Cryptanalysis of Block Tea

Markku-Juhani Saarinen
mjos@math.jyu.fi

October 20, 1998

Abstract

We describe an attack against the Block Tea encryption algorithm that solves the secret key with 2^{34} chosen ciphertext queries and little additional computation. The attack can be characterized as a differential attack on a nonsurjective mixing function.

We also present an algorithm that can be used to distinguish Block Tea 2, the proposed fix of Block Tea, from a random permutation with 2^{80} chosen plaintexts.

1 Introduction

In the Fast Software Encryption workshop of 1994, professors Roger Needham and David Wheeler (University of Cambridge) proposed a new cryptographic primitive called Tiny Encryption Algorithm, or TEA [1].

The most striking feature of TEA and it's successors is the exceptionally small size of the source code and the apparent simplicity of design. The TEA encryption and decryption source code is 11 C lines long.

In 1996 David Wagner discovered three related-key attacks against TEA, which were published in [4]. This attack motivated Needham and Wheeler to improve the algorithm, and out came not one, but two algorithms; the "Extended TEA" and "Block TEA" algorithms. These were published as a University of Cambridge research report [2].

Extended TEA, or XTEA is a 64-bit block cipher with Feistel structure. Block Tea is a novel widened Feistel-type design that allows arbitrary block sizes. Both ciphers have a 128-bit keysize. In this article, we will attack the Block Tea algorithm.

2 Breaking Block Tea

We will reproduce (an equivalent) source code of Block Tea encryption here for reference.

```
/*
    Block Tea 1 encryption
    v a pointer to an array of n longwords
    k a pointer to a key of four longwords (128 bits)
 */
```

```
void teab1_encrypt(long *v, long n, long *k)
{
  unsigned long z = v[n - 1], sum = 0, e;
  long p, q;

  for (q = 6 + 52 / n; q > 0; q--)
    {
      sum += 0x9e3779b9;
      e = sum >> 2 & 3 ;
      for (p = 0; p < n; p++)
        z = v[p] += (((z << 4) ^ (z >> 5)) + z) ^
            (k[(p & 3) ^ e] + sum);
    }
}
```

The Block Tea algorithm provides good propagation of differences in the forward direction (from plaintext to ciphertext), but the propagation of differences in reverse direction occurs at the rate of one word per round.

The mixing function uses $f(z) = ((z << 4) \oplus (z >> 5)) + z$ with $z = v[p-1]$ to modify $v[p]$. We note that $f(z)$ is not surjective. Here's some collisions for $f()$.

$z1$	$z2$	$f()$
000201e1	00020000	00221000
000403c2	00040000	00442000
00080784	00080000	00884000
00100f08	00100000	01108000
00201e10	00200000	02210000
00f803c2	10000000	10800000
01f00784	20000000	21000000
03e00f08	40000000	42000000
1e2e4a7d	00040000	00442000
3c23d319	00000020	00000221

The difference propagation in reverse direction does not occur when $v[p-1] = z1$ for one ciphertext and $v[p-1] = z2$ for another. The propagation is delayed by one round.

Our attack works as follows. We first try solve the key that is used to decrypt $v[n-1]$ at the first decryption round. We do this by decrypting a pair of ciphertexts that are the same, except that in the other $v[n-1]$ gets decrypted as 000201e1 and in the another as 00020000. If we have guessed the key word correctly, $v[q-1]$ will match in the plaintexts. This can be verified with another "collision pair" from the table above. Once this key word has been found, we can proceed to find the next key word using the same technique for $v[n-2]$.

Since each key word requires 2^{32} decryptions (average value), the entire 128-bit key can be recovered with 2^{34} decryptions.

3 A Distinguisher for Block Tea 2

Shortly after the author disclosed the attack of the previous section in the USENET newsgroup `sci.crypt.research`, Needham and Wheeler acknowledged that the attack is valid and proposed a “fixed” version. [3] We will use the name Block Tea 2 for this algorithm.

```
/*
    Block Tea 2 encryption
    v a pointer to an array of n longwords
    k a pointer to a key of four longwords (128 bits)
 */

void teab2_encrypt(long *v, long n, long *k)
{
  unsigned long z = v[n - 1], y, sum = 0, e;
  long p, q;

  for (q = 6 + 52 / n; q > 0; q --)
    {
      sum += 0x9e3779b9;
      e = sum >> 2 & 3;
      for (p = 0; p < n - 1; p++)
{
  y = v[p + 1];
  z = v[p] += (((z >> 5) ^ (y << 2)) + ((y >> 3) ^
      (z << 4))) ^ ((sum ^ y) + (k[(p & 3) ^ e] ^ z));
}
      y = v[0];
      z = v[n - 1] += (((z >> 5) ^ (y << 2)) + ((y >> 3) ^
(z << 4))) ^ ((sum ^ y) + (k[(p & 3) ^ e] ^ z));
    }
}
```

We set $n = 53$; there are $q = 6$ rounds. We keep $v[52]$ and $v[0 \ldots 45]$ fixed and choose the ten words $v[42..51]$ as random. Consider a vector $u[i]$ that has the value of $v[52]$ after encryption round i. $u[0]$ and $u[6]$ are the 53rd words of the plaintext and ciphertext, respectively. $u[6]$ does not affect other words; we can ignore it for now. We model $u[1 \ldots 5]$ as random.

The random vector $u[1..5]$ has $(2^{32})^5 = 2^{160}$ possible states. When we encrypt roughly 2^{80} plaintexts v that are chosen as described above, two of these encryptions can be expected to have matching $u[1 \ldots 5]$ vectors by the birthday paradox.

This means that the difference propagation to the right does not flip around from $v[52]$ to $v[0]$ in any round. Therefore the only difference propagation is from $v[42]$ to the left, at the rate of one word per round. This propagation only fills $v[36 \ldots 41]$. This means that the two different plaintexts have ciphertexts that match in the area $v[0 \ldots 35]$.

The chance that a random permutation would leave these over 1100 bits unchanged is very small compared to the probability (lower bound) of 2^{-80}

with Block Tea 2. This means that Block Tea 2 can be distinguished from a random permutation with an algorithm that requires 2^{80} effort.

We also note that the above attack can be turned into a key search shortcut that provides two-fold speedup compared to straightforward exhaustive search.

4 Conclusions

We have presented a practical attack on the Block Tea encryption algorithm and a potential flaw in the Block Tea 2 algorithm.

The main lesson learned from the analysis of these algorithms is that ciphers should be designed to be equally strong in both, encryption and decryption directions. It also rises some questions in the design of variable-length block ciphers.

We have also presented a new general method of attack for widened Feistel ciphers that use nonsurjective round functions and a distinguisher for Feistel ciphers that use feedback in both directions.

References

[1] R. Needham and D. Wheeler *TEA, a Tiny Encryption Algorithm.* Proceedings of FSE 94, Springer-Verlag, 1994, pp 363 – 366

[2] R. Needham and D. Wheeler *TEA Extensions.* `http://www.cl.cam.ac.uk/ftp/djw3/xtea.ps`, 1997

[3] R. Needham and D. Wheeler *Block Tea.* sci.crypt.research posting, September 21, 1998

[4] J. Kelsey, B. Schneier, and D. Wagner *Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, NewDES, RC2, and TEA* ICICS 97 Proceedigs, Springer-Verlag, 1997