Analýza a návrh riešenia

Kryptoanalýza šifier v mobilných sieťach

Bc. Ján Kotrady

Vedúci práce: RNDr. Rastislav Krivoš-Belluš, PhD.

Ciele práce:

1. Analyzovať a porovnať publikované útoky na šifry používané v mobilných sieťach.

2. Preskúmať praktické implementácie analyzovaných útokov.

3. Navrhnúť nové metódy kryptoanalýzy týchto šifier.

Literatúra:

1. E. Biham, O, Dunkelman: Techniques for Cryptanalysis of Block Ciphers (Information Security and Cryptography), Springer, 2017, ISBN 978-3642172311

2. W. Stallings: Cryptography and Network Security: Principles and Practice, 7th edition, Pearson, 2016, ISBN 978-0134444284

3. A. G. H. Naim: Cryptanalysis of Some Block Ciphers, PhD. thesis, University of London, 2014

Skratky:
USIM - Universal Subscriber Identity Module
UE - User Equipment
SRNC - Serving Radio Network Controller
VLR - Visitor Location Register
SQN - Sequence number
AuC - Authentication Centre
AUTN | S - Authentication token | Synchronization

Úvod

Práca je venovaná kryptoanalýze šifier, ktoré sú používane v mobilných sieťach. Prioritne ide o šifry ako A5/1, A5/2, A5/3 alebo aj KASUMI a taktiež šifru SNOW 3G. V súčasnosti sú šifry A5/1 a A5/2 prelomené a je možná kryptoanalýza v reálnom, respektíve skoro reálnom čase (do niekoľko minút až hodín). Šifra KASUMI je taktiež šifra, ktorá sa považuje sa prelomenú ale jej kryptoanalýza nie je možná v reálnom čase a v spôsobe použitia, ako je v súčasnosti použitá v mobilných zariadeniach. Totižto šifra KASUMI sa používa v counting mode, kde sa výsledok výstupu zašifrovaného počítadla pomocou xor funkciu pripojí k šifrovanej správe. Útoky ako related key rectangle attack je možný iba za znalosti veľkého počtu otvorených, nešifrovaných textov, kde je potrebná aj pomerne vysoká výpočtová sila. Poslednou spomínanou šifrou je šifra SNOW 3G. Práve táto šifra je časťou najnovšieho 4G štandardu, v ktorom sa okrem šifry SNOW 3G taktiež využíva aj šifra AES. V našej práci sa budeme venovať predovšetkým kryptoanalýze šifier A5/1 a A5/2, respektíve budeme simulovať útoky na tieto šifry a povieme si aj niečo o reálnych možnostiach útokov na tieto šifry v súčasných mobilných sieťach. Následne sa budeme venovať kryptoanalýze, predovšetkým diferenčnej kryptoanalýze šifre KASUMI a možnosti využití tejto kryptoanalýzy v praxi. Ako už bolo spomenuté, v reálnych podmienkach súčasné útoky nedokážu prelomiť šifru KASUMI tak, ako je používaná v mobilných sieťach a práve z tohto dôvodu budeme hľadať alternatívne útoky, skúmať aktuálne prístupy s možnosťou využitie týchto útokov v praxi. Poslednú spomínanú šifru SNOW 3G si necháme ako alternatívnu časť záverečnej práce, kde by sme taktiež chceli poukázať na možnosti útokov na túto šifru, prípadne jej slabiny.

1 Popis protokolov v mobilných sieťach a historické pozadie

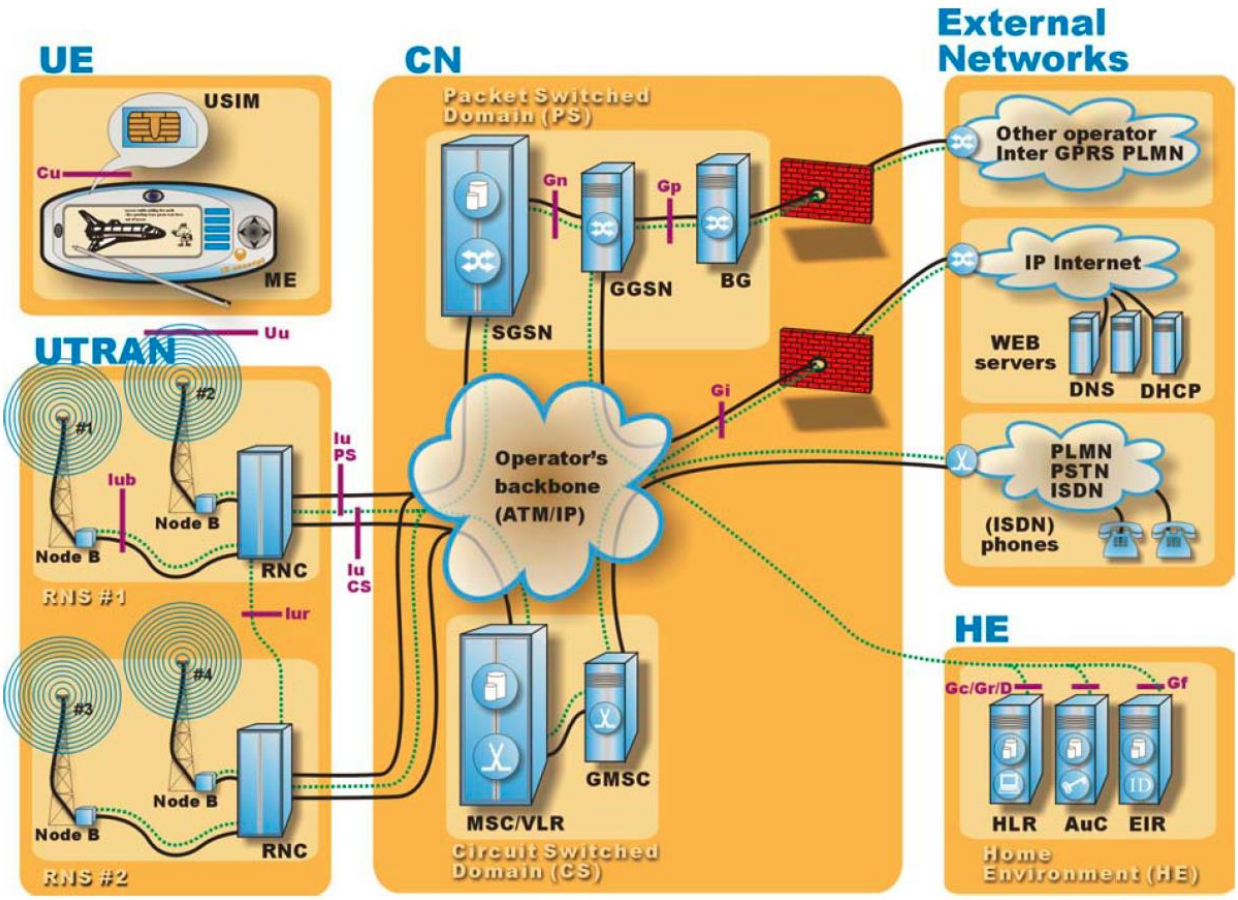
Počiatok komunikácie prostredníctvom mobilných telefónov sa datuje približne v 80 rokoch [1.1]. Vznikali rôzne dohody na štandardoch, prvé telefónne zariadenia a začiatkom 90 rokov prišiel druhý štandard, ktorý už nevyužíval analógový ale digitálny prenos dát a zvuku. S tým prišiel aj štandard 2G ktorý pretrval dodnes a je možné tento štandard, alebo aj lepšie povedané mód v súčasnosti využívať. My sa nebudeme zaoberať spôsobom

šírenia signálu, metódami ako funguje roaming, dátové prenosy ale pozrieme sa na systém autentifikácie a spôsoby overenia používateľa a dohody na kľúči, taktiež na približnú schému zariadení, ktoré vystupujú na pozadí celej mobilnej siete do takej miery, aby sme boli schopný porozumieť hlavným princípom.

1.2 Základné zariadenia a definovanie pojmov

Mobilná sieť pozostáva z niekoľkých častí, pre nás najdôležitejšie sú:

1. BTS – Ide o prvú stanicu, s ktorou je mobilné zariadenie pripojené a s ktorou priamo komunikuje. Na obrázku 1 v našom prípade ide o komplex zariadení Node B, niekde do daného pojmu spadajú aj zariadenia RNC. Pre účely tejto práce budeme postačovať s tým, že ide o zariadenie, ktoré stojí za prijatím a odosielaním správ priamo do mobilného zariadenia cez nezabezpečený kanál.
2. USIM/SIM – Každé mobilné zariadenie obsahuje takzvanú SIM kartu. V príade, že rozprávame o USIM, ide o novší typ SIM karty, ktorý sa používa už v 3G sieťach a zabezpečuje viac funkcií ako SIM karta.
3. AuC – Ide o autentifikačné centrum operátora. Každý operátor ma svoje vlastné autentifikačné centrum, ktoré je zodpovedné za autentifikáciu, autorizáciu a integritu dát (a mnoho iných vlastností). V prípade, že je používateľ pripojený v cudzej sieti (napríklad pri ceste do zahraničia) a nastáva autentifikácia, tak zahraničná mobilná sieť je povinná autentifikovať užívateľa s domovským AuC. Takže cudzia sieť musí posielať požiadavku domovskej siete. AuC taktiež obsahuje takzvané záznamy IMSI, autentifikačné a autorizačné dáta, taktiež aj zvolené šifrovacie kľúče.
4. UE, ME – User Equipment, Mobile Equipment označuje v prípade UE komplex mobilného zariadenia so SIM, resp. USIM kartou a ME označuje komplex mobilného zariadenia (mimo SIM kariet)

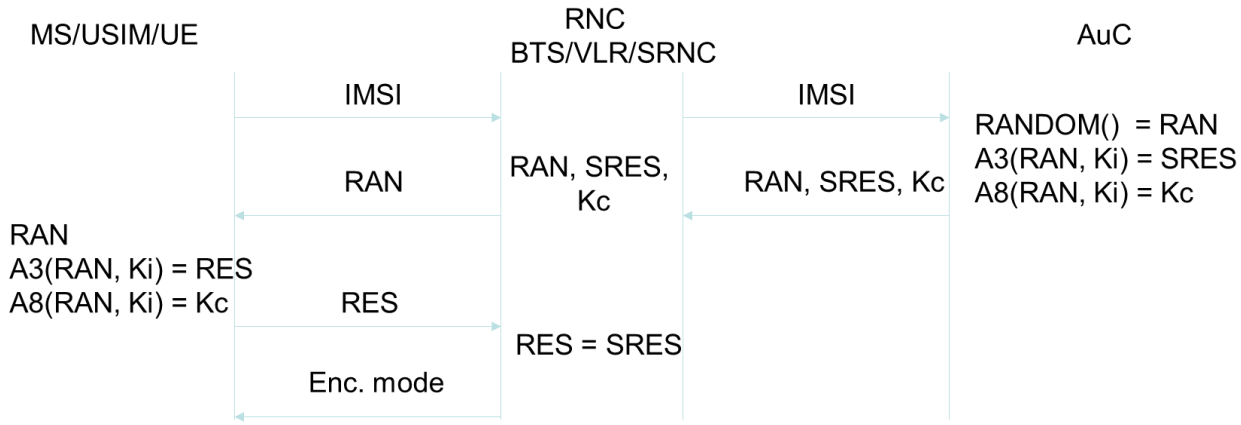


Obrázok 1: Schéma mobilnej siete

1.2 Protokol v móde 2G

Mobilné zariadenie (UE) je pripojené do mobilnej siete, či už domáceho alebo cudzieho operátora. Následne sa ohlasy svojim IMSI. Operátor na základe údajov, ktoré dostal od UE pošle prostredníctvom svojej vlastnej siete správu s obsahom IMSI autentifikačnému centru (AuC). Autentifikačné centrum vyberie zo zoznamu známych IMSI (tie sú uložené v autentifikačnom centre pri vyrábaní/vydávaní SIM karty, spolu s kľúčami) zdieľaný kľúč K_i, vygeneruje náhodné číslo RAN a následne použije funkciu A3 so vstupom RAN a K_i, výstup označíme SRES. Taktiež použije funkciu A8 so vstupom K_i a náhodným číslom RAN, výstup označíme K_C. Tieto dáta (RAN, SRES a K_C) prepošle AuC späť, k BTS stanici, od ktorej predtým požiadavka prišla. Následne stanica BTS vyzve UE tým, že jej prepošle náhodnú hodnotu RAN, vygenerovanú stanicou AuC. V prípade, že ide

o správneho používateľa, tak tento používateľ by mal mať znalosť K_i a v prípade, že bolo doručené správne RAN, UE je schopný zo znalosti K_i a RAN vygenerovať hodnoty SRES a K_C použitím funkcií A3 a A8 (o týchto funkciách budem písať neskôr, možno že až v práci, nie v článku, podľa počtu strán...). Následne UE prepošle vygenerovaný SRES BTS stanici. Ak stanici BTS bude doručený rovnaký SRES od UE ako bol doručený od AuC, stanica autentifikuje UE a následne prebieha šifrovaná komunikácia pomocou kľúča K_C. Následne je možné ešte určiť o aký mód šifrovania pôjde, tj. o žiadny alebo o šifrovanie šiframi A5/1 alebo A5/2, v oboch prípade kľúčom K_C [1.2].



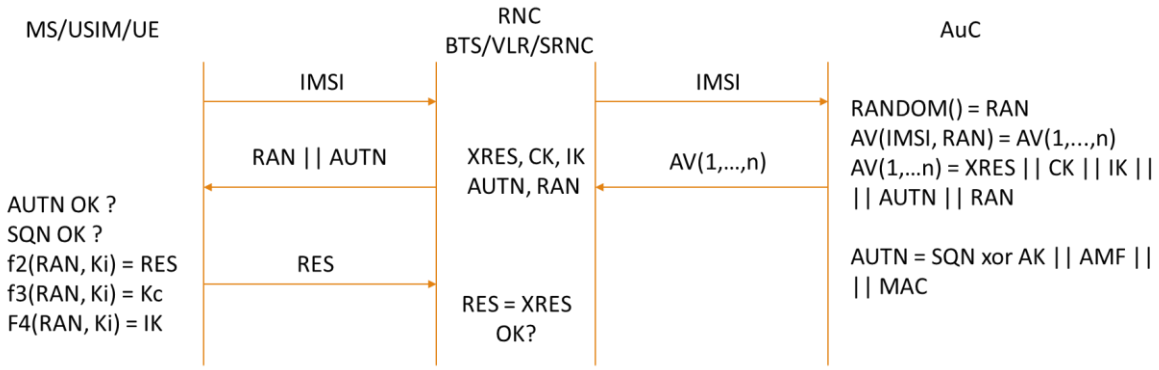
Obrázok 2: Protokol v móde 2G

Ako môžeme vidieť, v tomto prípade sa autentifikuje len mobilné zariadenie voči BTS stanici, nie naopak, takže je možné využiť takzvanú falošnú stanicu BTS. O tomto type útoku budeme písať nižšie. Totižto ak vytvoríme falošnú BTS stanicu, ktorá sa bude voči nefalšovanej BTS stanici tváriť ako UE zariadenie a voči napadnutému UE bude vystupovať ako BTS stanica, je možné týmto spôsobom vynútiť šifrovací režim v móde bez šifrovania a celú komunikáciu odpočúvať. Existujú aj iné druhy útokov na daný mód 2G, o ktorých sa ale budeme bližšie zmieňovať v záverečnej práci.

1.3 Protokol v móde 3G a 4G

Známe útoky popísane v kapitole vyššie sa snažili autori módov 3G a 4G odstrániť. Navrhli autentifikáciu aj BTS stanice voči zariadeniu UE. Celý protokol na autentifikáciu a dohodu

o kľúči prebieha podobne, ako v prípade módu 2G, no AuC generuje v tomto prípade celý autentifikačný vektor. Bližšie tento vektor popíšeme v záverečnej práci, keďže sa chceme aj zamerať na možnosti kryptoanalýzi ktoré priamo súvisia s dátami z autentifikačného vektora. V podstate ide o to, že AuC vygeneruje náhodné číslo a vypočíta na základe náhodného čísla a IMSI niekoľko ďalších hodnôt. V tomto prípade vygeneruje znova C_K ako kľúč, ktorý sa použije na šifrovanie dát, očakávanú XRES (očakávanú odpoveď od UE), kľúč integrity I_K a veľmi dôležitú časť, tzv. autentifikačný token, ktorý slúži na autentifikáciu BTS stanice voči UE. Práve týmto tokenom je zaručená autentifikácia BTS stanice a slúži ako ochrana voči útokom falošnými BTS stanicami. V rámci protokolu v móde 3G je implementovaná aj tzv MAC funkcia, ktorá slúži ako integritná ochrana správy. Totižto, ak UE prijme správu s nesprávnym AUTN, nebude považovať sieť za bezpečnú. Totižto, ak by sme aj v tomto prípade použili falošnú BTS stanicu, autentifikáciu by sme vyžiadali od správneho AuC, nedostali by sme sa k dátam, ktoré sa používajú na autentifikáciu správ (MAC kód), takže v našom prípade by sme neboli schopný poslať korektnú správu UE s tým, že volíme mód šifrovania taký, že sa správa nebude šifrovať. AUTN musí byť poslané UE, pretože ak by AUTN nebolo poslané UE mohlo by dôjsť k situácií, žeby UE bolo pripojené na falošnú BTS stanicu a boli by podvrhnuté dáta ako je RANDOM číslo, respektíve sekvenčné číslo, ktoré sa používa v autentifikačnom vektore [1.2]. Bližšie informácie, popis možných útokov a podrobné vysvetlenie celého systému budeme špecifikovať v záverečnej práci.



Obrázok 3: Protokol v móde 3G a 4G

Existujú ale aj v tomto prípade možnosti útokov, ktoré nie sú ale z implementačného hľadiska v našom prípade reálne. Ide o napríklad vytvorenie falošného mobilného operátora, ktorý by bol schopný dostávať správy od AuC. Tieto scenáre sú možné len v prípade, žeby sme povolili vytvorenie a zaregistrovanie osoby, subjektu ako mobilného operátora a potom by sme mohli využívať falošné BTS stanice.

Ak by sme ale možnosť vytvorenia falošného operátora zavrhli, vytvorenie falošnej BTS stanice by nám v tomto prípade nepomohlo, skôr naopak, fungovali by sme len ako repeater, tj. nejaké zariadenie, ktoré by šírilo signál bez možnosti reálneho útoku. V takomto prípade sme ale schopný dosiahnuť rovnakých výsledkov ak použijeme zariadenie ako RTL-SDR.

1.4 Nezabezpečená komunikačná linka operátora

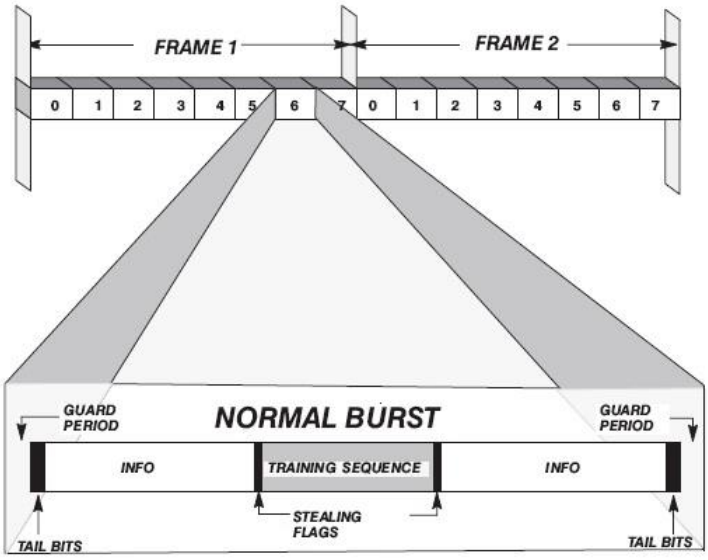
Mobilný operátor nemá povinnosť šifrovať dáta, ktoré prechádzajú jeho internou mobilnou sieťou a ani to v mnohých prípadoch nerobí. Dokonca zákon Slovenskej republiky (Zákon č. 351/2011 Z. z., §64, ods. 14) hovorí o tom, že je povinné pre operátora zabezpečiť také podmienky, aby bolo možné na žiadosť orgánov činných v trestnom konaní zachytiť nešifrovanú mobilnú komunikáciu a tým pádom ju aj odpočúvať. Práve z týchto dôvodov je akákoľvek mobilná komunikácia nešifrovaná na úrovni vnútornej siete operátora. Ak by bol niekto schopný dáta, ktoré sa šíria po vnútornej sieti mobilného operátora zachytiť, získal by priamy prístup k všetkým mobilným hovorom, ktoré sa na danej časti sieti uskutočňujú. Na druhej strane ten istý zákon hovorí o tom, že je operátor povinný zabezpečiť mobilnú sieť tak, aby nebolo možné odpočúvanie komunikácie, ak na to nebol daný súhlas alebo tento zákon nehovorí inač (v prípade vyžiadania odpočúvania mobilnej komunikácie zo strany orgánov činných v trestnom konaní).

1.5 Frame number, frame, burst

Komunikácia v GSM sieti používa tzv. Frame number, ktoré sú verejné známe. Každá BTS stanica vysiela správu v takzvaných frame a burst. Dĺžka burstu je 122 bitov, pričom

prenášaná informácia je dĺžky 114 bitov. Práve týchto 114 bitov je generovaných ako výstup zo šifry A5/1. Správa, ktorú prenášame GSM (napríklad SMS správa) je rozdelená medzi niekoľko burstov. Napríklad SMS správa, ktorá je kódovaná 7 bitovým kódovaním (štandard z pred pár rokov, dnes sa používa UTF-8) a obsahuje 120 znakov je rozdelená do 7 rôznych burstov. Okrem danej správy burst ešte obsahuje informácie od operátora, kontrolné súčty a niekedy aj rôzne nepredvídateľné dáta. Preto je potrebné danej problematike venovať pozornosť. Každá osmica burstov je odosielaná naraz, počas jedného frame. Takto je obslúžených naraz až 8 mobilných zariadení.

Frame number je verejné známe číslo, je nastavený na začiatku komunikácie (posiela sa na začiatku komunikácie ako ohlásenie, že takýto je frame number, správa typu: Next frame number is NNNN). Frame number je číslo veľkosti 22 bitov a inicializácia šifry A5/1, popísanej nižšie nezávisí od frame number. Ak sa pošle 8 burstov, počítač zvýši frame number o 1 (inkrementuje danú hodnotu) a šifra A5/1 sa reinicializuje s novým frame number no s pôvodným inicializačným kľúčom.



Obrázok 4: Burst a Frame

2 Aktuálne možnosti kryptoanalýzy

2.1 Rušenie signálu a prechod na nižší mód

V súčasnosti existujú rôzne protokoly a štandardy prenosu dát prostredníctvom mobilných zariadení. Ide predovšetkým o módy GSM, 2G, 3G a 4G. V prípade využívania GSM módu, ktorý sa aj inač označuje 2G (ide len o technické zlepšenie, ktoré záleží od lokality) ktorý dosahuje v súčasnosti najväčšie pokrytie, sú podporované iba šifry A5/1 a A5/2. Práve z tohto dôvodu je možné využiť rušenie signálu na frekvenciách 3G a 4G, keďže každý z tých módov (niekedy nazývaných aj protokolom, no v princípe ide o viac ako len o protokol, z tohto dôvodu volíme označenie mód) pracuje na vlastnej frekvencií. Napríklad GSM pracuje v pásme približne od 800 MHz do 900 MHz [2.1], pričom v mnoho prípadoch záleží od konkrétnej lokality, krajiny a regiónu. V prípade, že sa použitie zariadenie, ktoré dokáže nejakým spôsobom prerušiť komunikáciu na frekvenciach patriacim módom 3G a 4G, mobilné zariadenie bude nútené komunikovať prostredníctvom módu 2G (GSM) a bude musieť využiť šifry A5/1 a A5/2. A ako už bolo spomínané vyššie, dané šifry je možné prelomiť v reálnom čase. Napríklad v prípade šifry A5/1, pri využití približne 2TB tzv. rainbow tabuľky je možné šifru A5/1 prelomiť v priebehu pár sekúnd [2.2]. Príprava danej rainbow tabuľky trvala približne jeden mesiac a je k dispozícií širokej verejnosti v [2.3], pričom aktuálne prebieha sťahovanie týchto tabuliek. Zachytiť mobilnú komunikáciu na danej frekvencií je možné použitím takzvaných RTL-SDR rádio prijímačov, ktoré dokážu zachytiť signál na frekvencií od 15 MHz do 1300 MHz, aj v tomto prípade ale záleží na výrobcovi daného zariadenia.

Ak vezmeme v úvahu, žeby sme použili zariadenie na rušenie signálu na frekvenciách príslušných módov 3G a 4G, sme schopný pomocou jedného zariadenia RTL-SDR zachytiť a odpočúvať aktuálne jedno pásmo prislúchajúce módu 2G (GSM). Samozrejme, mobilné zariadenia využívajú v rámci jedného módu niekoľko ďalších špecifických kanálov, presných frekvencií, individuálne určených pre uplink a pre downlink. Z tohto dôvodu sa my budeme zameriavať na práve jednu frekvenciu. Mobilné zariadenia totižto častokrát prepínajú medzi jednotlivými frekvenciami v rámci určeného módu na zvýšenie

kvality prenosu signálu. Tento fakt ale nie je bezpečnostným prvkom a na jeho vyriešenie by sme použili len niekoľko naraz pripojených RTL-SDR zariadení, aby sme skenovali celé pásmo prislúchajúce módu 2G.

Práve týmto spôsobom je možné odpočúvanie mobilnej komunikácie aj bežným ľudom, potrebná je len vstupná investícia do rušičky signálu (ktorej ale použitie je trestné v právnej úprave Slovenskej republiky, na druhej strane je to v individuálnych lokálnych prípadoch nedokázateľné) a investícia do zariadenia typu RTL-SDR, ktorého hodnota nepresahuje niekoľko pár desiatok dolárov.

2.2 Simulovanie stanice BTS

Ďalšou alternatívou je možnosť simulácie takzvanej BTS stanice. Táto stanica slúži na príjem signálu od mobilného zariadenia a následne jeho šírenie v sieti operátora. BTS stanica je prvá časť siete operátora, ktorá príde do kontaktu s mobilným zariadením a je to jediná časť, s ktorou je schopné mobilné zariadenie komunikovať.

Existuje niekoľko ukážok simulácie takejto BTS stanice [2.4], kedy útočník prostredníctvom niekoľkých mobilných zariadení značky motorola dokáže simulovať takúto BTS stanicu. Obeť sa v tomto prípade pripojí priamo na stanicu útočníka, ktorý ďalej bude len preposielať signál skutočnej BTS stanici. V tomto prípade je možné zvoliť šifrovací mód v rámci módu 2G taký, ktorý nepoužíva žiadne šifrovanie (takýto mód šifrovania 2G siete podporujú, ale bežne nie je používaný) a následne stačí zaznamenať celú komunikáciu. Ide o útok typu man-in-the-middle, ktorý je možne predviesť prostredníctvom už spomínaných mobilných zariadení typu motorola a softvéru, ktorý je voľne k dispozícii, tzv openBSC. Taktiež je možné použiť zariadenia ako hackRF alebo BladeRF ako je spomínané v článku [2.4]. Simulácia takouto metódou nie je náročná, no je na finančné zdroje zaťažujúca a neposkytuje nám žiadnu pridanú hodnotu v použití kryptoanalýzi, ide skôr o útok na štandardy módu 2G a z tohto dôvodu sme sa rozhodli, že danou cestou nepôjdeme.

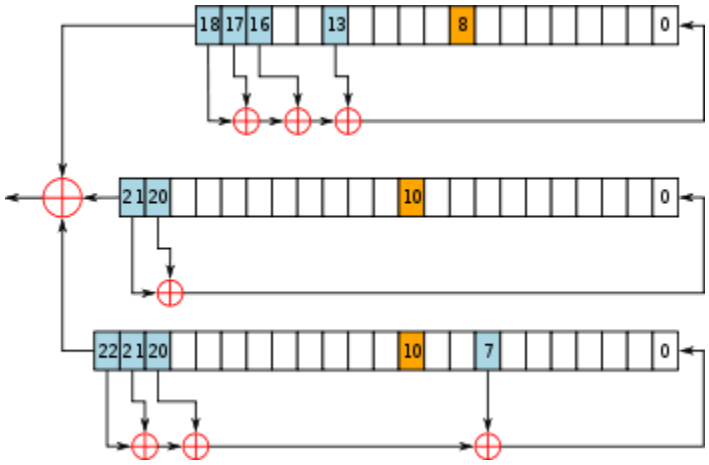
2.2 Kryptoanalýza šifry A5/1

Keďže šifra A5/2 a jej kryptoanalýza je z pohľadu kryptoanalytika jednoduchá a jej podpora sa v súčasnosti vytráca jak z mobilných zariadení, tak aj z BTS staníc, rozhodli sme sa, že sa budeme zaoberať iba kryptoanalýzou šifry A5/1.

Šifra A5/1 je z formálneho hľadiska v kategórii prúdových šifier. Jej algoritmus bol z počiatku utajovaný a reverzným inžinierstvom bol neskôr extrahovaný priamo zo SIM kariet. Šifra obsahuje 3 posuvné registre, takzvané LFSR registre, a niekoľko hradiel XOR [2.5]. Z matematického hľadiska je možné definova LFSR registre ako polynómy:

(shift register A) $X^{19} + X^5 + X^2 + X + 1$
(shift register B) $X^{22} + X + 1$
(shift register C) $X^{23} + X^{15} + X^2 + X + 1$

Princíp celej šifry je založený na posúvaní bitov v registroch. Pre bližší princíp viď. obrázok 5. Žltou sú vyznačené pozície, ktoré sa používaju na sťaženie kryptoanalýzi. Ide o to, že cez tieto pozície prejde do ľava len tá hodnota, ktorá sa vyskytuje vo väčšine pozícií. Napríklad ak je v registri 1 na pozícií 0, v registri 2 je na pozícií 10 hodnota 1 a v registri 3 je na pozícií 10 hodnota 0, tak cyklicky sa posunú len registre 1 a 2, pretože tie obsahujú majoritný bit.



Obrázok 5: Šifra A5/1 [2.7]

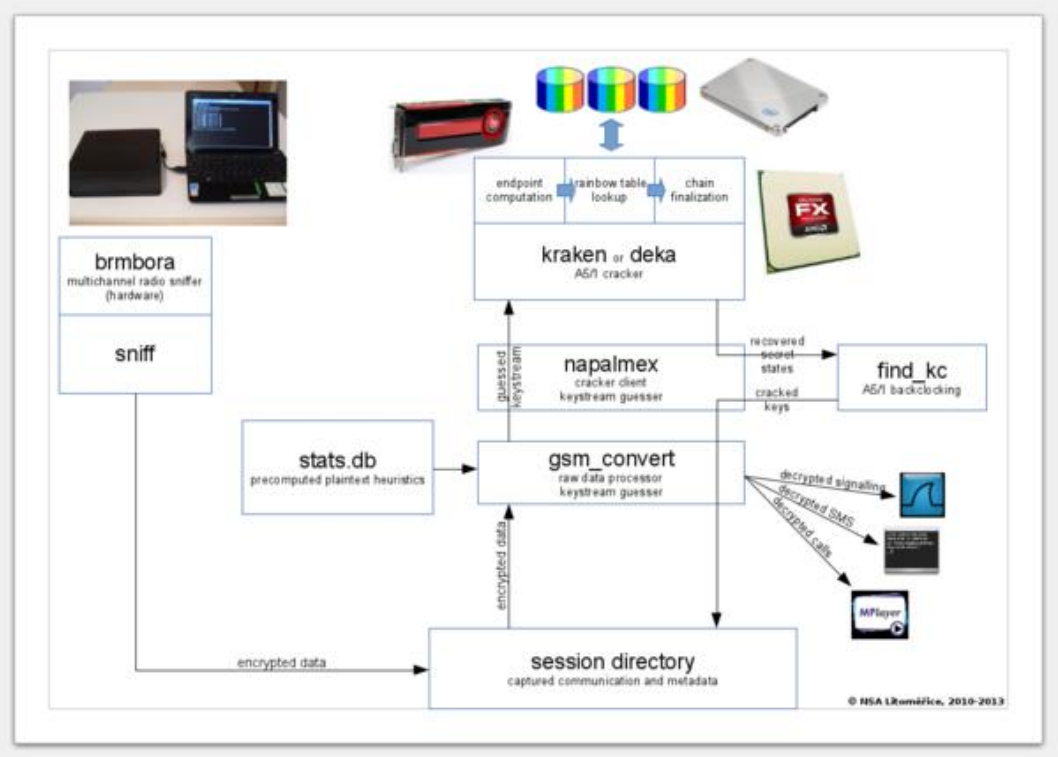
Bolo dokázane, že existujú stavy, do ktorých sa dané registre nedokážu dostať. V našom prípade ide o 2^{64} všetkých možností. Totižto v danej šifre existuje práve 64 hodnôt, každá

môže nadobúdať buď 0 alebo 1tku, takže celkovo 2^{64} všetkých možností. Matematicky sa podarilo stanoviť presný počet reálne možných dosiahnuteľných interných stavov, ktorých je už len $2^{61,5}$ je dosiahnuteľných [2.8,2.9]. Šifra A5/1 je taktiež ľahko reverzibilná v kontexte toho, že je možné celkom triviálne vypočítať spätný prúd. To znamená, že ak sa nám podarí odhaliť stav, v ktorom sa šifra nachádzala, vieme spätne dopočítať nejakú hodnotu vnútorného stavu, v ktorom sa šifra nachádzala v minulosti. Samozrejme, pri spätnom točený šifrou dochádza k vetveniu, no toto vetvenie nie je časovo príliš náročne [2.9].

Celá kryptoanalýza je postavená na základe rainbow table alebo aj inač dúhových tabuliek. Tie sú vytvorené ako možnosti vnútorných stavov, v ktorých sa mohli registre nachádzať. Napríklad povieme si, že všetky registre sú nastavené na 0 až na poslednú hodnotu v prvom registre, ktorá je nastavená na 1tku. Takýto vnútorný stav využijeme na to, aby sme vypočítali keystream, ktorý daná šifra vygeneruje. Následne využijeme známe vlastnosti rainbow tabuliek a využijeme redukčnú funkciu ktorá bude fungovať na princípe takom, že výstup z jedného vnútorného stavu bude vstup do ďalšieho vnútorného stavu. Konkrétne sa tento algoritmus opakuje pre jeden začiatok 4096 krát, približne, kým nedostaneme konečnú hodnotu ktorá je tvorená niekoľkými nulami na konci. Aby z toho vznikli aj skutočné rainbow tabuľky pridáme do tejto tabuľky aj takzvanú farbu. V redukčnej funkcií pridáme jednu funkciu XOR, ktorá bude na výstup z A5/1 aplikovať funkciu XOR s konštantou, takzvanou farbou. Tieto tabuľky sú predpočítané, verejné dostupné a boli predvedené na konferencií v Berlíne [2.2]. Následne pri zachytený správy, ktorú chceme dešifrovať, je potrebné spočítať niekoľko krát funkciu A5/1 kým sa dostaneme do vopred definovaného koncového stavu a pozrieme sa, či sa tento stav nenachádza v našej tabuľke. Ak sa tam nachádza, sme schopný spätným počítaním zistiť počiatočný, respektíve predchádzajúci vnútorný stav a tak dešifrovať celú komunikáciu ktorá predchádzala danému prelomeniu až do bodu, kým nebol zmenený vnútorný kľúč. Bližšie bude tento útok vysvetlený v záverečnej práci. Ak by sme sa chceli venovať len popisu tomuto útoku celý tento materiál by bol len opis daného útoku aj s vysvetlením jednotlivých častí. Využívajú sa rôzne nástroje. Jeden zo všeobecne známych je Kraken, ktorý je v súčasnosti

nepodporovaný na mnohých OS a hardware, preto sa skupina ľudí z Českej republiky rozhodla implementovať nový nástroj, takzvanú Deku, ktorá už využíva pokročilé technológie ako OpenCL.

Asi zostáva už len vysvetliť, ako je možné objaviť vnútorný stav, keď šifra A5/1 funguje na princípe generovanie keystreamu, ktorý sa funkciu XOR spojí so šifrovaným textom. Totižto, BTS stanica vysiela veľké množstvo zbytočných dát. Ak nemá čo povedať, bude vysielať dáta typu NOP, ktoré majú presne definovanú štruktúru. Z tohto dôvodu, ak zachytíme nejakú mobilnú komunikáciu, tak vieme odhadnúť, tipnúť si správy, ktoré boli posielané a približne 17 % správ sú správy typu NOP. Ak na celú zachytenú mobilnú komunikáciu aplikujeme, skúsime uhádnuť, že to boli správy typu NOP, máme pomerne vysokú šancu že sa trafíme. Celý tento princíp funguje na vlastnostiach funkcie XOR, pretože platí: keystream XOR plaintext = ciphertext, ciphertext XOR plaintext = keystream.



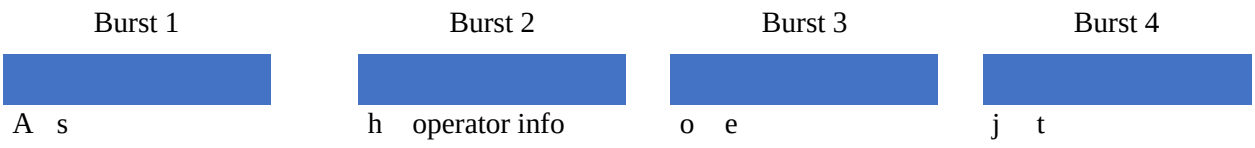
Obrázok 6: Postup kryptoanalýzi [2.6]

2.3 Priestorová veľkosť šifry a kľúčov

Ako už bolo popísané predtým, šifra A5/1 pracuje s 64 rôznymi bitovými hodnotami ako hodnoty posuvných registrov. Keďže veľkosť kľúča je taktiež 64 bitov, potom medzi dvoma rovnako veľkými množinami 2^{64} musí existovať inklúzia alebo rovnosť. Vieme, že existuje niekoľko nedosiahnuteľných stavov, takže množina kľúčov je nadmnožina množiny vnútorných stavov. Celková množina vnútorných stavov je 2^{61}, takže rozdiel je markantný. Množina všetkých kľúčov menšia byť nemôže. Z toho vyplýva, že $\frac{2^{64}}{2^{61}} = 8$ do každého stavu sa vieme dostať z 8 rôznych kľúčov (množina kľúčov je osemnásobné väčšia ako množina vnútorných stavov). Taktiež ako sa v danej šifre pri inicializácií najprv vytvorí vnútorný stav z 64 bitov kľúča a následné z množiny 22 bitov frame number, čo je známa hodnota, vyplýva, že všetky vnútorné stavy, aj z iných kľúčov (tj. z množiny 2^{64} kľúčov) sú dosiahnuté zamiešaním vnútorného stavu šifry A5/1 po použití frame number. Frame number ako kľúč nepridáva šifre zložitosti prelomenia, pretože aj keď sa použije frame number, šifra zkolabuje do nejakého stavu, ktorý by dosiahla použitím iného kľúča. Vyskúšanie 2^{22} rôznych frame number je z pohladu súčasnej informačnej doby (a jej výpočtovej sily) otázka niekoľkých minút na priemernom osobnom počítači. No z pohľadu kryptoanalýzy popísanej v roku 2010 nevieme tento fakt využiť.

2.4 Návrh útoku na šifru A5/1

Celý problém útoku na šifru A5/1 spočíva v tom, že otvorený text nie je súvisle distribuovaný medzi všetky bursty, respektíve frame. Taktiež tabuľky popísane vyššie pracujú len s vnútorným stavom, takže nie je možné efektívne vyskúšať nejaké kombinácie otvoreného textu.

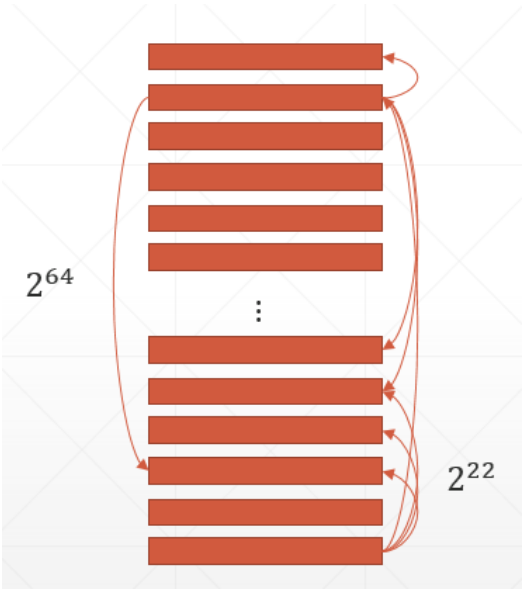


Napríklad, ak pošleme SMSku obeti s textom „Ahoj svet“, tento text nemusí byť obsiahnutý v jednom burste, ale je rozdistribuovaný v niekoľkých burstov.

Ako vidíme na obrázku vyššie, tak SMS správa s textom Ahoj svet je rozdistribuovaná v 4 burstov plus operátor pribalil nejaké ďalšie informácie, ktoré nám SMS správu ešte viac rozhádzali. V útoku popísanom vyššie máme možnosť vyskúšať všetky možnosti textu v danom burste (ak sa tam nachádza) a následne skúsiť využiť tento útok.

Čo je nám ale známe a vôbec sa v danom útoku (až na konci pri analýze, vtedy, keď už len potrebujeme dogenerovať kľúč) nevyužíva je tzv. frame number. Nastáva otázka, ako vieme frame number využiť pri útoku na šifru A5/1 ?

Vezmime si možnosť, že zostrojíme podobnú rainbow tabuľku, len s tým rozdielom, že tabuľka bude nastavená tak, že sa bude točiť šifra A5/1 nie podľa náhodného stavu vo vnútri, ale podľa frame number. Vezmime si, ak poznáme frame number, tak frame number vieme využiť tak, že skúsime uhádnuť text správy a využijeme naprv útok, ako sme použili predtým. Tým pádom dostaneme kľúč, ktorý sa použil pri šifrovaní danej správy ak sme vykonali hádanie správy dobre. Následne vieme tento odhad ešte vylepšiť. Vezmeme vnútorný stav šifry A5/1 a budeme spätným točením generovať také možné točenie, že točíme v rámci nejakého frame number. V podstate vieme zachovať, zafixovať frame number a vieme šifru A5/1 točiť (zatiaľ povedzme, že nejakým algoritmom) tak, aby sa zachovával frame number, menil sa iba vnútorný kľúč. Totižto pri každom vnútornom kľúči, ako je v rainbow tabuľke v pôvodnom útoku existuje rôzných 2^{22} prepojení do iného kľúča tak, že na zvolený frame number sa vieme dostať cez určitú množinu kľúčov.



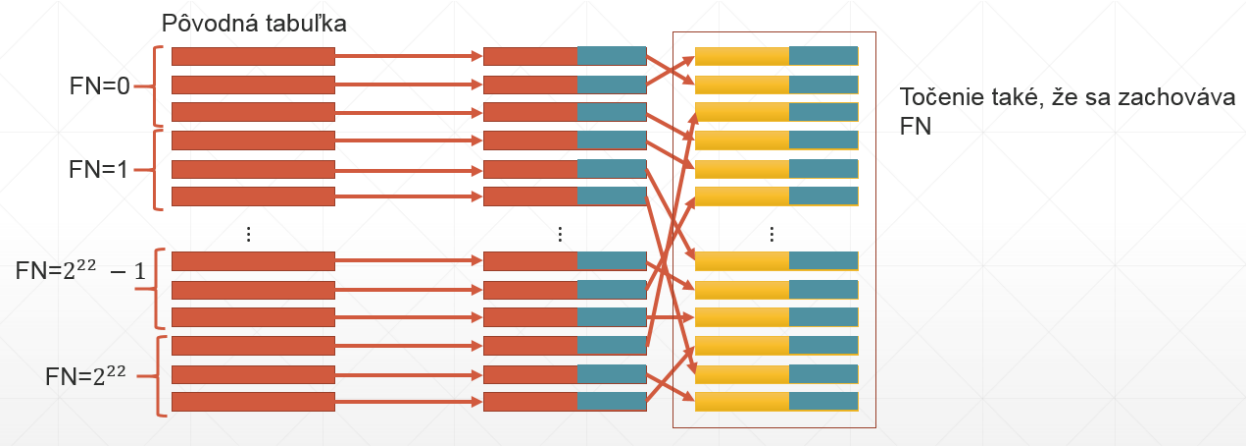
Obrázok FRAME vs kľúč

Obrázok FRAME vs kľúč znázornuje červenými štvorčekami vnútorné stavy, tzv. kľúče tak, že z každého kľúča vychádza práve 2^{22} rôznych frame number. Totižto frame number nám len presunie vnútorný nejaký kľúč do iného kľúča.

2.4.1 Generovanie novej tabuľky

Je možné ale takúto tabuľku vôbec vygenerovať ? Ak si vezmeme, že už máme nejakú kompletnú tabuľku, tak vieme sa z každého vnútorného stavu na základe spätného točenia vygenerovať prislúchajúci Kc, ktorý sa v šifre používa a potom zmeniť, vložiť konštantú hodnotu Frame number, klasickým dopredným točením šifry vieme vygenerovať nový vnútorný stav a tento postup opakovať až kým neprídeme do stavu, kedy meníme farbu. Ako ale do pôvodnej tabuľky zakomponovať tieto vlastnosti ? Potrebujeme implementovať zmenu farby. V pôvodnej tabuľke je zmena farby implementovaná tak, že sa k vnútornému stavu prixoruje nejaká konštanta. To vieme urobiť aj v tomto prípade a takým spôsobom vieme z každého začiatočného bodu vygenerovať nový, druhý koncový bod pôvodnej rainbow tabuľky. Ku každej jednej hodnote vnútorného stavu vieme ale priradiť až 2^{22} koncových stavou s rôznými frame number. To je ale zbytočné, pretože ako sme vraveli, tá množina vnútorných kľúčov je oveľa menšia ako množina vnútorných stavou rainbow

tabuľky. Preto by sme mohli algoritmus upraviť tak, žeby sme celú pôvodnú rainbow tabuľku vedeli rozdeliť na 2^{22} častí a každá časť by mala, bola by točená na základne fixovaného frame number.



Obrázok schéma novej rainbow tabuľky

2.4.2 Možnosti využitia danej rainbow tabuľky

Ako je už známe, tak rainbow tabuľku a šifru A5/1 vieme točiť aj spätne, čo vieme využiť práve v tomto útoku. Ak zachytíme správu, v ktorej sa domievame, že obsahuje podstrčenú napríklad SMS správu, vieme využiť Frame number danej správy a vieme zafixovať predpokladaný vnútorný stav a točiť šifrou A5/1 len na základe frame number. Ak dôjdeme do správneho konca, máme vyhraté, ak nie, môže nastať niekoľko situácií.

1. Buď sme mali zle rozdelenú tabuľku vnútorných štavov na základe frame number, naše pokrytie je zlé a musíme použiť inú rainbow tabuľku (iné pokrytie)
2. Predpokládali sme zlé rozloženie správy, správa je modifikovaná, rozdelená do viacerých burstov
3. Nastala kolízia rainbow tabuľky, interná chyba, alebo nie je možné jednoznačne dobre dotočiť danú tabuľku dokonca.

Chybu v prvom bode vieme eliminovať tak, že použijeme pre celé pokrytie práve pre každý vnútorný stav každý možný frame number. Síce sa zväčší kapacita, no kapacita nie je vysoká v porovnaní s brute force útokom a všetkými možnými stavmi.

Bod číslo 2 je to, k čomu týmto útokom smerujeme. Takto vieme overiť, že náš predpoklad je správny a vieme použiť tento predpoklad na ďalšie lámanie šifry a možný odposluch.

S kolíziou rainbow tabuľky veľa nenarobíme, vieme len využiť to, akú máme rainbow tabuľku a budeme potrebovať overiť, kedy a ako nastávajú kolízie. Tieto overenia sa robia automatizovane a príprava novej takejto tabuľky by si vyžadovala niekoľkoročnú prípravu.

Taktiež aby sme sa vyhli nedosiahnuteľným stavom, vieme modifikovať útok tak, že použijeme model točenia šifry A5/1 dopredu. Budeme predpokládať, že máme vnútorný stav ale pred tým, ako bol pridaný frame number. To sa v pôvodnom útok neukazuje. V podstate je možné pridať len konštanty k pôvodnej rainbow tabuľke ktoré budú vyjadrovať zmenu, respektíve aktívny frame number nakonci točenia.

Tento útok je možné využiť v kombinácií s pôvodným útokom a vieme takto dostať lepšie výsledky. Hlbšia analýza si vyžaduje viac času a viac informácií bude v mojej záverečnej práci. Aktuálny stav je taký, že vieme točiť šifru A5/1 spätne a na vygenerovaných náhodných dátach sa nám daný útok podarilo odskúšať. Generovanie takejto celej rainbow tabuľky je nemožné.

3 Kryptoanalýza šifry KASUMI

3.1 Popis šifry KASUMI

KASUMI je už blokovou šifrou, ktorá sa používa v mobilných sieťach. Ide o upravenú verziu šifry MISTY1 a autori šifry KASUMI tvrdili, že úprava šifry MISTY1 nemá vplyv na novú šifru KASUMI a je rovnako bezpečná ako šifra MISTY1, no časom sa opak potvrdil a šifra KASUMI bola z kryptoanalytického hľadiska prelomená. Daný útok ale nie je možný v mobilných sieťach tak, ako je v súčasnosti použitá šifra KASUMI. Šifra KASUMI využíva 8 kôl, z ktorých je každé kolo rekurzívne, príloha 1. V každom kole sa využívajú dve funkcie FL a FO. Vstupný šifrovaný text je o veľkosti 64 bitov a kľúč je o veľkosti 128 bitov [3.1]. Šifrovaný text sa rozdelí na dve rovnaké časti a každá časť je

buď v každom kole vstupom do funkcie FL a FO, alebo je vstupom do funkcie XOR, ktorá má na vstupe aj výstup z funkcií FL a FO. Funkcia FL obsahuje rotáciu o jeden bit v ľavo a bitový AND a OR s kľúčom. Funkcia FO je z pohľadu kryptoanalýzi kľúčová. Na vstup dostane 32 bitov, ktoré rozdelí na dve časti po 16 bitov. Na ľavú stranu vstupu aplikuje funkciu FI dva krát a na pravú stranu aplikuje funkciu FI jeden krát. Taktiež sa vo funkcií FO aplikuje XOR s kľúčom respektíve XOR pravej a ľavej strany funkcie. Ide o verziu Feistalovej siete. Ďalej funkcia FI je asymetrická Feistalová sieť. Funkcia FI rozdelí vstup na 9 a 7 bitové časti a aplikuje pomerne malý S-box, ktorý je predmetom diskusií a je jedným z hlavných cieľov kryptoanalýzi danej šifry [3.1].

Generovanie, respektíve odvodzovanie kľúča v každom kole je taktiež pomerne veľké zoslabenie šifry MISTY1. totižto kľúč sa generuje tak, že sa vezme pôvodný kľúč a na ten sa aplikuje len logický bitový posun vľavo a XOR funkcia s konštantou. Práve aj na takéto generovanie kľúča mnohí autori poukazovali, že je bezpečnostným rizikom [3.1].

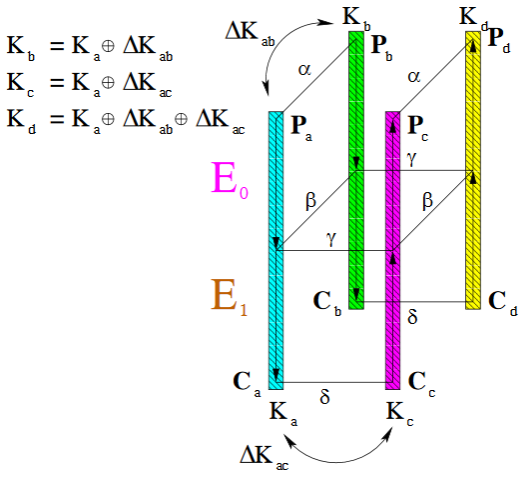
3.2 Diferenčná kryptoanalýza

Súčasné útoky, ktoré sú vedené voči šifre KASUMI sú formou diferenčnej kryptoanalýzi. Diferenčná kryptoanalýza je založená na princípe rozdielu otvorených a šifrových textov a následným hľadaním kľúča, ktorý sa použil pri šifrovaní. Pre viac detailov viď [3.1]. Celý princíp diferenčnej kryptoanalýzi je založený na jednoduchom pozorovaní:

- $d_{k_1}\left(e_{k_1}(m)\right) = m \oplus k_1 \oplus k_1 = m \oplus 0 = m$
- $(m_1 \oplus k_1) \oplus (m_2 \oplus k_2) = m_1 \oplus m_2$ (3.1)

Ak vezmeme v úvahu, že šifra šifruje triviálnym spôsobom a to tak, že otvorený text a kľúč aplikuje XOR funkciu, dešifruje takým istým spôsobom, tak ak vezmeme dve zašifrované text ako v (3.1), spravíme, respektíve aplikujeme funkciu XOR na tieto šifrované texty, tak dosiahneme diferenciu, rozdiel otvorený správ. Tento princíp sa ďalej rozvíjal a využívajú sa pritom rôzne techniky analýzy takýchto párov ako je filtrácia, charakteristiky a analýza S-Boxov ktoré sa v danej šifre vyskytujú.

3.3 Related-Key Boomerang attack



Obrázok 7: Related-Key Boomerang Attack [3.1]

Related key Boomerang attack vychádza z Boomerang attack a je modifikáciou pôvodného Boomerang attack o to, že sa pridá ešte predpoklad, že šifrovaný text je šifrovaný kľúčom s vopred danou diferenciou.

V našom prípade sa budeme zaoberať rôznymi metódami a modifikáciami Boomerang útoku. Boomerang útok je verzia diferenčnej kryptoanalýzi, ktorá pracuje s tým, že existujú dobré dlhe diferencie ale zlé krátke diferencie. To znamená, že je možné mať častokrát opakujúci sa rozdiel v šifrovaných textov niekde počas šifrovania. Pri Boomerangovom útoku si rozdelíme šifrovaný text na 2 časti počas šifrovania a položíme E_0 ako prvú časť šifrovania a E_1 ako druhú časť šifrovania. Ďalej budeme predpokládať, že existuje diferencia aj medzi kľúčami, ktorými bol tento šifrovaný text zašifrovaný a tú si označíme ako ΔK_{ab}. Vytvoríme 2 páry šifrovaných a dešifrovaných textov s vopred stanovenou diferenciou (Obrázok 6), tie následne zašifrujeme a využijeme niekoľko metód na spätné odhalenie kľúča. Celý algoritmus Boomerangovho útoku je možné zhrnúť v nasledujúcom „algoritme“ [3.1]:

- Vyber náhodne P_a a spočítaj $P_b = P_a \oplus \alpha$

- Požiadaj o šifrovanie $C_a = E_{K_a}(P_a)$ a $C_b = E_{K_b}(P_b)$
- Spočítaj $C_c = C_a \oplus \delta$ a $C_d = C_b \oplus \delta$
- Požiadaj dešifrovať $P_c = E_{K_c}^{-1}(C_c)$ a $P_d = E_{K_d}^{-1}(C_d)$
- Skontroluj, či $P_c \oplus P_d = \alpha$

V prípade, že posledná podmienka platí, využijeme nejakú metódu na spätne generovanie kľúča.

Ak sa zamyslíme nad celým princípom Boomerang attack je možné tento útok ešte zefektívniť rôznymi metódami výpočtu párov, takzvaných dvojíc a ich prehadzovaniu. Zefektívnenému Boomgerang útoku hovoríme aj Rectangle attack. Princíp fungovania Rectangle Boomgerang Attack je nasledovný [3.1]:

- $K_a, K_b = K_a \oplus K_{ab}, K_c = K_a \oplus K_{ac}, K_d = K_a \oplus K_{ad}$
- Vyber N otvorených párov (P_a,$P_b = P_a \oplus \alpha$) a požiadaj zašifrovanie P_a kľúčom K_a a P_b kľúčom K_b
- Vyber N otvorených párov (P_c,$P_d = P_c \oplus \alpha$) a požiadaj zašifrovanie P_c kľúčom K_c a P_d kľúčom K_d
- Najdi štvorice (P_a, P_b, P_c, P_d) a odpovedajúce (C_a, C_b, C_c, C_d) splňujúce $C_a \oplus C_c = C_b \oplus C_d = \delta$

V tomto prípade dostávame lepšie pravdepodobnosti útokov, to znamená, že je väčšia šanca, že sa v našom útoku podarí nájsť také dvojice, ktoré budú vyhovovať podmienkam. Celkovo po náročných výpočtoch dostávame, že pri šifrovaní N otvorených dvojíc dostávame približne správnych dvojíc $N^2 2^{-n} (pq)^2$.

3.4 Útok na šifru KASUMI

3.4.1 Diferencie

Je dokázané, že ak stanovíme diferenciu v kľúči rovnú $\Delta K_{ab} = (0,0,1,0,0,0,0,0)$, a vstupnú diferenciu otvorených textov $\alpha = (0_X, (0020\ 0000)_X)$, tak diferencia po prvých troch kôl bude $(0_X, (0020\ 0000)_X) \rightarrow (0_X, (0020\ 0000)_X)$ s pravdepodobnosťou 25%, čo naznačuje, že použitie Boomerangovho útoku by bolo prijateľnou možnosťou. Keďže KASUMI je ale 8 kolová šifra a naša diferencia sa vzťahuje len na 3 kolá, potrebujeme doplniť ešte jedno kolo šifry. Z princípu fungovania šifry KASUMI sa nám v štvrtom kole na ľavej strane šifrovaný text nezmení a na ľavej strane môžeme predpokladať, že dostaneme ľubovoľný výsledok, stanovíme si diferenciu po prvých štyroch kolách na

- $\alpha = (0_X, (0020\ 0000)_X) \rightarrow (y_X, (0020\ 0000)_X)$

S pravdepodobnosťou $2^{-34} = \frac{1}{4}\frac{1}{2^{32}}$. Ak položíme niektoré bity špecificky na jednotku alebo nulu, vieme odhadnúť aj dve bity v danej diferencií po štvrtom kole, čím vieme ešte zlepšiť pravdepodobnosti na $\frac{1}{2^{33}}$, pričom efektívna pravdepodobnosť je $\frac{1}{2^{17}}$. Diferenciu v kolách 5-7 stanovíme ako na začiatku, to znamená, že

- Vstupná diferencia $\gamma = (0_X, (0020\ 0000)_X)$
- $\gamma = (0_X, (0020\ 0000)_X) \rightarrow (0_X, (0020\ 0000)_X)$
- Pravdepodobnosť: $\frac{1}{4}$, efektívna je $\frac{1}{4}$

3.4.2 Popis útoku

Algoritmus vyžaduje veľké množstvo šifrovaných textov, základný algoritmus je popísaný nižšie. Najkritickejšou časťou algoritmu je úloha o tipnutí si kľúča a vydedukovaniu $KL_{8,2}$ a práve tejto časti sa chceme venovať aj v záverečnej práci. Totižto, všetky navrhnuté kryptoanalytické metódy pracujú práve s tým, ako rýchlo vydedukovať v jednom kole kľúča ak máme takéto dáta, ako v algoritme popísanom nižšie.

- Algoritmus:

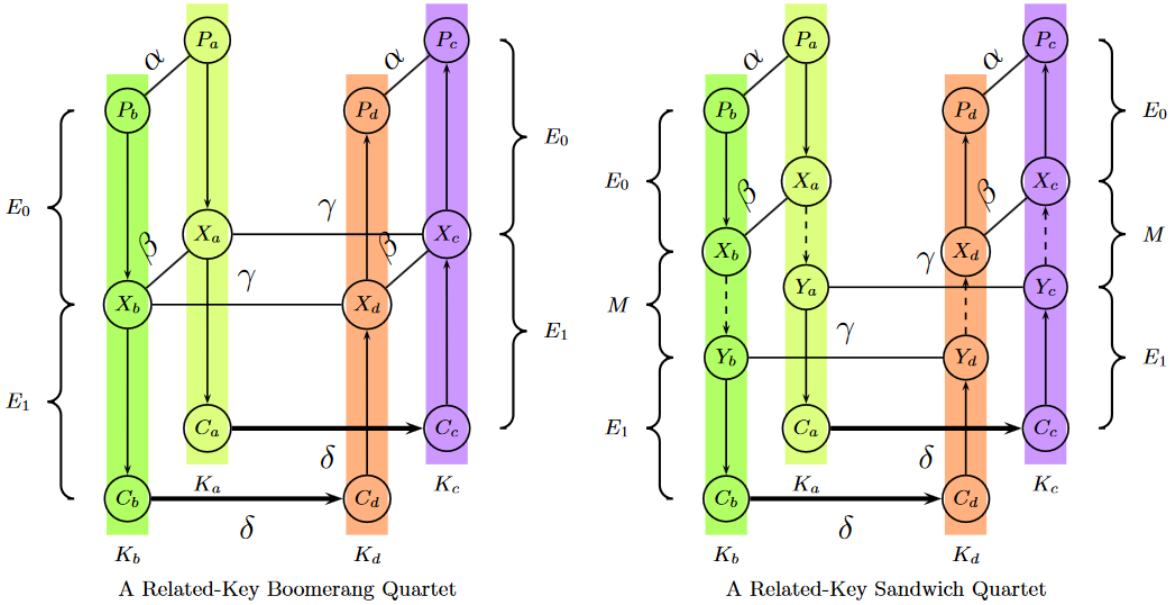
- 2^{51} šifrovaní: $(P_a, P_b = P_a \oplus \alpha)$, $P^0_{a_{LL}} = 0$, $P^1_{a_{LR}} = 1$, $\mathrm{E}(P_a, K_a)$, $\mathrm{E}(P_b, K_b)$
  Index: $\left(C_{a_{RL}} C_{a_{RR}} C_{b_{RL}} C_{b_{RR}}\right)$
- 2^{51} šifrovaní: $(P_c, P_d = P_c \oplus \alpha)$, $P^0_{c_{LL}} = 0$, $P^1_{c_{LR}} = 1$, $\mathrm{E}(P_c, K_c)$, $\mathrm{E}(P_d, K_d)$
  Index: $\left(C_{c_{RL}} \oplus 0020_X, C_{c_{RR}} C_{d_{RL}} \oplus 0020_X, C_{d_{RR}}\right)$ (zhoduje sa diferencia?)
- $\left(C_{c_{RL}} \oplus 0020_X, C_{c_{RR}} C_{d_{RL}} \oplus 0020_X, C_{d_{RR}}\right)$
  - Nájdeme (P_a, P_b), pokračujeme štvoricou (P_a, P_b, P_c, P_d)
- 2^{38} približne bude vyhovovať
- Tipneme si kľúč $(KO_{8,1}, KI_{8,1})$ a vydedukujeme
  $KL_{8,2}$ - je možné vypočítať vstupné a výstupne
  diferencie do OR funkcie (spor)
- Tipneme si kľúč $(KO_{8,3}, KI_{8,3})$ a vydedukujeme $KL_{8,2}$ - spočítame
  diferencie vstupu a výstupu
- Pre všetky vyhovujúce kombinácie urob šifrovanie a over výsledok

Tento algoritmus je možné ešte zefektívniť modifikáciami a vieme sa dostať na časové zložitosti pri šifrovaní $2^{52,6}$ vstupných textov $2^{86,6}$, analýzou až na časovú zložitosť $2^{76,1}$ [3.1].

3.4.3 Sandwich attack

Existuje ešte efektívnejší útok na šifru KASUMI. Pri tomto útoku sa v podstate taktiež využíva modifikovaný Boomerangov útok. Ide o útok typu Sandwich attack, ktorý ale znova pracuje s odhaľovaným kľúčov ako v prípade Boomerangového útoku popísaného vyššie. Bližšie o tomto útoku a jeho modifikáciách už v záverečnej práci. Jeho zložitosti sú ale uskutočniteľné v reálnom čase a boli aj experimentálne overené:

- 2^{32} časová zlozitosť
- 2^{30} pamäťová zlozitosť
- 2^{25} šifrových textov



Obrázok 8: Sandwitch Attack

3.4.4 Analýza a návrhy útoku 3.4.2

Dôležitou vlastnosťou šifry KASUMI je to, že šifra pracuje s otvoreným aj šifrovaným textom o veľkosti 2^{64}. Autor útoky si daný paradox nevšimol. V jeho útoku je výsledkom útok o veľkosti práve 2^{74} a nejaké drobné. Prečo ale potrebujeme útok, ktorý má horšiu časovú zložitosť ako veľkosť kľúča ? Je možné nejako využiť práve veľkosť kľúča ? Ak by sme počas útoku využili tento fakt, vedeli by sme spôsobiť to, že daný útok spravíme efektívnejšie ? Ak vezmeme do úvahy to, že po tom, ako máme diferenčné dvojice $C_i^8 \oplus C_j^8 = \left(C_i^{7L} \oplus C_j^{7L}\right) \oplus \left(C_i^{7Ln} \oplus C_j^{7Ln}\right)$, pričom C_i^{7Ln} je výstup z funkcie FL8, tieto majú veľkosť 32 bitov. Ak by sme pre každú jednu dvojicu vygenerovali každú druhú dvojicu, dostávame celkovo $2^{32}2^{32} = 2^{64}$ neznámych bitov. Pričom v útoku stále pracujeme s časovou zložitosťou 2^{74}. Následne v kroku útoku, kde odhadujeme správanie z feistalovej siete, kde budú chyby v AND a OR možnostiach, vieme použiť možnosti prepočítaných všetkých kombinácií správ,

konkrétne ich ľavých časti (výstupov z funkcie FL8). Túto analýzu ale dokončenú nemám, čo plánujem dokončiť v začiatkom letného semestra. Nejaké časti už som navrhol daného útoku, no neviem, či som sa nepomýlil a potrebujem si dané informácie overiť.

Týmto spôsobom je možné generovať a nie len hádať hodnoty ktoré sa vyskytujú vo funkcií FL8. Totižto ak uhádneme hodnoty vo funkcií FO tak ako je to popísané v útoku predtým, vieme využitím toho, že prejdeme, vyskúšame každú uhádnutú hodnotu funkcie FO dokončiť šifrovanie s tým, že dostaneme ďalší spor s tým, ako majú vyzerať hodnoty v záverečnej fáze funkcie FO, ktoré majú len keyspace 64 bitov. Tj. Prejdeme hodnoty približne 74 bitov ktoré hádame a takzvaným počítadlom zistíme, ktoré z 64 bitov tomu môžu odpovedať. Niečo ako keby sme využívali diferenčnú kryptoanalýzu v diferenčnej kryptoanalýze.

Záver

Popísali sme, aj keď len veľmi jednoducho možnosti útokov na aktuálne šifry, ktoré sa používajú v mobilných sieťach. Tento článok nie je bližšie venovaný šifre SNOW 3G ktorá sa používa v aktuálne najnovšom móde 4G, spolu s AES, no v prípade, žeby sa ukázala šifra SNOW 3G ako veľmi slabá, tak je možnosť túto šifru vypnúť a povoliť len šifrovanie pomocou AES. V našej práci sa budeme venovať útoku, ako je možné mobilnú sieť odpočúvať a taktiež aj kryptoanalýze a rozširovanie útokom ako Boomerangov útok a Sandwich útok, pri ktorých je ešte veľa nepreskúmaných možnosti. Zameriame sa na analýzu S-Boxov použitých v šifre KASUMI a na spätné generovanie kľúčov po úspešnom Boomerangovom útoku.

Generovanie rainbow tabuliek je možné otestovať a potrebujem dopočítať pokrytie a časové zložitosti, respektíve časové náročnosti generovania rainbow rabuliek. Možnosť daného útoku mám overenú, no potrebujem vedieť pokrytie, aby som vedel stanoviť či tento útok bude efektívnejší ako vyskúšanie všetkých možných správ (2^{114}) alebo nie. Taktiež potrebujem dotiahnuť návrh spätného točenia, respektíve daného algoritmu. Keďže algoritmus už funguje, ale neviem vyriešiť v ňom pár technických problémov.

V prípade šifry KASUMI vidím náznaky útoku na konci. Potrebujem taktiež dokončiť analýzu daného útoku, na čo počas semestra nebol čas. Keďže my budeme poznať správu, respektíve výstup z FL8 funkcie, vieme veľmi ľahko dopočítať hodnoty, ktoré sa nám tam budú vyskytovať počas výpočtu, respektíve odhadovania kľúča a úlohy v útoku ako odhadnúť kľúč nám budú vedieť zredukovať veľkosť množiny kľúčov, ktoré následne budeme potrebovať spracovať.

Zoznam použitej literatúry

[1.1] Citované online, 23.6.2017, zdroj: https://en.wikipedia.org/wiki/GSM#History

[1.2] Olaussen, L., S., Dohmen, J.,R.,: UMTS Authentication and Key Agreement, dostupné online, 23.6.2017: https://brage.bibsys.no/xmlui/bitstream/handle/11250/137418/master_ikt_2001_dohmen.pdf?sequence=1

[2.1] Citované online, 23.6.2017, zdroj: https://en.wikipedia.org/wiki/GSM_frequency_bands

[2.2] NOHL, Karsten. Attacking phone privacy. *Black Hat USA*, 2010, 1-6.

[2.3] https://opensource.srlabs.de/projects/a51-decrypt/files

[2.4] https://evilsocket.net/2016/03/31/how-to-build-your-own-rogue-gsm-bts-for-fun-and-profit/

[2.5] QUIRKE, Jeremy. Security in the GSM system. *AusMobile, May*, 2004, 1-26.

[2.6] Citované online, 25.6.2017, zdroj: https://brmlab.cz/user/jenda/gsm

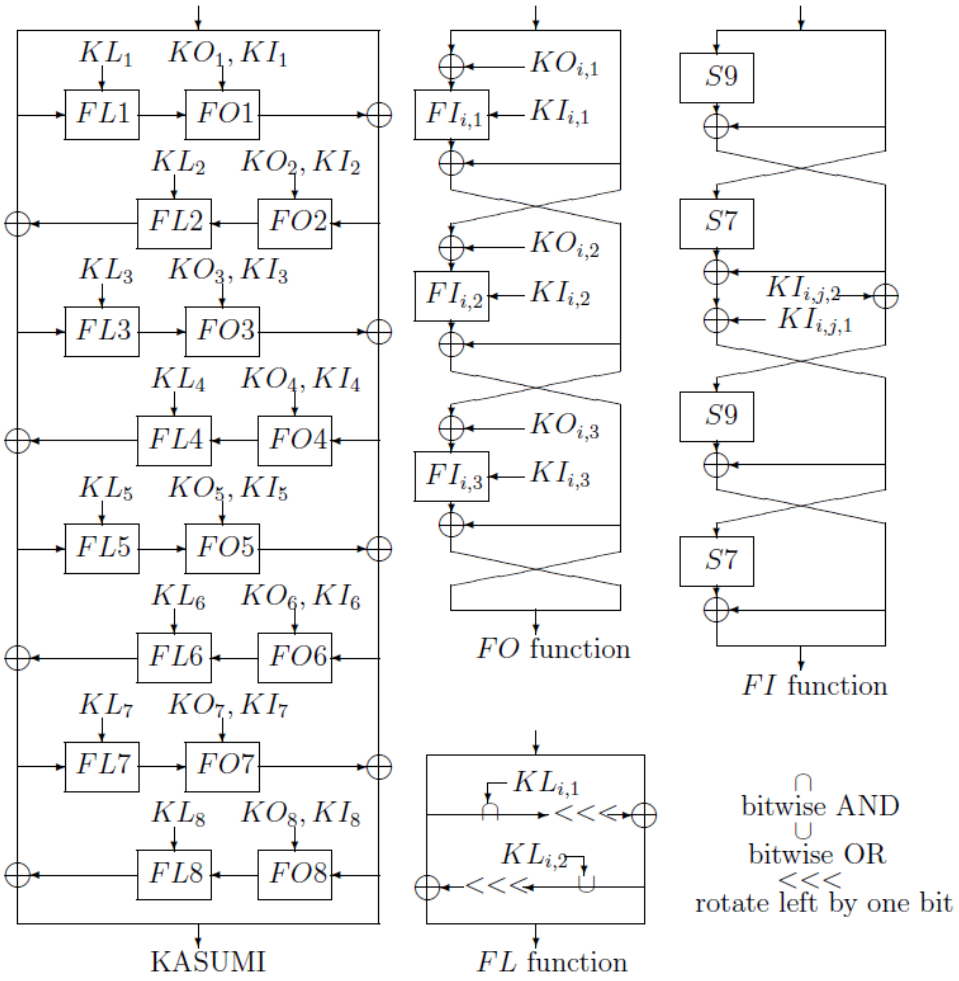
[2.7] Citované online, 25.6.2017, zdroj: https://en.wikipedia.org/wiki/A5/1

[2.8] BIRYUKOV, Alex; SHAMIR, Adi; WAGNER, David. Real Time Cryptanalysis of A5/1 on a PC. In: *International Workshop on Fast Software Encryption*. Springer Berlin Heidelberg, 2000. p. 1-18. dostupné online: https://www.researchgate.net/profile/Alex_Biryukov2/publication/2539606_Real_Time_Cryptanalysis_of_A51_on_a_PC/links/53d13d860cf228d363e5b625.pdf

[2.9] BIHAM, Eli; DUNKELMAN, Orr. Cryptanalysis of the A5/1 GSM stream cipher. *Progress in Cryptology—INDOCRYPT 2000*, 2000, 43-51. dostupné online: ftp://nozdr.ru/biblioteka/kolxo3/Cs/CsLn/Progress%20in%20Cryptology%20-%20INDOCRYPT%202000(LNCS1977,%20Springer,%202000)(ISBN%203540414525)(305s).pdf#page=53

[3.1] E. Biham, O. Dunkelman: Techniques for Cryptanalysis of Block Ciphers (Information Security and Cryptography), Springer, 2017, ISBN 978-3642172311

Príloha 1 [3.1]:


KL_1
KO_1, KI_1
$FL1$
$FO1$
KL_2
KO_2, KI_2
$FL2$
$FO2$
KL_3
KO_3, KI_3
$FL3$
$FO3$
KL_4
KO_4, KI_4
$FL4$
$FO4$
KL_5
KO_5, KI_5
$FL5$
$FO5$
KL_6
KO_6, KI_6
$FL6$
$FO6$
KL_7
KO_7, KI_7
$FL7$
$FO7$
KL_8
KO_8, KI_8
$FL8$
$FO8$
KASUMI
$KO_{i,1}$
$FI_{i,1}$
$KI_{i,1}$
$KO_{i,2}$
$FI_{i,2}$
$KI_{i,2}$
$KO_{i,3}$
$FI_{i,3}$
$KI_{i,3}$
FO function
$KL_{i,1}$
$KL_{i,2}$
FL function
$S9$
$S7$
$KI_{i,j,2}$
$KI_{i,j,1}$
$S9$
$S7$
FI function
bitwise AND
bitwise OR
rotate left by one bit

Príloha 2 [3.1]:

Round	$KL_{i,1}$	$KL_{i,2}$	$KO_{i,1}$	$KO_{i,2}$	$KO_{i,3}$	$KI_{i,1}$	$KI_{i,2}$	$KI_{i,3}$
1	$K_1 \lll 1$	K'_3	$K_2 \lll 5$	$K_6 \lll 8$	$K_7 \lll 13$	K'_5	K'_4	K'_8
2	$K_2 \lll 1$	K'_4	$K_3 \lll 5$	$K_7 \lll 8$	$K_8 \lll 13$	K'_6	K'_5	K'_1
3	$K_3 \lll 1$	K'_5	$K_4 \lll 5$	$K_8 \lll 8$	$K_1 \lll 13$	K'_7	K'_6	K'_2
4	$K_4 \lll 1$	K'_6	$K_5 \lll 5$	$K_1 \lll 8$	$K_2 \lll 13$	K'_8	K'_7	K'_3
5	$K_5 \lll 1$	K'_7	$K_6 \lll 5$	$K_2 \lll 8$	$K_3 \lll 13$	K'_1	K'_8	K'_4
6	$K_6 \lll 1$	K'_8	$K_7 \lll 5$	$K_3 \lll 8$	$K_4 \lll 13$	K'_2	K'_1	K'_5
7	$K_7 \lll 1$	K'_1	$K_8 \lll 5$	$K_4 \lll 8$	$K_5 \lll 13$	K'_3	K'_2	K'_6
8	$K_8 \lll 1$	K'_2	$K_1 \lll 5$	$K_5 \lll 8$	$K_6 \lll 13$	K'_4	K'_3	K'_7

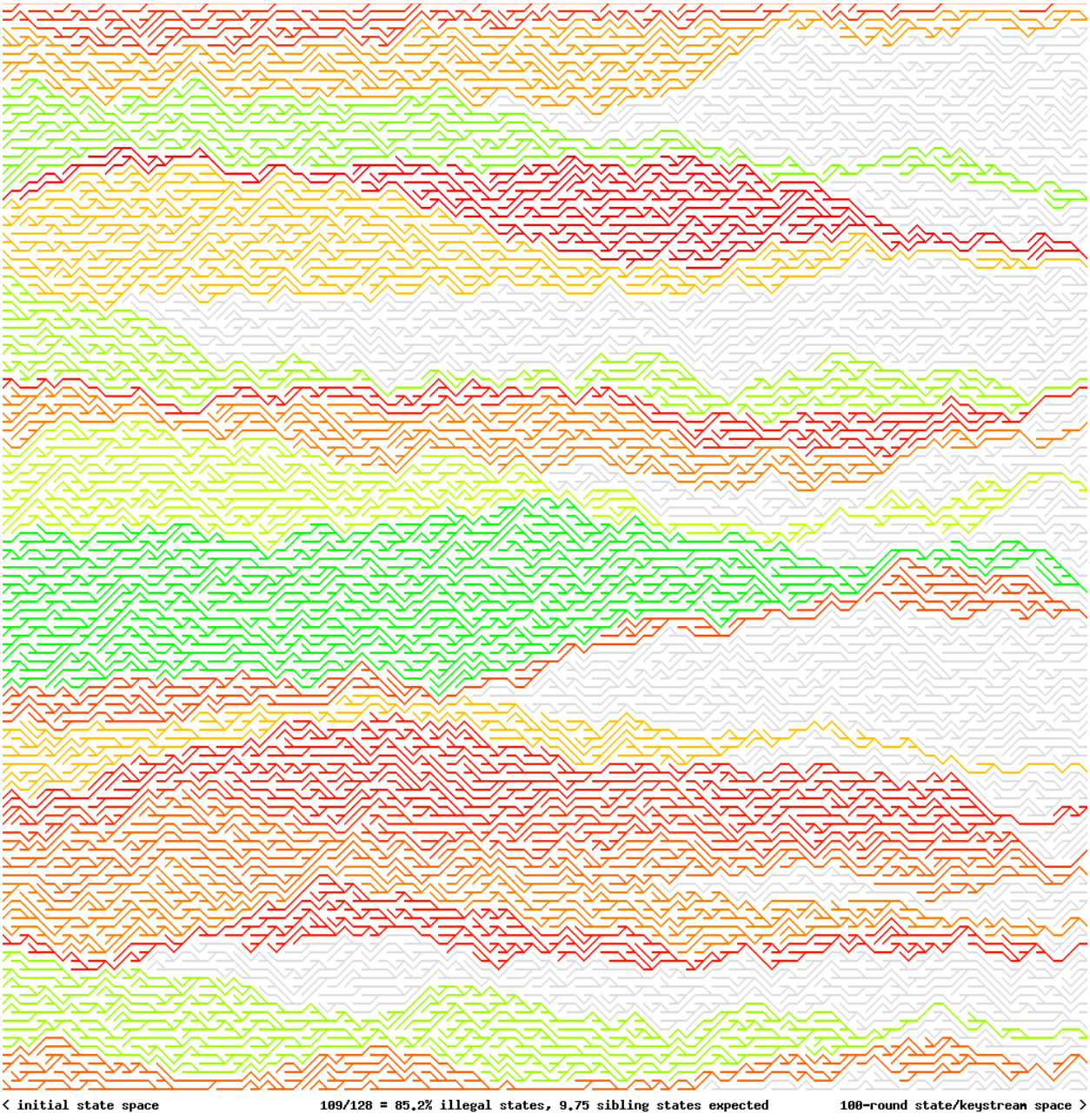
$X \lll i$ — X rotated to the left by i bits

Table C.3: KASUMI's Key Schedule Algorithm
KASUMI ly zegztnd izz aeyig mzixebl‘

Round	1	2	3	4	5	6	7	8
Constant	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
Value	0123_x	4567_x	$89AB_x$	$CDEF_x$	$FEDC_x$	$BA98_x$	7654_x	3210_x

Table C.4: KASUMI's Key Schedule Constants
KASUMI ly zegztnd izz aeyig mzixebl‘ yeniyay mireawd

Príloha 3:



Obrázok ukazuje ako sa mení keyspace v šifre A5/1. Šedým sú cesty ktoré nie sú dosiahnuteľné bežným točením šifry vpred. Zeleným sú cesty ktoré majú niekoľkých predchodcov ale len jeden výstup a červeným sú cesty, ktoré majú niekoľkých, ale iba pár, predchodcov a končia v jednom stave.