STUDI MENGENAI TINY ENCRYPTION ALGORITHM (TEA) DAN TURUNAN-TURUNANNYA (XTEA DAN XXTEA)

Khandar William – NIM : 13506022

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if16022@students.if.itb.ac.id

Abstrak

Roger M. Needham dan David J. Wheeler menciptakan Tiny Encryption Algorithm (TEA), XTEA, dan XXTEA yang terkenal dengan kesederhanaan implementasinya. Pertama mereka menciptakan TEA, lalu kelemahan dari TEA ditutupi oleh XTEA, setelah itu mereka juga menciptakan Block TEA, yaitu algoritma XTEA yang dimodifikasi untuk dapat menerima blok sebesar apapun, dan terakhir mereka menciptakan XXTEA yang menutupi kelemahan dari Block TEA.

Makalah ini akan membahas secara mendalam mengenai TEA, XTEA, dan XXTEA. Makalah ini juga membahas kelehaman dari TEA dan Block TEA yang mengakibatkan munculnya XTEA dan XXTEA. Selain itu, makalah ini juga membahas serangan-serangan yang mengekspoitasi kelemahan-kelemahan tersebut.

***Kata kunci**:* Tiny Encryption Algorithm, TEA, XTEA, XXTEA, enkripsi, dekripsi, *block cipher*.

1. Pendahuluan

Kebutuhan akan keamanan data semakin meningkat seiring dengan berkembangnya teknologi digital. Hampir semua kegiatan komunikasi dan transaksi sekarang dilakukan secara digital sehingga kekhawatiran akan adanya data yang dicuri semakin menjadi-jadi.

Kriptografi muncul untuk menjawab kebutuhan ini. Sudah banyak algoritma kriptografi yang ditemukan dan sudah banyak juga kriptanalisnya. Semakin tinggi tingkat keamanan suatu algoritma kriptografi biasanya disertai dengan meningkatnya beban proses enkripsi dan dekripsinya. Hal ini akan menjadi masalah besar jika diaplikasikan pada perangkat elektronik seperti *handphone* di mana penggunaan daya listrik dan *memory* sangat diperhatikan.

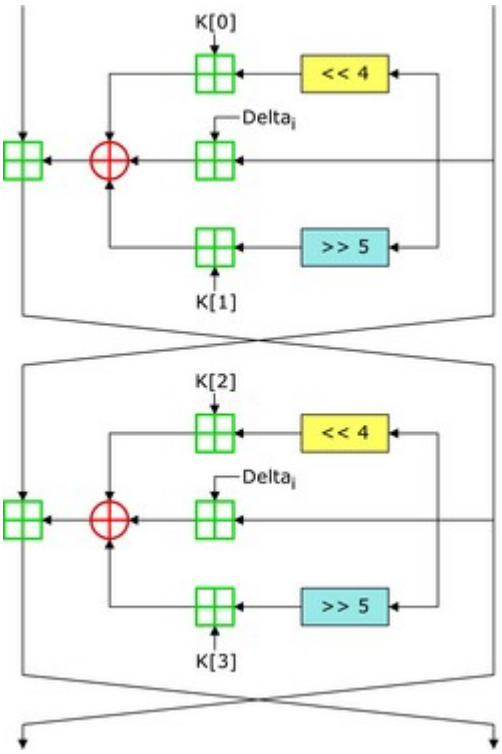
TEA dan turunannya diklaim mempunyai tingkat keamanan yang tinggi jika dibandingkan dengan algoritma kriptografi sejenis. Keunggulan utama dari TEA adalah keringanan prosesnya, operasi-operasi yang digunakan hanya berupa operasi bit biasa, tanpa substitusi, permutasi, ataupun operasi matrix.

2. TEA (Tiny Encryption Algorithm)

TEA adalah algoritma *block cipher* yang diciptakan oleh David J. Wheeler dan Roger M. Needham dari Cambridge University tahun 1994. Hal yang paling menonjol dari TEA adalah kesederhanaan implementasi, ketiadaan S-Box maupun P-Box dan kecepatan yang tinggi.

2.1 Algoritma TEA

TEA beroperasi dalam ukuran blok 64 bit dan panjang kunci 128 bit. TEA berbasiskan jaringan Feistel dan memiliki 32 putaran. Kunci K pertama-tama akan dibagi menjadi 4 kunci internal yaitu K[0..3] masing-masing panjangnya 32 bit. Setiap putaran TEA terdiri atas dua ronde Feistel (lihat gambar 1). Penjadwalan kunci TEA sangat sederhana, yaitu untuk ronde ganjil digunakan K[0] dan K[1], sedangkan untuk ronde genap digunakan K[2] dan K[3].



Gambar 1: Satu putaran enkripsi dalam jaringan Feistel milik TEA

Berikut potongan *source code* enkripsi dan dekripsi TEA.

```
unsigned long* TEA::encrypt() {
 unsigned long y = block[0],
  z = block[1],
  sum = 0,
  delta = 0x9e3779b9,
  n = 32; //jumlah ronde

 while (n-- > 0) {
  sum += delta;
y+=((z<<4)+key[0])^(z+sum)^((z>>5)+key[1]);
z+=((y<<4)+key[2])^(y+sum)^((y>>5)+key[3]);
 }
 block[0] = y;
 block[1] = z;
 return (block);
}
```

```
unsigned long* TEA::decrypt() {
 unsigned long y = block[0],
  z = block[1],
  delta = 0x9e3779b9,
  sum = delta << 5,
  n = 32; //jumlah ronde

 while (n-- > 0) {
z-=((y<<4)+key[2])^(y+sum)^((y>>5)+key[3]);
y-=((z<<4)+key[0])^(z+sum)^((z>>5)+key[1]);
  sum -= delta;
 }
 block[0] = y;
 block[1] = z;
 return (block);
}
```

Angka `delta` didapatkan dari rumus *golden number*:

$$\text{delta} = (\sqrt{5} - 1)2^{31}$$

2.2 Analisis Performa TEA

Berikut tabel berisi hasil enkripsi beberapa sampel *plaintext* (dalam heksadesimal).

Plaintext	***Key***	***Ciphertext***
00000000 00000000	00000000 00000000 00000000 00000000	41ea3a0a 94baa940
00000000 00000001	00000000 00000000 00000000 00000000	414091a7 a27f9c32
00000000 00000000	00000000 00000000 00000000 00000001	0c6d2a1d 930c3fab
ffffffff ffffffff	ffffffff ffffffff ffffffff ffffffff	319bbefb 016abdb2
efffffff ffffffff	ffffffff ffffffff ffffffff ffffffff	dd2617de fe524cc8

Berdasarkan tabel di atas terlihat bahwa TEA mengimplementasikan prinsip *diffusion* milik Shanon dengan baik karena perbedaan 1 bit pada *plaintext* mengakibatkan perubahan besar pada *ciphertext*.

Berikut tabel berisi waktu yang diperlukan untuk mengenkripsi *file* ukuran tertentu (mode ECB).

Besar *File*	**Lama Enkripsi**	**Lama Dekripsi**
~ 10 KB	0,078 detik	0,031 detik
~ 100 KB	0,421 detik	0,375 detik
~ 1 MB	3,687 detik	3,718 detik
~ 10 MB	36,109 detik	35,906 detik

Berdasarkan tabel di atas dapat disimpulkan bahwa waktu yang diperlukan untuk enkripsi dan dekripsi TEA relatif sama.

2.3 Serangan Terhadap TEA

David Wagner dan Kelsey pada tahun 1997 menemukan kerentanan TEA terhadap *equivalent key* dan *related key attack* karena kesederhanaan penjadwalan kuncinya.

Equivalent key yang dimiliki TEA adalah untuk setiap kunci terdapat tiga buah kunci lain yang menghasilkan *ciphertext* yang sama. Kunci-kunci tersebut didapatkan dengan membalik nilai *most significant bit* (MSB) pada K[0] dan K[1] atau K[2] dan K[3]. Sehingga panjang kunci 128

bit hanya akan menghasilkan 2^{126} kunci yang berbeda. Berikut adalah contoh bukti dari keberadaan *equivalent key* pada TEA.

Plaintext	***Key***	***Ciphertext***
00000000 00000000	80000000 00000000 00000000 00000000	9327c497 31b08bbe
00000000 00000000	00000000 80000000 00000000 00000000	9327c497 31b08bbe
00000000 00000000	80000000 00000000 80000000 80000000	9327c497 31b08bbe
00000000 00000000	00000000 80000000 80000000 80000000	9327c497 31b08bbe

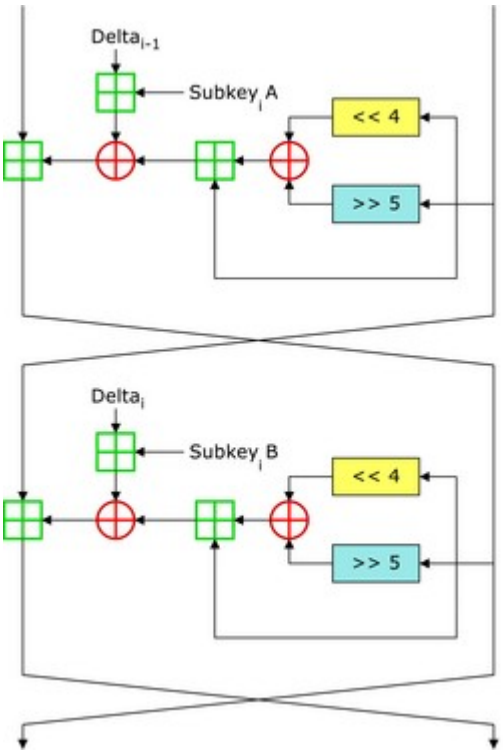
Related key attack yang ditemukan oleh Kelsey et al dapat memecahkan TEA menggunakan satu *related-key query* dan 2^{34} *chosen plaintext*.

3. XTEA (eXtended TEA)

Wheeler dan Needham menciptakan XTEA pada tahun 1997 untuk menutupi kelemahan pada TEA.

3.1 Algoritma XTEA

Sama seperti TEA, XTEA juga beroperasi dalam ukuran blok 64 bit dan panjang kunci 128 bit. Bentuk jaringan Feistel nya pun masih sama, yang membedakan adalah fungsi Feistel dan penjadwalan kunci yang digunakan. Pada XTEA, pada ronde ganjil digunakan `K[sum & 3]`, sedangkan pada ronde genap digunakan `K[sum >> 11 & 3]`.



Gambar 2: Satu putaran enkripsi dalam jaringan Feistel miliki XTEA

Berikut potongan *source code* enkripsi dan dekripsi XTEA.

```
unsigned long* XTEA::encrypt() {
 unsigned long y = block[0],
  z = block[1],
  delta = 0x9e3779b9,
  n = 32;
 unsigned long sum = 0;
 while (n-- > 0) {
  y+=(z<<4^z>>5)+z^sum+(key[sum&3]);
  sum += delta;
  z+=(y<<4^y>>5)+y^sum+(key[sum>>11&3]);
 }
 block[0] = y;
 block[1] = z;
 return (block);
}
```

```
unsigned long* XTEA::decrypt() {
 unsigned long y = block[0],
  z = block[1],
  delta = 0x9e3779b9,
  n = 32;
 unsigned long sum = 0xc6ef3720;
 while (n-- > 0) {
  z-=(y<<4^y>>5)+y^sum+(key[sum>>11&3]);
  sum -= delta;
  y-=(z<<4^z>>5)+z^sum+(key[sum&3]);
 }
 block[0] = y;
 block[1] = z;
 return (block);
}
```

XTEA menggunakan angka `delta` yang sama dengan TEA. Pada dekripsi, variabel `sum` diinisialisasi dengan nilai

```
0xc6ef3720 = delta*32 = delta<<5
```

3.2 Analisis Performa XTEA

Berikut adalah hasil enkripsi beberapa sampel *plaintext* (dalam heksadesimal).

Plaintext	***Key***	***Ciphertext***
00000000 00000000	80000000 00000000 00000000 00000000	057e8c05 50151937
00000000 00000000	00000000 80000000 00000000 00000000	4f190ccf c8deabfc
00000000 00000000	80000000 00000000 80000000 80000000	31c4e2c6 b347b2de
00000000 00000000	00000000 80000000 80000000 80000000	ed69b785 66781ef3

Berdasarkan tabel di atas dapat dilihat bahwa kelemahan *equivalent key* pada TEA sudah tertutupi sehingga membalik MSB pada tiap kunci sudah tidak menghasilkan *ciphertext* yang sama lagi.

Berikut tabel berisi waktu yang diperlukan untuk melakukan enkripsi *file* dengan ukuran tertentu (mode ECB).

Besar *File*	**Lama Enkripsi**	**Lama Dekripsi**
~ 10 KB	0,093 detik	0,046 detik
~ 100 KB	0,453 detik	0,468 detik
~ 1 MB	4,421 detik	4,406 detik
~ 10 MB	44,250 detik	44,203 detik

Berdasarkan tabel di atas dapat disimpulkan bahwa lama waktu enkripsi dan dekripsi XTEA relatif lama dan sedikit lebih lambat daripada TEA.

3.3 Serangan Terhadap XTEA

Beberapa kripanalisis atas XTEA sudah pernah dipublikasikan dan hampir semuanya adalah *differential attack*. Beberapa di antaranya:

- Moon et al (2002) mematahkan XTEA 14-ronde menggunakan *impossible differential attack*, membutuhkan $2^{62.5}$ *chosen plaintext*,
- Hong et al (2003) berhasil mematahkan XTEA 23-ronde menggunakan *truncated differential attack*, membutuhkan $2^{20.55}$ *chosen plaintext*,
- Ko et al (2004) berhasil mematahkan XTEA 27-ronde menggunakan *related-key differential attack*, membutuhkan $2^{20.5}$ *chosen plaintext*.

Sampai saat ini serangan terhadap XTEA masih efektif untuk XTEA yang rondenya dikurangi. XTEA yang paling umum digunakan adalah XTEA 32-ronde. Jadi, hingga saat ini XTEA masih dapat dianggap sebagai algoritma kriptografi yang aman asalkan ronde yang digunakan tidak kurang dari 32.

4. XXTEA (Corrected Block TEA)

Beriringan dengan dirilisnya XTEA, Wheeler dan Needham juga merilis Block TEA, yaitu algoritma *block cipher* berbasiskan XTEA yang dapat dioperasikan pada blok sebesar apa pun. Dengan sifat ini, maka Block TEA dapat digunakan tanpa mode operasi *cipher*. Namun, Saarinen (1998) mengungkapkan kelemahan yang terdapat dalam proses dekripsi Block TEA sehingga XXTEA dikeluarkan pada tahun 1998 untuk menutupinya.

4.1 Algoritma XXTEA

XXTEA merupakan turunan dari Block TEA sehingga memiliki beberapa perbedaan yang signifikan dengan TEA dan XTEA. XXTEA beroperasi dalam ukuran blok kelipatan 32 bit dan panjang kunci 128 bit. Karena XXTEA tidak memiliki batas ukuran blok, XXTEA dapat digunakan untuk mengenkripsi satu buah pesan utuh tanpa memerlukan mode operasi cipher.

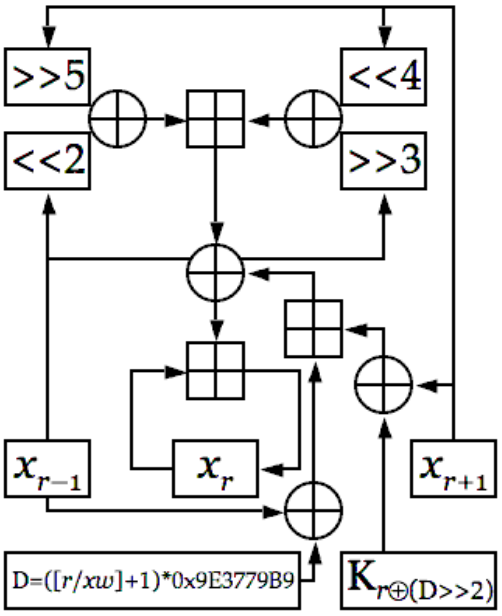
Meski XXTEA dapat mengenkripsi keseluruhan pesan atau *file* sekaligus, pada implementasinya XXTEA dapat dioperasikan dengan mode operasi untuk *file-file* yang berukuran sangat besar sehingga tidak bisa dibaca ke dalam *memory* sekaligus.

Jumlah putaran XXTEA bergantung pada panjang blok. Rumus jumlah putaran `p`:

```
p = 6+52/n
```

n : panjang blok dalam bit / 32

Berbeda dengan TEA dan XTEA, terdapat dua *loop* di dalam proses enkripsi dan dekripsinya, *loop* luar sama seperti *loop* pada TEA dan XTEA dilakukan sebanyak putaran, sedangkan *loop* dalam melakukan iterasi per 32 bit blok (disebut segmen). Pada setiap iterasi tersebut dilakukan operasi terhadap segmen X_r yang melibatkan segmen sebelum X_{r-1} dan sesudahnya X_{r+1}.



Gambar 3: Operasi enkripsi pada tiap segmen blok per putaran jaringan Feistel milik XXTEA

Berikut adalah potongan *source code* enkripsi dan dekripsi XXTEA.

```
unsigned long* XXTEA::encrypt() {
 unsigned long y = block[0],
  z = block[n-1],
  delta = 0x9e3779b9,
  sum = 0,
  e;
 unsigned long m,p,q;
 q = 6 + 52/n;
 while (q-- > 0) {
  sum += delta;
  e = sum >> 2 & 3;
  for (p=0; p<n-1; p++) {
   y = block[p+1];
   z = block[p] += MX;
  }
  y = block[0];
  z = block[n-1] += MX;
 }
 return (block);
}
```

```
unsigned long* XXTEA::decrypt() {
 unsigned long y = block[0],
  z = block[1],
  delta = 0x9e3779b9,
  sum = 0,
  e;
 unsigned long m,p,q;
 q = 6 + 52/n;
 sum = q*delta;
 while (sum != 0) {
  e = sum >> 2 & 3;
  for (p=n-1; p>0; p--) {
   z = block[p-1];
   y = block[p] -= MX;
  }
  z = block[n-1];
  y = block[0] -= MX;
  sum -= delta;
 }
 return (block);
}
```

4.2 Analisis Performa XXTEA

Berikut adalah hasil enkripsi beberapa sampel *plaintext* (dalam heksadesimal).

Plaintext	***Key***	***Ciphertext***
00000000 00000000	00000000 00000000 00000000 00000000	053704ab 575d8c80
00000000 00000000 00000000	00000000 00000000 00000000 00000000	5e3cd3f0 e109e3ce 79d7c945
00000000 00000000 00000000 00000000	00000000 00000000 00000000 00000000	e6c8d5ff 070fb6e4 98a534f7 ac03e399
80000000 00000000 00000000 00000000	00000000 00000000 00000000 00000000	ac42a409 ef2446d4 f8b99981 b72a2fab
00000000 00000000 00000000 00000001	00000000 00000000 00000000 00000000	3a931e30 24d0257c 68fc69da 21ceed21
00000000 00000000 00000000 00000000	80000000 80000000 00000000 00000000	aa89cca2 04037043 327e686c 000991d1

Berdasarkan tabel di atas dapat disimpulkan beberapa hal, di antaranya adalah XXTEA masih tetap mempertahankan *diffusion* yang dimiliki TEA meski panjang bloknya berbeda, selain itu XXTEA juga sudah tidak memiliki kelemahan *equivalent key* seperti TEA.

Berikut adalah waktu enkripsi dan dekripsi beberapa *file* berukuran tertentu dengan XXTEA (mode ECB).

Besar *File*	**Lama Enkripsi**	**Lama Dekripsi**
~ 100 KB	0,062 detik	0,015 detik
~ 1 MB	0,078 detik	0,062 detik
~ 10 MB	0,578 detik	0,500 detik
~ 100 MB	6,265 detik	5,859 detik
~ 1 GB	112,343 detik	111,218 detik

Berdasarkan tabel di atas dapat dilihat jelas satu keunggulan dari XXTEA yaitu kecepatannya. Waktu yang diperlukan bagi XXTEA untuk

mengenkripsi dan dekripsi hampir 100 kali lipat lebih cepat daripada XTEA.

4.3 Serangan Terhadap XXTEA

Saarinen (1998) pernah mengeluarkan teknik Distinguisher untuk memecahkan XXTEA 6-ronde dengan memerlukan 2^{80} *chosen plaintext*. Serangan ini tidak terlalu berbahaya karena belum bisa memecahkan XXTEA untuk jumlah ronde yang besar dan waktu yang dibutuhkan pun masih eksponensial.

Baru-baru ini, Elias Yarrkov (2008) mengirim tulisan ke sebuah milis di sci.crypt yang menjelaskan serangan *chosen plaintext attack* yang dia temukan. Menurut pengakuannya, serangan ini terbukti lebih cepat daripada *brute force* dan memerlukan 2^{110} chosen plaintext untuk memecahkan XXTEA 6-ronde. Berhubung serangan ini baru saja ditemukan, maka masih banyak potensi untuk mengembangkan dan mengoptimalkan serangan ini.

5. Kesimpulan

TEA adalah bapak dari XTEA dan XXTEA. Karakteristik dari TEA yang menonjol adalah *small, secure, simple, and fast*. Dengan karakteristik seperti ini, TEA layak untuk dijuluki "kecil-kecil cabe rawit".

Untuk penggunaan secara umum, lebih direkomendasikan untuk menggunakan XTEA atau XXTEA karena TEA sudah resmi dinyatakan tidak aman. Dan jika ingin memperoleh kecepatan yang tinggi, maka XXTEA bisa dijadikan pilihan utama.

XTEA dan XXTEA sangat cocok untuk diaplikasikan ke perangkat-perangkat elektronik *mobile* seperti *handphone* karena proses enkripsi dan dekripsinya tidak memakan *resource* terlalu berat.

DAFTAR PUSTAKA

[1] Munir, Rinaldi. (2006). Diktat Kuliah IF3054 Kriptografi.

[2] Wheeler, David J. & Needham, Roger M. (1994). TEA, a Tiny Encryption Algorithm. *http://edipermadi.files.wordpress.com/2008/06/tea-spec.pdf*. Tanggal akses: 12 Maret 2009 pukul 17:00.

[3] Wheeler, David J. & Needham, Roger M. (1997). Tea Extensions. *http://os2.zemris.fer.hr/algoritmi/simetricni/2007_marceta/XTEA/xtea.pdf*. Tanggal akses: 12 Maret 2009 pukul 17:00.

[4] Wheeler, David J. & Needham, Roger M. (1998). Correction to Xtea. *http://www.cix.co.uk/~klockstone/xxtea.pdf*. Tanggal akses: 12 Maret 2009 pukul 17:00.

[5] Shepherd, Simon (2006). The Tiny Encryption Algorithm. *http://143.53.36.235:8080/tea.htm*. Tanggal akses: 12 Maret 2009 pukul 17:00.

[6] Moon, Dujae et al. (2002). Impossible Differential Cryptanalysis of Reduced Round XTEA and TEA. *In Fast Software Encryption – Proceedings of the 9th International Workshop*.

[7] Russel, Matthew D. (2004). Tinyness: An Overview of TEA and Related Ciphers.

[8] Yarrkov, Elias. (2008). A Chosen Plaintext Attack for XXTEA. http://groups.google.co.za/group/sci.crypt/browse_thread/thread/5ed40f85e693ef04. Tanggal akses: 30 Maret 2009 pukul 22:00.

Catatan: *source code* yang digunakan dalam makalah ini dapat diperoleh dengan menghubungi saya lewat *email*.