INTERCEPTAR Y ESCUCHAR LLAMADAS EN REDES MOVILES GSM

Decrytping teléfono GSM llama

A5 / 1 es un cifrado de flujo utilizado para proveer sobre-el-aire privacidad de la comunicación en el estándar de teléfono celular GSM. Se mantuvo inicialmente en secreto, pero se hizo de conocimiento público a través de filtraciones e ingeniería inversa. Un número de graves deficiencias en el sistema de cifrado se han identificado. El primero es un ataque activo. Los teléfonos GSM se puede convencer a utilizar el A5 mucho más débil / 2 cifrado brevemente. A5 / 2 se puede romper fácilmente, y el teléfono utiliza la misma clave que el más fuerte A5 / 1 algoritmo. Un segundo ataque en A5 / 1 se esboza, un texto cifrado, sólo hora memoria ataque desventaja que requiere una gran cantidad de precálculo.

PARTE 2
GSM Descifrado
Grabación de las llamadas
Datos GSM se puede grabar fuera del aire utilizando, por ejemplo, una radio programable tal como el USRP. Gnuradio proporciona las herramientas para grabar canales mientras gsm-receptor decodifica Airprobe de control del tráfico y en escenarios donde no se usa cifrado o donde la clave de cifrado es conocido, también decodifica el tráfico de voz.
Cracking A5 / 1
Cuando GSM utiliza encriptación A5 / 1, la clave secreta puede ser extraído de tráfico registrado. Dadas dos cifrados mensajes conocidos de texto claro, la utilidad Kraken que se ejecuta en un PC encuentra la clave secreta con una probabilidad de alrededor del 90% en cuestión de segundos en un conjunto de tablas de arco iris. Un conjunto conocido tabla actual tardaron 2 meses para calcular y contiene 40 mesas para un total de 2 TB.
Defensas
Corto plazo parches de protocolo ya existe lo que hace mucho más difícil agrietamiento por no revelar texto plano conocido innecesariamente (). Estos parches deben desplegarse con alta prioridad. A largo plazo, GSM (2G) no proporciona una seguridad suficiente y alternativas más potentes tales como el UMTS (3G) y LTE (4G) debe ser preferido.
Instrumentos

Las siguientes herramientas se utilizan para analizar las llamadas de voz:
- GNU Radio
- Airprobe
- Kraken
GNU Radio se incluye en las recientes distribuciones de Linux. Registro de datos requiere un receptor de radio programable tal como el USRP.
Airprobe está disponible a través de: (línea de comandos) git clone git :// git.gnumonks.org / airprobe.git. Por favor, siga a descifrar el tráfico GSM con Airprobe.
Kraken está disponible a través de: (línea de comandos) git clone git :// git.srlabs.de / kraken.git. Kraken utiliza las tablas del arco iris que están disponibles a través de Bittorrent.
GNU Radio
Tres maneras de empezar con GNU Radio:
Versión estable oficial:
- Los paquetes binarios
- El código fuente estable de liberación (tarball)
Código de desarrollo inestable:
- Inestable código fuente borde de corte de git
Si usted nunca ha utilizado antes de GNU Radio, se recomienda comenzar con una de las versiones oficiales. Si su distribución tiene listo para utilizar los paquetes se le anima a empezar con ellos. Es más fácil de instalar y obtendrá una instalación utilizable GNU Radio para iniciar desde. Comunicados oficiales le dará un código base estable para el desarrollo a largo plazo. Si su distribución o el sistema operativo no tiene paquetes, sin embargo, usted puede construir usted mismo de GNU Radio fuente. Esto puede ser un poco difícil. Si desea obtener la última y mejor, necesitamos algún rasgo añadido recientemente, o si quiere contribuir a GNU Radio, usted puede obtener la última versión del repositorio git.
Las instrucciones del sistema operativo específica
- Ubuntu haga clic aquí ...
- Windows, haga clic aquí ...
Hardware
El Universal Software Radio Peripheral o USRP2 es el dispositivo recomendado para la conexión de Radio GNU con el mundo real El USRP ha sido desarrollado especialmente para GNU Radio, y está disponible en Ettus Investigación (http://www.ettus.com). Es posible utilizar Radio GNU con otras E / S de hardware. También es posible utilizar el USRP con otro software de Radio GNU.
Instrucciones download source
Estable
La última versión estable de GNU Radio se puede encontrar aquí:
http://ftp.gnu.org/gnu/gnuradio/gnuradio-3.3.0.tar.gz
Un archivo de versiones anteriores se pueden encontrar aquí:
http://ftp.gnu.org/gnu/gnuradio/Desarrollo
Si prefiere el último código en desarrollo, desean contribuir a GNU Radio, o desean trabajar con las características que aún no lo han hecho en la rama estable, usted puede sacar de la fuente desde el

repositorio git usando una de estas dos líneas:
$ Git clone git :/ / gnuradio.org / gnuradio
o
$ Git clone http://gnuradio.org/git/gnuradio.git
Actualizar el repositorio local
$ Git pull
Edificio GNU Radio
Para compilar e instalar GNU Radio, usted puede descargar un archivo comprimido de liberación, o puede utilizar el software de cliente de git de revisar el código de nuestro repositorio git. Algunos sistemas operativos pueden tener los paquetes binarios de instalación disponible. Ver Sistemas Operativos Instrucciones específicas para Ubuntu y Windows. Para enterarse de las últimas http://gnuradio.org/releases/gnuradio estable código de liberación
En general, si está desarrollando aplicaciones GNU Radio que deben depender de la estabilidad de las características y API de los componentes de GNU Radio, se recomienda que utilice la serie 3.3 versión estable. Para tener una idea de lo que está pasando, clonar el repositorio (si no lo ha hecho ya), a continuación, ejecute "qgit" o uno de los otros espectadores git en él. Se le mostrará todas las ramas y uniones, diferencias, etc Asegúrese de que usted ha cumplido con las dependencias indicadas en el nivel superior.
Mayoría de los sistemas GNU / Linux vienen con nuestras dependencias ya envasados. Es posible que tenga que instalarlos fuera de la instalación del CD / DVD o en la red. Para compilar, hay 5 pasos. Comience por cd'ing al directorio gnuradio, a continuación, complete los siguientes comandos:
$. / Bootstrap # No realice este paso si usted está construyendo a partir de un archivo tar.
$. / Configure
$ Make
$ Make check
$ Sudo make install
Se realizará todos los controles de configuración y seleccionar para crear, probar e instalar todos los componentes que pasan.
Hacer frente a viejos problemas de compilación
Desinstalar gnuradio:
$ Sudo make uninstall
Para restaurar los archivos git originales de los archivos agregados durante una antigua construcción:
$ Git clean-d-x-f
A continuación, puede arrancar, configure, make, etc
La Radio GNU Build System Configuration
La Radio de código fuente GNU Build System utiliza el método estándar:
$. / Configure
$ Make
$ Sudo make install
o el uso de un suplente directorio de construcción:
$ Mkdir build

$ Cd build
$ .. / Configure
$ Make
$ Sudo make install
para construir e instalar el software. Este es un punto de partida suficiente para la mayoría de usuarios.
GNU Radio consta de un conjunto de bibliotecas de componentes, llamado simplemente "componentes" en lo que sigue. Cada componente vive en un directorio aparte de la raíz del árbol de código fuente, y tiene una serie de requisitos previos para que las pruebas de script 'configure'. Dependiendo del resultado de estos controles de configuración, el sistema de construcción ya sea selecciona o anula la selección de componentes individuales para crear, probar e instalar. El comportamiento por defecto es llevar a cabo todas las pruebas de configuración, crear todos los componentes que pasan, e ignorar las que no lo hacen.
Algunos usuarios pueden desear tener un mayor control sobre lo que se construye y lo que no, y el sistema de generación proporciona una facilidad para lograrlo. El sistema de construcción también permite la instalación incremental de componentes individuales.
Construir Opciones de configuración del sistema
Existen tres opciones para cada componente en el árbol de código fuente GNU Radio:
-Enable-foo
-Disable-foo
-With-foo [= arg]
donde * foo * es el nombre del directorio del componente a ser afectados, y para la última opción * arg * es la ruta completa al archivo pkg-config para foo ('foo.pc').
Hay también:
-Enable-todos los componentes-
-Disable-all-componentes
Esto se aplica a todos los componentes del árbol que no se sigue concretando en la línea de comando configure script.
Comportamiento predeterminado
Control de componentes individuales utilizando-enable-foo, foo-disable-, y-con-foo [= arg]
Al especificar-enable-foo hace que el sistema de compilación para considerarlo un error si fallan los controles de configuración para el componente * foo *, y salir con un error cuando eso ocurre ('error out'). Esta opción es adecuada para garantizar que un determinado componente se construye e instalado, o para entender por qué configurar con error cuando los controles fallan.
Al especificar-disable-foo impide que el componente de ser construido o instalado incluso si sus controles de configuración pasar. Esta opción es adecuada para los componentes que usted no está interesado o sabe de antemano que no va a pasar controles de configuración.
Al especificar-enable-all-componentes o-disable-all-componentes se aplica el comportamiento anterior a todo lo que no se especifique lo contrario más adelante en la línea de comandos.
Al especificar-con-foo hace que el sistema de compilación para buscar el archivo pkg-config, utilizando la variable de entorno PKG_CONFIG_PATH proporciona, por * foo * como evidencia de que el componente * foo * ya está instalado. Si * foo * aún no está instalado, el error será impreso una

configuración y se detendrá. Importantes esfuerzos se han hecho para asegurar que si-con-foo se especifica, y * foo * está instalado, las rutas a * foo * 's de las bibliotecas y archivos de cabecera no va a interferir con los del-enable'd componentes.
Speficying-con-foo = arg es el mismo que-con-foo, excepto que el componente * foo * se busca únicamente utilizando PKG_CONFIG_PATH = arg.
NOTA: Especificación de ambos-enable-foo y foo-con-dará como resultado un error.
NOTA: Cuando se utiliza-con-foo [= arg]: Debido a que el sistema de construcción utiliza pkg-config para localizar los componentes, PKG_CONFIG_PATH debe incluir la ruta completa (por ejemplo, "/ opt / local / lib / pkgconfig") de los paquetes a menos que sean instalado en el prefijo-siempre de configurar, ya que el camino está internamente antepone a PKG_CONFIG_PATH.

PARTE 3
Casos de uso común
Comportamiento predeterminado:
$. / Configure
Seleccione para construcción y todo lo que pasa instalación comprobaciones de configuración.
Comprobación de Todo:
$. / Configure-enable-all-componentes
Seleccionar todo para construcción e instalación, a excepción de la salida con un mensaje de error si algún cheque configuración no pasa. Es probable que esto no para todos, porque algunos controles de configuración específicas de la plataforma son mutuamente excluyentes (por ejemplo, los distintos módulos de audio para sistemas diferentes). Selectivo Contextura:
$. / Configure-enable-all-components \
-Disable-foo1 \
-Disable-foo2
donde * foo1 * y * foo2 * son componentes que pueden fallar y no les importa, o que no quieren construir lo que pase. Si no desea que la compilación falle si falla todo lo demás comprobaciones de configuración. Por ejemplo, en un sistema Linux que de lo contrario tiene todas las bibliotecas dependientes de construir instalados:
$. / Configure-enable-all-components \
-Disable-gr-audio-osx \
-Disable-gr-audio-windows \
-Disable-gr-audio-oss
tendrá en cuenta los módulos de la plataforma no Linux, así como el material de audio OSS (que de otra manera recopilar ok). Construir con componentes preinstalados:
$. / Configure-disable-todos los componentes-\
-Enable-foo1 \
-Con-foo2
a tratar de construir componente * foo1 * usando sólo la biblioteca ya instalados y los archivos de cabecera para el componente * foo2 * como se encuentra por pkg-config en la PKG_CONFIG_PATH proporcionada. Construir con componentes preinstalados en un lugar específico:

$. / Configure-disable-todos los componentes-\
-Enable-foo1 \
-With-foo2 = bar
a tratar de construir componente * foo1 * usando sólo la biblioteca ya instalado y los archivos de cabecera para el componente * foo2 * como se encuentra por pkg-config en PKG_CONFIG_PATH = bar.
Componente Mínimo Construye
Para seleccionar un solo componente para construir e instalar utilizar:
$. / Configure-disable-todos los componentes-\
-Enable-foo
donde * foo * es un directorio único componente. Esto hará que la compilación falle si * foo * no pasar los controles de configuración, ignorando a los otros paquetes.
ADVERTENCIA: Los componentes individuales de Radio GNU generalmente dependen de otros componentes (por ejemplo, * gnuradio-core *) para compilar con éxito. Cuando se utiliza-disable-all-componentes de cada dependencia componente debe ser especificado a través de un a habilitar o con la opción. El sistema de construcción se producirá un error si no se cumplen las dependencias, y lo más probable es que construyen componente se saltará.
Por ejemplo, el componente * gr-USRP * depende de los componentes * USRP * y * gnuradio-core *. El primero depende de los componentes * omnithread *, * MBLOCK *, y * pmt *. Este último depende del componente * omnithread *, que es redundante con las dependencias * USRP * 's. Por lo tanto, a partir de un código fuente fresca construir sin nada instalado, habría que utilizar la siguiente línea de comandos con el fin de tratar de construir * gr-USRP *:
$. / Configure-disable-todos los componentes-\
-Enable-gr-USRP \
-Enable-gnuradio-core \
-Enable-USRP \
-Enable-pmt \
-Enable-MBLOCK \
-Enable-omnithread
El orden de estos en la línea de comandos de configuración no es importante, el sistema de generación conoce el orden correcto en el que para construirlos. Como otro ejemplo, supongamos que los componentes * USRP * (y sus dependientes) ya están instalados, a través de los comandos:
$. / Configure-disable-todos los componentes-\
-Enable-USRP \
-Enable-pmt \
-Enable-MBLOCK \
-Enable-omnithread
$ Make
$ Sudo make install
Entonces uno podría usar esos componentes pre-instalados y construir * gr-USRP * a través del comando:

$. / Configure-disable-todos los componentes-\
-Enable-gr-USRP
-Enable-gnuradio-core \
-Con-USRP \
-With-pmt \
-Con-MBLOCK \
-Con-omnithread
Haciendo Tarballs distribución
Independientemente de lo que se está habilitado, deshabilitado, o incluido via-con-foo [= arg], el 'make dist' y 'make distcheck' operaciones creará un tarball fuente de distribución de la colección completa de los componentes.
Airprobe
Información general sobre el proyecto se puede encontrar en el Wiki. El código fuente está en el git.
Cómo usarlo:
$ Git clone git :// / svn.berlin.ccc.de / airprobe
Echa un vistazo a la hoja de ruta para los hitos actuales que necesitan su contribución. Siéntase libre para generar las entradas para estos hitos y trabajar en ellas.
Estructura
AirProbe se divide en tres subproyectos principales: adquisición, demodulación y análisis.
Adquisición
El módulo de adquisición es dependiente del hardware y contiene todo lo que tiene que ver con la recepción y digitalización de la interfaz aérea. Esta es la parte que necesita ser reescrita para el hardware receptor diferente, por lo que debe ser pequeña y limitada a las funciones necesarias. La mayoría de las partes debe ser heredado de gnuradio, para mantener la carga de trabajo limitada.
Demodulación
El módulo de demodulación contiene todo el código necesario para que los bits de la señal captada por Adquisición. Es en principio independiente del hardware, pero debe ser abierto para utilizar DSPs se desea.
Análisis
Este módulo contiene todas las interpretaciones protocolo y decodificación. Wireshark se puede utilizar para manejar partes de la visualización y tareas de interfaz de usuario. Una parte importante del módulo de análisis no es en tiempo real descifrado A5 basado en una CPU rápida genérico. Dercyption A5 en tiempo real o casi en tiempo real, no es un objetivo del proyecto. Para efectos de análisis de protocolo y la demostración de inseguridades, no en tiempo real de descifrado es suficiente.
Hardware
Descripción de algunos dispositivos que podrían ser utilizados para el proyecto.
USRP
Universal Software Radio Peripheral es un dispositivo que le permite crear una radio software. El dispositivo está conectado al ordenador a través de USB o puerto ethernet Gigagbit. El procesamiento de señales se realiza con software (GNU Radio) en el ordenador. USRP está diseñado para permitir a las computadoras de propósito general para funcionar como radios de alto ancho de banda de software,

que está haciendo todo el procesamiento de formas de onda específica, como la modulación y demodulación, en el equipo host. USRP tiene un diseño abierto y todo el software, incluidos los controladores, es opensource.
USRP constan de una placa base y tarjetas hija USRP varios USRP, que sirven como interfaz de RF. Doughterboards son receptores y transmisores y transceptores. No puede ser desde 2 hasta 4 placas secundarias conectadas a USRP. Hay dispositivo USRP y USRP2, el primero está disponible desde septiembre de 2008.
USRP funciona bajo Linux (kernel 2.6), Windows, Max OS X (PPC e Intel), FreeBSD y NetBSD.
GNU Radio es un conjunto de herramientas de desarrollo de software libre que proporciona el entorno de ejecución de procesamiento de señales y bloques de procesamiento para implementar radios de software. Aplicaciones GNU radio son principalmente escrito utilizando el lenguaje de programación Python, mientras que la suministrada, el rendimiento crítico ruta de procesamiento de la señal se implementa en C + + con extensiones para procesador de punto flotante cuando estén disponibles. Por lo tanto, el desarrollador puede implementar en tiempo real, de alto rendimiento sistemas de radio de una forma sencilla de usar, rápido desarrollo de aplicaciones medio ambiente. Aunque no es principalmente una herramienta de simulación, GNU Radio sí apoyan el desarrollo de algoritmos de procesamiento de señales a partir de datos previamente grabados o generados, evitando la necesidad de hardware real a la radiofrecuencia. En Ubuntu se puede instalar por gnuradio paquete.
Así que para la recepción de señales GSM que usted necesita:
- USRP (700 USD)
- DBSRX placa hija, que puede recibir frecuencias de 800 MHz a 2,4 GHz (150 USD)
- LP0926 900 Mhz - 2.6 Ghz Antena Log periódica PCB (35 USD)
- SMA-M a SMA-M Cable (30 USD)
- GNU Radio software (sin costo)

PARTE 3
Airprobe uso
Este es un howto principiantes que ilustra los pasos de trabajo con airprobe primera y la USRP. Está dirigido a los principiantes a ser un poco más fammiliar con airprobe y la interfaz de radio GSM.
Herramientas que usted debe tener:
- Gnuradio, Airprobe y USRP
- Baudline
- Usrp_bimbo
- Arfcncalc
- La última versión de desarrollo de Wireshark

Preparación de la USRP
Para el registro de datos GSM-su USRP debe estar equipado con el DBSRX-Board en RX A. Usted también debe tener una antena adecuada, sino un trozo de cable también debe hacer el trabajo. Es

bueno vivir en una gran ciudad, porque en las zonas urbanas que tendrá un montón de células pequeñas en una malla thight. Usted debe compilar e instalar la versión más reciente gnuradio.

Ensayos adecuados fuente de señal

Es bueno tener una fuente de señal de prueba tal como un GSM-estación base o una unidad de prueba GSM-Radio. Productivos redes GSM son buenas fuentes de señal también, pero olfateando ellos podría ser un delito punible. Por lo tanto, se recomienda llevar a cabo todas las pruebas en un entorno protegido profesional.

Seguimiento del espectro

La primera de ellas debe hacer es controlar el espectro para ver lo que está pasando en el barrio. La mejor manera de hacer esto con éxito es tomar cualquier número ARFCN del espectro que desea supervisar. Siga estos pasos para utilizar "gsm-receptor" de Airprobe para depurar sin saltos, las células individuales ARFCN en dirección descendente.

Capturar parte del tráfico de enlace descendente de un no-hopping, célula ARFCN único utilizando el USRP o USRP2 (tasa de diezmado recomendado para la USRP2 es 174, la tasa de diezmado por defecto para el USRP es 112).

Una captura USRP2 muestra está aquí:

http://reflextor.com/vf_call6_a725_d174_g5_Kc1EF00BAB3BAC7002.cfile.gz

Iniciar una reciente versión Wireshark escuchando el puerto GSMTAP (UDP 4729).

NOTA: "174" en los siguientes ejemplos es la tasa de diezmado utilizado para la captura USRP2, "go_ usrp2.sh" se utiliza porque es una captura de USRP2. Para el USRP1 "go.sh" se utiliza en lugar de "go_usrp2.sh" y "112" en lugar de "174".

Decode TS0 como "Non-combinado" configuración de BTS ("0B" parámetro). Para un "combinado" de configuración especifica "0 º C". Si no está seguro, pruebe ambos y tomar el que tiene más sentido o mirar "Información del sistema 3", "Descripción del canal de control".

./go_usrp2.sh vf_call6_a725_d174_g5_Kc1EF00BAB3BAC7002.cfile 174 0B> vf_call6

Busque la sección "Asignación inmediata" en la traza Wireshark GSMTAP.

==> SDCCH / 8 sobre TS1 se asigna

Decode TS1 como "SDCCH / 8" ("1s" parámetro)

./go_usrp2.sh vf_call6_a725_d174_g5_Kc1EF00BAB3BAC7002.cfile 174 1S> vf_call6

La parte no cifrada de la SDCCH / 8 se muestra en Wireshark. La parte cifrada de SDCCH / 8 figura en el archivo "vf_call6", esos son los marcos donde se muestra un error (por ejemplo, "no puede decodificar fnr = 0x0d288b (862.347) ts = 1").

Archivo "vf_call6" contiene el estallido de los marcos, para el primer fotograma codificado las ráfagas son:

C1 862344 1332354: 00111100011000010010111000011011111100010111100010100101111100110010011001001101100000010000010101001100100001001

P1 862344 1332354: 00111100011000010010111000011011111100010111100010100101111100110010011001001101100000010000010101001100100001001

S1 862344 1332354: 00000000000000000000000000000000000000000000000000000000000000000000

00000000000000000000000000000000000000000000000000
C0 862345 1332387: 0100011000011110000100101111110001011011000110101010100010111 0
0010101101001000100101010110001011011010110111001010
P0 862345 1332387: 0100011000011110000100101111110001011011000110101010100010111 0
0010101101001000100101010110001011011010110111001010
S0 862345 1332387: 00000000000000000000000000000000000000000000000000000000000000 0
00000000000000000000000000000000000000000000000000000
C0 862346 1332420: 1101000010100101011001100011010010101000110010111000100101100 0
0001101011111000011100111011000001100000111010010101 1
P0 862346 1332420: 1101000010100101011001100011010010101000110010111000100101100 0
0001101011111000011100111011000001100000111010010101 1
S0 862346 1332420: 00000000000000000000000000000000000000000000000000000000000000 0
00000000000000000000000000000000000000000000000000000
C0 862347 1332453: 1111100001111110101001101010000001001011000010000111110111000 0
1111010001101111001001111001101111111001110101001000
P0 862347 1332453: 1111100001111110101001101010000001001011000010000111110111000 0
1111010001101111001001111001101111111001110101001000
S0 862347 1332453: 00000000000000000000000000000000000000000000000000000000000000 0
00000000000000000000000000000000000000000000000000000
error: SACCH: Error de paridad (-1 fn = 862.347)
no puede decodificar fnr = 0x0d288b (862.347) ts = 1

"Cx" son los bits codificados explosión, "px" son los bits de ráfaga descifrados y "SX" son los bits (XOR ristra de bits de cifrado descifrados bits). No tenemos en estos momentos para descifrar los bits de ráfaga descifrados son los mismos que los bits de ráfaga cifrados. Si "x" es "1" que se trata de la primera explosión de un marco.

El segundo número es el número de trama, el tercer número es el "modificado" número de trama como es requerido por la A5 / algoritmo 1. Elija una ráfaga donde se sabe que el contenido de la trama y lo utilizan para encontrar Kc. Utilice la herramienta de Kraken para encontrar Kc. El Kc resultante es 1EF00BAB3BAC7002.

NOTA: este paso no es tan fácil como parece. Por lo general, captar algunas llamadas de su propio teléfono donde se sabe que el Kc (se puede leer desde la tarjeta SIM o muestre en la pantalla del modo de Ingeniería de algunos teléfonos) y buscar conocidos de texto sin formato candidatos. Un ejemplo son "5/6/5ter Información del sistema" en la SACCH o "U LAPDM, func = interfaz de usuario" frames. También hay que tener en cuenta que podría haber bits erróneos en una explosión debido a la distorsión.

Decode TS1 como "SDCCH / 8" ("1s" parámetro) y descifrar (Kc se especifica como parámetro)

./go_usrp2.sh vf_call6_a725_d174_g5_Kc1EF00BAB3BAC7002.cfile 174 1S 1EF00BAB3BAC7002> vf_call6

Busque el "Comando de asignación" en la traza Wireshark GSMTAP.

==> TCH / F en TS5 se asigna
Decodificar y descifrar el tráfico de voz sobre TS5 ("5T" parámetro, Kc se especifica)
./go_usrp2.sh vf_call6_a725_d174_g5_Kc1EF00BAB3BAC7002.cfile 174 5T 1EF00BAB3BAC7002> vf_call6
El archivo "speech.au.gsm" contiene el tráfico de voz. Se puede convertir en "tostadas" (http://www quut.com / gsm /.) A un archivo de audio:
brindis-d speech.au.gsm
El archivo resultante "speech.au" se pueden reproducir., /

Rainbow Tablas
Clasificación y uso compartido de tablas de arco iris
Si ya calculado tablas de arco iris con las herramientas indicadas más adelante, entonces es hora de hacer sus tablas utilizables a través de ordenarlos. Por favor, también comparten las tablas a través de Bittorrent. El proceso de clasificación se explica en este tutorial rápido (y requiere de este programa). Actualmente no hay ningún código fuente de trabajo para compilar para otras arquitecturas.

Creación de A5 / 1 Tablas Rainbow
Para comenzar a contribuir tiempo de cálculo, por favor póngase única tabla de parámetros
http://reflextor.com/cgi-bin/a51/a51id.cgi
Obtener una copia de trabajo del generador de tablas mediante: Descarga de binarios
http://reflextor.com/trac/a51 o compilar el programa (código disponible a través de:. Svn co
https://svn.reflextor.com/tmto-svn Entonces ejecutar el programa

Opciones de configuración
NOTA: La única manera de que el software se debe utilizar en este momento es correr con un conjunto fijo de opciones sin desviación. Como aún es software alfa, otros usos para revelar ciertamente bugs. Algunas de las características documentadas no se aplican en realidad, pero será en el futuro próximo. La única razón para dar de software incompleta es el hecho de que la generación de tabla lleva mucho tiempo y debe iniciarse tan pronto como sea posible.
Estas son las opciones que se deben utilizar ahora para generar tablas. Cada tabla que se genera tiene que tener una semilla diferente inicial para el generador de funciones redonda, que se administra en los argumentos de la opción antes de la sub-roundfunc: Opción de generador. Como se puede ver la semilla es cero en el ejemplo. Utilice el cgi arriba para solicitar una presentadas. En esta fase de despliegue, los desarrolladores podrán responder a todas sus preguntas y comprobar pequeñas porciones de las tablas generadas en contra de una implementación de referencia (juego las cadenas de opción del comando generate a algo así como 1024 y enviar en la tabla generada, por ejemplo mediante la creación de un billete trac).

Con la construcción, en el archivo de configuración por defecto, la invocación es el siguiente:
$ A51table de avance XXX-consumir archivo: prefix = datos: append = red apodo (nombre nombreDeServidor): password = (your_passwd): host = reflextor.com: port = 80-512 operaciones generar
Escrito con todo detalle que equivale a:
-Estado rondas: rondas = 32
-Roundfunc xor: Estado = distinguished_point :: bits = 15: = generador LFSR :: tablesize = 32 :: avance = 0
La red apodo = su_nombre: password = (your_passwd): host = reflextor.com: port = 80
-Implementación SHAREdmem
-Algoritmo A51
-Device cuda: Operaciones = 512
-Trabajar al azar: prefix = 11,0
-Consumir archivo: prefix = datos: append
Normal-logger
generar
Cadenas 270000000
Longitud de cadena-3000000
Intermedio filtro: longitud de recorrido = 512

Formatos de tabla
En el curso del proyecto 3 diseños diferentes en el listado de las principales fueron utilizados. Los 2 primeros fueron similares en que ambos utilizan los 64 bits inmediatos que se generaron por la A5 / motor 1 como entrada para la siguiente ronda. El tercero primero registró 100 pasos hacia adelante y luego los siguientes 64 bits se utilizaron. El primer formato fue dp15k32, puntos destacados del tamaño de 15 bits con 32 rondas. Más tarde se descubrió que la asunción de un marco único LAPDm de texto plano conocido era bastante conservador, así que cambiaron a dp15k8. Debido al mayor número de muestras disponibles ristra, más se podía hacer búsquedas y las tablas necesarias para cubrir menos del espacio de claves conjunto. Manteniendo nuestro objetivo de tamaño de la tabla 2 TB, apuntaban a reducir los tiempos de búsqueda, junto con la evidente menor tiempo de generación de la tabla. Ese objetivo se cumplió con la reducción en el número de funciones circulares.
Por último, con el descubrimiento de que sólo el 16% del espacio de claves era aún accesible en el mundo real después de los 100 marcajes que forman parte del A5 / 1 Funcionamiento en redes GSM, cambiaron allí formato dp 12k8e100, 12 puntos trozos distinguidos, rondas 8 y 100 marcajes adicionales. El espacio de claves cubierto con estas nuevas tablas era 8 veces más pequeño (como resultado de ir de DP15 DP12 a), que es aproximadamente en alineación con el hecho de que sólo 16% del espacio de claves importa de todos modos. Junto con el hecho de que cada ronda toma ciclos 164/64 = 2,5 veces más para calcular, tiempo de generación y tiempo de búsqueda se redujo en un factor de 8/2.5 == 3,2 en comparación con el formato dp15e8.

Los dos formatos más antiguos no son óptimas, pero si usted puede permitirse los módulos flash de memoria adicionales, puede ser utilizado en la parte superior de la 2TB de nuevas tablas están generando.

Tiempo de búsqueda

Hay 2 momentos diferentes: el tiempo mínimo para una búsqueda única y tiempo de búsqueda promedio para un flujo infinito de búsquedas. De cada muestra keystream, 8 cadenas de búsqueda se pueden generar para cada tabla, a partir de una de las rondas. Cada una de dichas cadenas se calcula entonces hasta el punto final, la búsqueda se realiza a la tabla en el dispositivo de almacenamiento y el valor de arranque del disco se utiliza entonces para calcular una nueva cadena hasta el punto en que se inició la cadena de búsqueda. Las 2 partes de la cadena de búsqueda son tan largos como una cadena almacenada en el disco, y un poco más (alrededor de la mitad de una longitud DP). Suponiendo que hay 40 tablas y 400 muestras de corriente de clave para buscar: si 40 * 8 * 400 = 128000 cadenas pueden ser procesadas por un motor de hardware en particular en paralelo, entonces el tiempo de búsqueda mínimo puede ser alcanzado. Si el hardware no es lo suficientemente ancha, entonces algunas de las cadenas tienen que ser calculado después de que otros han terminado. Si la anchura de hardware es mayor que 128000 entonces el tiempo de búsqueda promedio será menor que el tiempo de búsqueda mínimo. Con una sola GPU HD5870, 320 núcleos físicos operan en rebanadas de 32 bits de ancho, por lo que el ancho de hardware es 10240. Debido a que cada núcleo físico puede funcionar en 4 rebanadas de 32 bits en el mismo tiempo como uno solo, es seguro decir que el HD5870 tiene 1280 núcleos y puede operar en 40960 cadenas a la vez. Para llegar a tiempo de búsqueda mínimo, 3 HD5870 son obligatorios. Pero debido a que el tiempo necesario para procesar hilos 5120 es menos de 4 veces el tiempo necesario para procesar las roscas 1280, una HD5870 único sería más rápido que el tiempo mínimo de 3 *