(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 777 930**

(51) Int. CI.:

H04L 9/18 (2006.01)
H04L 9/28 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **30.04.2004 E 13169371 (5)**

(97) Fecha y número de publicación de la concesión europea: **18.12.2019 EP 2663019**

(54) Título: **Método y sistema de criptoanálisis GSM**

(30) Prioridad:

30.04.2003 IL 15567103

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
06.08.2020

(73) Titular/es:

BARKAN, ELAD (100.0%)
12 Habanim Street
Kfar-Sirkin 49935, IL

(72) Inventor/es:

BARKAN, ELAD

(74) Agente/Representante:

ELZABURU, S.L.P

DESCRIPCIÓN

Método y sistema de criptoanálisis GSM

Campo técnico

La presente invención se refiere a dispositivos y métodos de criptoanálisis, y más particularmente a criptoanálisis solamente de texto cifrado de comunicaciones encriptadas GSM recibidas fuera del aire.

Antecedentes de la invención

Esta sección detalla la necesidad de la presente invención, los métodos de criptoanálisis de la técnica anterior y el método de cifrado utilizado ahora en GSM.

GSM es el método más ampliamente extendido de comunicaciones celulares. Incluye una medida de protección de datos mediante cifrado, que a veces puede ser deseable descifrar.

Por ejemplo, las agencias de aplicación de la ley, tales como la policía, pueden desear escuchar las comunicaciones celulares, sin una conexión física a la infraestructura celular. Este proceso a menudo requiere permiso del tribunal, y a veces es denominado interceptación legal.

Los clientes tienen una sensación de seguridad cuando utilizan el teléfono móvil, lo que a veces no está justificado. Los espías pueden escuchar una conversación, jaquear una llamada o hacer llamadas telefónicas a expensas del usuario. Puede ser deseable probar el nivel de seguridad del sistema realizando intentos de atacar el sistema. Así se puede evaluar el nivel real de seguridad de la red. Tales pruebas pueden ser realizadas por el proveedor de la red celular, por entidades locales de apoyo o agencias de protección al cliente.

Lo anterior, así como otras aplicaciones, requieren la realización de criptoanálisis en tiempo real, en un período de tiempo corto y utilizando una cantidad razonable de memoria digital, tal como no se ha conseguido en la técnica anterior.

GSM es la tecnología celular más ampliamente utilizada. Para diciembre de 2002, más de 787,5 millones de clientes GSM en más de 191 países formaban aproximadamente el 71% del total del mercado inalámbrico digital. GSM incorpora mecanismos de seguridad. Los operadores de red y sus clientes confían en estos mecanismos para la privacidad de sus llamadas y para la integridad de la red celular. Los mecanismos de seguridad protegen la red autenticando a los clientes en la red, y proporcionan privacidad a los clientes cifrando las conversaciones mientras son transmitidas por el aire.

GSM utiliza el cifrado para proteger las señales transmitidas. Hay dos métodos básicos en uso ahora, A5/1 y A5/2, el primero es utilizado principalmente en el Medio Oriente y el segundo generalmente para el resto del mundo. El A5/1 es más difícil de descifrar sin un conocimiento previo de la clave que ha sido utilizada.

Por lo tanto, para escuchar transmisiones GSM, es necesario descifrar los mensajes. El salto de frecuencia en GSM hace que el problema sea aún más difícil.

Hay tres tipos principales de algoritmos criptográficos utilizados en GSM: A5 es un algoritmo de cifrado de secuencia utilizado para el cifrado, A3 es un algoritmo de autenticación y A8 es el algoritmo de acuerdo clave. El diseño de A3 y A8 no se ha especificado en las especificaciones de GSM, solo se ha especificado la interfaz externa de estos algoritmos. El diseño exacto del algoritmo puede ser seleccionado por los operadores de forma independiente. Sin embargo, muchos operadores utilizaron el ejemplo, llamado COMP128, que figura en el memorando de entendimiento GSM (MoU).

Los métodos de criptoanálisis de la técnica anterior plantean demandas poco realistas, tales como unos pocos minutos de conversación conocida sobre los bits, véase la lista de referencias a continuación.

Briceno, Goldberg, y Wagner han realizado un criptoanálisis del COMP128 encontrado, que permite encontrar la clave (maestra) compartida del móvil y la red, permitiendo así la clonación. La descripción del algoritmo A5 es parte de las especificaciones de GSM, pero nunca se hizo pública. Hay dos versiones utilizadas actualmente de A5: A5/1 y A5/2. A5/1 es la versión "fuerte" de exportación limitada. A5/2 es la versión que no tiene limitaciones de exportación, sin embargo, es considerada la versión "débil".

El diseño exacto tanto de A5/1 como de A5/2 fue diseñado por ingeniería inversa por Briceno a partir de un teléfono GSM real en 1999 y se verificó contra los vectores de prueba conocidos. Una nueva versión adicional, que está estandarizada pero que no se ha utilizado aún en redes GSM es A5/3. Fue elegida recientemente y está basada en el cifrado de bloque KASUMI.

GPRS (Servicio General de Paquetes por Radio) es un nuevo servicio para las redes GSM que ofrece servicios de datos basados en paquetes y contenido basado en Internet, de mayor capacidad, 'siempre encendidos', habilita servicios tales como navegación en color por Internet, correo electrónico en movimiento, potentes comunicaciones visuales, mensajes multimedia y servicios basados en la ubicación. GPRS utiliza su propio cifrado, sin embargo, la clave para el cifrado GPRS es creada por el mismo algoritmo A3A8 en la tarjeta SIM del abonado, utilizando el mismo K_i que es utilizado para crear claves de cifrado para A5/1, A5/2 y A5/3. Utilizaremos este hecho para atacarlo más tarde. A5/1 fue inicialmente

criptoanalizado por Golic, y luego por: Biryukov, Shamir y Wagner, Biham y Dunkelman, y recientemente por Ekdahl y Johansson.

Después de que el A5/2 fue diseñado por ingeniería inversa, fue criptoanalizado inmediatamente por Goldberg, Wagner y Green. Su ataque es un ataque de texto sin formato conocido que requiere la diferencia en el texto sin formato de dos tramas GSM, que están separadas exactamente por 2^{11} tramas (con una separación de aproximadamente 6 segundos). La complejidad temporal media de este ataque es de aproximadamente 2^{16} productos de punto de vectores de 114 bits.

Aparentemente, este ataque no es aplicable (o falla) en aproximadamente la mitad de los casos, ya que en la primera trama necesita que el 11º bit de R4 sea cero después de la inicialización del cifrado. Un trabajo posterior de Petrovic y Fúster-Sabater sugiere tratar el estado interno inicial del cifrado como variables, escribir cada bit de salida del algoritmo A5/2 como una función cuadrática de estas variables, y linealizar los términos cuadráticos. Mostraron que la salida de A5/2 puede ser predicha con una probabilidad extremadamente alta después de unos pocos cientos de bits de salida conocidos. Sin embargo, este ataque no descubre la clave de sesión de A5/2 (Kc).

Por lo tanto, no es posible utilizar este ataque como un bloque de construcción para ataques más avanzados, como los que presentaremos más adelante. La complejidad temporal de este resultado posterior es proporcional a 2^{17} eliminaciones de Gauss de matrices de tamaño (estimado) de aproximadamente 400 x 719.

Goldberg, Wagner y Green presentaron el primer ataque contra A5/2. La complejidad temporal de este ataque es muy baja. Sin embargo, requiere el conocimiento de XOR de los textos sin formato en dos tramas que están separadas por 2^{11} tramas. Su ataque muestra que el cifrado es bastante débil, aunque podría resultar difícil implementar tal ataque en la práctica. El problema es conocer la XOR exacta de los textos sin formato en dos tramas que están separadas por 6 segundos.

Otro aspecto es el tiempo transcurrido desde el comienzo del ataque hasta su finalización. Su ataque lleva al menos 6 segundos, porque lleva 6 segundos completar la recepción de los datos. El nuevo método descrito en la presente solicitud mejora en gran medida la velocidad del ataque.

El ataque de texto sin formato conocido de Petrovic y Fúster-Sabater tiene requisitos de datos similares a los de nuestro ataque, sin embargo, no recupera la clave de sesión (Kc) y, por lo tanto, puede no ser adecuado para los ataques activos que describimos más adelante.

El estado de la técnica anterior puede ser revisado en las siguientes referencias:

1. Una implementación pedagógica (en lenguaje de programación C) de A5/1 y A5/2:

Marc Briceno, Ian Goldberg, David Wagner, *A pedagogical implementation of the GSM A5/1 y A5/2 "voice privacy encryption algorithms,*http://cryptome.org/gsm-a512.htm (Originalmente en www.scard.org), 1999.

2. Descripción y criptoanálisis de COMP128, utilizado por muchos operadores GSM como A3A8:

Marc Briceno, Ian Goldberg, David Wagner, *An implementation of the GSM A3A8 algorithm,* http://www.iol.ie/kooltek/a3a8.txt, 1998.

Marc Briceno, Ian Goldberg, David Wagner, *GSM Cloning*, http://www.isaac.cs.berkeley.edu/isaac/gsm-faq.html, 1998.

3. Criptoanálisis de Texto Sin Formato Conocido de A5/1:

Eli Biham, Orr Dunkelman, *Cryptoanalysis of the A5/1 GSM Stream* Ciipher, Progress in Cryptology, proceedings of Indocrypt'00, Lecture Notes in Computer Science 1977, Springer-Verlag, págs. 43-51, 2000.

Alex Biryukov, Adi Shamir, Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers, Advances in Cryptology, proceedings of Asiacrypt'00, Lecture Notes in Computer Science 1976, Springer-Verlag, págs. 1-13, 2000.

Alex Biryukov, Adi Shamir, David Wagner, *Real Time Cryptanalysis of A5/1 on a PC*, Advances in Cryptology, proceedings of Fast Software Encryption'00, Lecture Notes in Computer Science 1978, Springer-Verlag, págs. 1-18, 2001.

Patrik Ekdahl, Thomas Johansson, *Another Attack on A5/1*, que será publicado en IEEE Transactions on Information Theory, http://www.it.lth.se/patrik/publications.html, 2002.

Jovan Golic, *Cryptanalysis of Alleged A5 Stream Cipher*, Advances in Cryptology, proceedings of Eurocrypt'97, LNCS 1233, págs. 239-255, Springer-Verlag, 1997.

4. Información relacionada con A5/2:

Ian Goldberg, David Wagner, Lucky Green, *The (Real-Time) Cryptanalysis of A5/2*, presentado en la Rump Session de Crypto'99, 1999.

Grupo de Expertos en Algoritmos de Seguridad (SAGE), *Report on the specification and evaluation of the GSM cipher algrithm A5/2*, http://cryptome.org/espy/ETR278e01p.pdf, 1996.

Slobodan Petrovic, Amparo Fúster-Sabater, *Cryptanalysis of the A5/2 Algorithm*, Cryptology ePrint Archive, Report 2000/052, Disponible en línea en http://eprint.iacr.org, 2000.

Descripción de los Antecedentes de Seguridad A5/2 y GSM

En esta sección describimos la estructura interna de A5/2 y la forma en que es utilizada, véase la Figura 4. A5/2 consta de 4 LFSR de longitud máxima: R1, R2, R3, y R4. Estos registros tienen una longitud de 19 bits, 22 bits, 23 bits, y 17 bits, respectivamente. Cada registro tiene varios "tap" ("punto de acceso terminal") y una función de retroalimentación. Sus polinomios irreducibles son: $x^{19} \oplus x^5 \oplus x^2 \oplus x \oplus 1$, $x^{22} \oplus x \oplus 1$, $x^{23} \oplus x^{15} \oplus x^2 \oplus x \oplus 1$ y $x^{17} \oplus x^5 \oplus 1$, respectivamente.

Téngase en cuenta que damos los bits en los registros en orden inverso, es decir, en nuestro esquema de numeración, x^1 corresponde a un tap en el índice *len-i-1*, donde *len* es la longitud absoluta del registro. Por ejemplo, cuando R4 está sincronizado, se calcula el XOR de R4 [17-0-1 = 16] y R4 [17-5-1 = 11]. Luego, el registro es desplazado un lugar hacia la derecha, y el valor del XOR es colocado en R4[0].

En cada operación de A5/2 los registros R1, R2 y R3 están sincronizados de acuerdo con un mecanismo de sincronización que se ha descrito más adelante. Luego, el registro R4 está sincronizado. Después de realizar la sincronización, un bit de salida está listo en la salida de A5/2. El bit de salida es una función no lineal del estado interno de R1, R2, y R3.

Después de la inicialización, 99 bits de salida son descartados, y los siguientes 228 bits de salida son utilizados como secuencia de claves de salida. Algunas referencias indican que A5/2 descarta 100 bits de salida, y que la salida es utilizada con un retraso de un bit. Esto es equivalente a afirmar que descarta 99 bits de salida, y que la salida es utilizada sin demora.

Señalar $K_c[i]$ como el i-ésimo bit de la clave de sesión de 64 bits K_c, Rj[i] el i-ésimo bit de registro j, y *f*[i] el i-ésimo bit del número de trama públicamente conocido de 22 bits.

La generación de secuencia de claves es como sigue:

1. Inicializar con K_c y número de trama.

2. Forzar los bits R1[15], R2[16], R3[18], R4[10] para que sean 1.

3. Ejecutar A5/2 para 99 relojes e ignorar la salida.

4. Ejecutar A5/2 para 228 relojes y utilizar la salida como secuencia de claves.

El primer bit de salida es definido como el bit que está en la salida después de que se realizó la primera sincronización.

La inicialización es realizada de la siguiente manera:

- Establecer todos los LFSR en 0 (R1 = R2 = R3 = R4 = 0).
- Para i: = 0 a 63 do

1. Sincronizar los 4 LFSR.

2. $R1[0] \leftarrow R1[0] \oplus K_c[i]$

3. $R2[0] \leftarrow R2[0] \oplus K_c[i]$

4. $R3[0] \leftarrow R3[0] \oplus K_c[i]$

5. $R4[0] \leftarrow R4[0] \oplus K_c[i]$

- Para i: = 0 a 21 do

1. Sincronizar los 4 LFSR.

2. $R1[0] \leftarrow R1[0] \oplus f[i]$

3. $R2[0] \leftarrow R2[0] \oplus f[i]$

4. $R3[0] \leftarrow R3[0] \oplus f[i]$

5. $R4[0] \leftarrow R4[0] \oplus f[i]$

En la Figura 4 se ha mostrado la estructura interna del algoritmo A5/2.

El mecanismo de sincronización funciona como sigue: el registro R4 controla la sincronización de los registros R1, R2 y R3. Cuando se va a realizar la sincronización de R1, R2 y R3, los bits R4[3], R4[7], y R4[10] son la entrada de la unidad de sincronización. La unidad de sincronización realiza una función mayoritaria en los bits. R1 está sincronizado si y solo si R4[10] está de acuerdo con la mayoría. R2 está sincronizado si y solo si R4[3] está de acuerdo con la mayoría. R3 está sincronizado si y solo si R4[7] está de acuerdo con la mayoría. Después de estas sincronizaciones, R4 está sincronizado.

Una vez que se realizó la sincronización, un bit de salida está listo. El bit de salida es calculado como sigue:

salida = R1[18]⊕maj(R1[12],R1[14]⊕1,R1[15])⊕R2[21]⊕maj(R2[9],R2[13] R2[16]⊕1)⊕R3[22]⊕maj (R3[13]⊕1,R3[16], R3[18]), donde maj (·,·,·) es la función de mayoría, es decir, fuera de cada registro, hay 3 bits cuya mayoría es cifrada en XOR para formar la salida (cuando un bit de cada triplete es invertido), además del último bit de cada registro. Téngase en cuenta que la función mayoritaria es cuadrática en su entrada: maj(a,b,c) = a·b⊕b·c⊕c·a.

A5/2 se basa en una infraestructura algo similar a A5/1. Las funciones de retroalimentación de R1, R2 y R3 son las mismas que las funciones de retroalimentación de A5/1. El proceso de inicialización de A5/2 también es algo similar al de A5/1. La diferencia es que A5/2 también inicializa R4, y que después de la inicialización, un bit en cada registro es forzado para que sea 1. Entonces A5/2 descarta 99 bits de salida mientras que A5/1 descarta 100 bits de salida. El mecanismo de sincronización es el mismo, pero los bits de entrada al mecanismo de sincronización son de R4 en el caso de A5/2, mientras que en A5/1 son de R1, R2, y R3. Los diseñadores tenían la intención de usar bloques de construcción similares para ahorrar hardware en el móvil.

Este algoritmo genera 228 bits de secuencia de claves. El primer bloque de 114 bits es utilizado como una secuencia de claves para cifrar el enlace de la red al cliente, y el segundo bloque de 114 bits es utilizado para cifrar el enlace del cliente a la red. El cifrado es realizado como un XOR simple del mensaje con la secuencia de claves.

Aunque A5 es un cifrado de secuencia, es utilizado para cifrar "bloques" de 114 bits. Cada uno de tales bloques es la carga útil de una ráfaga GSM, que es una unidad de datos de interfaz de aire GSM. Téngase en cuenta que cada trama está construida con 8 ráfagas consecutivas, que atienden a 8 clientes en paralelo. A cada cliente le es asignado un índice de ráfaga. Todas las ráfagas en este índice están designadas para ese cliente. Las tramas están numeradas secuencialmente, y cada trama tiene asociado un número de trama conocido públicamente de 22 bits con ella. Este número de trama es utilizado al inicializar A5. Dado que el enfoque está siempre en un solo cliente, utilizamos los términos "ráfaga" y "trama" indistintamente.

Uno podría preguntarse por qué GSM utiliza un cifrado de secuencia y no un cifrado de bloque de tamaño de bloque de 114 bits. Una explicación posible es que GSM realiza la corrección de errores y luego el cifrado. Supóngase que un bit en un bloque es dado la vuelta debido a un error. Descifrar ese bloque con un cifrado de bloque daría como resultado un bloque que parecería aleatorio, y que los códigos de corrección de errores no tienen posibilidad de corrección. Sin embargo, cuando se utiliza un cifrado de secuencia, un bit dado la vuelta causa exactamente un bit dado la vuelta después del descifrado.

Antecedentes de seguridad GSM

La siguiente es una descripción más detallada sobre la utilización y la especificación de los algoritmos A3 y A8.

A3 proporciona autenticación del móvil a la red, y A8 es utilizado para el acuerdo de clave de sesión. La seguridad de estos algoritmos está basada en una clave secreta específica del usuario K_i que es común para el móvil y la red. Las especificaciones GSM no especifican la longitud de K_i, por lo tanto, es dejada a elección del operador, pero habitualmente es una clave de 128 bits. La autenticación de los clientes a la red es realizada utilizando el algoritmo de autenticación A3 como sigue: La red desafía al cliente con un valor elegido aleatoriamente de 128 bits RAND. El cliente calcula una respuesta de 32 bits de longitud SRES = A3 (K_i, RAND) y envía SRES a la red, que luego puede verificar su validez.

La clave de sesión K_c es obtenida mediante el algoritmo A8 como sigue: K_c = A8 (K_i, RAND). Téngase en cuenta que A8 y A3 siempre son invocados juntos y con los mismos parámetros. En la mayoría de las implementaciones, hay un algoritmo con dos salidas, SRES y K_c. Por lo tanto, son denominados habitualmente como A3A8.

La descripción anterior del cifrado de la técnica anterior en GSM es transmitida en la descripción detallada de la invención a continuación.

En esta invención, el término criptoanálisis es utilizado para describir el proceso de ser capaz de cifrar/descifrar la comunicación sin el conocimiento previo de la clave de sesión utilizada. En algunos casos, el criptoanálisis puede recuperar la clave de sesión que es utilizada. En otros casos, la clave de sesión no es recuperada, sin embargo, aún podría ser posible descifrar o cifrar mensajes de la misma manera que hubiera sido si se hubiera utilizado el cifrado relevante que utiliza la clave de sesión. A veces, en esta invención, el término descifrado también es utilizado en el significado de criptoanálisis.

El texto sin formato conocido significa que el atacante tiene acceso a los mensajes cifrados, así como a los mensajes que fueron cifrados.

El texto cifrado solo significa que el atacante solo tiene acceso a los mensajes cifrados, y no tiene acceso a los mensajes antes de que fueran cifrados.

En esta invención, el término teléfono debería entenderse en el sentido más amplio de un dispositivo celular que utiliza la red GSM.

Sumario de la invención

De acuerdo con la presente invención, se ha proporcionado un método y un sistema para realizar un criptoanálisis efectivo de comunicaciones cifradas GSM. El método utiliza criptoanálisis solamente de texto cifrado. El sistema no necesita estar conectado por cable a la infraestructura celular, sino que puede recibir mensajes transmitidos por el aire.

Se han descrito nuevos métodos para atacar el cifrado GSM y los protocolos de seguridad. Estos métodos son mucho más fáciles de aplicar y mucho más rápidos.

Básicamente, para A5/2 GSM, un sistema atacante móvil recibe los mensajes cifrados, realiza un criptoanálisis eficiente y permite escuchar los mensajes GSM y/o revisar la información relacionada. Cuando se realiza en un ordenador personal, el proceso puede tomar menos de un segundo.

En principio, se puede aplicar un método similar al GSM A5/1, sin embargo, en este caso, el cifrado es más complejo y puede requerir unos 5 minutos de mensajes de comunicación para descifrar. Es posible que se requiera un sistema complejo, que puede ser difícil de implementar, ya que debe realizar un seguimiento del salto de frecuencia en GSM.

De acuerdo con otro aspecto de la presente invención, para GSM A5/1, el sistema atacante crea una pequeña célula alrededor de sí mismo, cuya célula incluye el teléfono GSM objetivo. El sistema suplanta la red celular para el teléfono objetivo, y el teléfono objetivo para la infraestructura GSM. Esto requiere una capacidad de transmisión en el sistema atacante, sin embargo, el descifrado es simplificado en gran medida y es mucho más rápido.

Además, se han presentado nuevas mejoras en las redes GSM. Estas incluyen mejoras en los algoritmos y protocolos criptográficos. Tales mejoras pueden ser realizadas, por ejemplo, por operadores GSM.

Incluso las redes GSM que utilizan el nuevo A5/3 sucumben a nuestro ataque, en la forma en que A5/3 está integrado en GSM. La presente divulgación incluye cambios en la forma en que A5/3 es integrado para proteger las redes de tales ataques.

Realizando tales pruebas o ataques en la red celular, se puede conseguir y mantener un mayor nivel de seguridad. Se pueden detectar puntos débiles presentes y futuros y se pueden tomar acciones correctivas. La estructura de la red GSM en sí misma se puede así ser mejorada para aumentar su seguridad.

La presente invención podría no estar limitada a la red celular GSM: por ejemplo, una versión similar de A5/3 también es utilizada en redes celulares de tercera generación.

Otros objetos, ventajas y otras características de la presente invención serán evidentes para los expertos en la técnica al leer la descripción expuesta a continuación.

Breve descripción de los dibujos

La Figura 1 ilustra una célula GSM con una estación base, un abonado y un sistema atacante.

La Figura 2 detalla un diagrama de bloques del sistema atacante.

La Figura 3 detalla un diagrama de bloques de otra realización del sistema atacante.

La Figura 4 detalla el diseño interno A5/2 (técnica anterior).

La Figura 5 detalla un método para el ataque solamente de texto cifrado.

La Figura 6 detalla un Ataque de Texto Sin Formato Conocido en el método A5/2.

Descripción detallada de la invención

Se describirá ahora una realización preferida de la presente invención a modo de ejemplo y con referencia a los dibujos adjuntos.

La Figura 1 ilustra una célula GSM 11 con una estación base 12, un abonado 13 y un sistema atacante 14. Hay enlaces inalámbricos 21, 22, 23 entre estas unidades.

La Figura 2 detalla un diagrama de bloques del sistema atacante. El sistema puede ser utilizado para implementar los métodos detallados en la presente descripción. El sistema atacante comprende un primer transceptor 31 con una antena 32, que se comunica con un conjunto de abonado objetivo, y un segundo transceptor 33 con una antena 34, que se comunica con una estación base. El sistema también incluye un ordenador/controlador 36, que controla el funcionamiento del sistema, es controlado por el operador y muestra los resultados del descifrado. El ordenador 36 también permite al operador escuchar las comunicaciones del teléfono objetivo.

La Figura 3 detalla un diagrama de bloques de otra realización del sistema atacante. Incluye un primer transceptor 31 que está en una ubicación diferente que el transceptor 33 - el primero está ubicado cerca de un abonado objetivo, el último - cerca de la estación base.

El sistema incluye además un medio 38 de interfaz, que permite colocar el primer transceptor 31 en una ubicación remota.

Alternativamente, el sistema puede utilizar antenas direccionales dirigidas cada una hacia un abonado o hacia la estación base, respectivamente.

Aunque los ejemplos aquí se refieren en su mayoría a GSM A5/2, A5/1, A5/3 y GPRS, pueden estar adaptados también a otras redes, utilizando la presente invención.

Los ejemplos en la presente descripción detallan un criptoanálisis solamente de texto cifrado de comunicación cifrada GSM. Los ataques funcionan en redes GSM que emplean, por ejemplo, A5/1 o A5/2 e incluso el recién elegido A5/3.

El ataque a A5/2 requiere aproximadamente 40 milisegundos de conversación celular cifrada fuera del aire y encuentra la clave correcta en menos de un segundo en un ordenador personal. Se ha mostrado cómo aprovechar fácilmente nuestro ataque contra A5/2 para ataques activos contra redes que utilizan A5/1 o A5/3. Los ataques anteriores a GSM requerían información irreal, como largos períodos de texto sin formato conocidos. Nuestros ataques son los primeros ataques prácticos en redes GSM y no requieren conocimiento acerca del contenido de la conversación.

Estos ataques permiten a los atacantes aprovechar cualquier conversación y descifrarla o bien en tiempo real o bien en cualquier momento posterior. También mostramos cómo montar ataques activos, como el jaqueo de llamadas, la alteración de mensajes de datos y el robo de llamadas. Incluso cuando tales ataques activos son aplicados, no pueden ser identificados por el operador de red utilizando métodos y sistemas de la técnica anterior.

El A5/3 también es utilizado en redes celulares de tercera generación, por lo tanto, la presente invención no está limitada a GSM, sino que también puede ser utilizada con otros sistemas celulares.

La presente descripción ilustra un método para montar un ataque solamente de texto cifrado en A5/2. En las pruebas que realizamos, nuestro ataque encontró la clave en menos de un segundo en un ordenador personal. Se ha mostrado que el ataque que proponemos en A5/2 puede ser aprovechado para montar un ataque activo incluso en redes GSM que utilizan A5/1 y A5/3, realizando así un ataque activo en tiempo real en redes GSM, sin ningún conocimiento previo requerido.

Método para atacar solamente texto cifrado

El nuevo método de ataque completo comprende, véase por ejemplo la Figura 5:

1. Un ataque de texto sin formato conocido eficiente en A5/2 que recupera la clave de sesión. Este primer ataque es de naturaleza algebraica. Aprovecha el bajo orden algebraico de la función de salida A5/2. Representamos la salida de A5/2 como una función multivariante cuadrática en el estado inicial de los registros. Luego, construimos un sistema sobredefinido de ecuaciones cuadráticas que expresa el proceso de generación de secuencia de claves y resolvemos las ecuaciones.

2. Mejorar el ataque de texto sin formato conocido a un ataque solamente de texto cifrado en A5/2. Observamos que GSM emplea códigos de Corrección de Errores antes del cifrado. Mostramos cómo adaptar el ataque a un ataque solamente de texto cifrado en A5/2 utilizando esta observación.

3. Aprovechamiento de un ataque en A5/2 a un ataque activo en redes GSM A5/1 y A5/3, y también a GPRS. El presente inventor ha encontrado que, debido al diseño de la interfaz de los módulos de seguridad GSM, la clave que es utilizada en A5/2 es la misma clave que en A5/1 y A5/3. Y el mismo mecanismo que establece la clave en el cifrado A5, es decir, A3A8 es utilizada para establecer la clave para GPRS. Se ha mostrado cómo montar un ataque activo en cualquier red GSM.

Fin del método.

Nota: Véase la descripción de los Antecedentes de Seguridad de A5/2 y de GSM en la sección de Antecedentes de la presente descripción.

Ataque de Texto Sin Formato Conocido en los Métodos A5/2

En esta sección presentamos un nuevo ataque de texto sin formato conocido (ataque de secuencia de claves conocido) en A5/2. Dando una secuencia de claves, dividida en tramas, y los números de trama respectivos, el ataque recupera la clave de sesión.

En comparación con los ataques de la técnica anterior, el nuevo método de ataque podría parecer que requiere más información, sin embargo, funciona dentro de solo unos pocos milisegundos de datos. Luego, mejoramos nuestro ataque a un ataque solamente de texto cifrado que requiere solo aproximadamente 40 milisegundos de datos cifrados, desconocidos. Por lo tanto, nuestro ataque es muy fácil de implementar en la práctica. Hemos simulado nuestro ataque de texto sin formato conocido en un ordenador personal y verificado los resultados. Esta simulación recupera la clave en menos de un segundo.

El tiempo de cálculo y la complejidad de la memoria de este ataque son similares al ataque de Goldberg, Wagner y Green.

Por lo tanto, el método comprende, véase la Figura 6:

1. Conocer el estado interno inicial de los registros R1, R2, R3 y R4, y el número de trama inicial, la clave de sesión puede ser recuperada utilizando operaciones algebraicas simples. Esto se debe principalmente a que el proceso de inicialización es lineal en la clave de sesión y el número de trama inicial. Por lo tanto, en el ataque nos enfocamos en revelar el estado interno inicial de los registros.

2. Dejar que k_0, k_1, k_2, ... sean la salida del algoritmo A5/2 dividido en tramas. Téngase en cuenta que cada k_j es la secuencia de claves de salida para una trama completa, es decir, cada k_j tiene 114 bits de longitud. Dejar que f, $f + 1$, $f + 2$, ... sean los números de trama asociados con estas tramas, donde f es el número de trama inicial. Indicamos como $k_j[i]$ el bit i-ésimo de la secuencia de claves en la trama j. El estado interno inicial del registro Ri en la trama j es indicado como Ri_j. Este es el estado interno después de la inicialización pero antes de las 99 sincronizaciones. Téngase en cuenta que esta indicación es algo imprecisa, ya que la salida es en realidad 228 bits, cuando la primera parte es utilizada para cifrar el enlace de red a móvil, y la segunda parte de 114 bits es el enlace de móvil a red.

3. Suponer que se conoce el estado inicial $R4_0$ del registro R4 en la primera trama. Una observación importante es que R4 controla las sincronizaciones de los otros registros, y dado que se conoce R4, también se conoce el número exacto de veces que cada registro ha sido sincronizado desde su estado inicial. Cada registro tiene una retroalimentación lineal, por lo tanto, una vez dado el número de veces que un registro es sincronizado, cada bit de su estado interno puede ser expresado como una combinación lineal de bits del estado interno original.

4. La salida del algoritmo A5/2 es un XOR de los últimos bits de los registros R1, R2, y R3, y tres funciones de mayoría de bits de R1, R2 y R3 (véase la Figura 4 para los detalles exactos). Por lo tanto, la función resultante es cuadrática, cuando las variables son los bits en el estado inicial de estos registros. Aprovechamos este bajo grado algebraico de la salida. El objetivo en los siguientes párrafos es expresar cada bit de la salida completa del cifrado (constituido por varias tramas) como una función multivariada cuadrática en el estado inicial. Luego, construimos un sistema sobredefinido de ecuaciones cuadráticas que expresa el proceso de generación de secuencia de claves y lo resuelve.

5. Dando un número de trama f, hay una descripción algebraica de cada bit de salida. Realizamos la linealización a los términos cuadráticos en esta descripción algebraica. Observamos que cada función de mayoría opera en bits de un único registro. Por lo tanto, tenemos términos cuadráticos que consisten en variables solamente del mismo registro. Teniendo en cuenta que un bit en cada registro es establecido en 1: R1 aporta 18 variables lineales más todos sus $(17 \cdot 18)/2 = 153$ productos. Del mismo modo, R2 aporta $22 + (22 \cdot 21)/2 = 22 + 231$ variables y R3 contribuye $22 + (22 \cdot 21)/2 = 22 + 231$ variables. Hasta ahora hay $18 + 153 + 21 + 210 + 22 + 231 = 655$ variables después de la linealización. También se necesita una variable que tome el valor constante de 1. En total tenemos un conjunto de 656 variables. Indicamos el conjunto de estas 656 variables por V_0. De estas variables, $18 + 21 + 22 = 61$ variables describen directamente el estado inicial completo de R1, R2, y R3.

6. Cada bit de salida que tenemos, añade una ecuación en variables de V_0. Una trama consta de 114 bits. Por lo tanto, obtenemos 114 ecuaciones de cada trama. La solución del sistema de ecuaciones revela el valor de las variables en V_0, y entre ellas las variables lineales que describen directamente el estado interno inicial de R1, R2, y R3. Sin embargo, no hay suficientes ecuaciones en esta etapa para resolver eficientemente el sistema.

La observación principal es que, dadas las variables en V_0 definidas en la trama f, los bits de cualquier otra trama pueden ser descritos en términos lineales de las variables en el conjunto V_0. Cuando se mueve a la siguiente trama, el número de trama es incrementado en 1 y el estado interno es reiniciado. Suponemos que se conoce el valor del registro $R4_0$. Debido al método de inicialización, donde el número de trama es cifrado en XOR bit por bit en los registros (véase la descripción de A5/2), conocemos el valor de $R4_1$. Dado que los valores $R1_0$, $R2_0$ y $R3_0$ no se conocen, no conocemos el valor de los registros $R1_1$, $R2_1$ y $R3_1$ tampoco, pero conocemos la diferencia XOR entre $R1_0$, $R2_0$, $R3_0$ y $R1_1$, $R2_1$, $R3_1$ respectivamente.

7. Definimos el conjunto de variables que describen su estado y la linealización de estas variables como V_1, de la misma

manera que lo hicimos con la primera trama para crear el conjunto V_0. Debido al método de inicialización, para cada registro *i* sabemos la diferencia entre Ri_1 y Ri_0. Conociendo la diferencia, podemos describir las variables en el conjunto V_1 en términos lineales de las variables en el conjunto V_0. Es decir, incluyendo los términos cuadráticos! Para ver esto, se supone que $a_1 \cdot b_1$ es un término cuadrático en V_1, naturalmente $a_0 \cdot b_0$ es un término cuadrático en V_0, y se conoce la diferencia d_a y d_b, de tal manera que: $a_1 = a_0 \oplus d_a$ y $b_1 = b_0 \oplus d_b$.

8. Por lo tanto, $a_1 \cdot b_1 = (a_0 \oplus d_a) \cdot (b_0 \oplus d_b) = a_0 \cdot b_0 \oplus a_0 \cdot d_b \oplus b_0 \cdot d_a \oplus d_a \cdot d_b$. Dado que se conocen d_b y d_a, esta ecuación es lineal en las variables en V_0. Este hecho permite utilizar los bits de salida en la segunda trama para obtener ecuaciones lineales adicionales en las variables de V_0. Lo mismo sigue para cualquier otra trama.

Está claro que una vez que se han obtenido 656 ecuaciones linealmente independientes, el sistema puede ser resuelto fácilmente utilizando la eliminación de Gauss. Sin embargo, es prácticamente muy difícil recopilar 656 ecuaciones linealmente independientes. Esto es un efecto de los reinicios frecuentes, y del bajo orden de la función de mayoría. En realidad no es necesario resolver todas las variables, es decir, es suficiente para resolver las variables lineales del sistema, ya que las otras variables son definidas como sus productos. Hemos probado experimentalmente y encontramos que después de obtener aproximadamente 450 ecuaciones de forma secuencial, las variables lineales originales en V_0 pueden ser resueltas utilizando la eliminación de Gauss.

Fin del método

Este ataque puede ser resumido como sigue: se prueban todos los valores posibles para $R4_0$, y para cada uno de tales valores se ha resuelto el sistema linealizado de ecuaciones que describe la salida. La solución de las ecuaciones da el estado interno de R1, R2 y R3. Junto con R4, se conoce el estado interno completo que da una sugerencia para la clave.

La complejidad temporal del ataque es como sigue: hay 2^{16} posibles conjeturas del valor de $R4_0$. Esta cifra debería ser multiplicada por el tiempo que lleva resolver un sistema binario lineal de 656 variables para una suposición específica, es decir, aproximadamente $656^3 \approx 2^{28}$ XOR, o aproximadamente 2^{44} XOR en total.

Resultado: hemos implementado con éxito este algoritmo, tarda unos 40 minutos en nuestro ordenador personal Linux 800MHz PIII. El requisito de memoria es insignificante: mantener el sistema linealizado en la memoria requiere 656^2 bits ≈ 54 KB. Cuando se implementa el algoritmo en un ordenador personal, aprovechamos el hecho de que una máquina de PC puede realizar el XOR de 32 bits con otros 32 bits en una operación.

Optimización del Ataque de Texto Sin Formato Conocido en el Método A5/2

Una posible optimización es filtrar valores incorrectos de $R4_0$, y resolver el sistema de ecuaciones solo para el valor correcto de $R4_0$. El filtrado está basado en la observación de que el sistema de ecuaciones para cada sugerencia de $R4_0$ contiene líneas linealmente dependientes. Este filtrado ahorra una cantidad considerable de tiempo, ahorrando la resolución relativamente costosa de los sistemas de ecuaciones.

1. Hay un sistema diferente de ecuaciones para cada valor diferente de $R4_0$. Nuestra técnica de etapa de filtrado requiere una etapa de cálculo previo que resuelva los 2^{16} sistemas posibles de antemano. Dando la matriz *S* que describe el sistema, y para cualquier salida *k*, es decir, $S \cdot V_0 = k$, calculamos una "matriz de resolución" *T* del sistema.

2. La matriz *T* es calculada tomando la matriz de unidad que tiene el mismo número de filas que la matriz *S*, y aplicando la misma serie de operaciones elementales que son realizadas durante una eliminación de Gauss de *S*. Multiplicación por T a la izquierda de *S* tiene el impacto de aplicar la eliminación de Gauss a *S*:

$$T \cdot S = \begin{pmatrix} V_s \\ 0 \end{pmatrix},$$

donde V_s es una matriz cuyas líneas son linealmente independientes, y las filas debajo de la matriz V_s son todas líneas cero. Las líneas cero son el resultado del sistema de ecuaciones que contiene líneas linealmente dependientes. Lo que nos interesa es aprovechar las líneas linealmente dependientes de *S*.

3. Aprovechamos esta ventaja utilizando bits linealmente dependientes de la salida de A5/2:

$$T \cdot k = T \cdot S \cdot V_0 = \begin{pmatrix} V_s \\ 0 \end{pmatrix} \cdot V_0.$$

4. Nos gusta verificar la suposición del valor de $R4_0$, es decir, filtra las suposiciones erróneas de $R4_0$. Se pueden utilizar las líneas de *T* que una vez multiplicadas por la salida *k* dan como resultado el valor cero. En una suposición correcta, todas estas líneas dan como resultado un cero después de la multiplicación anterior. En una suposición errónea, cada línea tiene una probabilidad de aproximadamente la mitad de ser cero una vez multiplicada por *k*. Por lo tanto, en promedio, se deben calcular aproximadamente dos líneas (productos de punto) para cada suposición errónea de $R4_0$. Durante el cálculo previo, mantenemos para cada valor posible de $R4_0$ solo aproximadamente 16 de las líneas de *T* que

obtienen el valor 0 una vez multiplicado por *k*. Cuando se realiza el ataque, las suposiciones incorrectas de $R4_0$ son filtradas multiplicando las líneas guardadas por *k*.

5. Cuando el resultado de todas las multiplicaciones para un $R4_0$ supuesto son cero, tenemos un sistema de ecuaciones candidato, que es en realidad un candidato para un valor para $R4_0$. Dando la sugerencia para $R4_0$, resolvemos el sistema de ecuaciones sugerido y calculamos el estado interno inicial de R1, R2, y R3. Junto con la suposición de $R4_0$, se puede determinar fácilmente K_c. La etapa de filtrado está diseñada de modo que la suposición correcta de $R4_0$ sobreviva. Téngase en cuenta que el número de valores de $R4_0$ que sobrevive a la etapa de filtrado es aproximadamente uno, es decir, el valor correcto para $R4_0$.

Fin del método

Resultado: la complejidad de memoria es de aproximadamente $2^{27,8}$ bytes (menos de 250 MB) necesarios para almacenar los vectores de fila anteriores.

El resultado anterior se aplica cuando se utiliza texto sin formato conocido del enlace inalámbrico que se origina desde la red hacia el teléfono móvil. Cuando se utiliza el texto sin formato conocido del enlace que se origina desde el teléfono móvil hacia la red, se necesitan unas pocas ecuaciones más para alcanzar un estado en el que existen líneas linealmente dependientes. Esto es porque en el enlace del teléfono hacia la red, se ha utilizado el segundo bloque de 114 bits de los 228 bits de la salida de A5/2. Estos bits se ven menos afectados por los reinicios frecuentes y, por lo tanto, un poco menos dependiente linealmente.

Téngase en cuenta que cuando se utiliza esta optimización se necesita algún compromiso.

Dado que se requieren cuatro tramas de texto sin formato conocidas, el XOR entre el número de trama *f* y cada uno de *f*+1, *f*+2, y *f*+3 debe ser conocido de antemano, antes de conocer el valor exacto de *f*. Esta diferencia XOR es requerida con el fin de expresar los bits de secuencia de claves de las tramas como términos lineales sobre el conjunto V_0, y para calcular el sistema de ecuaciones. En otras palabras, el sistema de ecuaciones depende no solo de $R4_0$ sino también de la diferencia XOR.

El problema aquí es la operación de suma, por ejemplo, *f*+1 puede resultar en una carga que se propagaría a través de *f*, no permitiendo así el cálculo de la diferencia XOR de antemano. Para facilitar el cálculo, requerimos que *f* tenga un bit específico establecido en 0. Este requisito impide que una carga se propague más allá del bit específico. Tenemos en cuenta que necesitamos calcular la diferencia XOR para una suma del número 3 al número de trama *f*, por lo tanto, necesitamos que el valor del tercer bit menos significativo de *f* sea cero, y también necesitamos requerir que los dos últimos bits en *f* tengan un valor constante ya que cualquier combinación de estos bits da como resultado una diferencia XOR diferente después de la suma.

Estos requisitos son suficientes para permitir calcular de antemano las diferencias anteriores. Para permitir cualquier valor constante de los dos bits inferiores de *f*, se realiza el cálculo previo para cada valor posible. Hay cuatro valores posibles. Este hecho multiplica la complejidad de la memoria por un factor de cuatro, y la complejidad del tiempo de cálculo previo también por un factor de cuatro. La complejidad de la memoria anterior ya incluye este factor. Podemos eliminar el requisito de que el tercer bit sea 0, en el caso de que los dos bits inferiores sean cero, debido al hecho de que en este caso suma de hasta tres no puede provocar una carga fuera de los primeros dos bits.

Por lo tanto, fuera de los ocho valores posibles para los tres bits inferiores de *f*, permitimos cinco. Hacemos hincapié en que esta limitación sobre los posibles valores de *f* no tiene implicaciones prácticas serias, ya que es necesario esperar como máximo 3 tramas para obtener un número de trama que cumpla con los requisitos. El ataque instantáneo solamente de Texto Cifrado que describimos está basado en este ataque y necesita funcionar en 4 bloques de tramas. Téngase en cuenta que en este caso, si el primero del número de trama, fuera de cuatro tramas consecutivas no cumple con los requisitos. Si eso sucede, se asegura que el primer número de trama en el siguiente bloque de 4 tramas cumple con los requisitos.

Analizamos la complejidad temporal de este ataque optimizado como sigue: dando un valor del número de trama *f*, por cada suposición errónea de $R4_0$ necesitamos probar dos productos de punto en promedio. Una vez que tenemos el valor $R4_0$ correcto, el tiempo necesario para resolver el sistema de ecuaciones para el valor correcto es de aproximadamente 2^{28}, lo cual es insignificante. Por lo tanto, la complejidad temporal media de este ataque optimizado es de aproximadamente 2^{16} productos de punto.

Analizamos la complejidad temporal del cálculo previo como sigue: en la etapa de cálculo previo calculamos el sistema de ecuaciones *S* y su matriz *T* para cada valor $R4_0$, de los 2^{16} valores posibles y para cada diferencia XOR permitida de *f*. Para cada uno de tales sistemas, solo conservamos aproximadamente 16 de las líneas de *T* que obtienen el valor 0 una vez multiplicado por *k*. Para calcular *T*, realizamos la eliminación de Gauss sobre *S*. La complejidad temporal para la eliminación de Gauss es de aproximadamente 2^{28} XOR. Cuando se multiplican las cifras anteriores, obtenemos 2^{44}. Dado que repetimos el proceso para cada una de las cuatro diferencias XOR requeridas de *f*, multiplicamos esta cifra por cuatro. Por lo tanto, la complejidad temporal de procesamiento previo es de 2^{46} XOR.

Hemos implementado este ataque optimizado en nuestro ordenador personal, y nos lleva menos de un segundo

recuperar K_c. El cálculo previo de una sola vez tarda aproximadamente 160 minutos.

Un Ataque Instantáneo Solamente de Texto Cifrado en el Método A5/2

En esta sección mostramos un ataque en A5/2. Un factor importante que nos facilita convertir el ataque de "Ataque de Texto Sin Formato Conocido en A5/2" en un ataque solamente de texto cifrado contra A5/2 es que en GSM se han empleado códigos de corrección de errores antes del cifrado. Por lo tanto, el texto sin formato del cifrado tiene una redundancia altamente estructurada.

Existen varios tipos de métodos de corrección de errores que son utilizados en GSM, y son utilizados diferentes esquemas de corrección de errores para diferentes canales de datos. Por simplicidad, nos centramos en los canales de control, y específicamente en los códigos de corrección de errores del Canal de Control Asociado Lento (SACCH). Téngase en cuenta que este código de corrección de errores es el único código que es utilizado en el inicio de una conversación. Por lo tanto, es suficiente concentrarse en este código. Utilizando este código de corrección de errores, montamos un ataque solamente de texto cifrado que recupera la clave. Sin embargo, el nuevo método de ataque también se puede aplicar a otros códigos de corrección de errores.

En el SACCH, el mensaje que ha de ser codificado con códigos de corrección de errores tiene un tamaño fijo de 184 bits. El resultado es de 456 bits de largo. Este mensaje de 456 bits es intercalado en 4 ráfagas.

La operación de codificación y la operación de intercalado pueden ser modeladas juntas como una matriz de 456 x 184 sobre *GF*(2), que indicamos mediante *G*. El mensaje que ha de ser codificado es considerado como un vector binario de 184 bits, *P*. El resultado de la operación de codificación-entrelazado es: $M = G \cdot P$. El vector resultante *M* es dividido en 4 ráfagas. En el proceso de cifrado cada ráfaga es cifrada con XOR con la salida de A5/2 para la ráfaga respectiva.

Dado que la matriz *G* es una matriz binaria de 456 x 184, existen 456-184 = 272 ecuaciones que describen el núcleo de la transformación inversa. En otras palabras, dando el vector $M = G \cdot P$, hay 272 ecuaciones linealmente independientes en sus elementos. Dejar que K_G sea una matriz que describe estas ecuaciones lineales, es decir, $K_g \cdot M = 0$ para cualquiera de tales *M*.

Indicamos los bits de secuencia de salida de A5/2 para una duración de 4 tramas mediante $k = k_j \| k_{j+1} \| k_{j+2} \| k_{j+3}$, donde ‖ es el operador de concatenación. El texto cifrado *C* es calculado por $C = M \oplus K$. Utilizamos las mismas 272 ecuaciones en *C*, concretamente:

$$K_G \cdot (M \oplus k) = K_G \cdot M \oplus K_G \cdot k = 0 \oplus K_G \cdot k = K_G \cdot k.$$

Dado que se conoce el texto cifrado *C*, en realidad obtenemos ecuaciones lineales sobre los elementos de *k*.

Téngase en cuenta que las ecuaciones que obtenemos son independientes de *P* -- solo dependen de *k*. Sustituimos cada bit en *k* con su descripción como términos lineales sobre V_0 (véase nuestra descripción del ataque instantáneo de texto sin formato conocido), y así obtenemos ecuaciones en variables de V_0. Cada bloque de codificación de 456 bits, proporciona 272 ecuaciones. El resto de los detalles del ataque y su complejidad temporal son similares al caso optimizado en la sección anterior, cuando sustituimos *k* por $K_G \cdot k$.

Mientras que en el ataque de texto sin formato conocido, cuatro tramas de datos son suficientes para lanzar el ataque, en el ataque solamente de solo texto cifrado necesitamos ocho tramas, dado que de cada trama cifrada, obtenemos solo la mitad de la información en comparación con el ataque de texto sin formato conocido. Cuando se analiza el tiempo y la complejidad de memoria de este ataque solamente de texto cifrado, tenemos en cuenta que restringimos los cuatro bits inferiores del número de trama *f*. Solo permitimos 9 fuera de los 16 valores posibles para estos cuatro bits. Esta restricción duplica la complejidad de la memoria en comparación con el ataque optimizado de texto sin formato conocido, y también duplica la complejidad del cálculo previo.

Fin del método

Resumimos la complejidad del ataque solamente de texto cifrado como sigue: la complejidad temporal media de este ataque solamente de texto cifrado es de aproximadamente 2^{16} productos de punto. La complejidad de la memoria es de aproximadamente $2^{28,8}$ bytes (menos de 500 MB), la complejidad del tiempo de cálculo previo es de aproximadamente 2^{47} XORS. Nuestra implementación en un ordenador personal recupera K_c en menos de un segundo, y se tarda aproximadamente 320 minutos en completar el cálculo previo.

También hemos mejorado con éxito el ataque de Goldberg, Wagner, y Green y el ataque de Petrovic y Fúster-Sabater a un ataque solamente de texto cifrado utilizando nuestros métodos. Cuando se da la descripción actual, la mejora mencionada anteriormente debería ser obvia para los expertos en la técnica.

Ataque Directo contra el Método A5/1

El siguiente es un ejemplo de tal ataque directo: Dando un bloque de varias tramas que están cifradas, utilizamos los métodos de las secciones anteriores, para calcular $K_G \cdot k$. Estos bits que obtenemos solo dependen de la salida de A5/1 en varias tramas. Llamamos a estos bits de salida la secuencia codificada. Supongamos que se sabe que el número de

trama de la primera trama de estas tramas ha de ser dividido entre cuatro sin resto. Todo el proceso puede ser visto como una función desde el estado interno de A5/1 hasta la secuencia codificada. Dejar que f() sea esa función, cuando solo se consideran 64 bits de salida, por lo tanto, f() asigna 64 bits a 64 bits.

Entonces f() es una función que toma un estado interno de A5/1 después de la inicialización, y emite la secuencia codificada. Invertir f() revelará el estado interno, y romperá el cifrado. Téngase en cuenta que debemos hacer una suposición con respecto al número de trama, de lo contrario, f() depende del número de trama. Podemos aplicar una de las compensaciones de datos de memoria de tiempo conocidas en la técnica, por ejemplo, las que se han descrito por Biryukov y Shamir en su documento "Cryptanalytic Tem/Memory/Data Tradeoffs for Stream Ciphers", Advances in Cryptology, proceedings of Asiacrypt'00, Lecture Notes in Computer Science 1976, Springer-Verlag, págs. 1-13, 2000.

Utilizamos sus anotaciones para la compensación, es decir, N es el espacio de estados internos, T es el número de evaluaciones de f(), D es el número de puntos de datos disponibles, M es el número de líneas de memoria. En este caso, $N = 2^{64}$, y cada línea de memoria tiene 16 bytes de longitud. Por ejemplo, en la curva de compensación $N^2 = D^2M^2T$, $T>D^2$, y $N = 2^{64}$, un punto es $D = 2^8$, que es aproximadamente 8 segundos de datos fuera del aire, $M = 2^{39}$, que es aproximadamente 8,8 Tera-Bytes (Puede ser almacenado en 44 discos duros de 200 GB). Si tomamos una codificación similar a las secciones anteriores, eso significa que tenemos que multiplicar los datos por 4 para compensar los diferentes números de trama. Por lo tanto, se necesitan 176 discos duros de 200 GB.

El tiempo que lleva un ataque real es por lo tanto, $T = 2^{34}$ valoraciones de f(). Suponiendo que f() puede ser calculado 2^{20} veces en un segundo en un solo ordenador personal, el cálculo requiere 2^{14} segundos. En una red con 1000 ordenadores, tarda aproximadamente 16 segundos. El resultado será aproximadamente 2^{17} accesos aleatorios al disco, se puede acceder aleatoriamente a cada disco aproximadamente 200 veces por segundo, por lo tanto, el tiempo de acceso es de aproximadamente 655 segundos, pero hay 176 discos duros, por lo tanto, el tiempo total de acceso es 3,76 segundos, que son realizados en segundo plano cuando se realizan los otros cálculos.

El cálculo previo tarda N/D, que son 2^{50} evaluaciones de f(), que tarda 2^{36} segundos en un ordenador personal. Necesitamos calcularlo cuatro veces. En total, en una red de 10.000 ordenadores, esta tarea debería ser completada en aproximadamente 10 meses. En un trabajo distribuido, esa red requiere un ancho de banda de aproximadamente 1,35 Mbyte/Segundo. Este es un cálculo factible a través de Internet, por ejemplo.

Téngase en cuenta que aumentar los datos disponibles disminuye los otros requisitos dramáticamente. Si tuviéramos 5 minutos de tales datos, que son 37,5 veces más que 8 segundos, entonces $D = 2^{13}$, solo se deben utilizar aproximadamente 44 discos duros de 200 MB en total, es decir $M = 2^{37}$, y el tiempo sería $T = 2^{28}$, que tarda aproximadamente 5 minutos para calcular en un solo PC. Esto significa que el ataque se lleva a cabo en tiempo real, pero solo uno de los pocos miles de tramas es una trama que permite que el ataque tenga éxito. Cuando se encuentra una trama, el ataque encuentra casi instantáneamente si esta trama es, de verdad, "la correcta" o no.

El ataque requiere 2^{14} accesos aleatorios al disco, que tardan aproximadamente 81 segundos, pero son realizados en 44 discos duros en paralelo, lo que lleva en total aproximadamente 1,86 segundos, que son gastados en el antecedente del cálculo. El cálculo previo es reducido a $N/D = 2^{51}$ evaluaciones de f(), lo que lleva aproximadamente 2^{31} segundos, o aproximadamente 3 meses en una red de 1000 ordenadores personales.

Si se permite 1 hora de tales datos, solo se necesitan aproximadamente 270 GB de memoria, que pueden ser almacenados en uno o dos discos duros. El tiempo de ataque real dura aproximadamente una hora en un ordenador personal, lo que significa que en realidad es en tiempo real, el tiempo de acceso al disco duro es de aproximadamente 5 minutos, lo cual es insignificante. El cálculo previo puede ser completado en una red de 40 PC en aproximadamente 10 meses.

Incluso si A5/2 ya no es utilizado en redes GSM, pero A5/1 permanece, este ataque directo en A5/1 puede ser utilizado para aprovechar un ataque en GPRS, utilizando el hecho de que su clave es creada con el mismo mecanismo (es decir, A3A8), y por lo tanto, cuando se le da la misma entrada (es decir, RAND y K_i) la salida, que es la clave resultante, es la misma. Este es un ejemplo, que puede ocurrir en otros cifrados, siempre que dos cifrados compartan el mismo acuerdo de clave, y un ataque activo puede ser montado más fácilmente en uno de estos cifrados.

Briceno descubrió que muchas redes GSM utilizan solo 54 bits fuera de los 64 bits de clave, estableciendo 10 bits de clave en 0. La técnica anterior no aprovechó este hecho cuando empleó criptoanálisis. Observamos que cuando los bits son establecidos en un valor constante, el ataque directo A5/1 puede ser mejorado dramáticamente. Cuando N disminuyó de 2^{64} a 2^{54}. Solo 54 bits de salida deben ser considerados fuera de f(). Por lo tanto, cada línea de memoria ahora tiene solo 54 veces 2 bits de longitud = 13,5 bytes, supongamos 14 bytes.

En la curva de compensación, $N^2 = D^2M^2T$, $T>D^2$ y $N = 2^{54}$, considera el ejemplo que mostramos anteriormente, donde $D = 2^8$, que es aproximadamente 8 segundos de datos fuera del aire, pero ahora $M = 2^{33}$, que es aproximadamente 500 GB, puede ser almacenado en tres discos duros de 200 GB. $T = 2^{26}$, esto tarda solo aproximadamente de un minuto de cálculo en un SOLO PC! Este cálculo puede ser realizado en paralelo en algunas computadoras que alcanzan un ataque directo en tiempo real en A5/1. El cálculo previo de una sola vez tarda $N/D = 2^{46}$, que son aproximadamente 3100 días de ordenador en total, que pueden ser calculados en una red de 10 PC en aproximadamente 10 meses.

Aprovechar el Ataque a redes que requieren A5/1 o A5/3 pero aceptan menos

Algunas redes pueden preferir que el teléfono móvil funcione con A5/1, pero si no es posible funcionar con A5/2. Cuando un teléfono móvil accede a la red, le dice a la red cuáles son sus capacidades, incluyendo qué algoritmo de cifrado puede utilizar. Un ataque simple de marca de clase sería cambiar la información que obtiene la red, por lo que cree que el teléfono puede funcionar bien en A5/2 o bien en A5/0. Si la red acepta el cifrado con A5/2, entonces se pueden encontrar las claves de cifrado utilizando el método detallado anterior. Un ataque de marca de clase similar podría ser montado cuando la red prefiere A5/3 pero acepta menos (o bien A5/1 o bien A5/2).

Aprovechar los ataques a Cualquier Método de Red GSM

El ataque mostrado en "Un Ataque Instantáneo Solamente de Texto Cifrado en el Método A5/2" supone que el algoritmo de cifrado es A5/2. Utilizando ese ataque, es fácil recuperar K_c en tiempo real a partir de unas pocas decenas de milisegundos de texto cifrado.

Hacemos la pregunta, qué sucede cuando el algoritmo de cifrado no es A5/2, sino que es A5/1 o el recientemente elegido A5/3 o incluso GPRS. La sorprendente respuesta es que se aplica casi el mismo ataque. Todo lo que se necesita para que el nuevo ataque tenga éxito es que el teléfono móvil sea compatible con A5/2, pero este es en realidad un requisito obligatorio de GSM para permitir la itinerancia a las redes que utilizan A5/2.

El siguiente ataque recupera la clave de cifrado que utiliza la red cuando se emplea A5/1 o A5/3. La clave es descubierta por un hombre en mitad del ataque al cliente víctima. En este ataque, el atacante desempeña dos funciones. Suplanta la red, hasta donde ve el cliente, y suplanta al cliente, hasta donde ve la red. Téngase en cuenta que este tipo de ataque es relativamente fácil de montar en un entorno celular.

Durante la inicialización de una conversación, la red puede enviar la solicitud de autenticación al atacante, el atacante la envía a la víctima. La víctima calcula *SRES* y lo devuelve al atacante, que lo envía de vuelta a la red. Ahora el atacante está "autenticado" en la red. A continuación, la red le pide al cliente que inicie el cifrado con A5/1.

En nuestro ataque, dado que el atacante suplanta al cliente, la red le pedirá realmente al atacante que inicie el cifrado con A5/1. El atacante aún no tiene la clave y, por lo tanto, no es capaz de iniciar el cifrado. El atacante necesita la clave antes de que le pida que la utilice. Para conseguirlo, el atacante le pide a la víctima que cifre con A5/2 justo después de que la víctima haya devuelto los *SRES*, y antes de que el atacante devuelva la información de autenticación a la red.

Esta solicitud considera a la víctima como una solicitud legítima, dado que la víctima ve al atacante como la red. Luego, el atacante emplea el criptoanálisis para recuperar la clave de cifrado del A5/2 que es utilizado por la víctima. Solo entonces, el atacante envía la información de autenticación a la red. La clave solo depende de *RAND*, eso significa que la clave recuperada a través del ataque A5/2 es la misma clave que ha de ser utilizada cuando se ha utilizado A5/1 o incluso cuando se ha utilizado A5/3 de 64 bits! Ahora el atacante puede cifrar/descifrar con A5/1 o A5/3 utilizando esta clave.

Uno puede sospechar que la red puede identificar este ataque, identificando un pequeño retraso en el tiempo que lleva a completar el procedimiento de autenticación. Sin embargo, el estándar GSM permite 12 segundos para que el móvil complete sus cálculos de autenticación y devuelva una respuesta. El retraso incurrido por este ataque es menos de un segundo. También, los mensajes de señalización GSM pueden tardar normalmente un tiempo en desplazarse entre la red y el móvil, debido al retraso de protocolo de capa 2. En total, hay un retraso pero es insignificante.

Muchas redes rara vez inician el procedimiento de autenticación, y utilizan la clave creada en la última autenticación que es guardada en la SIM del cliente. Esta clave está numerada por la red con un número en el intervalo de cero a seis. Un atacante puede descubrir estas claves almacenadas suplantando la red para el móvil de la víctima. Luego, el atacante inicia una sesión de radio con la víctima, y le pide al móvil de la víctima que inicie el cifrado utilizando el algoritmo A5/2 y el número clave correspondiente. El atacante emplea entonces el ataque y recupera la clave y luego termina la sesión de radio. El propietario del móvil y la red no tendrán indicación del ataque.

Uno puede preguntarse si el operador de red puede descubrir el ataque porque el ataque transmite, y la interfaz puede ser causada. Aunque esto es generalmente cierto, ambos ataques requieren menos transmisión de la que se podría esperar a primera vista. En el primero, puede parecer que el atacante necesita transmitir durante todo el tiempo de conversación. Sin embargo, después del primer segundo de comunicación, el atacante ya tiene la clave de cifrado y realmente no necesita continuar con el hombre activo en mitad del ataque.

Un atacante podría querer detener el ataque activo, dejar que la red y la víctima continúen comunicándose, y aprovechar la conversación encriptada utilizando la clave que había descubierto. El primer paso para hacerlo, es cambiar el cifrado que la víctima utiliza para satisfacer los requisitos de la red. El atacante debería pedirle a la víctima que cambie el cifrado a ningún cifrado, y luego que cambie al cifrado que es utilizado por la red, es decir, A5/1. Un atacante podría hacer que la red ordene un traspaso de la conversación a otra frecuencia. Al mismo tiempo, el atacante solicita al móvil de la víctima que realice un traspaso a la misma frecuencia.

Téngase en cuenta que GSM realmente no transmite en una sola frecuencia. En su lugar, GSM emplea un esquema de

salto de frecuencia. Por simplicidad, relacionamos con una cierta secuencia de salto como una sola frecuencia. Esto no tiene implicaciones en los ataques que presentamos. En la mayoría de las conversaciones GSM, un traspaso es iniciado por la red poco después del comienzo de la conversación. Dado que sucede de todos modos, le ahorra al atacante la necesidad de "hacer" que la red solicite un traspaso. De esta manera, el atacante puede detener su transmisión, mientras que aún es capaz de escuchar a escondidas la conversación. En el segundo escenario, el atacante ataca en el momento que elija, y todo el ataque puede ser completado en unos segundos como máximo. Cuando estas claves son utilizadas posteriormente, el atacante no necesita realizar ninguna transmisión.

En los escenarios que se han descrito a continuación, se ha mostrado un ataque en el que el atacante puede aprovechar cualquier conversación, mientras transmite solo por un corto período de tiempo en un momento posterior a su elección, posiblemente después de que se haya completado la llamada.

El aprovechamiento del ataque se basa en el hecho de que la misma clave es cargada en A5/2 y A5/1 e incluso en A5/3 de 64 bits (en el escenario donde A5/3 es utilizado en GSM, de acuerdo con los estándares GSM). Por lo tanto, descubrir la clave para A5/2 revela la clave para A5/1 y A5/3 de 64 bits.

Este ataque también se aplica a GPRS, debido a razones similares. Cuando la red desafía al móvil por una nueva clave GPRS, utilizando un valor aleatorio RAND de 128 bits, el atacante puede utilizar "hombre-en-el-medio" e iniciar sesión de radio con la víctima, iniciar solicitud de autenticación utilizando el mismo valor RAND, y luego pedirle que cifre con A5/2. Entonces encuentra la clave utilizando el ataque solamente de texto cifrado que presentamos. La clave que es recuperada será la misma que la clave GPRS, esto se debe al hecho de que ambos son creados utilizando el mismo algoritmo A3A8 y la misma K_i; por lo tanto, cuando se da el mismo RAND, se produce la misma clave de sesión.

El atacante puede abstenerse de un ataque "hombre-en-el-medio", y grabar la comunicación GPRS y más tarde descifrarla utilizando un ataque similar, en el que le pide a la víctima que su autenticación con la misma RAND que se utilizó en la sesión, y luego pedirle a la víctima que cifre con A5/2 y recupere la clave. Incluso si GPRS cambia la clave varias veces utilizando una nueva RAND, cada vez que el atacante registra las comunicaciones y la RAND puede repetir este proceso más tarde en el ataque contra la víctima, encontrar la clave y descifrar las comunicaciones. Estos ataques pueden ser utilizados para suplantación, de manera similar. Téngase en cuenta que, aunque A5/3 puede ser utilizado con longitudes de clave de 64-128 bits, el estándar GSM permite la utilización de solo A5/3 de 64 bits.

Escenarios de Consecuencias Desafortunadas para GSM

Los ataques presentados pueden ser utilizados para emular ataques de la vida real en varios escenarios. En esta sección se han presentado cuatro. Estos ataques funcionan para varios algoritmos de cifrado pueden ser utilizados, por ejemplo: A5/1, A5/2 o A5/3, e incluso GPRS.

Ataque de Escuchas Telefónicas

Un escenario simple que uno podría anticipar es escuchar conversaciones a escondidas. Las comunicaciones que están cifradas utilizando GSM pueden ser descifradas y escuchadas a escondidas por un atacante, una vez que el atacante tiene la clave de cifrado. Tanto las conversaciones de voz como los datos, por ejemplo mensajes SMS, pueden ser interceptados.

Otro posible ataque de escuchas telefónicas es que el atacante graba la conversación cifrada. El atacante debe asegurarse de conocer el valor RAND que creó la clave que está en uso. En un momento posterior, cuando sea conveniente para el atacante, el atacante suplanta la red a la víctima. Luego, el atacante inicia una sesión de radio, pide a la víctima que realice la autenticación con el RAND anterior, y recupera la clave de sesión que se utilizó en la conversación grabada. Una vez que el atacante tiene la clave, simplemente descifra la conversación y puede escuchar su contenido.

Téngase en cuenta que un atacante puede grabar muchas conversaciones y, con ataques posteriores, recuperar todas las claves. Este ataque tiene la ventaja de transmitir solo en el tiempo que sea conveniente para el atacante. Posiblemente incluso años después de la grabación de la conversación, o cuando la víctima está en otro país, o en un lugar conveniente para el atacante.

Otro ataque es encontrar la clave antes de la conversación encontrando la clave almacenada como describimos. Encontrar la clave antes de la conversación es efectivo, si la red no le pide al abonado que realice la autenticación con una RAND diferente al comienzo de la conversación.

Ataque de Jaqueo de llamadas

Aunque una red GSM puede realizar la autenticación al inicio de la llamada, el cifrado es el medio de GSM para impedir la suplantación en etapas posteriores de la conversación. La suposición subyacente es que un impostor no tendría K_c, y por lo tanto, no sería capaz de realizar comunicaciones cifradas. Se ha mostrado cómo obtener claves de cifrado. Una vez que un atacante tiene las claves de cifrado, puede cortar la conversación de la víctima, y suplantar a la víctima para la otra parte. Por lo tanto, es posible jaquear la conversación después de la autenticación.

Algunas personas pueden afirmar que sería difícil aplicar este ataque en la práctica, debido a la dificultad de transmitir los datos requeridos en el aire. Se ha enfatizado que la suplantación es un ataque que es relativamente fácil de montar en un entorno celular. La transmisión GSM es llevada a cabo por radiofrecuencia, lo que hace que este tipo de ataque muy fácil de realizar y difícil de detectar. Por ejemplo, un atacante podría asegurarse de que su señal es recibida en la antena de la célula con una potencia mucho mayor que la señal de la víctima. El atacante también puede causar molestias asegurándose de que una señal de ruido es recibida en alta potencia en la antena de la víctima.

El jaqueo puede ocurrir durante la configuración temprana de la llamada, incluso antes de que el teléfono de la víctima comience a sonar. El operador difícilmente puede sospechar que hay un ataque. La única pista de un ataque es un momento de interfaz electromagnética algo aumentada.

Otra forma de jaquear las llamadas entrantes, es montar una especie de ataque de "hombre en el medio", pero en lugar de reenviar la llamada a la víctima, el atacante recibe la llamada.

Ataque de Alteración de Mensajes de Datos (SMS)

Una vez que se ha jaqueado una llamada, el atacante decide el contenido. El atacante puede escuchar el contenido de un mensaje que está siendo enviado por la víctima y enviar su propia versión. El atacante puede detener el mensaje, o enviar su propio mensaje SMS. Esto compromete la integridad del tráfico GSM.

Ataque de Robo de Llamada

GSM se creía que era seguro contra el robo de llamadas, debido a los procedimientos de autenticación de A3A8.

Sin embargo, debido a las debilidades mencionadas, un atacante puede hacer llamadas salientes a expensas de la víctima. Cuando la red solicita autenticación, entonces un hombre en el medio del ataque, similar al que describimos al aprovechar el ataque para cualquier red GSM, tendría éxito. El atacante inicia en paralelo una llamada saliente a la red celular, y una sesión de radio a una víctima. Cuando la red le solicita autenticación al atacante, el atacante le pide autenticación a la víctima, y retransmite la autenticación resultante de vuelta a la red.

El atacante también puede recuperar K_c como se ha descrito en la presente descripción. Ahora el atacante puede cerrar la sesión de radio con la víctima, y continuar la llamada saliente de forma regular. Este ataque es difícilmente detectable por la red, ya que lo ve como un acceso normal. El teléfono de la víctima no sonará, y la víctima no tendrá indicios de que él/ella es una víctima. Al menos hasta que llegue su factura mensual.

Otras realizaciones diferentes de métodos de ataque se les ocurrirán a personas expertas en la técnica tras la lectura de la presente descripción. Los métodos detallados anteriormente pueden ser ampliados adicionalmente, por ejemplo:

1. Un método de criptoanálisis que comprende

A. Solicitar un teléfono para cifrar con A5/2;

B. Utilizar los resultados para descifrar las comunicaciones que están cifradas con A5/2, A5/1, A5/3 o GPRS.

Por lo tanto, un atacante afecta la decisión con respecto al método de cifrado que ha de ser utilizado, en este caso de una manera que facilite su descifrado posterior.

2. Un método de ataque de marca de clase de criptoanálisis que comprende:

A. el atacante hace que la red decida que el teléfono no es capaz de cifrar con A5/1 sino solo con A5/2;

B. esto permite al atacante utilizar el ataque y descifrar las comunicaciones

3. Un método de criptoanálisis que comprende:

A. Realizar un criptoanálisis directo solamente de texto cifrado de A5/1;

B. Utilizar los resultados de la Operación (A) para facilitar el descifrado y/o cifrado de comunicaciones adicionales que sean consistentes con el cifrado que utiliza la clave de sesión y/o descifrado que utiliza la clave de sesión.

En el método anterior, el criptoanálisis puede considerar que parte de los bits de la clave de sesión tienen un valor fijo conocido.

Además, el criptoanálisis también puede encontrar la clave de sesión.

4. Un método de criptoanálisis que comprende:

A. realizar un criptoanálisis directo solamente de texto cifrado de A5/2;

B. Utilizar los resultados de la Operación (A) para facilitar el descifrado y/o cifrado de comunicaciones adicionales que

sean consistentes con el cifrado que utiliza la clave de sesión y/o el descifrado que utiliza la clave de sesión.

En el método de criptoanálisis anterior, el criptoanálisis puede considerar que parte de los bits de la clave de sesión tienen un valor fijo conocido.

Además, el método de criptoanálisis también puede encontrar la clave de sesión.

5. Un método para proteger las comunicaciones GSM que comprende realizar autenticación GSM repetidamente durante una sesión en curso.

En el método anterior para proteger las comunicaciones GSM, la clave de cifrado también puede ser cambiada como resultado de aplicar el procedimiento de autenticación GSM.

Además, un atacante puede emitir transmisiones de radiofrecuencia.

Los métodos de criptoanálisis activo GSM anteriores también pueden incluir:

A. la transmisión del atacante hace que la red decida que la marca de clase del teléfono es tal que el teléfono no es capaz de cifrar con A5/1 sino solo con A5/2;

B. esto hace que la red solicite cifrado solo con A5/2, permitiendo al atacante utilizar el ataque y descifrar las comunicaciones.

Los métodos de criptoanálisis activo GSM anteriores también pueden incluir:

A. la transmisión del atacante hace que la red decida que la marca de clase del teléfono es tal que el teléfono no es capaz de cifrar con A5/3 sino solo con A5/2 o A5/1;

B. esto hace que la red solicite cifrado solo con A5/1, permitiendo al atacante utilizar el ataque y descifrar las comunicaciones.

Los métodos de criptoanálisis activo GSM anteriores también pueden incluir:

A. La transmisión del atacante hace que el teléfono decida que la transmisión se ha originado desde la red, y solicita que el teléfono cifre con A5/2;

B. El teléfono responde con datos que están cifrados con A5/2;

C. Utilizar la clave de sesión que resulta del criptoanálisis para descifrar y/o cifrar comunicaciones en el enlace inalámbrico entre el atacante y el teléfono, y/o descifrar y/o cifrar comunicaciones en el enlace inalámbrico entre el atacante y la red, que está cifrada con A5/2, A5/1, A5/3 o GPRS, y/o descifra y/o cifra las comunicaciones en el enlace inalámbrico entre el teléfono y la red, que está cifrada con A5/2, A5/1, A5/3 o GPRS.

Los métodos de criptoanálisis activo GSM anteriores también pueden incluir:

A. La transmisión del atacante hace que el teléfono decida que la transmisión se ha originado desde la red, y solicita que el teléfono cifre con A5/1;

B. El teléfono responde con datos que están cifrados con A5/1;

C. Utilizar la clave de sesión que resulta del criptoanálisis para descifrar y/o cifrar comunicaciones en el enlace inalámbrico entre el atacante y el teléfono, y/o descifrar y/o cifrar comunicaciones en el enlace inalámbrico entre el atacante y la red, que está encriptada con A5/2, A5/1, A5/3 o GPRS, y/o descifrar y/o cifrar las comunicaciones en el enlace inalámbrico entre el teléfono y la red, que está cifrada con A5/2, A5/1, A5/3 o GPRS.

En el método de criptoanálisis anterior, el criptoanálisis solamente de texto cifrado puede comprender:

A. Realizar un ataque eficiente de texto sin formato conocido en A5/1 que recupera la clave de sesión;

B. Mejorar el ataque de texto sin formato conocido a un ataque solamente de texto cifrado en A5/1.

Mejoras en el método y sistema de red GSM

Varias mejoras en los sistemas celulares se les ocurrirán a las personas expertas en la técnica tras la lectura de los posibles nuevos métodos de ataque detallados en la presente descripción.

Los ejemplos relacionados con GSM incluyen:

1. Los operadores GSM debería reemplazar los algoritmos y protocolos criptográficos que se utilizan ahora lo antes posible, para proteger la privacidad de sus clientes.

2. Incluso las redes GSM que utilizan el nuevo A5/3 sucumben al ataque presentado aquí, en la forma en que A5/3 está actualmente integrado en GSM. Por consiguiente, se sugiere realizar cambios en la forma en que está integrado A5/3 para proteger las redes de tales ataques. Una posible corrección es hacer que las claves utilizadas en A5/1 y A5/2 no estén relacionadas con las claves que son utilizadas en A5/3. Este cambio también debería hacerse en GPRS. En general, es preferible crear claves no relacionadas para diferentes cifrados, de modo que una debilidad en una de ellas sería capaz de infligir en las otras.

3. Incluso si GSM implementa tamaños de clave más grandes para A5/3, la forma trivial para que GSM lo implemente, es utilizar los mismos primeros bits de la clave para A5/1 y A5/2, y tener algunos bits de clave adicionales añadidos. Dicha implementación hará que nuestro ataque descubra fácilmente 64 bits clave de la clave que es utilizada en A5/3, reduciendo así la seguridad considerablemente.

4. El presente ataque solamente de texto cifrado es facilitado por el hecho de que los códigos de corrección de errores son empleados ahora antes del cifrado. En el caso de GSM, la adición de tal redundancia estructurada antes de que se realice el cifrado, reduce fatalmente la seguridad del sistema. Se ha propuesto un método y una estructura para corregir este defecto.

5. Modificar el estándar GSM, para permitir la utilización de más de 64 bits A5/3, ofrecería una mayor seguridad, y limitaría la efectividad del ataque.

6. Realizar la autenticación más a menudo, incluso durante una sesión en curso, puede probar que es una buena protección. Probaría que el abonado "real" sigue siendo el que está al otro lado del canal. También, la autenticación en GSM cambia la clave, lo que obligaría a un espía a realizar el ataque desde el principio. Incluso si la clave no es cambiada durante la conversación, se asegurará de que no haya suplantación.

7. Detener la utilización de A5/2, especialmente en teléfonos. Detener la utilización de A5/2, dejaría el ataque directo en A5/1, que es más costoso y difícil de realizar. El ataque contra A5/1 aún podría ser aprovechado contra GPRS (como el ataque A5/2), pero el coste sería mucho más alto de realizar. La desventaja de este cambio, es que obligaría a actualizar la infraestructura en las redes que emplean A5/2. Otra opción es crear una serie de teléfonos que no sean compatibles con A5/2. No tendrán cifrado cuando estén en itinerancia a redes A5/2 (pero como mostramos, A5/2 no es seguro de todos modos), pero aumentarán la seguridad en las redes A5/1 o A5/3, dado que el ataque A5/2 no funcionaría. Por lo tanto, el A5/1 directo tendría que ser empleado, y este ataque es mucho más costoso y difícil de realizar.

8. Utilizar más bits disponibles para el cifrado, es decir, utilizar los 64 bits completos de clave disponibles. En el futuro, cuantos más bits disponibles estén disponibles – más bits pueden ser utilizados.

Se reconocerá que lo anterior no es más que un ejemplo de un aparato y método dentro del alcance de la presente invención y que a los expertos en la materia se les ocurrirán diferentes modificaciones tras la lectura de la descripción expuesta anteriormente.

REIVINDICACIONES

1. Un dispositivo para recuperar una clave de cifrado procesando una comunicación digital cifrada que comprende:

un receptor configurado para recibir un primer texto cifrado que es cifrado de acuerdo con A5/1, red a móvil o móvil a red, o con un A5/2, red a móvil o móvil a red, esquema de cifrado en el que el primer texto cifrado es un cifrado de un primer mensaje tras el cual se empleó un esquema de código de corrección de errores GSM antes del cifrado;

y un circuito de procesamiento adaptado para recuperar una clave de cifrado utilizada en el cifrado del primer texto cifrado realizando un criptoanálisis solamente de texto cifrado del primer texto cifrado,

caracterizado por que realizar un criptoanálisis solamente de texto cifrado del primer texto cifrado implica bits de XOR basados en una matriz G asociada con el esquema de codificación de corrección de errores GSM.

2. El dispositivo según la reivindicación 1, en el que realizar un criptoanálisis solamente de texto cifrado del primer texto cifrado implicada además generar ecuaciones independientes del primer mensaje.

3. El dispositivo según la reivindicación 1 o 2, en donde dicho circuito de procesamiento está adaptado además para cambiar la información de la red o del teléfono móvil incluyendo qué esquema de cifrado GSM puede o debería ser utilizado, así la red o el móvil aceptan uno de los cifrados A5/1 y A5/2, y para realizar el criptoanálisis solamente de texto cifrado en el primer texto cifrado que es cifrado de acuerdo con uno de dichos cifrados A5/1 y A5/2.

4. Un método para recuperar una clave de cifrado procesando una comunicación digital cifrada que comprende:

recibir un primer texto cifrado que es cifrado de acuerdo con un esquema de cifrado en el que el primer texto cifrado es un cifrado de un primer mensaje tras el cual se empleó un esquema de codificación de corrección de errores antes del cifrado;

y recuperar la clave de cifrado utilizada en el cifrado del primer texto cifrado realizando un criptoanálisis solamente de texto cifrado del primer texto cifrado,

caracterizado por que realizar un criptoanálisis solamente de texto cifrado del primer texto cifrado implica bits de XOR basados en una matriz G asociada con el esquema de codificación de corrección de errores GSM.

5. El método según la reivindicación 4, en donde realizar un criptoanálisis solamente de texto cifrado del primer texto cifrado implica generar ecuaciones independientes del primer mensaje.

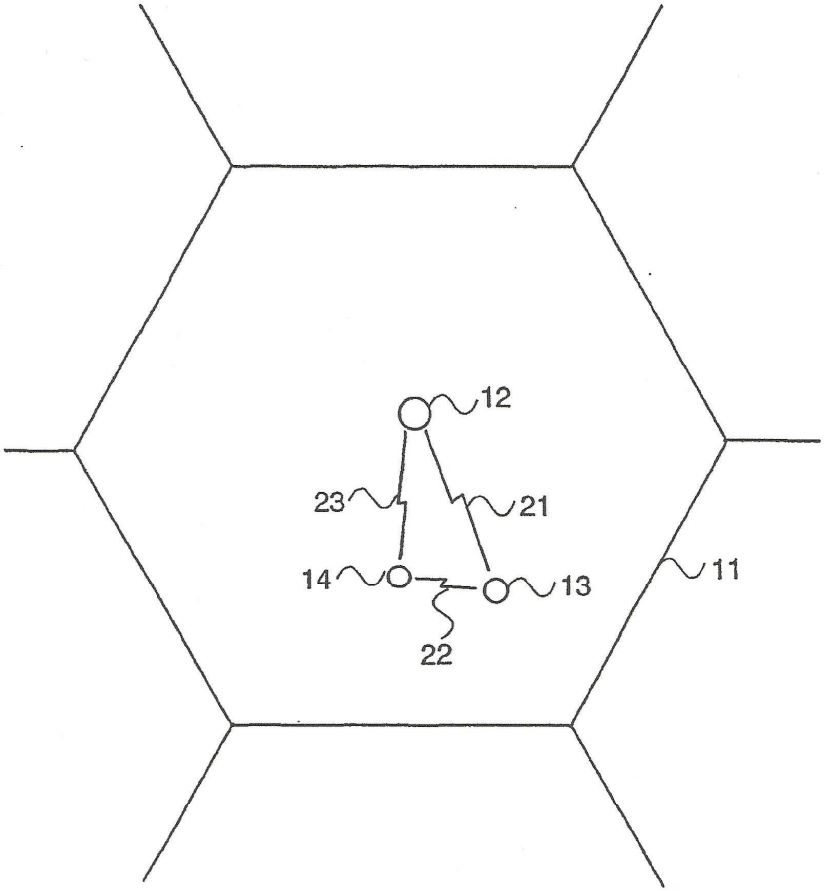

Fig. 1

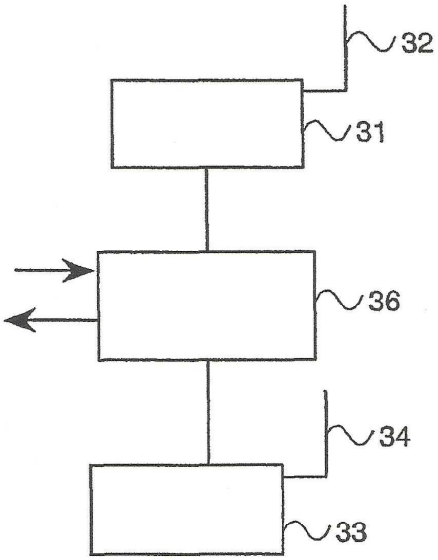

Fig. 2

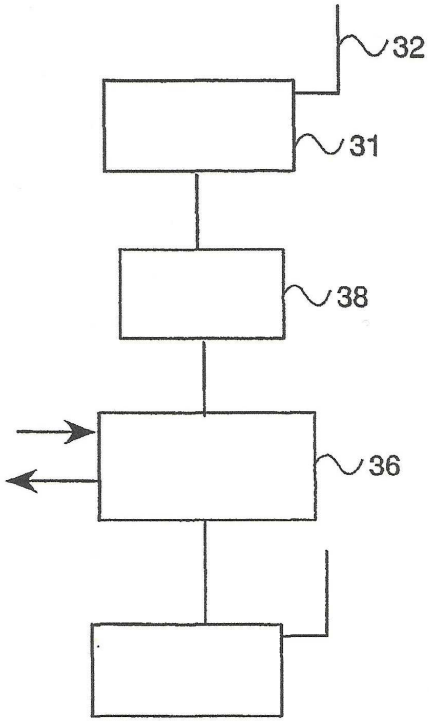

Fig. 3

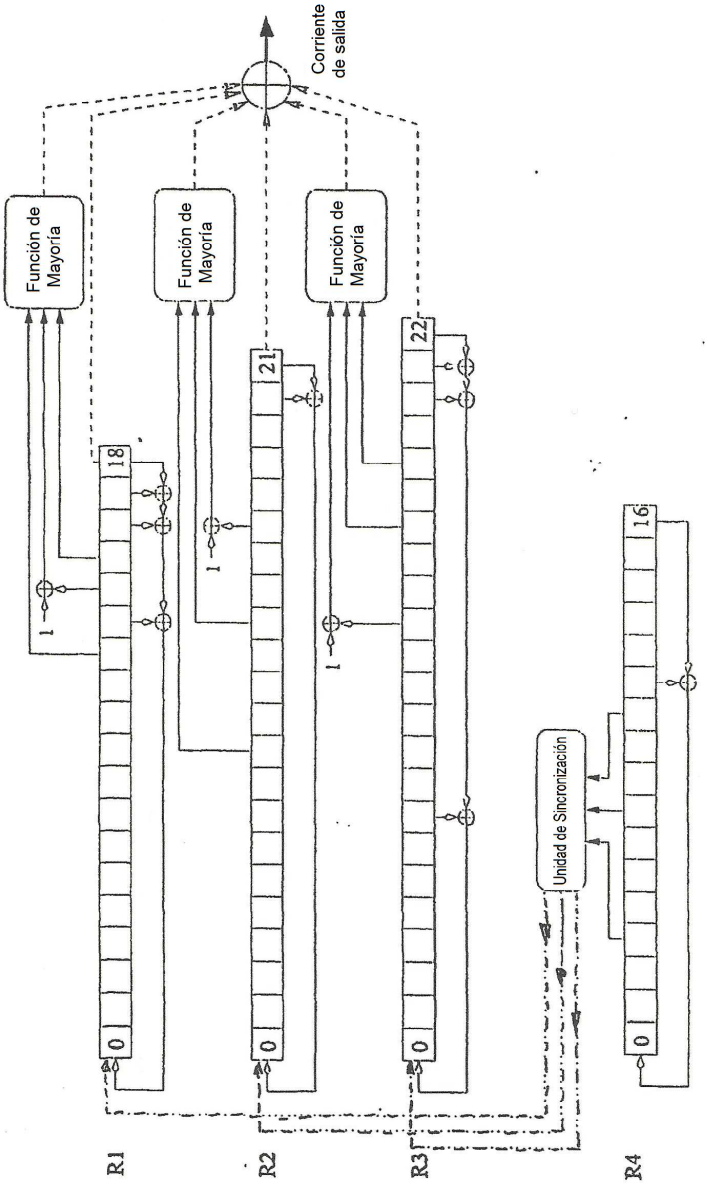
Corriente de salida
Función de Mayoría
Función de Mayoría
Función de Mayoría
Unidad de Sincronización
R1
R2
R3
R4
0
18
21
22
16
1

Fig. 4

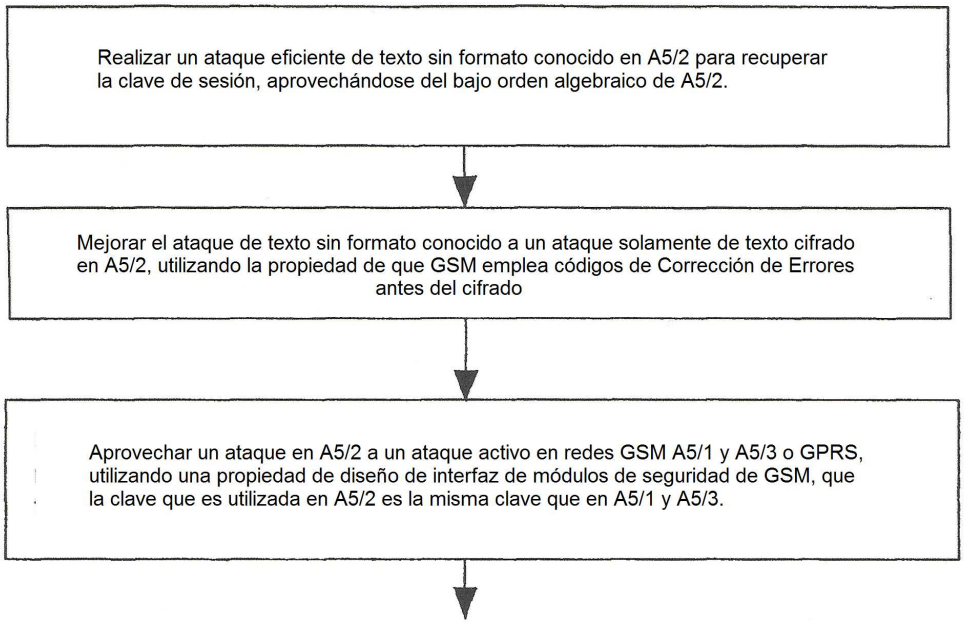
Realizar un ataque eficiente de texto sin formato conocido en A5/2 para recuperar la clave de sesión, aprovechándose del bajo orden algebraico de A5/2.
Mejorar el ataque de texto sin formato conocido a un ataque solamente de texto cifrado en A5/2, utilizando la propiedad de que GSM emplea códigos de Corrección de Errores antes del cifrado
Aprovechar un ataque en A5/2 a un ataque activo en redes GSM A5/1 y A5/3 o GPRS, utilizando una propiedad de diseño de interfaz de módulos de seguridad de GSM, que la clave que es utilizada en A5/2 es la misma clave que en A5/1 y A5/3.

Fig. 5

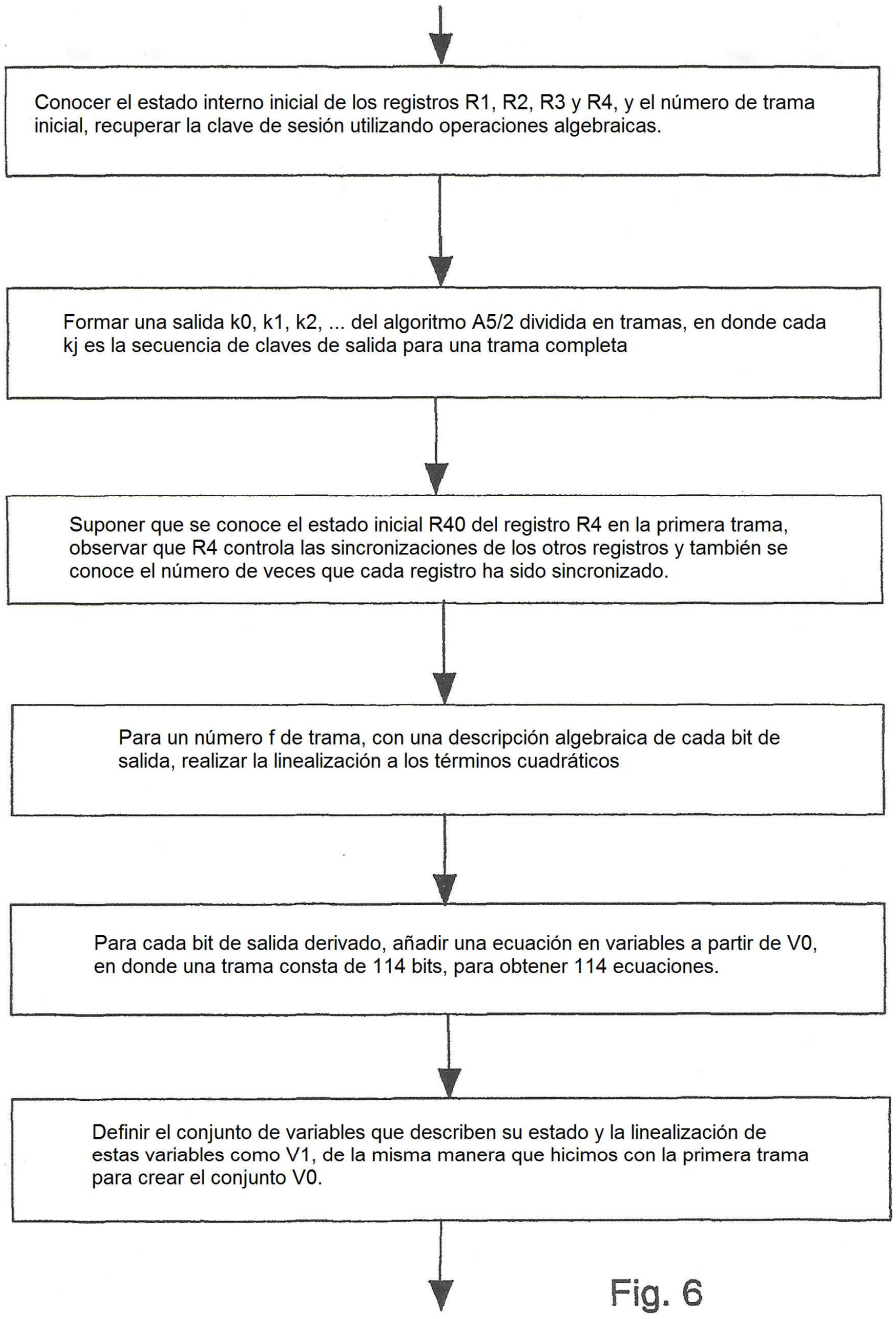
Conocer el estado interno inicial de los registros R1, R2, R3 y R4, y el número de trama inicial, recuperar la clave de sesión utilizando operaciones algebraicas.
Formar una salida k0, k1, k2, ... del algoritmo A5/2 dividida en tramas, en donde cada kj es la secuencia de claves de salida para una trama completa
Suponer que se conoce el estado inicial R40 del registro R4 en la primera trama, observar que R4 controla las sincronizaciones de los otros registros y también se conoce el número de veces que cada registro ha sido sincronizado.
Para un número f de trama, con una descripción algebraica de cada bit de salida, realizar la linealización a los términos cuadráticos
Para cada bit de salida derivado, añadir una ecuación en variables a partir de V0, en donde una trama consta de 114 bits, para obtener 114 ecuaciones.
Definir el conjunto de variables que describen su estado y la linealización de estas variables como V1, de la misma manera que hicimos con la primera trama para crear el conjunto V0.

Fig. 6