Tea extensions

Roger M. Needham and David J. Wheeler

Notes October 1996
Revised March 1997
Corrected*October 1997

Introduction

The two minor weaknesses of TEA pointed out by David Wagner [1] are eliminated (if desired) by some minor changes.

Extensions to TEA

The mixing portion of TEA seems unbroken but related key attacks are possible even though the construction of 2^{32} texts under two related keys seems impractical. The second weakness, that the effective length of the keys is 126 bits not 128 does affect certain potential applications but not the simple cypher decypher mode.

To cater for these weaknesses, we can readily adjust the algorithm, while trying to retain the original objectives of little set up time, simplicity and using a large number of rounds to prevent attacks, and avoid complicated analysis.

The first adjustment, is to adjust the key schedule, and the second is to introduce the key material more slowly. This results in the algorithm given below which repairs the two minor weaknesses and retains almost the same simplicity.

Using two bits of sum to select the keywords saves using a separate count for n, while the shift of 11 causes the sequence to be irregular. However 11 is chosen so that all 4 keywords are used in the the first two cycles. The cycle uses roughly the same formula to ensure “mixing” at the same rate.

It is true that in order to gain these advantages, the time to complete change diffusion is delayed by one cycle, however it seems unnecessary to increase 32 to 33, as it is still believed that the number needed is about 16, and the safety factor against inadequate analysis is much the same.

*The declaration in teab has been changed from sum to sum=0 on page 3. Error found by Keith Lockstone.

[1] Private Communication, possibly due to be published at Eurocrypt 1997, email David Wagner ¡daw@cs.berkeley.edu¿

Coding and Decoding Routine

v gives the plain text of 2 words,
k gives the key of 4 words,
N gives the number of cycles, 32 are recommended,
if negative causes decoding, N must be the same as for coding,
if zero causes no coding or decoding.
assumes 32 bit "long" and same endian coding or decoding

```
tean( long * v, long * k, long N) {
unsigned long y=v[0], z=v[1], DELTA=0x9e3779b9 ;
if (N>0) {
       /* coding */
   unsigned long limit=DELTA*N, sum=0 ;
   while (sum!=limit)
     y+= (z<<4 ^ z>>5) + z ^ sum + k[sum&3],
     sum+=DELTA,
     z+= (y<<4 ^ y>>5) + y ^ sum + k[sum>>11 &3] ;
         }
else  {


           /* decoding */
    unsigned long sum=DELTA*(-N) ;
    while (sum)
      z-= (y<<4 ^ y>>5) + y ^ sum + k[sum>>11 &3],
      sum-=DELTA,
      y-= (z<<4 ^ z>>5) + z ^ sum + k[sum&3] ;
       }
v[0]=y, v[1]=z ;
return ;                                }
```

Block TEA

The algorithm can be modified to cater for larger block sizes. The intrinsic mixing is kept similar to TEA, but we run cyclically round the block. This means the mixing of various stages is done together, and we get a potential saving of time, and a natural binding together of a complete block of N words. The key scheduling is also slightly changed, but probably has the same characteristics.

Thus we do the mix operation along the words of a block and do the whole block operation a number of times.

The operation is v[n]+= mix(v[n-1],key]

The operation rolls around the end. If we assume that the mixing is sufficient in TEA with 64 operations, and that we need at least six operations on each word, so the flow of data into the mth word is at least as much as the key, then we need to operate on each word 6+52/n times. This gives a minimum of 6 mixes on each word. A differential attack exists for 3 mixes, which utilises the changes from the last word to the first word. Also the bandwidth of changes into one word from the previous word is about 6 times 32, which may give sufficient margin against solving a "cut".

We now find the work needed for the longer blocks is about 6 mix operations per word and that when $n > 4$ the block algorithm is faster than TEA in spite of the increased overheads. For n=2 the routine is about twice as slow as TEA. In fact for n=2-10 the time is roughly independent of block size in one implementation.

The question of whether it as secure remains open. The same device can be used on DES, although as it has only 16 rounds and not 64, the gains are not as spectacular.

coding and decoding routine

teab is a block version of tean.
It will encode or decode n words as a single block where $n > 1$.
v is the n word data vector,
k is the 4 word key.
n is negative for decoding,
if n is zero result is 1 and no coding or decoding takes place,
otherwise the result is zero.
assumes 32 bit "long" and same endian coding and decoding.

```
long teab( long * v, long n , long * k ) {
unsigned long z=v[n-1], sum=0,e,
    DELTA=0x9e3779b9 ;
long m, p, q ;
if ( n>1)  {
   /* Coding Part */
  q=6+52/n ;
  while (q-- > 0)           {
    sum += DELTA ;
    e = sum>>2&3 ;
    for (p=0 ; p<n ; p++ )
      z=v[p] += (z<<4 ^ z>>5) + z ^ k[p&3^e] + sum ;
                            }
  return 0 ; }
```

```
  /* Decoding Part */
else if (n<-1) {
  n= -n ;
  q=6+52/n ;
  sum=q*DELTA ;
  while (sum != 0)   {
    e= sum>>2 & 3 ;
    for (p=n-1 ; p>0 ; p-- )      {
      z=v[p-1] ;
      v[p] -= (z<<4 ^ z>>5) + z ^ k[p&3^e] + sum ;
                                  }
    z=v[n-1] ;
      v[0] -= (z<<4 ^ z>>5) + z ^ k[p&3^e] + sum ;
    sum-=DELTA ; }
  return 0 ; }
  return 1 ; }  /* Signal n=0 */
```

Comments

For ease of use and general security the large block version is to be preferred when applicable for the following reasons.

- A single bit change will change about one half of the bits of the entire block, leaving no place where the changes start.
- There is no choice of mode involved.
- Even if the correct usage of always changing the data sent (possibly by a message number) is employed, only identical messages give the same result and the information leakage is minimal.
- The message number should always be checked as this redundancy is the check against a random message being accepted.
- Cut and join attacks do not appear to be possible.
- If it is not acceptable to have very long messages, they can be broken into chunks say of 60 words and chained analogously to the methods used for DES.