Algoritmo de cifrado de flujo: TRIVIUM

MILITELLO, NICOLAS URIEL
41926805
nmilitello@alumno.unlam.edu.ar

RODRÍGUEZ, EZEQUIEL NICOLAS
40135570
ezeqrodriguez@alumno.unlam.edu.ar

PIZZO FEDERICO DANIEL
36697346
fpizzo@alumno.unlam.edu.ar

Resumen

El algoritmo de cifrado de flujo trivium, al igual que otros cifrados de flujo, se basa en generar una secuencia pseudoaleatoria que, junto a un vector de inicialización, se combina con la clave y de esta forma cifrar el mensaje que se desee. Es un algoritmo que fue presentado en la competencia eSTREAM al perfil II (Hardware), presenta una seguridad muy robusta y solo unos pocos ataques son capaces de romperlo.

Introducción

Este informe tiene la finalidad de presentar y detallar el funcionamiento del algoritmo de cifrado de flujo "Trivium", además de describir el nivel de seguridad que proporciona este algoritmo, sus usos y aplicaciones.

Trivium es un cifrador de flujo sincrónico, esto implica que la clave es generada de forma independiente del texto plano y del texto cifrado y utiliza una configuración de 288 bits para sus operaciones de cifrado y descifrado.

Algoritmos de cifrado de flujo.

Los cifradores de flujo, a diferencia de los cifradores de bloque, cifran bit a bit el mensaje. Estos utilizan la función XOR para cifrar los mensajes y utilizan una clave secreta compartida por el emisor y receptor junto a un algoritmo generador determinístico para producir una secuencia cifrante binaria y pseudoaleatoria. Los algoritmos de cifrado de flujo deben cumplir con los postulados de Golomb para asegurar la correcta generación de la secuencia pseudoaleatoria. Estos postulados indican que en la secuencia cifrante deberá existir igual número de ceros que de unos, se acepta como máximo una diferencia de uno entre unos y ceros; Deberá contener uno o más bits iguales entre dos bits distintos y, además, la autocorrelación AC(K) deberá ser constante para todo valor de desplazamiento de K bits.

Existen distintos algoritmos que permiten la generación de secuencias cifrante como por ejemplo los generadores de congruencia lineal y los generadores con registros de desplazamiento. Trivium utiliza un generador con registros de desplazamiento que a su vez que un generador lineal: LFSR.

El algoritmo Linear Feedback Shift Register (LFSR) utiliza registros de desplazamiento donde cada bit del extremo de la izquierda pasa a ocupar la primera posición desde la derecha y el resto de bits se corren un lugar hacia la izquierda. Este generador utiliza polinomios primitivos de n celdas y tendrá una complejidad lineal igual a n. Genera todos los estados lineales posibles del cuerpo de trabajo n, cumple con los postulados de Golomb y entrega una secuencia cifrante con periodo $T = 2^n - 1$. Sin embargo, es susceptible al ataque de Berlekamp-Massey, este ataque hace uso de la complejidad lineal de una secuencia binaria finita S^n de longitud n para medir la robustez de un cifrador de flujo.

Preparación del algoritmo

Trivium utiliza una clave de 80 bits, un vector de inicialización de 80 bits y una configuración interna de 288 para generar una secuencia pseudoaleatoria de hasta 2^{64} bits de salida.

La configuración interna se compone de 3 registros de desplazamiento debido a que usa el algoritmo LFSR para generar la difusión y dispersión de la clave, además de otras operaciones. El primer registro tiene un tamaño de 93 bits, el segundo de 84 bits y, el tercero y último, permite almacenar 111 bits.

La generación de la secuencia pseudoaleatoria se divide en tres fases: Fase de inicialización, Fase de mezcla y Fase de generación.

Fase de Inicialización

Se carga el primer registro de 93 bits con los 80 bits de la clave y se completan con 0 las 13 posiciones siguientes. Luego, se cargan los 80 bits del vector de inicialización en el segundo registro más 4 bits que se completan en 0. Por último, el tercer registro se completa todo con 0 a excepción de los últimos tres bits que se completan con 1.

Fase de mezcla:

La fase de mezcla comprende las primeras 1152 rondas del algoritmo que se utilizan para lograr la dispersión y difusión de la clave.

Fase de generación:

La fase de generación comienza en la ronda 1153 y hasta que sea necesario para lograr la secuencia pseudoaleatoria que son tantas rondas como bits tenga el mensaje a cifrar. Los pasos a realizar en esta fase son los mismos que la fase de mezcla con la diferencia que el bit de salida se utiliza para generar la secuencia pseudoaleatoria.

Cifrado

Luego de haber generado la secuencia pseudoaleatoria, se procede a realizar las operaciones de XOR bit a bit entre la secuencia pseudoaleatoria y el mensaje a cifrar para así lograr el texto cifrado: C = K ^ T

Descifrado

Debido a que la secuencia pseudoaleatoria no es completamente aleatoria, se necesita de la misma clave y del mismo IV para realizar el descifrado. El vector de inicialización es un dato público en este algoritmo, y además el destinatario debería estar en posesión de la clave secreta, por lo tanto, es completamente factible volver a generar la misma secuencia pseudoaleatoria y así descifrar el criptograma realizando la operación inversa: T = K ^ C

Cómo funciona TRIVIUM

Como ya fue explicado anteriormente, Trivium utiliza tres registros de 93, 84 y 111 bits respectivamente, una clave y un vector de inicialización (IV) de 80 bits cada uno y una secuencia cifrante de hasta 2^{64} bits.

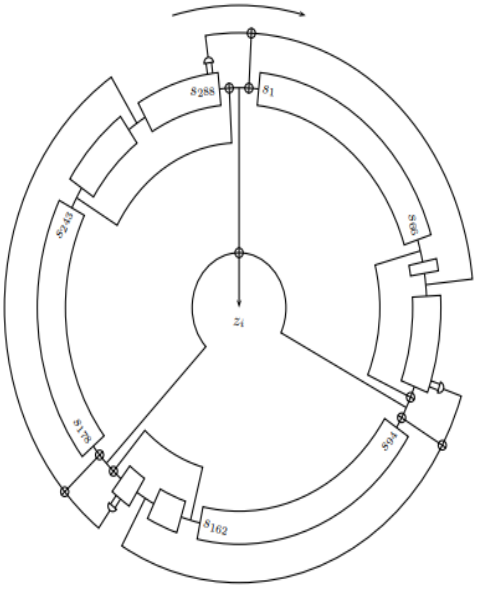

Figura 1. Representación gráfica del algoritmo Trivium.

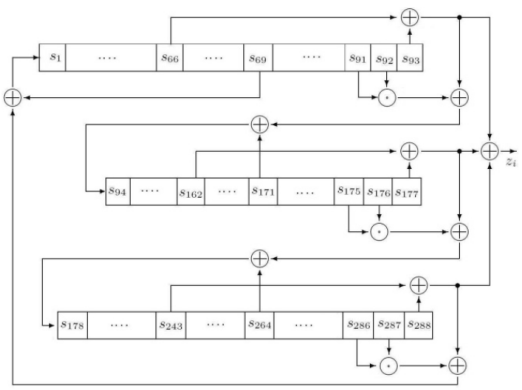

Figura 2. Otra representación gráfica del algoritmo Trivium.

Generación de la secuencia pseudoaleatoria:

Trivium tiene un estado interno de 288 bits (los bits están numerados de S_1 a S_{288}), hay tres registros de cambio de tamaño diferentes.

1) Copiar los 80 bits de la clave en los primeros 80 bits del primer registro, completar con 0 los 13 bits restantes
2) Copiar los 80 bits del IV en los primeros 80 bits del segundo registro y completar con 0 los 4 bits restantes.
3) Completar con 0 los primeros 108 bits del tercer registro y completar los últimos tres bits con 1.

4) Una vez inicializados los registros, el estado es rotado 4 ciclos completos, es decir 4 * 288 = 1152, siendo 288 el total de bits de los tres registros que conforman la configuración interna del algoritmo.

```
for i = 1 to 4 * 288 do:
   t1 ← s66+s91^s92+s93+s171
   t2 ← s162+s175^s176+s177+s264
   t3 ← s243+s286^s287+s288+s69
   (s1,s2,...,s93)←(t3,s1,...,
s92)
   (s94,s95,..,s177)←(t1,s94,..
.,s176)
   (s178,s279,...,s288)←(t2,s17
8,...,s287)
end for
```

Figura 3. Configuración del estado del algoritmo.

5) Luego de haberse obtenido un fuerte nivel de dispersión y distorsión de la clave, comienza el proceso de generación de la secuencia pseudoaleatoria. Siendo N el tamaño del mensaje original:

```
for i = 1 to N do
      t1←s66+s93
      t2←s162+s177
      t3←s243+s288
      zi←t1+t2+t3
      t1←t1+s91^s92+s171
      t2←t2+s175^s176+s264
      t3←t3+s286^s287+s69
      (s1,s2,...,s93)←(t3,s1,...
, s92)
      (s94,s95,..,s177)←(t1,s94,
...,s176)
      (s178,s279,...,s288)←(t2,s
178,...,s287)
end for
```

Figura 4. Generación de la secuencia cifrante.

6) Teniendo finalmente la secuencia cifrante se procede al cifrado del mensaje utilizando la función XOR y calculando su resultado entre la secuencia pseudoaleatoria y el mensaje original.

Seguridad

Hasta la fecha, no se ha descubierto ninguna vulnerabilidad crítica en Trivium que ponga en peligro su seguridad. A continuación, se explican posibles ataques que se consideran teóricos y requieren condiciones específicas para tener éxito:

Ataque de flujo conocido:

Este ataque ocurre cuando el atacante conoce parte del flujo de salida generado por Trivium. Utilizando esta información, el atacante puede intentar deducir partes de la clave secreta. Si se revela una cantidad suficiente de información del flujo de salida, es posible realizar un análisis algebraico para recuperar la clave completa. Sin embargo, este tipo de ataque requiere una gran cantidad de flujo conocido para ser efectivo y, en general, se considera poco probable en escenarios reales.

Ataque de texto claro conocido:

En este ataque, el atacante conoce el texto plano correspondiente a una parte del flujo de salida cifrado. Al analizar las diferencias entre los flujos de salida cifrados, es posible extraer información sobre la clave secreta. Sin embargo, este tipo de ataque también requiere una cantidad significativa de texto claro conocido y es poco probable en situaciones prácticas.

Ataque de inyección de flujo:

En este tipo de ataque, el atacante intenta inyectar flujo adicional en el cifrador para alterar su comportamiento. Si el atacante puede controlar o manipular parte del flujo interno utilizado por Trivium, puede comprometer la seguridad del cifrado. Sin embargo, este tipo de ataque es altamente sofisticado y requeriría un conocimiento profundo del diseño y la implementación del cifrador.

Vulnerabilidades/Ataques conocidos

A partir de abril de 2015, no se conocen ataques criptoanalíticos mejores que los ataques de fuerza bruta, pero varios ataques se acercan:

El **ataque del cubo** requiere 268 pasos para romper una variante de Trivium donde el número de rondas de inicialización se reduce a 799. Otro tipo de ataque recupera el estado interno (y por lo tanto la clave) del cifrado completo en alrededor de 289,5 pasos (donde cada paso es aproximadamente el costo de una sola prueba en una búsqueda exhaustiva). En aquellas variantes reducidas del cifrador que utilizan los mismos principios de diseño se han roto mediante una técnica de resolución de ecuaciones. Estos ataques mejoran el conocido ataque de compensación de espacio-tiempo en cifrados de flujo, que con el estado interno de 288 bits de Trivium tomaría 2144 pasos, y además se demuestra que una variante de Trivium que no hizo ningún cambio excepto aumentar la longitud de la clave más allá de los 80 los bits, exigidos por eSTREAM perfil II, no serían seguros.

Proyecto eSTREAM

Trivium fue uno de los candidatos seleccionados para el perfil de cifrador de flujo en el proyecto eSTREAM, una iniciativa de la Unión Europea para evaluar y seleccionar nuevos algoritmos de cifrado de flujo. En este proceso de competencia, Trivium compitió con otros candidatos, como Grain, HC-128, Rabbit, Salsa20 y SOSEMANUK, entre otros.

Trivium fue seleccionado como uno de los cinco cifradores finales en el perfil de cifrador de flujo en el proyecto eSTREAM debido a varias razones:

Seguridad: Trivium demostró una resistencia sustancial a diversos ataques criptográficos conocidos. Durante el proceso de evaluación, no se descubrieron debilidades graves que pusieran en peligro su seguridad.

Rendimiento: se destacó por su eficiencia en términos de velocidad de cifrado y descifrado. Proporcionó un rendimiento satisfactorio en una amplia gama de plataformas, incluidos dispositivos con recursos limitados, como tarjetas inteligentes.

Diseño sencillo: se caracteriza por su diseño relativamente simple y elegante. Utiliza tres registros internos de tamaño fijo y una función de retroalimentación no lineal para generar el flujo de salida, lo que facilita su implementación y análisis criptográfico.

Flexibilidad: Trivium es un cifrador versátil que puede adaptarse a diferentes requisitos de seguridad y aplicaciones. Su longitud de clave y vector de inicialización proporcionan un espacio adecuado para aplicaciones prácticas.

Candidate	Key Size (bits)	Max IV Size (bits)	Data radix d (bits/cycle)	Separate key register	Separate IV register
DECIM v2	80	64	0.25	no	yes
DECIM 128	128	128	0.25	no	no
Edon80	80	64	1	yes	yes
F-FCSR-H v2	80	80	8	no	no
F-FCSR-16	128	128	16	no	no
Grain v1	80	64	1	no	no
Grain 128	128	96	1	no	no
MICKEY 2.0	80	80	1	no	no
MICKEY 128 2.0	128	128	1	no	no
Moustique	96	104	1	yes	no
Pomaranch	80 /128	108 /162	1	yes	yes
Trivium	80	80	1	no	no

Tabla 1. Candidatos de hardware eSTREAM

Conclusión

El algoritmo trivium, a pesar de estar diseñado para implementarse sobre hardware, es igualmente efectivo cuando se lo implementa en software. Sumado a esto, debido a ser un algoritmo de cifrado de flujo y los detalles de su diseño explicados anteriormente, los procesos de cifrado y descifrado conllevan una velocidad relativamente elevada.

Desde el punto de vista de seguridad, este cifrado ha resistido una enorme cantidad de ataques, y los que no ha resistido se produjeron con condiciones específicas. Como ejemplos, se puede señalar dos ataques de los años 2015 y 2022, específicamente Key Recovery Attacks; los cuales requirieron una complejidad de aproximadamente 2^62 optimizado para 799 rondas y el segundo ataque, con complejidad 2^53 para 820 rondas. Sin embargo, el uso normal y recomendado para Trivium es de 1152 rondas. De esta forma dejando por eliminación a los ataques por fuerza bruta como la única forma de ataque efectiva contra la versión completa de Trivium (1152 rondas). No es invencible, pero cumple con el principio que el descifrado de un mensaje entre un emisor y receptor por parte de un tercero requiera demasiados recursos y tiempo tal que no vería una ganancia neta positiva.

Repositorio de código

https://github.com/MilitelloN/Trivium

Referencias

[1] https://en.wikipedia.org/wiki/Trivium_(cipher)

[2]https://www.ecrypt.eu.org/stream/p3ciphers/trivium/trivium_p3.pdf

[3]" The Stream Cipher Trivium Explained" - https://www.youtube.com/watch?v=YCnUKCki_rg&list=WL&index=5

[4]https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642006000300023

[5]https://www.ecrypt.eu.org/stream/papersdir/2007/003.pdf

[6]https://www.ecrypt.eu.org/stream/e2-trivium.html