Šifrování v mobilní telefonii

Referát do předmětu M0170 Kryptografie: Jiří Kalina, UČO 150824

1. Úvod

Bezdrátové komunikační systémy s větším počtem přijímačů a vysílačů komunikujících na nezávislých linkách lze v principu rozdělit na tři základní skupiny: FDMA (Frequency Division Multiple Access), TDMA (Time Division Multiple Access) a CDMA (Code Division Multiple Access), ve kterých se kapacita komunikačních kanálů rozděluje mezi jednotlivé účastníky po řadě na principu frekvence, času a datového kódu [@2].

Nejstarším standardem pro mobilní telefonní komunikaci v českých zemích se stal v roce 1991 spuštěný FDMA systém NMT (Nordic Mobile Telephone) společnosti Eurotel. Mobilní telefony tohoto standardu využívaly analogový signál na frekvenci 450 MHz a umožňovaly pouze hlasové přenosy, později doplněné o přenosy textových zpráv a dat DMS. Na totožné frekvenci pracuje dodnes používaný systém pro datové přenosy CDMA (U:fon). Zpočátku neumožňovaly sítě NMT žádné šifrování a hovor mohl být poměrně snadno odposloucháván, později byla doplněna možnost analogového scramblingu.

Provoz NMT sítě byl v ČR ukončen v roce 2006, mezitím však byla již od roku XXXX budována druhá generace mobilních sítí TDMA GSM (původní zkratka Groupe Spécial Mobile), která je v současnosti globálně i v ČR zdaleka nejrozšířenějším standardem mobilní hlasové komunikace. Sítě GSM využívají od počátku jednotně šifrovací algoritmy rodiny A5 na rozhraní Um, tedy mezi základnovou stanicí BTS a mobilním telefonem. Při navázání komunikace se podle úrovně technologie podporované vysílačem a telefonem ustanoví spojení nešifrované (A5/0) nebo šifrované podle některého z pokročilejších algoritmů.

Při použití nejrozšířenějšího algoritmu A5/1 generátor vytváří proudovou šifru, která se bit po bitu sčítá s původními daty. Šifra tedy pracuje na první fyzické vrstvě, kterou lze přímo odposlechnout [@ 1]. Záměrně zjednodušenou variantou algoritmu A5/1 je algoritmus A5/2 vyvinutý pro šifrování mobilní komunikace ve východoevropských a asijských zemích, do nichž nesměla být technologie šifrování A5/1 z politických důvodů vyvezena [@5]. Algoritmus A5/2 je extrémně slabý a byl prolomen během necelého měsíce od zveřejnění v roce 1999, od roku 2006 se na základě doporučení GSMA (GSM Association) od použití algoritmu A5/2 plošně upouští. Současné telefony nabízí použití pokročilejších algoritmů A5/3 a A5/4 využitelných také pro šifrování datových přenosů.

2. A5/1

Algoritmus A5/1 využívá poněkud komplikovaného mechanizmu, kdy je každých 4,615 ms ke zdrojovým datům rozděleným do burstů o délce 114 bitů + synchronizační údaje burstu [@7] přičítán kód generovaný pomocí trojice lineárních posouvacích registrů, tajného 64 bitového klíče a známého 22 bitového čísla TDMA rámce, v němž právě komunikace probíhá. Tři lineární registry mají délku 19, 22 a 23 bitů v uspořádání po řadě $x^{19}+x^{18}+x^{17}+x^{14}+1$, $x^{22}+x^{21}+1$ a $x^{23}+x^{22}+x^{21}+x^{8}+1$. Výsledný bit na výstupu je dán exkluzivním součtem (xor) výstupů všech tří lineárních posouvacích registrů. Komplikace však nastává v mechanizmu posouvání registrů – po vygenerování každého bitu na výstupu, jsou porovnány tři pulzní pozice (kloky) v registrech (po řadě 8. bit, 10. bit a 10. bit) a pokud se jeden bit na této pozici odlišuje od zbývajících dvou, není v daném cyklu registr posunut. Zbývající registry se posunou a dochází k vygenerování bitu na výstupu. Snadno se lze přesvědčit, že pravděpodobnost posunutí každého registru je rovna ¾.

V praxi jsou na počátku všechny tři lineární posouvací registry nulové a po 64 cyklů jsou na vstup všech tří registrů přičítány po řadě příslušné bity tajného klíče. V těchto 64 cyklech dochází k posunu registru vždy, nezávisle na hodnotách v pulzních pozicích (klocích). V dalších 22 cyklech se stejným způsobem přičítají bity veřejného rámce. Následuje 100 cyklů, jejichž výstup není žádným způsobem zpracováván a poté konečně dva 114 bitů dlouhé úseky výsledného kódu, z nichž jeden je přičten k příchozímu, druhý k odchozímu datovému burstu.

Ve starších aplikacích bylo využito pouze 54 pozic 64 bitového klíče, neboť původní implementace Comp128v1 přibližně do roku 2007 [@7] přidávala na začátek klíče 10 nul.

Útoků na šifru A5/1 bylo v historii publikováno již několik. Zatímco v roce 2008 představili David Hulton a Steve Muller jako převratnou novinku zařízení a algoritmus, které dovedou pasivně dešifrovat zachycené hovory v síti GSM během 30 minut při nákladech na pořízení přibližně 1 000 USD [@3], bylo již na konci roku 2011 možné většinu komunikace v sítích GSM dešifrovat za pomoci stolního počítače s výkonnější grafickou kartou.

Jeden ze servisních burstů má vždy známou podobu, útočník tedy zná původní a zachytí podobu šifrovanou (tzv. known-plaintext attack). To mu umožní sestavit tabulku použitých klíčů a kódovou knihu.

Touto cestou se v roce 2009 vydal Karsten Nohl [@3]. Kódová kniha šifry A5/1 má hrubou velikost 128 PiB a její výpočet by běžnému procesoru trval řádově desítky tisíc let. Při využití 40 grafických procesorů založených na architektuře CUDA trvalo sestavení kódové knihy asi tři měsíce. Celková velikost se díky optimalizaci a tzv. rainbow tables smrskla na celkovou velikost pouhé 2 TiB. Výsledné tabulky jsou sdíleny na bittorentu.

3. A5/2

Zejména z důvodu možného použití stejných hardwarových prvků byl algoritmus A5/2 snižující úroveň bezpečnosti algoritmu A5/1 postaven na stejném základě. Lineární posouvací registry byly zachovány, jejich posouvání (klokování) však není zajištěno pomocí porovnání pulzních pozic, ale čtvrtým lineárním posouvacím registrem o délce 17 bitů a uspořádání $x^{17}+x^5+1$. Tento čtvrtý registr je posouván v každém cyklu vždy po posunutí prvních tří registrů slouží výhradně k jejich klokování tím způsobem, že jsou (stejně jako v A5/1) porovnány pulzní pozice prvních tří registrů (po řadě 8. bit, 10. bit a 10. bit) a je určena převažující hodnota (buď 0 nebo 1). První registr je v daném cyklu klokován pouze, pokud se převažující hodnota rovná hodnotě třetího bytu ve čtvrtém registru, obdobně druhý, resp. třetí registr jsou klokovány pouze v případě, že se převažující hodnotě rovnají sedmý, resp. desátý bit čtvrtého registru.

Další drobnou odlišností je průběh inicializace, kdy jsou po proběhnutí prvních 64 cyklů odpovídajících tajnému klíči a 22 cyklů odpovídajících číslu rámce nastaveny ve všech registrech po řadě (v každém jeden) 15., 16., 18. a 10. bit na 1 a následně neprobíhá 100 cyklů, ale pouze 99 cyklů naprázdno. Pak teprve následuje generování výsledného kódu.

Výstupní bit algoritmu je dán exkluzivním součtem (xor) tří hodnot, z nichž každá představuje pro jeden registr většinovou hodnotu dvou vybraných bitů a opačné hodnoty třetího vybraného bitu. V prvním registru jde o 16. a 13. bit a opačnou hodnotu 15. bitu, v druhém registru jde o 10. a 14. bit společně s opačnou hodnotou 17. bitu a ve třetím registru o 17. a 19. bit a opačnou hodnotu 14. bitu [@5].

Generování dvou 114 bitových sekvencí je dále totožné s průběhem algoritmu A5/1.

4. A5/3

Přestože byl pro 3. generaci mobilních telefonů (po FDMA analogových NMT a TDMA digitálních GSM jde o širší škálu obvykle CDMA standardů pro rychlý přenos dat, např. UMTS, CDMA2000 aj.) vyvíjen nový šifrovací algoritmus, vzhledem k tlaku na rychlé uvedení sítí 3. generace do provozu rozhodl nakonec ETSI (European Telecommunications Standards Institute) v roce 2005 o využití tou dobou již užívaného šifrovacího algoritmu MISTY, vyvinutého japonskou společností Mitsubishi Electric Corporation [zdroj].

Algoritmus byl částečně upraven, rozšířen a přejmenován na KASUMI, což znamená japonsky mlha, stejně jako anglické mist.

Základním, ve struktuře algoritmu několikrát opakovaným, principem A5/3 je Feistelova síť – šifrovací algoritmus, který umožňuje data zašifrovat i dešifrovat víceméně totožným postupem, což je výhodné zejména vzhledem k hardwarovým nárokům. Vypuštěním lineárních posouvacích registrů a využitím Feistelovy sítě se tak algoritmus zásadně liší od svých předchůdců A5/1 a A5/2.

Páteří A5/3 je osmirundová[1] Feistelova síť, která šifruje 64 bitové datové bloky pomocí 128 bitového klíče. V první řadě je vygenerováno pro každou z 8 rund 8 rundových klíčů $KL_{i,1}$, $KL_{i,2}$, $KO_{i,1}$, $KO_{i,2}$, $KO_{i,3}$, $KI_{i,1}$, $KI_{i,2}$, $KI_{i,3}$ o délce 16 bitů, kde i označuje pořadí rundy, následujícím způsobem [@8]:

1. 128 bitový klíč je rozdělen na osm 16 bitových podklíčů K_1 až K_8.
2. Ke každému podklíči K_j je dopočítán doplňkový podklíč K'_j exkluzivním sečtením (xor) s příslušnou konstantou C_j. Konstanty C_j vzniknou po řadě rozdělením konstanty C=0123456789ABCDEFFEDCBA9876543210 na osm 16 bitových konstant.
3. Rundový klíč $KL_{i,1}$ vznikne z podklíče K_i posunutím o 1 bit vlevo.

   Rundový klíč $KL_{i,2}$ je roven doplňkovému podklíči K'_{i+2}.

   Rundový klíč $KO_{i,1}$ vznikne z podklíče K_{i+1} posunutím o 5 bitů vlevo.

   Rundový klíč $KO_{i,2}$ vznikne z podklíče K_{i+5} posunutím o 8 bitů vlevo.

   Rundový klíč $KO_{i,3}$ vznikne z podklíče K_{i+6} posunutím o 13 bitů vlevo.

   Rundový klíč $KI_{i,1}$ je roven doplňkovému podklíči K'_{i+4}.

   Rundový klíč $KI_{i,2}$ je roven doplňkovému podklíči K'_{i+3}.

   Rundový klíč $KI_{i,3}$ je roven doplňkovému podklíči K'_{i+7}.

   Přičemž pokud dolní index u některého subklíče překročí číslo 8, uvažuje se dolní index o 8 menší.

Po vygenerování rundových klíčů algoritmus vstoupí do první rundy: 64 bitový datový blok I_0 je rozdělen na poloviny L_0 a R_0. V každé rundě je pak provedena záměna levého bloku za pravý $R_{i+1}=L_i$ a levá polovina nového datového bloku vznikne exkluzivním sečtením (xor) pravé strany s výsledkem funkce F, která je rozvedena níže: $L_{i+1}=F_i(KL_i,KO_i,KI_i,L_i) \oplus R_i$. Výstupem algoritmu je datový blok z osmé rundy $I_8=L_8 \,||\, R_8$.

Definice funkce F_i vnáší do algoritmu hlavní komplikaci. Funkce F_i je definována pomocí trojice funkcí FL, FO a FI jinak v sudých a lichých rundách:

[1] Použití slova runda je jazykovým přesahem původního německého die Rund, tj. kolo, cyklus, do angličtiny běžně překládaného jako the round.

- v lichých rundách $F_i = FO(FL(L_i, KL_i), KO_i, KI_i)$,
- v sudých rundách $F_i = FL(FO(L_i, KO_i, KI_i), KL_i)$.

Zbývá uvést definici funkcí FL, FO a FI:

- ve funkci $FL(L_i, KL_i)$ je 32 bitový vstup L_i rozdělen na dvě poloviny $LL_{i,0}$ a $LR_{i,0}$ a v dalším kroku je vypočtena:
    - nová pravá strana $LR_{i,1}$ jako exkluzivní součet (xor) původní pravé strany a o 1 bit doleva posunutého součinu (and) levé strany a rundového klíče $KL_{i,1}$ $LR_{i,1} = LR_{i,0} \oplus rol(LL_{i,0} \cap KL_{i,1}, 1)$, kde funkce rol() odpovídá posunutí o daný počet bitů, konkrétně zde o 1,
    - nová levá strana $LL_{i,1}$ jako exkluzivní součet (xor) původní levé strany a o 1 bit doleva posunutého součtu (or) nové pravé strany a rundového klíče $KL_{i,2}$ $LL_{i,1} = LL_{i,0} \oplus rol(LR_{i,1} \cup KL_{i,2}, 1)$, kde funkce rol() odpovídá posunutí o daný počet bitů, konkrétně zde o 1,

  výsledek je pak dán spojením nové levé a pravé strany $FL(L_i, KL_i) = LL_{i,1} \,||\, LR_{i,1}$;
- ve funkci $FO(L_i, KO_i, KI_i)$ je 32 bitový vstup L_i rozdělen na dvě poloviny $LL_{i,0}$ a $LR_{i,0}$ a vstupuje do třírundové Feistelovy sítě:
    - nová levá strana $LL_{i,j+1}$ je rovna předchozí pravé straně $LR_{i,j}$,
    - nová pravá strana $LR_{i,j+1}$ je rovna exkluzivnímu součtu (xor) předchozí pravé strany a výsledku funkce FI, jejímiž argumenty jsou předchozí levá strana $LL_{i,j}$ a rundové klíče $KO_{i,j}$, $KI_{i,j}$, kde index j značí pořadí rundy ve vnořené třírundové Fesitelově síti $LR_{i,j+1} = FI(LL_{i,j} \oplus KO_{i,j}, KI_{i,j}) \oplus LR_{i,j}$.

  výsledek je pak dán spojením levé a pravé strany ze třetí rundy $FO(L_i, KO_i, KI_i) = LL_{i,3} \,||\, LR_{i,3}$;
- funkce $FI(LL_{i,j}, KO_{i,j}, KI_{i,j})$ je reprezentována třírundovou nesymetrickou Feistelovou sítí, kde je 16 bitový vstup $LL_{i,j}$ rozdělen na 9 bitovou levou část $LLL_{i,j,0}$ a 7 bitovou pravou část $LLR_{i,j,0}$. Následně probíhají tři rundy:
    - první pravá strana je rovna exkluzivnímu součtu (xor) původní pravé strany rozšířené zleva o dva nulové bity a původní levé strany posunuté podle substituční tabulky s9 (viz níže): $LLR_{i,j,1} = (00 \,||\, LLR_{i,j,0}) \oplus s9(LLL_{i,j,0})$,
    - první levá strana je rovna exkluzivnímu součtu (xor) původní pravé strany posunuté podle substituční tabulky s7 (viz níže) a sedmi levých bitů první pravé strany $LLL_{i,j,1} = s7(LLR_{i,j,0}) \oplus ls7(LLR_{i,j,1})$, kde funkce ls7() odpovídá prvním sedmi bitům zleva,
    - ve druhé rundě je mezivýsledek $LLL_{i,j,1} \,||\, LLR_{i,j,1}$ xkluzivně sečten s klíčem $KI_{i,2}$ a výsledek je rozdělen na 7 bitovou druhou levou stranu $LLL_{i,j,2}$ a 9 bitovou druhou pravou stranu $LLR_{i,j,2}$,
    - třetí pravá strana je rovna exkluzivnímu součtu (xor) druhé levé strany rozšířené zleva o dva nulové bity a druhé pravé strany posunuté podle substituční tabulky s9 (viz níže): $LLR_{i,j,3} = (00 \,||\, LLL_{i,j,2}) \oplus s9(LLR_{i,j,2})$,
    - třetí levá strana je rovna exkluzivnímu součtu (xor) druhé levé strany posunuté podle substituční tabulky s7 (viz níže) a sedmi levých bitů druhé pravé strany $LLL_{i,j,3} = s7(LLL_{i,j,2}) \oplus ls7(LLR_{i,j,3})$, kde funkce ls7() odpovídá prvním sedmi bitům zleva,

výsledek je pak dán spojením levé a pravé strany ze třetí rundy $FI(LL_{i,j}, KO_{i,j}, KI_{i,j}) = LLL_{i,j,3} \,||\, LLR_{i,j,3}$.

5. Zdroje

[@2] Scrambling Techniques for Cdma Communications Od autorů: Byeong Gi Lee,Byoung-Hoon Kim

[@1] http://www.mobilmania.cz/clanky/bezpecnost-gsm-ohrozena-odposlechne-vas-i-amater/sc-3-a-1124204/default.aspx

[@3] http://www.mobilmania.cz/bleskovky/zabezpeceni-site-gsm-prolomeno-s-pristrojem-za-tisic-dolaru/sc-4-a-1118303/default.aspx

[@4] http://web.archive.org/web/20040712061808/www.ausmobile.com/downloads/technical/Security+in+the+GSM+system+01052004.pdf

[@5] http://cryptome.org/gsm-crack-bbk.pdf

[@6] http://cryptodox.com/A5/2

[@7] http://crypto-world.info/casop10/crypto02_08.pdf

[@8] http://www.3gpp.org/ftp/Specs/archive/35_series/35.202/35202-a00.zip