The Twofish block cipher and an observation on its key schedule

Fauzan Mirza

Information Security Group
Royal Holloway University of London

Contents

In this talk I will give

- A very brief introduction to block ciphers
- A brief description of the Twofish block cipher
- An explanation of how to find subkey collisions in Twofish

The work on the Twofish subkey collisions was also due to Sean Murphy (Royal Holloway, University of London).

Block Ciphers – An introduction

Simply, a **block cipher** takes a **plaintext** message block and processes it to a **ciphertext** block under the control of a secret **key**.

The plaintext and ciphertext blocks are measured in bits and are meant to be large to make it difficult for an attacker to exploit plaintext redundancy.

The key is also measured in bits and should be large enough to preclude exhaustive search.

Block Ciphers – Iterated ciphers

An **iterated cipher** processes the message block a fixed number of times with a relatively simple function called the **round function**.

The round function must be invertible in order for the cipher to be able decrypt.

The round function processes the message block using a parameter that depends on the key. These are called round **subkeys** and they are generated by a **key schedule** algorithm.

Block Ciphers – The Key Schedule

The key schedule is supposed to generate a sequence of distinct round subkeys from the key.

The key schedule can be as sensitive as the round function because a flaw in the key schedule of a cipher may weaken the cipher.

- DES has a complementation property
- SAFER-K caused the round subkeys to interact with the cipher structure in a way which could be exploited (due to the byte oriented design)
- TEA has equivalent keys (only a quarter of the key space is effective)

Block Ciphers – Feistel Ciphers

The plaintext $P = (L_0, R_0)$ can be encrypted to ciphertext $C = (R_n, L_n)$ by iterating the following round function:

$$\begin{aligned} L_{i+1} &= R_i \\ R_{i+1} &= L_i \oplus F(R_i, K_{i+1}) \end{aligned}$$

where K_i $(i = 1, \ldots, n)$ are the round subkeys and F is the **cipher function**.

An encryption algorithm that uses this general construction is called a **Feistel cipher**.

Feistel ciphers have the advantage that the construction is easily invertible (the same rules for encryption are also used for decryption with a few simple differences).

The strength of a Feistel cipher is largely dependent on the design of the cipher function and the key schedule.

The DES Block Cipher

The Data Encryption Standard (DES) is a Feistel cipher.

The cipher function F expands the message block by repeating some message bits and then xors the round subkey to it. Eight parallel nonlinear subsitution boxes (**S-boxes**) are applied to the result and then a bitwise permutation follows.

$$F(M, K) \;=\; P(S(E(M) \oplus K))$$

where P is the bitwise permutation, S is the substitution function (the eight parallel S-boxes) and E is the message expansion function.

The S-boxes and the bitwise permutation are fixed and neither depend on the message nor the key.

The Advanced Encryption Standard

DES is over 20 years old and is considered insecure because the key length is too short – an exhaustive search of the DES key space is within the capability of existing computers.

The US Government (specifically NIST – the National Institute for Science and Technology) asked the public to submit algorithms to replace DES as the new encryption standard and 15 candidates were submitted that fulfilled their initial requirements.

These initial requirements required a complete description of the cipher, source code and test vectors for encryption and decryption. No cryptanalysis was required at that stage.

The Twofish Block Cipher

The Twofish block cipher is one of the AES candidates, designed and submitted primarily by Counterpane Systems (USA). One of the Twofish designers also designed the Blowfish algorithm which is fielded in commercial and public domain encryption software.

- It is a Feistel cipher with 16 rounds
- The message block length is 128 bits
- The key length can be 128, 192 or 256 bits
- The cipher function is key dependent and is byte oriented

Twofish – Key schedule

Let the 128-bit key $K = (W, X, YZ)$

The key schedule can be described as follows

$$\begin{aligned} A_i &= \mathcal{Q}(2i) \\ B_i &= \mathcal{Q}(2i+1) \\ C_i &= Q(A_i \oplus Y) \oplus W \\ D_i &= Q(B_i \oplus Z) \oplus X \\ (K_{2i}, K_{2i+1}) &= H(C_i, D_i), \end{aligned}$$

where

$$\mathcal{Q}(x) = (q_0(x), q_1(x), q_0(x), q_1(x))$$

and the functions q_0, q_1 are (key independent) bijective S-boxes with one byte inputs.

The cipher function depends on the key

$$\begin{aligned} S_0 &= (W, X) \begin{pmatrix} T \\ T^2 \end{pmatrix} \\ S_1 &= (Y, Z) \begin{pmatrix} T \\ T^2 \end{pmatrix} \end{aligned}$$

where T is a 4×4 matrix and the matrix multiplication is in a finite field.

Twofish – Round function

Let $+$ denote a pair of modulo 2^{32} additions, and let $\theta = (e, \rho)$ and $\theta' = (\rho^{-1}, e)$, where e is the identity transformation on 32 bits and ρ is a left rotation by one place of 32 bits.

A Twofish encryption of $P = (P_L, P_R)$ under key $K = (K_L, K_R)$ to give ciphertext $C = (C_L, C_R)$ is given by

$$
\begin{aligned}
L_0 &= P_L \oplus (K_0, K_1) \\
R_0 &= P_R \oplus (K_2, K_3) \\
L_{i+1} &= (R_i\theta \oplus (g_{S_0,S_1}(L_i) + (K_{2i+8}, K_{2i+9})))\theta' \qquad [i \\
R_{i+1} &= L_i \qquad [i \\
C_L &= R_{16} \oplus (K_4, K_5) \\
C_R &= R_{16} \oplus (K_6, K_7).
\end{aligned}
$$

Twofish – A reduced version

Assume that the S-boxes in Twofish are fixed (known and/or unkeyed) – then (S_0, S_1) is fixed. We call this variant **reduced Twofish**.

By exploiting a feature of the key schedule we can find **subkey collisions** – a pair of distinct Twofish keys which cause exactly one round to be identical.

Fixing (S_0, S_1) imposes conditions on the 128-bit key. In particular, if the key $K = (W, X, Y, Z)$ then

$$\begin{aligned} W &= X \cdot T \oplus S_0 \cdot T^{-1} \\ Y &= Z \cdot T \oplus S_1 \cdot T^{-1} \end{aligned}$$

Reduced Twofish – Subkey collisions

Choose a round i

$$\begin{aligned} C_i &= Q(A_i \oplus Y) \oplus W = Q(A_i \oplus Y') \oplus W' \\ D_i &= Q(B_i \oplus Z) \oplus X = Q(B_i \oplus Z') \oplus X' \end{aligned}$$

For example, let $(S_0, S_1) = (0, 0)$ so $W = X \cdot T$ and $Y = Z \cdot T$

$$\begin{aligned} X \cdot T \oplus X' \cdot T &= Q(A_i \oplus Z \cdot T) \oplus Q(A_i \oplus Z' \cdot T) \\ X \oplus X' &= Q(B_i \oplus Z) \oplus Q(B_i \oplus Z') \end{aligned}$$

Rearranging gives

$$\begin{aligned} Q(A_i \oplus Z \cdot T) \oplus Q(B_i \oplus Z) \cdot T &= \\ Q(A_i \oplus Z' \cdot T) \oplus Q(B_i \oplus Z') \cdot T \end{aligned}$$

Find a pair (Z, Z') for which the above equation holds, pick any X, solve for X' and we have a pair of distinct reduced Twofish keys which have exactly one round the same (identical cipher function and identical subkeys in round i).

Twofish – Subkey collision example

Reduced Twofish key = (00000000 000006F5)
with (S_0, S_1) = (00000000 00000000)

Twofish key =
(00000000 00000000 82CD3758 000006F5)

$$
\begin{aligned}
(K_0, K_1) &= \texttt{ACDDC058 0E970B77} \\
(K_2, K_3) &= \texttt{42628A3D CE8AFD02} \\
(K_4, K_5) &= \texttt{AE3D830B 9DCA15BE} \\
(K_6, K_7) &= \texttt{322A4105 E03E7A87} \\
(K_8, K_9) &= \texttt{C82616C0 9FB7D001} \star \\
(K_{10}, K_{11}) &= \texttt{32B110EC B81277A4} \\
(K_{12}, K_{13}) &= \texttt{276B9B3E A1A4ABB5} \\
\vdots &= \vdots \\
(K_{34}, K_{34}) &= \texttt{BC7C8FE8 4BE393E0} \\
(K_{36}, K_{37}) &= \texttt{37E37854 FBBC838E} \\
(K_{38}, K_{39}) &= \texttt{775C23F9 7FA0DDDF}
\end{aligned}
$$

Twofish – Subkey collision example

Reduced Twofish key = (0015FB5C 000311C3)
with (S_0, S_1) = (00000000 00000000)

Twofish key =
(1ADE0FAC 0015FB5C AFB89707 000311C3)

$$
\begin{aligned}
(K_0, K_1) &= \texttt{49D9344F 42C53AA9} \\
(K_2, K_3) &= \texttt{F0305D62 2D6DB59B} \\
(K_4, K_5) &= \texttt{58E653DF C055DD84} \\
(K_6, K_7) &= \texttt{B6D643B4 4AE68FD8} \\
(K_8, K_9) &= \texttt{C82616C0 9FB7D001} \star \\
(K_{10}, K_{11}) &= \texttt{8A0DE8A6 6429C34D} \\
(K_{12}, K_{13}) &= \texttt{9FD3BC10 F2E9E072} \\
\vdots &= \vdots \\
(K_{34}, K_{35}) &= \texttt{21593134 C8F09749} \\
(K_{36}, K_{37}) &= \texttt{8EE63DE1 1524A48F} \\
(K_{38}, K_{39}) &= \texttt{178DA3CB E77C2055}
\end{aligned}
$$

Summary

The Twofish key can be regarded as two parts

- one part selects a reduced Twofish cipher (i.e., fixes (S_0, S_1))
- the other part generates the subkeys for the reduced Twofish using an unbalanced (non-surjective for 128-bit keys) key schedule

Whilst we have not found a way to exploit this

- it is a strange property of a cipher and hence undesirable
- the use of two separate parts in a cipher has often led to divide-and-conquer attacks on the keyspace