Analisis Mengenai Serangan-serangan Terhadap *Stream Cipher*

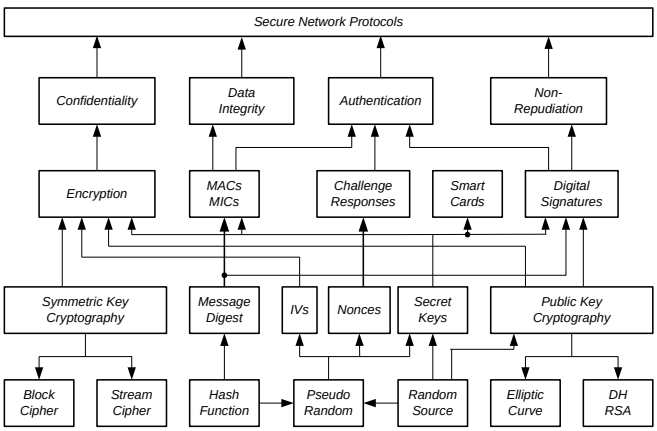
Muhamad Rizky Yanuar - 13508015[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]if18015@if.itb.ac.id

***Abstrak*—Stream cipher merupakan salah satu jenis kriptografi modern. Stream cukup popular digunakan dalam kehidupan sehari-hari, contohnya yaitu pada *wireless connection* dan kriptografi dalam bidang militer. Hal tersebut karena memang stream cipher terbukti memiliki tingkat keamanan yang cukup baik dan sulit ditembus oleh serangan-serangan dari para hacker dan cracker. Namun hal tersebut bukan berarti stream cipher merupakan algoritma enkripsi yang kebal dan tanpa celah. Pada makalah ini penulis akan memaparkan dan memberikan analisis mengenai stream cipher dan serangan-serangannya.**

***Kata kunci*—Stream cipher, *stream cipher attack*, kriptografi modern, streamkey generator, streamkey.**

I. PENDAHULUAN

I.I Algoritma Kriptografi Modern



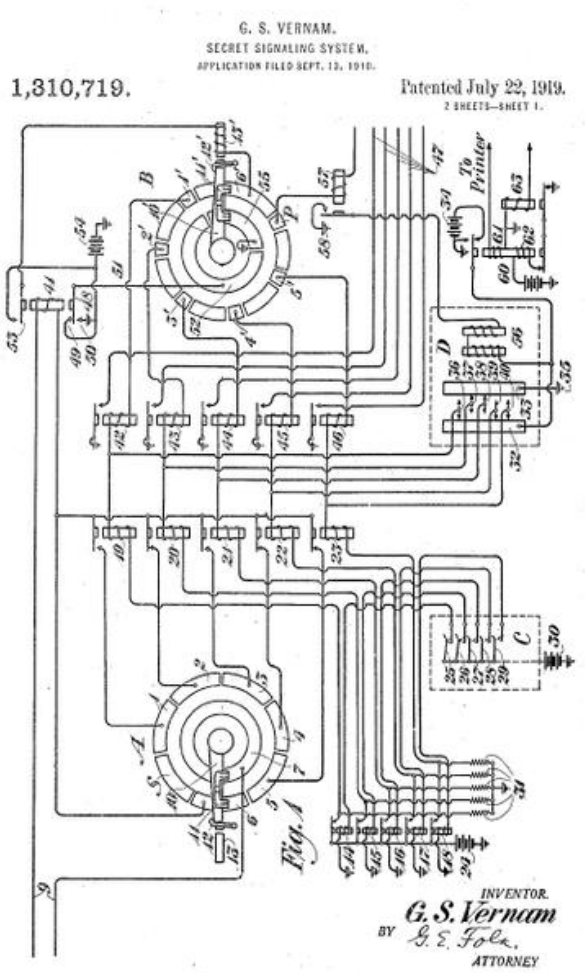
Gambar 1. Diagram Blok Kriptografi Modern

Dahulu, sebelum komputer digital digunakan secara umum, kriptografi menggunakan karakter sebagai dasar dari proses penyembunyian pesan. Saat ini, seiring dengan berkembangnya teknologi, proses enkripsi dan dekripsi untuk penyembunyian pesan tidak lagi dengan sebatas memanipulasi karakter saja, namun juga dengan mamanipulasi bit-bit dari plain text yang ingin dienkripsi ataupun cipher text yang ingin didekripsi. Namun dalam prosesnya, teknik-teknik pada algoritma kriptografi klasik tetap digunakan, yaitu substitusi dan transposisi, namun akan jauh lebih sulit dipecahkan. Operasi yang umum digunakan dalam kriptografi modern ini yaitu operasi bit XOR.

I.II Stream Cipher

Stream cipher, atau dikenal juga dengan nama state cipher, merupakan salah satu jenis algoritma kriptografi modern. Algoritma ini beroprasi pada bit atau byte tunggal, berbeda dengan block cipher yang beroprasi pada blok bit dengan ukuran tertentu. Sehingga proses enkripsi ataupun dekripsi pun dilakukan secara bit per bit (1 bit setiap kali transformasi) atau byte per byte (1 byte setiap kali transformasi).

Dalam sejarahnya, algoritma stream cipher ini pertama kali diperkenalkan oleh Vernam, yang di kemudian hari beliau juga menjadi salah satu penemu dari one-time pad cipher. Saat itu, Vernam mengusulkan untuk dibuat sebuah teletype cipher yang telah diset dengan suatu kunci yang kemudian akan digabungkan dengan masukan plain text karakter per karakter, sehingga menghasilkan suatu cipher text hasil enkripsi. Proses dekripsi dilakukan dengan melakukan hal yang sama, namun dengan masukan berupa cipher text. Proses pengombinasian karakter tersebut menggunakan operasi XOR yang diaplikasikan pada bit yang digunakan oleh teletype yang direpresentasikan dalam Baudot teletype code. Saat itu sebenarnya Vernam belum menggunakan istilah "XOR", namun dia mengimplementasikannya sebagai operasi dalam relay logic. Sebagai contoh, misalkan plain text yang akan dienkripsi adalah A, diartikan sebagai "+ + - - -" dalam Baudot, lalu key-nya adalah B yang diartikan sebagai "+ - - + +" dalam Baudot. Maka cipher text hasil enkripsinya menjadi "- + - + +" yang diartikan sebagai G oleh Baudot. Dengan menggabungkan G dengan key B, maka akan menghasilkan "+ + - - -", yaitu plain text A.



Gambar 2. Vernam's Patent

Proses enkripsi pada Vernam cipher bisa dituliskan sebagai berikut:

$$c_i = (p_i + k_i) mod\ 2 = p_i \oplus k_i$$

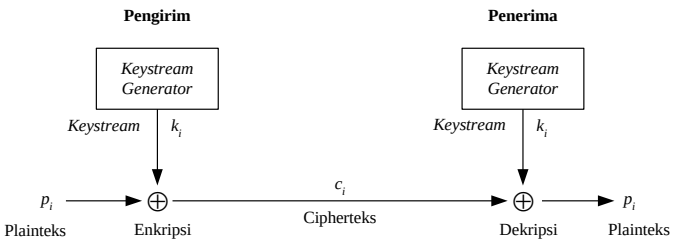
c_i : bit cipher text

p_i : bit plain text

k_i : bit kunci

Proses dekripsi pada Vernam cipher bisa dituliskan sebagai berikut:

$$p_i = (c_i + k_i) mod\ 2 = c_i \oplus k_i$$



Gambar 3 Konsep Stream Cipher

Bit-bit kunci yang digunakan sebagai kunci untuk melakukan proses enkripsi atau dekripsi disebut keystream. Keystream tersebut dibangkitkan oleh keystream generator. Keystream inilah yang menjadi faktor utama tingkat keamanan suatu stream cipher. Oleh karena itu diperlukan suatu keystream generator yang baik sehingga bisa menghasilkan keystream yang baik.

Sebelumnya, mari kita tinjau kasus-kasus yang mungkin terjadi dari suatu keystream yang dibangkitkan oleh keystream generator.

1. Keystream seluruhnya 0
2. Keystream berulang secara periodik
3. Keystream acak

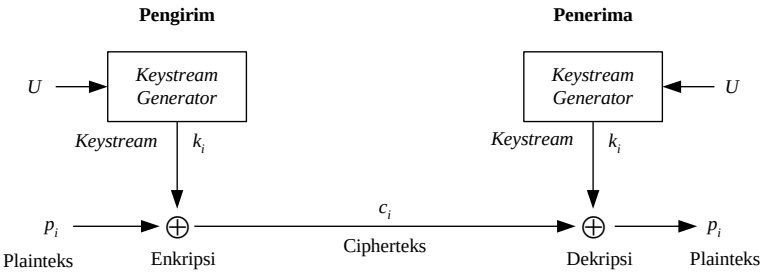
Jika pembangkit mengeluarkan keystream yang seluruhnya nol, maka hasil akhir suatu proses enkripsi menjadi cipher text yang sama persis dengan plain text. Hal tersebut mengakibatkan proses enkripsi menjadi sia-sia.

Jika pembangkit mengeluarkan keystream yang berulang secara periodik, maka algoritma enkripsi yang dihasilkan akan menjadi sangat sederhana dan akan sangat mudah dipecahkan oleh kriptoanalis.

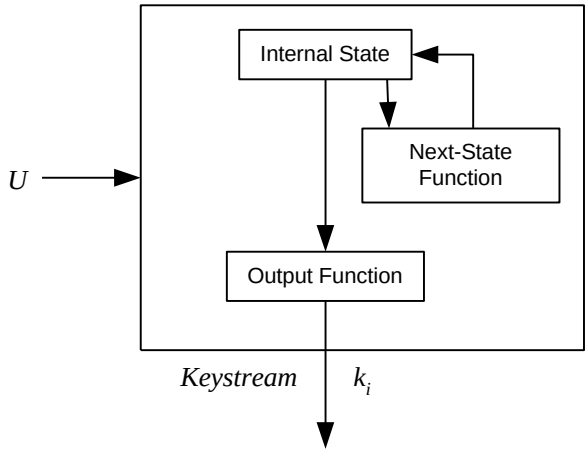
Jika pembangkit mengeluarkan keystream yang benar-benar acak, maka algoritma enkripsinya menjadi algoritma enkripsi yang dikenal dengan nama one-time pad, yaitu algoritma enkripsi dengan tingkat keamanan yang sempurna.

Oleh karena itu, keystream generator diharapkan bisa menghasilkan bit-bit kunci yang kuat, seperti pada contoh kasus 3.

Dalam prosesnya, keystream generator memiliki prosedur untuk menerima masukan sebuah kunci U. Keluarannya merupakan fungsi dari U yang sudah berupa keystream. Dalam hal ini, pengirim dan penerima harus memiliki kunci U yang sama dan bisa dijaga kerahasiaannya dari pihak luar.

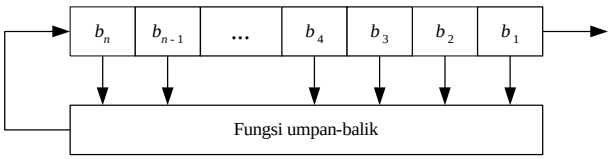


Gambar 4. Stream Cipher dengan Keystream Generator yang bergantung pada kunci U



Gambar 5. Proses di Dalam Keystream Generator

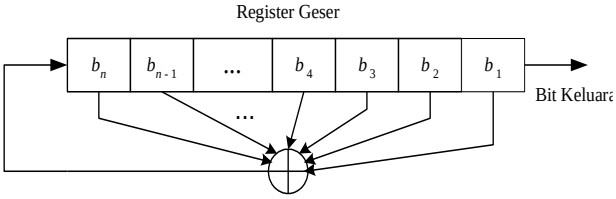
Salah satu contoh keystream generator adalah FSR atau Feedback Shift Register.



Gambar 6. Feedback Shift Register

FSR terdiri dari dua komponen penting, yakni *shift register* dan *feedback function*. Kedua komponen tersebut memegang peranan penting dalam FSR, yaitu sebagai formula dalam keacakan keystream yang dihasilkan.

Salah satu contoh dari FSR adalah Linear Feedback Shift Register (LFSR).



Gambar 7. LFSR

Salah satu contoh dari FSR adalah Linear Feedback Shift Register (LFSR).

Dapat diambil kesimpulan bahwa algoritma stream cipher yang baik sebaiknya memenuhi faktor-faktor berikut, yakni.

Algoritma stream cipher yang baik harus memiliki periode output yang semaksimum mungkin. Kunci yang tidak maksimum bisa meciptakan terjadinya perulangan kunci yang membuat proses penjebolan menjadi makin mudah.

- Kompleksitas yang dimiliki harus semaksimum mungkin.
- Tahan terhadap berbagai serangan.
- Sulit dipecahkan secara komputasi.
- Mudah diimplementasikan dalam software maupun hardware
- Dapat diaplikasikan secara real-time.

II. SERANGAN TERHADAP STREAM CIPHER

Stream cipher memang dikenal memiliki tingkat keamanan yang cukup baik, terutama jika operasi XOR saat melakukan proses enkripsi bit-per-bit-nya. Namun jika digunakan secara tidak tepat, kunci bisa dengan sangat mudah dibongkar. Berikut merupakan beberapa metode yang bisa digunakan untuk membongkar stream cipher.

II.I Exhaustive Search

Merupakan serangan pencarian key dengan cara brute force. Serangan ini efektif jika diketahui kunci merupakan kumpulan karakter yang sangat pendek serta jangkauan karakter untuk key tersebut diketahui. Namun serangan ini dinilai tidak efektif dalam penerapannya di dunia nyata, karena kunci yang lemah seperti disebutkan di atas hampir tidak mungkin dipakai sebagai key untuk stream cipher.

II.II Dictionary Search

Serangan ini merupakan salah satu metode lain yang hampir mirip denganexhaustive search. Namun pada cara ini domain pencarian dipersempit dengan sebatas kata-kata yang ada pada kamus saja. Metode ini tentu saja tidak efektif jika karakter yang dihasilkan oleh streamkey generator merupakan karakter acak yang tidak membentuk suatu makna.

II.III Known-Plaintext Attack

Serangan seperti ini bisa terjadi jika ada pihak lain yang tidak diinginkan mengetahui potongan dari yang plaintext dan ciphertext yang saling berkoresponden. Jika hal tersebut terjadi, maka membongkar stream cipher jadi sangat mudah. Itu karena akan terjadi hal sebagai berikut.

$$\begin{aligned} P \oplus C &= P \oplus (P \oplus K) \\ &= (P \oplus P) \oplus K \\ &= 0 \oplus K \\ &= K \end{aligned}$$

Contoh penggunaan kasus asli untuk formula diatas adalah sebagai berikut. Dilakukan percobaan proses XOR pada suatu karakter.

P	01100110	(karakter ‘f’)
K	00110101	(karakter ‘5’)
C	01010011	(karakter ‘S’)

P	01100110	(karakter ‘f’)
K	00110101	(karakter ‘5’)

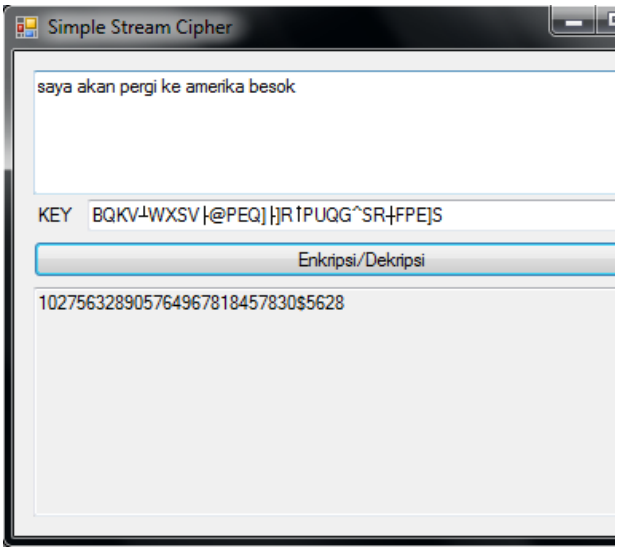
Berikut adalah pengujian bagi known-plaintext attack seperti diatas dalam suatu kasus sederhana. Misalkan diketahui suatu ciphertext sebagai berikut.

“BQKVWXSV@PEQ]]RPUQG^SRFPE]S”

Lalu diketahui ternyata plaintext yang berkesesuaian dengan cipher text tersebut, yaitu

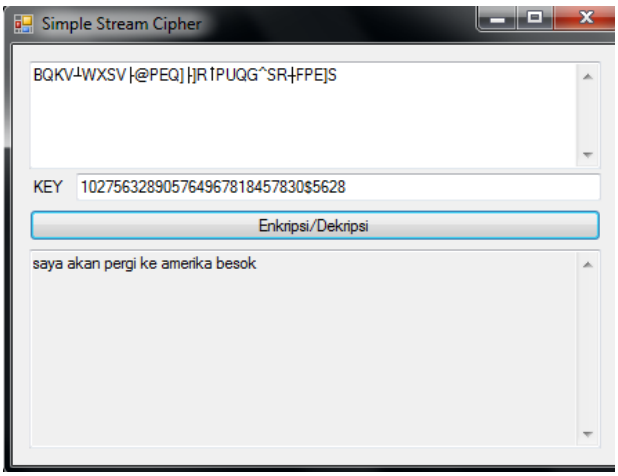
“saya akan pergi ke amerika besok”

Maka key-nya bisa diketahui dengan mudah, yaitu dengan melakukan operasi XOR pada plain text dengan ciper text yang diketahui tersebut.



Gambar 8. Proses XOR pada plain text dan cipher text yang bersesuaian.

Kebenaran teori ini bisa dibuktikan dengan melakukan proses dekripsi pada cipher text tersebut dengan kunci yang telah diketahui.



Gambar 9. Pembuktian Kunci yang Ditemukan dengan Known-Plaintext Attack

Hanya saja dengan metode ini, keseluruhan plain text dengan cipher text harus diketahui. Itu pun jika memang streamkey generator selalu mengeluarkan pola yang sama untuk streamkey yang dikeluarkannya. Jika tidak, maka butuh usaha lebih lagi untuk membongkar streamkey generator tersebut.

Dalam aplikasinya di dunia nyata, known-plaintext attack banyak digunakan untuk membongkar arsip ZIP yang sudah dienkripsi. Hal tersebut dikarenakan sang kriptoanalis hanya tinggal mengekstrak salah satu file dalam ZIP tersebut untuk mendapat plain text-nya. Dengan berbekal itu, kriptoanalis hanya tinggal menggunakan software-software pembantu untuk mengkalkulasikan kunci yang digunakan untuk mengenkripsi keseluruhan arsip.

Namun known-plaintext attack tidak mempan terhadap beberapa modern chipher seperti AES (Advanced Encription Standard).

II.IV Ciphertext-only Attack

Ciphertext-only attack bisa dilakukan untuk membongkar suatu stream cipher jika suatu streamkey generator **mengeluarkan satu key yang sama untuk mengenkripsi dua plain text yang berbeda.** Teknik ini biasa disebut juga dengan teknik keystream reuse attack.

Misalnya kriptanalis mempunyai dua potongan cipher text yang berbeda, kita sebut saja C_1 dan C_2 dan keduanya dienkripsi oleh bit-bit kunci yang sama, sebut saja K. Maka jika kedua cipher text tersebut di XOR-kan, akan menjadi seperti berikut.

$$\begin{aligned} C_1 \oplus C_2 &= (P_1 \oplus K) \oplus (P_2 \oplus K) \\ &= (P_1 \oplus P_2) \oplus (K \oplus K) \\ &= (P_1 \oplus P_2) \oplus 0 \\ &= (P_1 \oplus P_2) \end{aligned}$$

Saat P_1 atau P_2 diketahui atau berhasil ditebak oleh sang kriptanalis, maka dengan meng-XOR-kan salah satu plain text tersebut dengan cipher text-nya untuk mendapatkan K yang bersesuaian.

$$P_1 \oplus C_1 = P_1 \oplus (P_1 \oplus K) = K$$

Dari kunci ini bisa ditemukan P_2, dengan kalkulasi sebagai berikut.

$$C_2 \oplus K = P_2$$

Dengan berbekal P_1 dan P_2, dua plain text yang ter-XOR pun bisa diketahui dengan menggunakan nilai statistik.

Dalam pengaplikasiannya, serangan dengan known-ciphertext attack ini pernah menjebol berbagai macam aplikasi. Contohnya adalah sebagai berikut.

- Versi awal dari PPTP virtual private network

yang dibuat oleh Microsoft. Software ini rentan oleh known-ciphertext attack karena terkadang menggunakan streamkey yang sama.

- WEP (Wired Equivalent Privacy), yaitu merupakan protokol keamanan pertama untuk Wi-Fi.
- Beberapa modern cipher juga ternyata rentan oleh serangan ini, seperti akelarre.

II. V Flip-bit Attack

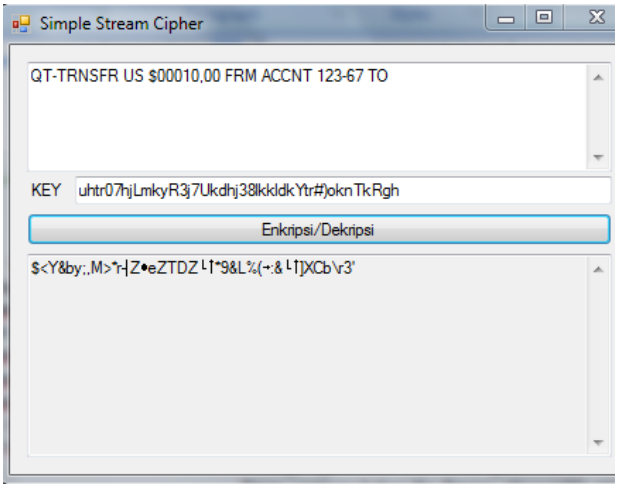
Flip-bit attack merupakan serangan pada stream cipher dengan tujuan untuk mengubah hasil dekripsi dengan cara mengubah bit cipher text tertentu. Proses pengubahan tersebut dilakukan dengan melakukan proses flip (membalikkan) bit tertentu pada cipher text. Maksud dari proses flip tersebut adalah mengubah 0 jadi 1 atau 1 jadi 0.

Contoh dari flip-bit attack, misalanya pada plain text seperti berikut.

"QT-TRNSFR US $00010,00 FRM ACCNT 123-67 TO"

Plain text di atas kemudian dienkripsi oleh algoritma stream cipher. Lalu keystream generator mengeluarkan runtutan kunci sebagai berikut.

"uhtr07hjLmkyR3j7**U**kdhj38lkkldkYtr#)oknTkRgh"



Gambar 10. Hasil Enkripsi Stream Cipher

Seperti pada gambar diatas, maka didapat hasil enkripsi sebagai berikut.

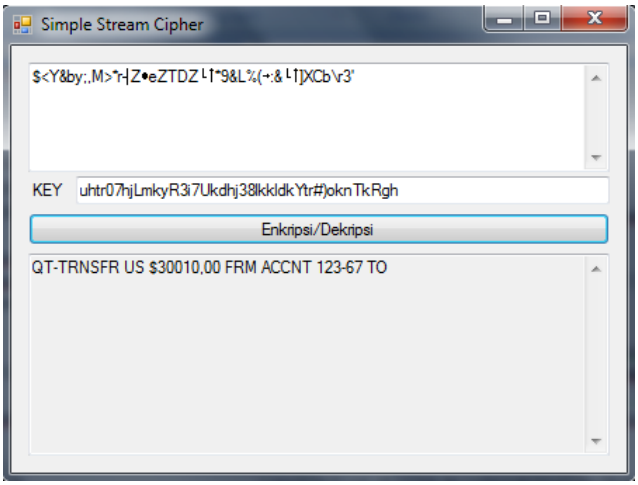
$<Y&by;,-M>*r Z eZTDZ *9&L%(:&]XC b\r3'

Hanya dengan mengetahui tata letak dari stream cipher tersebut, seorang kriptoanalis bisa meraba-raba letak tiap cipher text yang mempengaruhi plain text yang bersangkutan. Sebagai pengujian, karakter j pada key dibawah diubah menjadi i

"uhtr07hjLmkyR3**i**7Ukdhj38lkkldkYtr#)oknTkRgh"

menjadi

"uhtr07hjLmkyR3**i**7Ukdhj38lkkldkYtr#)oknTkRgh"



Gambar 11. Hasil Dekripsi yang telah Diserang oleh Flip-Bit Attack

Terlihat bahwa pesan dalam plain text yang menyampaikan nilai uang yang akan ditransfer berubah hingga hampir 3 kali lipatnya. Hal tersebut bisa sangat merugikan jika berhasil diaplikasikan pada suatu pesan terkait masalah kauangan seperti di atas.

II.VI Algebraic Attacks

Secara Umum algebraic attack memakai prinsip yaitu menggunakan suatu persamaan pada cipher manapun dengan bit dari key yang tidak diketahui. Berikutnya variabel dan konstanta yang diketahui akan diisi, lalu persamaan tersebut tinggal dipecahkan.

Namun metode ini bisa dibilang cukup rumit, karena persamaan yang muncul adalah persamaan non linear dengan derajat yang tinggi. Selain itu ada masalah lain yakni pencarian yang benar-benar bertumpu hanya pada cipher, serta penebakan keystream bits yang tidak trivial.

III. KESIMPULAN

- Stream cipher merupakan salah satu algoritma kriptografi modern yang cukup banyak digunakan dalam kehidupan sehari-hari.
- Stream cipher memiliki keunggulan dalam kecepatan dan kesederhanaannya dibanding block cipher.
- Streamkey generator harus selalu menghasilkan streamkey yang acak dan tidak pernah sama agar mencapai tingkat keamanan yang sempurna.

- Stream cipher bisa dipecahkan dengan berbagai metode, seperti exhaustive search, dictionary search, known-plaintext attack, known-ciphertext attack, bit-flip attack, dan lain sebagainya.

REFERENSI

http://en.wikipedia.org/wiki/Stream_cipher_attack
http://en.wikipedia.org/wiki/Stream_cipher
http://en.wikipedia.org/wiki/Vernam_cipher
http://www.goldamedia.com/miscellaneous/289-serangan-terhadap-sistem-stream-cipher.html
www.tcs.hut.fi/Studies/T-79.514/slides/S6.Kiviharju-alg.pdf
Munir, Rinaldi, "Slide mata kuliah Kriptografi : Algoritma Kriptografi Modern (Bab2) Baru".

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 22 Maret 2011

ttd

Muhamad Rizky Yanuar
13508015