(2)

BEZPEČNOST MOBILNÍCH TELEFONŮ GSM

Šifra v GSM prolomena!

V PŘEDCHOZÍ ČÁSTI JSME VÁS INFORMOVALI O NOVÉM ÚTOKU NA ALGORITMUS A5/1, KTERÝ SE V SÍTÍCH GSM POUŽÍVÁ K OCHRANĚ VZDUCHEM PŘENÁŠENÝCH DAT. UKÁZALI JSME SI KONCEPCI A HLAVNÍ MYŠLENKY CELÉHO POSTUPU. DNES NAVÁŽEME PODROBNĚJŠÍM VÝKLADEM JEDNOTLIVÝCH ČÁSTÍ LUŠTICÍHO PROCESU.

breakthrough

Na úvod je vhodné poznamenat, že za uplynulý měsíc se nám z dobře informovaného zdroje podařilo zjistit, že všichni tři operátoři působící na území našeho státu používají jako šifrovací algoritmus A5 právě zde popisovanou variantu A5/1. Toto zjištění znamená přece jen trochu uklidňující informaci, neboť alespoň víme, že naše sítě neběží pod A5/2, takže na tom — kulantně řečeno — nejsme nejhůř. Pomineme-li nešifrující variantu označovanou jako A5/0, je totiž A5/2 to nejhorší, co by nás mohlo potkat (viz minulý díl). Jak už bylo řečeno, budeme se nyní zabývat rozpracováním určitých detailů. Půjde nám přitom zejména o způsob synchronizace vnitřních stavů A5/1, o realizaci zpětného chodu a konečně si také ukážeme, jaké máme praktické možnosti provedení KPA v potřebném rozsahu.

HLEDÁME VNITŘNÍ STAV

Teoretické podložení správnosti dále popsaného postupu sahá poměrně hluboko do teorie automatů, takže si je zde uvedeme bez podrobnějšího důkazu. Prakticky jde o to, že máme nějaký automat Mooreova typu, který na základě svého aktuálního vnitřního stavu (označme jej q_i) produkuje jeden bit výstupu y_i a s každým hodinovým cyklem se přesune do jiného vnitřního stavu q_{i+1}, v němž produkuje na výstupu novou hodnotu y_{i+1}. Zabývejme se nyní vlastnostmi n-bitového náhodného vektoru představujícího posloupnost produkovanou od okamžiku *i*, tedy $W_i = (y_i, y_{i+1}, \ldots, y_{i+n-1})$. Vzhledem k tomu, že po automatech typu A5/1 je požadováno, aby jejich výstup co nejlépe simuloval náhodný zdroj s rozdělením $p[y_i = 1] = 0{,}5$, můžeme předpokládat, že i vektor W_i má uniformní rozdělení s pravděpodobností $p[W_i = X] = 2^{-n}$. Budeme-li se zajímat o posloupnosti generované ze všech možných vnitřních stavů (nechť jich je 2^m), potom střední hodnota počtu výskytů zvoleného vektoru X bude rovna $2^m * 2^{-n}$.

Odtud plyne, že pro korektní určení příslušného vnitřního stavu nám stačí znát jeho produkci v délce *m* bitů. Tato posloupnost by totiž měla být (ve střední hodnotě) generována právě jedním stavem q_i, a měla by jej tedy jednoznačně identifikovat.

Pro náš případ, kdy víme, že automat odpovídající A5/1 má celkem 2^{64} možných počátečních stavů, potřebujeme pro jejich správnou identifikaci znát počáteční stavy odpovídající všem 64bitovým výstupním posloupnostem. Takový požadavek je samozřejmě prakticky nerealizovatelný. Díky jistým anomáliím (viz například zakázané stavy) v chování A5/1 se však autorům útoku podařilo vypracovat postup, při kterém je možné vnitřní stavy rozlišovat pomocí 51 bitů, ze kterých nám navíc postačí pamatovat si „pouhých" 35 bitů.

Praktický postup vypadá takto: zvolíme si 16bitovou hodnotu α (budeme ji nazývat prefix) tak, abychom neměli potíže s jejím „zacyklováním" uvnitř sama sebe (např. 101010... není vhodná, zatímco třeba α = 1000....0000 vhodná je). Nyní zaznamenáme všechny vnitřní stavy A5 (tj. 64 bitů naplnění registrů) vedoucí k produkovanému heslu začínajícímu právě prefixem α. Takových stavů je přibližně $2^{64} * 2^{-16} = 2^{48}$. Jednou z klíčových myšlenek je, že tyto stavy A5, které nazveme *červené body*, **umíme vypočítat**, aniž bychom zkoušeli všech možných 2^{64} stavů A5 (návod: vyzkouší se pouze všechny buňky registru R1 a buňky registrů R2 a R3 vpravo od řídící buňky, tj. dohromady 41 bitů a 2^{41} zkoušek; 12 známých bitů vpravo od řídících buněk nám dává znalost krokování pro dalších cca $12*4/3 = 16$ kroků; ostatní se dopočítá). Z těchto stavů vybereme 2^{35} tzv. těžkých *červených bodů* (vysvětlíme později) a ty uložíme na disk. Navíc, aby se ušetřilo místo na disku, byla vyvinuta metoda, jak tyto vnitřní stavy efektivně reprezentovat pouze 40 místo 64 bity (toto je další klíčová myšlenka: ukládá se jen podstatné, ostatní se za cenu mírného zvýšení výpočetní zátěže dopočítá).

Nyní zjistíme, kolik červených bodů (každý reprezentovaný 5 bajty) se vejde na dva 73GB disky. Je to $2 * 73 * 2^{30} / 5 = 2^{35}$ bodů. Takové číslo napovídá, že bychom mohli ukládat úplný výčet 35bitového řetězce. U každého červeného bodu si tedy můžeme uložit ještě jeho produkci 35 bitů, následovanou po povinném řetězci α. Dvojici dat (35bitová produkce, 40bitový červený bod) uspořádáme na disk podle první položky, takže tu nemusíme ukládat. Na disk uložíme za sebou pouze druhou

část dvojice — pětibajtové červené body. To nám také později usnadní vyhledávání červených bodů podle zachycené produkce hesla. Celkem tak jeden červený bod znamená stav, od něhož A5 generuje 16 + 35 = 51 specifikovaných bitů.

Útok

Přejděme k vlastnímu útoku. V 228bitové posloupnosti hesla (zatím předpokládejme, že známe hodnotu pro uplink i downlink) sledujeme pouze výskyt řetězce α = 10000...000 (metoda se dá použít i pro jiné řetězce α). Dejme tomu, že to právě nastalo. Zachytili jsme tedy stav, kdy A5 vyprodukovala signální řetězec α a víme, že tato produkce začala jako P-tý bit hesla. Po ukončení řetězce α zaznamenáme následujících 35 bitů. Potom nahlédneme do databáze na disk, kam ukazuje těchto 35 bitů, a zjistíme příslušný červený bod. Od červeného bodu teď zařadíme reverzní chod A5 a odkrokujeme zpět P + 100 bitů. Tím se dostaneme k počátečnímu nastavení. Z něj (je to bod, kdy se smíchá klíč Kc a číslo rámce TDMA) a ze známého čísla rámce pak triviálně určíme klíč Kc.

Správnost klíče Kc ověříme dopředným chodem buď na další produkci hesla v tomtéž rámci, nebo v dalších rámcích. Toto ověření je nutné, neboť reverzní chod není jednoznačný a nabídne nám několik desítek kandidátů. Jejich prověření uvedeným způsobem je však dílem okamžiku. Zůstane nám jediný klíč Kc a jsme hotovi.

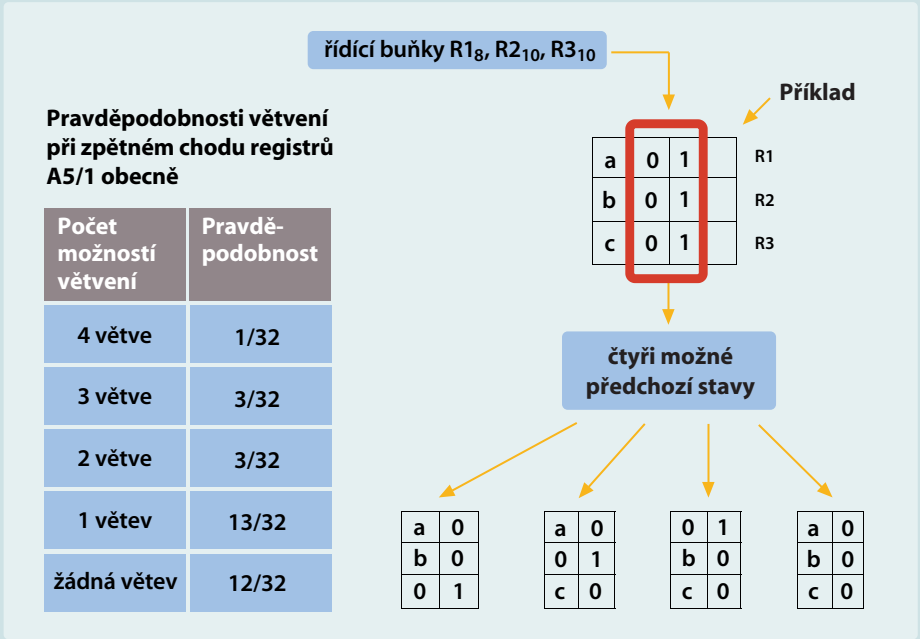
Reverzní chod A5/1

Reverznímu chodu A5 brání „jen" jeho nelineární řízení. Pokud by totiž nebylo použito, posunuli bychom při zpětném chodu každý registr o jeden bit (zpět) doprava a bit nejvíce vlevo bychom jednoznačně vypočetli z nejnižšího (vypadávajícího) bitu a ze zpětnovazebních buněk. To bychom udělali u všech tří registrů. Jinými slovy, každý stav A5 by měl jeden předchozí a jeden následující stav. Pokud bychom si stavy znázornili graficky, vytvořily by jeden cyklus o délce $(2^{19} - 1)*(2^{22} - 1)*(2^{23} - 1)$, tj. cca 2^{64}, zahrnující všechny možné nenulové kombinace vnitřních stavů registrů.

Jak to vypadá v případě nelineárního řízení, ukazuje názorně obrázek 1. Soustředíme se v něm opět jen na řídicí buňky a jejich levé sousedy. Vyjdeme-li ze stavu, kdy na řídicích místech jsou jedničky a vlevo nuly u všech registrů, můžeme zjistit, co se v předchozím kroku stalo. Majoritní bit musel být nula, protože alespoň dva z původních řídicích bitů se posunuly doleva, a tam jsou jen nuly. Podle pravidla řízení se tedy posunuly dva nebo tři registry — a v tom je právě ta nejednoznačnost. V našem případě se mohly posunout libovolné dva nebo všechny tři registry. Celkem tak máme ne jeden, ale 4 možné předchozí stavy. Z každého tohoto stavu bychom nyní udělali opět jeden krok zpět a pravděpodobně by u každého z nich došlo k dalšímu větvení. Můžeme ovšem také dospět do stavu, který nemá předchůdce. Takových stavů je dokonce požehnaně, viz minule vyjmenované zakázané stavy.

Pravděpodobnosti větvení při zpětném chodu registrů A5/1 obecně

Počet možností větvení	Pravděpodobnost
4 větve	1/32
3 větve	3/32
2 větve	3/32
1 větev	13/32
žádná větev	12/32



Obr. 1. Zpětný chod A5/1

Kdybychom takto probrali všechny stavy A5, dostali bychom celou množinu takových stromů, jaký vidíte na obrázku 2, přičemž znázorněný cyklus může být i prázdný. Zdálo by se, že počet větví narůstá exponenciálně. A5/1 je však příkladem zvláštního typu větvícího se procesu, který má jen lineární nárůst větví (synů). Praxe navíc, oproti tomuto teoreticky příznivému očekávání, ukázala, že tento počet je konkrétně u A5/1 ještě menší. V reálném luštění se při chodu o 100 kroků zpět nikdy nevyskytlo více než 120 synů! Zpětný chod je tedy velmi rychlý.

Efektivita luštění

Zabývejme se teď otázkou úspěšnosti popisovaného útoku. K tomu musíme nejdříve definovat pojem *zelený bod*. Červené body už známe a jejich pomocí definujeme zelené body jako takové vnitřní stavy A5/1, které po *i* krocích, $101 \le i \le 278$, přejdou do některého z červených bodů, tj. vygenerují posloupnost začínající zvoleným prefixem α.

Za zelené body tedy považujeme všechny počáteční stavy automatu A5/1, které jsme schopni v námi zachycené části hesla identifikovat na základě známého prefixu α. Velikost množiny všech zelených bodů je zhruba $178 * 2^{48}$.

Nyní si ukážeme výpočet úspěšnosti konkrétně. Budeme přitom stále předpokládat, že máme k dispozici známou obousměrnou komunikaci v délce dvou minut. Během ní dojde k šifrování 2 * 60 * 1000 / 4,6 = 26087 rámců. V každém rámci sledujeme prvních 178 bitů (228 - 51 + 1), na nichž očekáváme začátek řetězce α. Pravděpodobnost jeho výskytu je 2^{-16}, takže v posloupnosti 178 * 26087 bitů můžeme v průměru očekávat 178 * 26087 / 65536 = 71 výskytů řetězců α. Na disk se proto budeme v průměru dívat právě 71krát, což při přístupové době na disk 6 ms dává čas půl sekundy.

Jakmile řetězec zachytíme, z následujících 35 bitů zjistíme pointer na disk a přečteme si zaznamenaný vnitřní stav A5 — červený bod. Teď zařadíme zpětný chod A5 a cestujeme stromem možností až k zeleným bodům. Z nich vypočteme kandidáty na klíč Kc a na jiném kousku posloupnosti hesla falešné kandidáty vyloučíme.

Pokud zaznamenáme na disk všechny červené body (bylo by jich cca 2^{48}), tento postup nemá téměř žádnou chybu a bude vždy úspěšný. Autoři však šetřili místem a přišli na to, že mezi červenými body jsou rozdíly. Některé z nich mají za sebou velmi řídký strom možností, tj. s málo početnou množinou zelených bodů (váha). Ty druhé červené body můžeme nazývat těžkými, protože na nich visí velmi košatý strom zelených bodů. Na disk se proto zazna-

Obr. 2. Ilustrace stromové struktury stavů A5/1

menávají jen tyto těžké červené body. Na dostatečné pokrytí grafu, a tudíž pro úspěch zpětného chodu bylo vybráno pouze 2^{35} těchto bodů s průměrnou vahou 12500.

Jejich určení už je řemeslná matematická záležitost. Jejich celková váha je $2^{35} * 12500$ zelených bodů a všech možných zelených bodů je, jak už jsme si řekli, $178 * 2^{48}$. Těžké červené body nám tedy pokryly $2^{35} * 12500 / (178 * 2^{48}) = 0{,}0086$ celého grafu. Abychom se do této části s reálně zachyceným červeným bodem (vedoucím k bodu zelenému) „trefili“, musíme to zkoušet vícekrát. Při 71 pokusech je pravděpodobnost úspěchu už $71 * 0{,}0086 = 0{,}61$, tedy více než poloviční.

Jak se dostat k posloupnosti hesla?

Na první pohled se útok, tak jak je popsán, možná moc reálný nejeví, protože potřebuje znát dvouminutovou konverzaci (26 087 rámců), a to ještě v digitální podobě. Uvědomme si ale, že to je předpoklad pro vysvětlení teoretického útoku. Praxe je poněkud prozaičtější, což shrnují následující body:

1) Šetřili jsme na paměti pevných disků, takže jsme ukládali jen těžké červené body. Abychom se do nich se skutečně použitým heslem „trefili“, vyžadovalo to větší počet známých rámců. Pokud budeme mít uloženy všechny červené body (cca 2^{48}), postačí nám jeden jediný rámec s výskytem řetězce α! Abychom ho určitě zachytili, musíme v průměru nasbírat cca 65536 / 178 = 369 známých rámců.

2) V diskusích na internetu se uvádí, že minimálně v první desetině vteřiny mobilní telefon z určitých důvodů šifruje „ticho“. Než začne vůbec někdo hovořit, je tedy zašifrováno minimálně 0,1 * 1000 / 4,6 = 22 rámců. Jinak řečeno, známe 22 rámců hesla na začátku jakéhokoliv hovoru. Na získání 369 rámců bude tedy potřeba zaznamenat pouze 369 / 22 = 17 hovorů.

3) Skutečné „ticho“ ve skutečnosti může přijít mnohokrát i během hovoru (občas se musíme také nadechnout). Jistě, nevíme, na kterém bitu bude takové „ticho“ začínat, ale lušticí metodě je to úplně jedno. Lušticímu počítači budeme prostě tvrdit, že „ticho“ nastalo na každém bitu. Když se strefíme, pak je vše v pořádku, když ne, lušticí proces si bude myslet, že narazil na lehký červený bod, a tento alarm bude ignorovat. K úspěšnosti metody potřebujeme jenom, aby mezi předanými alarmy bylo 71 skutečných výskytů řetězců a a abychom znali jeho následujících 35 bitů, tj. celkem alespoň 16 + 35 = 51 bitů „ticha“.

4) „Ticho“ zde necháváme stále v uvozovkách, protože tento řetězec nemusí se skutečným tichem vůbec souviset. Může ho nahradit jakýkoliv jiný nám známý služební řetězec, o němž víme, že bude v šifrované komunikaci přenášen. Lušticí metoda by šla modifikovat i pro případ, že by „ticho“ nebyl souvislý řetězec. Je tu jen požadavek na délku, která by měla být alespoň 51 bitů. Čím delší, tím lépe.

5) Důležité je také si uvědomit, že útok, o kterém jsme až doteď uvažovali, byl chápán jako ryze pasivní. Pro správně „odrzlého“ hackera bude takový předpoklad jistě směšný. Proč? Jednoduše proto, že majiteli napadeného účtu před začátkem útoku zcela chladnokrevně a anonymně (stále nevíte, na co jsou předplacené kupony?) zavolá a bude s ním dvě minuty „jen tak“ konverzovat. Pokud bude alespoň trochu šikovný, podaří se mu takovou komunikaci s přehledem udržet. A výsledek? Porovnáním dat odchycených z jeho mobilu s tím, co vysílala a přijímala stanice napadeného uživatele, provede KPA v potřebném rozsahu a účet je „jeho“!

Pro výslednou efektivitu luštění jsou tedy klíčové následující předpoklady: buď máme dostatek červených bodů, nebo ne. Buď víme, jak přesně vypadá „ticho“, nebo to nevíme, a konečně buďto jsme drzí, nebo nejsme. Na základě těchto předpokladů potom můžeme přesněji vyjádřit naše vyhlídky na úspěch.

V případě extrémně ztížených podmínek závisí náš úspěch na pravděpodobnosti, že během hovorů šifrovaných jedním klíčem Kc (vzpomeňme na jeho dlouhou životnost) dojde k přenosu dvou minut „ticha“. Druhý extrém potom vede k luštění na základě aktivního KPA, kdy musíme umět napadené stanici podstrčit necelé dvě minuty známých dat.

Závěr

V tomto dílu jsme popsali další klíčové části útoku na algoritmus A5/1, jak jej prezentovali pánové Biryukov a Shamir z Weizmannova institutu v Izraeli. Hlavním cílem bylo přitom konkrétněji ukázat, na jakých myšlenkách je útok založen, a odtud odvodit klíčové faktory určující jeho efektivitu.

Rozdělíme-li si s trochou nadhledu typy v současnosti prezentovaných útoků na teoretické (tj. takové, co nevedou přímo k „rozbití“ systémů na bázi napadeného algoritmu) a praktické (tj. ty, co daný systém rovnou „odepíšou“), patří zde popsaná metoda luštění rozhodně mezi ty praktické. Ačkoliv se laikům může zdát, že k jejímu úspěšnému provedení je třeba mít nějakou extra zvláštní techniku, není to vůbec pravda. Je to jen otázka nabídky a poptávky. Pokud někdo bude vědět, že monitorováním příslušné stanice získá informace, které pak velmi výhodně prodá, potom nebude váhat do jejich získání investovat nemalé prostředky.

Navíc je třeba mít na zřeteli, že tato investice je jednorázová. Pak už může útočník atakovat lukrativní stanice doslova jako na běžícím pásu — a vzpomeňte si na minulý díl — kdekoliv na světě. Už začínáte cítit nebezpečnost a moc této techniky, která se vejde do několika kufříků? Co asi bude pro takové lidi znamenat nákup jednoho až dvou digitálních skenerů, podplacení pár techniků a pořízení několika běžných diskových polí? Naprosto nic.

Útočník může atakovat lukrativní stanice jako na běžícím pásu kdekoliv na světě!

Nechceme vás samozřejmě zrazovat od používání GSM techniky. Chceme vás pouze důrazně upozornit, že pokud patříte k těm, jejichž hovory mají cenu zlata, potom dnešním dnem počínaje nevěřte bezpečnosti svého mobilního telefonu o nic víc než bezpečnosti veřejného automatu.

Vlastimil Klíma, v.klima@decros.cz
Tomáš Rosa, t.rosa@decros.cz