \\ &\dots \\ (x_{18}, y_{18}) &= (P_{18} \oplus y_{17}, P_{17} \oplus x_{17}) \end{aligned}$$

The author has not worked through all of the details concerning the two versions of Blowfish being equivalent. The XOR operation is commutative, and Kara and Manap state that this property of XOR is what allows the modification to the original algorithm. Since the function F is not commutative, the modification can be carried out only so far. Kara and Manap examine the formulation of (x_3, y_3) and

(x_4, y_4) closely, and prove that if $P_1 = P_4$ and $P_2 = P_3$, then two rounds formulating (x_3, y_3) and (x_4, y_4) have 2^{32} fixed points. That is, there are 2^{32} plaintexts that are left unchanged by these two steps.

Since the next two rounds of the algorithm, the rounds that formulate (x_5, y_5) and (x_6, y_6) , do not make any use of the P array, the same result concerning the abundance of fixed points holds for these rounds. The rest of the steps in the algorithm cycle through rounds that are similar to the third and fourth rounds and then the fifth and sixth rounds.

Since the equivalent but reformulated Blowfish algorithm consists mainly of the two types of blocks mentioned above, the results about the number of fixed points can be extended to a good portion of the Blowfish algorithm. Examining the reformulated Blowfish algorithm, if the very first step and the last step are left out, there are certain keys that will cause the algorithm to have 2^{32} fixed points. Restated, that means there are 2^{32} plaintexts (ordered pairs (x, y) where x and y are 32 bit strings) that will be left unchanged by these rounds. Any key that will allow this to happen is called a *reflectively weak key*.

4 THE ATTACK

Kara and Manap's attack has two parts. The first part is concerned with identifying a key as a reflectively weak key. Once a reflectively weak key is identified, the second part of the attack is determining the P array. Once the P array is recovered, the key can be reconstructed.

Suppose that a reflectively weak key is used in the Blowfish algorithm. This means that if the first and last round of the reformulated Blowfish algorithm are left out, there will be 2^{32} fixed points. If an attacker wants to test to see if a reflectively weak key has been used, the attacker must know many plaintext-ciphertext pairs. Let (x, y) denote the plaintext and let (x', y') denote the corresponding ciphertext, and suppose that (x, y) is one of the fixed points of the algorithm. Going back to the reformulated description of Blowfish, the steps would look as follows:

$$\begin{aligned}(x_1, y_1) &= (P_1 \oplus x, P_2 \oplus y) \\ (x_2, y_2) &= \dots\end{aligned}$$

Fixed Point \Rightarrow Nothing really happens here!

...

$$\begin{aligned}(x_{17}, y_{17}) &= (P_1 \oplus x, P_2 \oplus y) \\ (x', y') &= (P_{18} \oplus P_1 \oplus x, P_{17} \oplus P_2 \oplus y)\end{aligned}$$

It follows that $x' \oplus x = P_1 \oplus P_{18}$ and that $y' \oplus y = P_2 \oplus P_{17}$. If an attacker has enough plaintext-ciphertext pairs, then the attacker can test to see if a reflectively weak key has been used. For all the plaintext-ciphertext pairs, calculate $(x' \oplus x, y' \oplus y)$. If a reflectively weak key has been used, then the pair $(P_1 \oplus P_{18}, P_2 \oplus P_{17})$ will occur more frequently because this will be the result for every one of the 2^{32} fixed points. Granted, this takes a great deal of plaintext-ciphertext pairs. Kara and Manap state that 2^{34} such pairs are needed. The author has tried to determine exactly why 2^{34} such pairs are needed, but has no results to show for the time invested.

Knowing that a reflectively weak key is being used automatically yields information concerning the P array. Recall that the small blocks that have fixed points have portions of the P array that are identical. This yields half of the 18 elements of the P array. Kara and Manap state that "by guessing $\frac{r}{2} + 1$ subkeys we can determine remaining $\frac{r}{2} + 1$ subkeys and obtain the whole P array," but they give no information on a method to guess the remaining subkeys. Once all of the elements of the P array are determined, an attackers would only need to perform 9 encryptions to recover the key.

5 CONCLUSION

Kara and Manap state that the properties of Blowfish that allow this attack are the similarity of the round functions. In fact, in the standard description of the Blowfish algorithm, all of the rounds are exactly the same. One approach may be to alter the action of each round based on the P array. For instance, use different or multiple elements of the P array at each round, perhaps chosen randomly. One downside to this approach is that in order to decrypt, the number and order of the elements of the P array used at each round would have to be

recorded. This seems to be a costly requirement, especially when Schneier wanted this algorithm to be implemented easily.

REFERENCES

- [1] B. Schneier, Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish), *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 191-204
- [2] Schneier, Bruce, Schneier on Security, <http://www.schneier.com>
- [3] Vaudenay, Serge, On the weak keys of Blowfish, *Lecture Notes in Computer Science*, vol. 1039, pp. 27-32, 1996
- [4] O. Kara and C. Manap, A New Class of Weak Keys for Blowfish, *Lecture Notes in Computer Science*, vol. 4593, pp. 167-180, 2007
- [5] O. Kara, Reflection Attacks on Product Ciphers, *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 191-204