Capitolul 5

Securitatea comunicaţiilor GSM

5.1 Descrierea sistemului GSM

GSM (Global System for Mobile Communications) a fost dezvoltat de $3GPP$ (*3rd Generation Partnership Project*) ca standard pentru comunicaţiile celulare digitale. Lucrările de proiectare au început în anul 1987, sistemul fiind realizat în forma finală prin cooperarea a 17 ţări europene; punerea în funcţiune a primelor reţele GSM s-a realizat în 1991.

Deşi iniţial GSM era destinat doar ca un standard european, el s-a răspândit rapid în întreaga lume; de remarcat că la nivel mondial există de asemenea şi ADC (*American Digital Cellular*) – sistemul digital nord-american şi JDC (*Japonese Digital Cellular*) – sistemul digital din Japonia. Se estimează că în acest moment 82% din piaţa mobilă la nivel global foloseşte standardul GSM. La sfârşitul anului 2005 sistemul dispunea de $1,5$ miliarde utilizatori, iar astăzi numai câteva ţări – printre care Japonia şi Coreea de Sud – nu se află sub acoperirea sistemului GSM.

Printre motivele care au stat la baza acestei expansiuni putem aminti îmbunătăţirile din domeniul tehnologiei telecomunicaţiilor, precum şi reducerile constante de preţuri – atât pentru infrastructură cât şi pentru telefoanele mobile. Sistemul GSM este bazat pe o tehnologie celulară de a doua generaţie – $2G$ (oferă voce digitală, spre deosebire de tehnica analogică folosită în sistemele anterioare). Mulţi operatori continuă să investească în dezvoltarea reţelelor GSM, acest lucru neîmpiedicând preocupările de a introduce noi funcţionalităţi şi de a micşora costul lor operaţional.

Standardul GSM s-a dezvoltat continuu, aşa încât versiunea ´97 a fost îmbogăţită cu $GPRS$ – serviciul de transfer pe pachete de date – care permite ca informaţia digitală să circule printr-o reţea de telefonie mobilă. Sistemele celulare $2G$ combinate cu $GPRS$ sunt de multe ori descrise ca $2.5G$. Acesta completează transferul rapid de date (CDS) şi SMS-ul, ajungând până la o viteză teoretică de 171.2 *kbps* (kilobiţi pe secundă). O viteză şi mai mare pentru transmiterea de date în GSM a fost introdusă de $EDGE$ (tehnologie a reţelei $GSM/GPRS$, destinată să îmbunătăţească calitatea serviciilor de date), care asigură o viteză de până la 236 *kbps*.

5.1.1 Cerinţe pentru sistemul GSM

Trecerea la sistemele celulare digitale s-a datorat în principiu:

1. Capacităţii reduse a sistemelor analogice.
2. Necesităţii de a pune la dispoziţia utilizatorilor un sistem compatibil cu alte sisteme digitale cu servicii integrate.

Creşterea capacităţii sistemelor digitale, faţă de sistemele analogice, se explică, pe de o parte prin creşterea capacităţii de trafic a macrocelulelor şi – pe de altă parte – prin posibilitatea reducerii dimensiunilor celulelor, adică prin introducerea microcelulelor (sistemul celular GSM are trei tipuri de celule: macrocelule, ce acoperă o rază de 15 km, microcelule – rază de 500 m şi picocelule – pe distanţe mai mici de 100 m).

Iniţial, sistemul GSM a fost proiectat să îndeplinească următoarele cerinţe:

1. Calitate bună a transmisiei,
2. Costuri mici pentru servicii şi terminal,
3. Suport pentru deplasări internaţionale,
4. Capacitate de a suporta terminale portabile,
5. Suport pentru servicii şi facilităţi noi,
6. Eficienţă spectrală şi compatibilitate cu sistemul $ISDN$ (reţea digitală cu servicii integrate, care permite interconectarea mai multor dispozitive: calculatoare, telefoane, aparate faximile etc).

5.1.2 Serviciile oferite de GSM

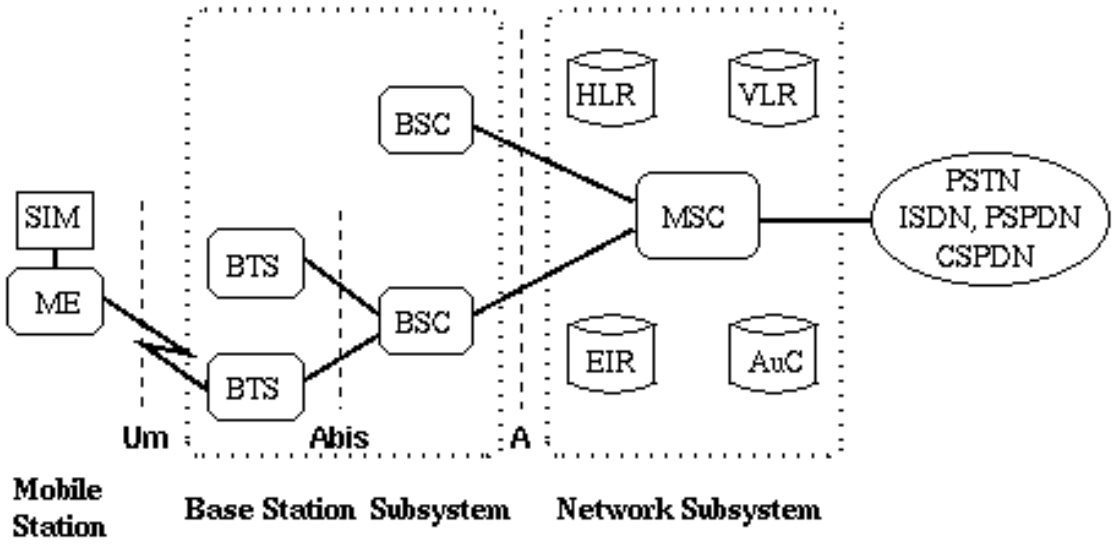
Serviciile de telecommunicaţii se împart în servicii de transfer de date, teleservicii, şi servicii suplimentare. Teleserviciul de bază suportat de GSM este telefonia. Semnalul vocal este codificat digital şi transmis prin reţeaua GSM ca un flux de semnal digital.

Un serviciu specific oferit de GSM – inexistent în sistemele analogice – este *Serviciul de Mesaje Scurte* (SMS). Alte servicii suplimentare sunt oferite în specificaţiile curente, care includ mai multe forme de transfer al apelului; ele includ identificarea apelantului, apel în aşteptare, conversaţii multiple (conferinţe) etc.

Reţeaua pe care se construieşte sistemul GSM se împarte în:

1. *Staţia mobilă (MS - Mobile Station)*: este componenta aflată la abonat.
2. *Subsistemul Staţie de bază (BSS - Base Station Subsystem)*: controlează legătura radio cu staţia mobilă.

3. *Subsistemul Reţea (Network Subsystem)*: principala sa componentă este *Centrul de comutaţie a serviciilor mobile (MSC - Mobile services Switching Center).*



SIM Subscriber Identity Module	BSC Base Station Controller	MSC Mobile services Switching Center
ME Mobile Equipment	HLR Home Location Register	EIR Equipment Identity Register
BTS Base Transceiver Station	VLR Visitor Location Register	AuC Authentication Center

Vom detalia fiecare din aceste componente.

5.1.3 Staţia mobilă

Staţia mobilă constă dintr-un echipament mobil (cunoscut şi sub denumirea de "*terminal*") şi un smartcard *SIM* (*Subscriber Identity Module*).

Echipamentul mobil

Cel mai cunoscut echipament mobil este telefonul celular – folosit iniţial numai pentru apeluri vocale; astăzi însă, tendinţa este de a-i asocia tot mai multe dispozitive mobile în care telefonul este doar o componentă:

- *PDA* cu telefon mobil pentru transmisii vocale şi de date.
- Console pentru jocuri cu telefon mobil integrat pentru voce şi date.
- Telefoane mobile pentru voce cu interfaţă Bluetooth integrată, care permite dispozitivelor de tip *PDA* sau notebook să folosească telefonul pentru conexiune la Internet.

Echipamentul mobil este identificat în mod unic de $IMEI$ (*International Mobile Equipment Identity*) – un cod care se găseşte de obicei tipărit pe telefon, sub bateria de alimentare şi este folosit strict pentru identificare, neavând o relaţie permanentă cu abonatul.

Componenta principală a unui telefon mobil este procesorul – care conţine o unitate centrală $RISC$ (cu set minimal de instrucţiuni) şi un procesor cu semnal digital (DSP). Procesorul $RISC$ este responsabil de următoarele acţiuni:

1. Manevrarea informaţiei primite prin diversele canale ($BCCh$, PCh etc).
2. Stabilirea apelului şi administrarea mobilităţii (căutarea reţelei, reactualizarea locaţiei, decalajul în timp etc).
3. Conexiunile prin interfeţele externe ca *Bluetooth*, USB etc.

Cum multe dintre aceste sarcini trebuie efectuate în paralel, procesorul $RISC$ foloseşte un sistem de operare multitasking în timp real, capabil să furnizeze informaţii pentru transmisiile prin interfaţa radio conform structurii şi coordonării în timp impusă de GSM.

Toate celelalte sarcini – ca manevrarea tastaturii, actualizarea ecranului şi interfaţa grafică – au o prioritate redusă.

Cartela SIM

Cartela SIM (Subscriber Identity Module) este un smartcard care stochează informaţii esenţiale, cum ar fi $IMSI$ (*International Mobile Subscriber Identity*), MSI – un număr unic asociat fiecărui abonat, K_i (cheia secretă folosită pentru autentificare) şi alte informaţii despre utilizator. În general, informaţiile sunt protejate printr-un număr personal de identificare (PIN).

De fapt, un card SIM este mai mult decât o simplă memorie; el conţine un microcontroller care poate fi folosit pentru scopuri adiţionale, cum ar fi generarea valorii $SRES$ în procesul de autentificare (descris ulterior). Este obligatoriu ca $SRES$ să fie calculat în interiorul SIM-ului şi nu în telefonul mobil, pentru a proteja cheia secretă K_i, care intervine de asemenea în procesul de autentificare.

Din punct de vedere logic, datele sunt stocate pe SIM în directoare şi fişiere într-o manieră similară stocării pe hard-ul unui PC.

În specificaţia tehnică $3GPP$ ($3GPP\ TS$), directorul *root* se numeşte fişierul principal (MF). Următoarele directoare se numesc *fişiere dedicate* (DF) iar fişierele normale se numesc *fişiere elementare* (EF). Datorită faptului că memoria unui SIM este extrem de limitată, fişierele nu pot fi identificate prin numele directorului şi fişier. Sunt folosite numere hexazecimale care ocupă numai 2 octeţi de memorie fiecare. Standardul asignează identificatori pentru aceste numere dar nu le stochează pe SIM.

Exemplul 5.1. *Directorul root este identificat, de exemplu, prin* $0x3F00$*, directorul* GSM *prin* $0x7F20$*, iar fişierul care conţine* $IMSI$*, de pildă, este identificat prin* $0x6F07$.

Pentru a citi IMSI-ul de pe cartela SIM, staţia mobilă trebuie să deschidă fişierul având calea

$$0x3F00 \;\; 0x7F20 \;\; 0x6F07.$$

Fişierele de pe *SIM* sunt protejate prin diferite drepturi de acces. Cu aceste atribute, producătorul poate controla dacă un fişier este citit sau scris numai dacă este accesat de telefonul mobil.

5.1.4 Subsistemul Staţie de bază

Dacă funcţionalitatea subsistemului reţea pentru *GSM* este în cea mai mare parte definită de soft-ul adiţional, subsistemul *Staţie de bază* este specific. Vechile generaţii de sisteme se bazau pe transmisii analogice prin interfaţă radio.

Subsistemul *Staţie de bază* este responsabil pentru asigurarea traficului şi a semnalului între staţia mobilă şi centrul de comutaţie a serviciilor mobile. El este compus din:

Banda de frecvenţe

În sistemul *GSM* ea este formată din două sub-benzi:

1. $890 - 915$ MHz: comunicarea de la Staţia Mobilă la Staţia de Bază (uplink);
2. $935 - 960$ MHz: comunicarea de la Staţia de Bază la Staţia Mobilă (downlink).

Aceste sub-benzi (fiecare de câte 25 MHz) sunt împărţite în 124 perechi de frecvenţe purtătoare, fiecare pereche având alocată o bandă de 200 kHz.

Fiecare frecvenţă purtătoare este utilizată pentru transportul a 8 canale telefonice distincte, multiplexate în timp (*TDMA*).
Deci numărul de căi telefonice este de $8 \times 124 = 992$.
Acest lucru este posibil deoarece datele şi semnalele audio sunt codate şi transmise digital. Transmisia se face în pachete în interiorul intervalului de timp alocat, cu o rată de 271 kbps.

Datorită cererii tot mai mari din ţările europene, ulterior s-a adăugat un interval suplimentar de frecvenţe pentru *GSM*, care foloseşte pentru uplink banda de frecvenţe $1710 - 1785$ MHz iar pentru downlink $1805 - 1880$ Mhz. În locul lăţimii de bandă de 25 MHz – normală frecvenţei 900 MHz – frecvenţa 1800 Mhz oferă o lăţime de bandă de 75 Mh, care permite încă 375 canale adiţionale[1].

GSM foloseşte diverse canale pentru transmiterea datelor. Acestea sunt împărţite în

[1]În America de Nord – datorită existenţei reţelelor de telefonie mobilă analogică – la apariţia *GSM*-ului, intervalele 900 MHz şi 1800 MHz erau deja ocupate; de aceea s-au deschis frecvenţe noi de bandă pe 1900 MHz şi 850 MHz. Prin urmare, multe telefoane mobile *GSM* din *SUA* nu pot fi folosite în Europa şi vice versa.

- Canale fizice (determinate de timesloturi),

- Canale logice: folosite pentru transmiterea datelor utilizatorului şi a datelor de semnalizare.
  Dacă datele dintr-un canal logic sunt dedicate unui singur utilizator, atunci canalul este numit "canal dedicat". Dacă avem în vedere transmiterea de date pentru mai mulţi utilizatori, canalul este numit "canal comun".

Canalele dedicate sunt clasificate în:

- *Canalul de trafic* (TCh): canal pentru datele utilizatorilor. Poate fi folosit pentru transmisia de semnal vocal digital până la 14.4 kbps.

- *Canale asociate rapide de control* ($FACCh$): ascund timeslot-uri din traficul alocat şi sunt folosite pentru cereri neprogramate de control, cereri transmise ocazional (de exemplu handover-urile).

- *Canale asociate lente de control* ($SACCh$): folosite în direcţia uplink pentru a raporta măsurile de calitate a semnalului din celula deservită şi din cele vecine. În sens invers, canalul este folosit pentru transmisia de comenzi de control de putere către staţia mobilă.

- *Canale de control dedicate* (DCh): canale multiplexate într-un canal de trafic standard. Sunt folosite pentru înregistrare, reactualizarea locaţiei, autentificare şi apelare.

- *Canal comun descendent* ($AGCh$): utilizat pentru a transmite mobilului mesaje de alocare a unui canal dedicat.

  Pe lângă canalele dedicate – asignate unui singur utilizator – există şi canale comune, dedicate tuturor utilizatorilor dintr-o celulă:

- *Canal de sincronizare* (SCh): folosit de staţiile mobile în timpul căutării reţelei şi a celulei.

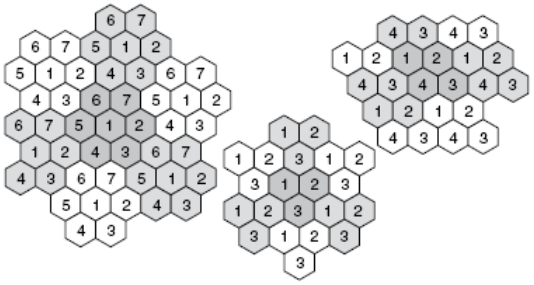
- *Canale de control emis* (BCh): canale logice necesare transmisiei periodice a informaţiilor generale.

- *Canal de paging* (PCh): canal logic care transportă mesajele de difuzare pe interfaţa radio; este folosit în principal pentru anunţarea mobilului despre apelurile primite.

- *Canal de acces aleator* ($RACh$): canal de accesare a reţelei $MS-BTS$. Este folosit de MS pentru a cere alocarea unui canal dedicat.

Base Transceiver Station (BTS)

Staţiile de bază sunt componentele cele mai vizibile din reţeaua GSM. Ele înlocuiesc conexiunile prin cablu din telefonia fixă prin conexiuni fără fir cu utilizatorul şi reprezintă componenta cea mai răspândită din reţeaua de telefonie mobilă.



Arhitectura celulară (celulele etichetate cu acelaşi număr folosesc aceeaşi frecvenţă)

BTS conţine echipamentul necesar transmiterii şi primirii semnalului, antene şi echipament pentru criptarea şi decriptarea comunicaţiilor cu BSC (*Base Station Controler*). De obicei, un BSC are sub control până la 100 BTS. Reţelele sunt structurate astfel ca să aibă mai multe BSC distribuite în regiuni apropiate de BTS-urile lor, conectate la grupuri centralizate de MSC.

Teoretic, o staţie de bază (BTS) poate acoperi o suprafaţă cu raza de până la 35 km, numită "celulă". Deoarece un BTS poate deservi simultan un număr limitat de utilizatori celulele sunt de fapt mult mai mici, în special în mediile urbane foarte dense. Aici, celulele acoperă arii de rază $3-4$ km în zonele rezidenţiale şi coboară până la raze de numai 100 m în zonele comerciale foarte frecventate.
Chiar şi în zonele rurale, aria de acoperire a unei celule este de obicei sub 15 km; puterea redusă de transmisie a staţiei mobile este – în acest caz – factorul limitator.

Având în vedere că emisiile staţiilor de bază diferite dintr-o reţea nu trebuie să interfereze, toate celulele învecinate trebuie să emită pe frecvenţe diferite (vezi figura). Cum o celulă este înconjurată de multe altele, o staţie de bază poate folosi doar un număr limitat de frecvenţe diferite pentru a-şi mări capacitatea.

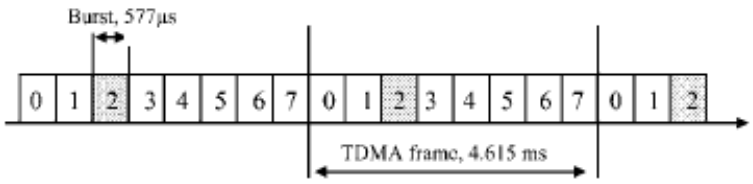
Interfaţa radio

Legătura între BTS şi terminalul mobil este denumită *interfaţă radio* sau *interfaţă* Um. Există două metode care permit staţiei de bază să comunice simultan cu mai mulţi utilizatori.

1. Acces multiplu prin diviziune în frecvenţă ($FDMA$): utilizatorii pot comunica cu staţia de bază pe frecvenţe diferite, fără a interfera unii cu alţii.

2. Acces multiplu prin diviziune în timp ($TDMA$ –Time Division Multiple Access). GSM foloseşte frecvenţe cu lăţimea de bandă de 200 kHz prin care 8 utilizatori pot

comunica simultan cu staţia de bază. Utilizatorii sunt multiplexaţi în timp prin împărţirea în frame-uri cu durata de $4,615$ ms. Fiecare frame conţine 8 *timeslot*-uri independente, fiecare timeslot fiind folosit pentru comunicarea cu alt utilizator. Durata unui timeslot este de $0,577$ ms şi cuprinde 148 biţi, cu o perioadă de gardă[2] de $8,25$ biţi între slot-uri.



Frame TDMA

Prin combinarea procedeelor $FDMA$ şi $TDMA$ se poate calcula capacitatea totală a unei staţii de bază.

Controllerul Staţiei de bază (BSC)

În timp ce staţia de bază este un element de interfaţă care conectează staţia mobilă cu reţeaua, controlerul staţiei de bază (*Base Station Controller* - BSC) este responsabil de stabilirea, întreruperea şi menţinerea conexiunilor celulelor care sunt conectate la el.

Funcţii îndeplinite de un BSC:

- Dacă un utilizator doreşte să efectueze un apel vocal, să trimită un SMS etc, staţia mobilă trimite un mesaj de cerere de canal către BSC, care verifică dacă există vreun canal $SDCCh$ disponibil, şi în caz afirmativ, îl activează. Apoi BSC trimite un mesaj de asignare staţiei mobile prin $AGCh$, care include şi numărul canalului $SDCCh$ asignat.

- Stabileşte canalele de semnalizare pentru apelurile primite sau mesajele scurte.

- Menţine conexiunea. Cum utilizatorii se pot plimba liberi în reţea în timpul unei convorbiri, se poate întampla ca ei să depăşească acoperirea celulei în care a fost iniţiat apelul. În acest caz BSC intervine, direcţionând apelul către cea mai apropiată celulă.

- Pentru a reduce interferenţa, răspunde de controlul puterii de transmisie pentru fiecare conexiune prin interfaţa radio.
  Pentru staţia mobilă, un control activ al puterii poate reduce puterea de transmisie

[2]Perioada de gardă este o perioadă în care nu se transmite informaţie; deoarece distanţa unui utilizator faţă de staţia de bază se poate schimba, se evită astfel suprapunerea între semnale.

în condiţii de recepţie favorabile. Dacă puterea de transmisie a staţiei mobile trebuie să crească/scadă, BSC trimite un mesaj de control către BTS. În practică, controlul puterii şi adaptarea se fac la fiecare $1-2$ secunde. În timpul stabilirii apelului, staţia mobilă foloseşte nivelul cel mai înalt admis de putere; acesta este apoi redus treptat de către reţea.

- Are în grijă controlul decalajului în timp. $TDMA$ solicită ca semnalele de la toate mobilele care folosesc acelaşi canal (de paging) să atingă Staţia de Bază la momente corecte de timp, fără să se suprapună.

Exemplul 5.2. *Dacă Staţia de Bază furnizează un semnal de referinţă, mobilele aflate în apropierea sa vor răspunde mai devreme decât cele aflate la marginea ariei celulei.*

GSM poate permite celulelor să se extindă până la 35 *Km de Staţia de Bază. Timpul luat de semnalul radio pentru a străbate* 70 *Km (până la perimetru şi înapoi) este de* $0,23$ *ms; astfel, pentru fiecare timeslot va fi oferită o perioadă de gardă de această lungime.*

Cum acest interval este clar ineficient, GSM va informa de fiecare dată mobilul cu cât trebuie să-şi avanseze transmisia astfel încât să se sincronizeze corect cu Staţia de Bază.

Se ajunge în final la o perioadă de $0,03$ *ms adică* $8,25$ *biţi.*

5.1.5 Subsistemul reţea

Are ca principale responsabilităţi:

- Stabilirea apelului;
- Controlul apelului;
- Rutarea apelurilor între diferite centre de comutaţie fixe/mobile şi alte reţele.

Componentele sale sunt:

Centrul de comutare a serviciilor

Mobile Switching Center (MSC) este elementul central al reţelei[3], deoarece el controlează toate conexiunile între utilizatori.

Activităţile de administraţie pentru stabilirea şi menţinerea unei conexiuni cuprind următoarele sarcini:

- Înregistrarea utilizatorilor: când staţia mobilă este pornită, ea se înregistrează în reţea, devenind accesibilă tuturor utilizatorilor.
- Stabilirea apelului şi rutarea acestuia între doi utilizatori.

[3]Este denumit şi *Reţea de telefonie pe teritoriu public* - $PLMN$ (Public Land Mobile Network).

- Transmiterea mesajelor scurte SMS.

Deoarece utilizatorii dispun de o mobilitate foarte mare în retea, MSC este responsabil şi de gestionarea următoarelor acţiuni:

- Autentificarea utilizatorilor în momentul stabilirii conexiunii.
- Dacă nu există nici o conexiune activă între reţea şi staţia mobilă, MSC trebuie să raporteze reţelei schimbarea locaţiei mobilului, făcându-l astfel disponibil pentru apeluri primite şi mesaje scurte.
- Dacă utilizatorul îşi schimbă locaţia în timpul stabilirii unei conexiuni cu reţeaua, MSC asigură continuitatea conexiunii şi rutarea ei către următoarea celulă – procedură numită *handover*.
- Asigură partea de facturare a apelurilor. MSC reţine informaţii cum ar fi numărul apelatului şi numărul apelantului, ID-ul celulei din care a fost iniţiat apelul, durata iniţierii apelului, durata apelului etc, pe care apoi le transmite serverului de facturare.

Registru de localizare a vizitatorilor

Visitor Location Register (VLR) reţine informaţii despre fiecare utilizator care este servit la momentul curent de MSC; aceste informaţii sunt copii ale informaţiilor originale stocate în HLR. Scopul principal al folosirii VLR este reducerea numărului de mesaje între MSC şi HLR.

Când un utilizator ajunge în zona unui MSC, datele sunt copiate în VLR-ul aferent, fiind disponibile local pentru orice conexiune. Când acesta părăseşte zona, informaţiile respective sunt copiate din HLR în VLR-ul noului MSC, fiind şterse din VLR-ul anterior.

Deşi este posibilă implementarea lui VLR ca o componentă hardware independentă, în majoritatea cazurilor el este o componentă software din MSC.

Registrul de localizare (HLR)

Home Location Register (HLR) reprezintă baza de date cu utilizatori a reţelei GSM. El conţine informaţii despre serviciile disponibile pentru fiecare utilizator în parte.

Exemplul 5.3. *$IMSI$-ul – care identifică un utilizator – este stocat atât pe SIM cât şi în HLR; el reprezintă cheia către orice informaţie despre utilizator. Cum $IMSI$ este unic la nivel internaţional, utilizatorul poate folosi telefonul înafara ţării sale de origine (desigur, dacă există un acord cu operatorul din ţara sa). Atunci când telefonul este deschis, se recuperează $IMSI$ de pe SIM şi este transmis către MSC care – la rândul său – poate cere informaţii din HLR referitoare la utilizatorul respectiv.*

Numărul de telefon al utilizatorului – numit şi numărul $ISDN$ al utilizatorului în standardul GSM – are o lungime de maxim 15 cifre; el conţine codul ţării, codul naţional destinaţie (corespunzător operatorului respectiv) iar restul cifrelor reprezintă numărul utilizatorului.

Centrul de autentificare (AC)

AC deţine o cheie secretă K_i pentru fiecare utilizator, care este o copie a cheii K_i de pe cartela sa SIM. Pentru anumite operaţii din reţea (cum ar fi stabilirea unui apel), utilizatorul este identificat prin K_i.

În AC se află de asemenea cheile de autentificare şi de criptare pentru toţi utilizatorii din HLR şi din VLR-urile aflate în reţeaua furnizorului. În particular, de aici sunt trimise triplete de tipul $(RAND, SRES, K_c)$ necesare pentru procesul de autentificare.

Registrul de identificare a echipamentului (EIR)

EIR este o bază de date care stochează o listă cu toate echipamentele mobile valide în reţea, fiecare telefon fiind identificat prin $IMEI$ (*International Mobile Equipment Identity*). Un $IMEI$ este marcat ca fiind invalid dacă a fost declarat furat sau dacă tipul său este unul neaprobat.

EIR menţine actualizate trei liste:

1. **Lista albă**: conţine echipamentele mobile conforme cu cerinţele impuse de operatorul de reţea.
2. **Lista neagră**: conţine echipamentele mobile care au fost raportate ca furate sau cele care afectează bunul mers al reţelei; acestea nu au voie sa acceseze reţeaua.
3. **Lista gri**: conţine echipamentele mobile care nu sunt conforme standardelor; ele atent supravegheate, dar au voie să acceseze reţeaua.

5.2 Rutarea apelurilor în GSM

Utilizatorii telefoanelor mobile sunt – prin definiţie – mereu în mişcare; de aceea mecanismele pe care le foloseşte reţeaua GSM pentru a-i localiza sunt foarte importante. Să detaliem modul de rutare al apelurilor către staţia mobilă.

5.2.1 Înregistrarea locaţiei

Actualizarea locaţiei este cel mai important instrument care permite găsirea telefoanelor mobile în reteaua GSM. Locaţia unei staţii mobile este identificată în mod unic de *codul ţării respective (mobile country code)*(MCC), *codul reţelei mobile* (MNC) şi *identitatea*

locaţiei (LAI). MCC-ul este o valoare de trei cifre ce identifică ţara în care este situată reţeaua. MNC-ul este o valoare de două cifre care identifică reţele (concurente) din aceeaşi ţară. LAI-ul identifică regiunea fizică în care este localizată staţia mobilă (o regiune constă dintr-una sau mai multe celule fizice). Aceste trei valori formează împreună IMSI-ul care identifică în mod unic un utilizator.

O reţea poate avea mai multe centre de comutare a serviciilor mobile (MSC) dar numai cele conectate la un ISDN sau PSTN sunt folosite ca gateway pentru centrele de comutare ($GMSC$) dintr-o reţea împreună cu un HLR. Apelurile sosite pentru telefoanele mobile pot fi rutate corect numai după ce este verificată (în HLR) locaţia mobilului ţintă precum şi serviciile autorizate ale acestuia. $GMSC$ şi MSC au fiecare propriul VLR. După ce s-a efectuat actualizarea locaţiei, mesajul de la staţia mobilă ce conţine informaţia despre locaţia sa este trimis prin MSC la VLR-ul ataşat. Acesta verifică dacă există deja o înregistrare cu acea staţie mobilă, şi îi actualizează locaţia – dacă aceasta este diferită de cea existentă. Dacă staţia mobilă este necunoscută pentru VLR, atunci acesta va avertiza HLR despre locaţia staţiei mobile şi despre informaţiile de rutare, emiţând anterior o cerere despre datele de bază ale utilizatorului, pe care VLR le va folosi pentru a finaliza iniţierea apelului.

Există trei tipuri diferite de proceduri disponibile pentru actualizarea locaţiei:

1. **Înregistrarea**: are loc atunci când este deschisă o staţie mobilă. După câteva iniţializări interne, ea va căuta reţele disponibile; când va găsi o astfel de reţea, staţia mobilă va citi informaţii despre locaţie şi va trimite $IMSI$-ul său reţelei respective.

2. **Actualizarea periodică a locaţiei**: este efectuată după o perioadă de timp predefinită de reţea şi este trimisă constant tuturor staţiilor mobile active care monitorizează canalul de control.
   Dacă staţia mobilă nu se înregistrează în acest timp (de pildă, datorită faptului că utilizatorul a ajuns într-un spaţiu cu semnal foarte slab), atunci reţeaua va presupune că staţia mobilă nu mai este disponibilă şi o va marca drept "inaccesibilă" în $HLSR$ şi VLR. Ca urmare, apelurile primite vor fi blocate în $GMSC$ în loc să fie rutate spre regiunea în care se află localizată. Apelul poate fi direcţionat către alt număr – dacă utilizatorul a setat în prealabil această opţiune.

3. Când staţia mobilă detectează o *schimbare a regiunii* în care se află, va anunţa reţeaua despre această schimbare.

5.2.2 Stabilirea apelului în GSM

Apelurile în GSM sunt clasificate astfel:

1. (În cazul în care apelul este iniţiat de reţea) Reţeaua notifică telefonul cu un PAGING REQUEST prin $IMSI$-ul corespunzător pe canalul de paging.
   Dacă apelul este iniţiat de telefonul mobil, se trece direct la pasul următor.

2. Procedura de alocare directă:

   (a) Telefonul – trimite pe canalul $RACh$ – un mesaj RANDOM REQUEST de 8 biţi. El este format dintr-o valoare aleatoare pe 5 biţi, iar restul de 3 biţi conţine "establishment cause" (tipul de apel care urmează a fi stabilit). Acest mesaj solicită staţiei de bază alocarea resurselor radio pentru realizarea conexiunii.

   (b) Reţeaua trimite un mesaj IMMEDIATE REQUEST pe canalul $PAGCh$.
   Acesta conţine valoarea aleatoare primită de la telefon, detalii despre canalul alocat telefonului mobil, împreună cu alte informaţii tehnice.
   Mobilul se va racorda imediat la canalul de trafic alocat.

3. Cererea de servicii:

   (a) Mobilul trimite un mesaj de cerere a serviciilor, incluzând $TMSI$-ul său şi versiunile de $A5$ suportate

   (b) Reţeaua certifică acest mesaj şi repetă $TMSI$-ul.

4. Autentificarea:

   (a) Reţeaua trimite o cerere de autentificare care include valoarea $RAND$ şi un număr care va stoca cheia K_c rezultată;

   (b) Mobilul răspunde cu valoarea $SRES$ calculată;

   (c) Reţeaua cere mobilului – prin comanda $CIPHMOD$ – să înceapă criptarea; ea poate specifica algoritmul de criptare folosit şi – eventual – cheia de criptare.
   Reţeua începe să decripteze informaţiile primite.
   Mesajul poate fi folosit şi pentru a cere mobilului să trimită $IMEI$-ul şi versiunea de software.

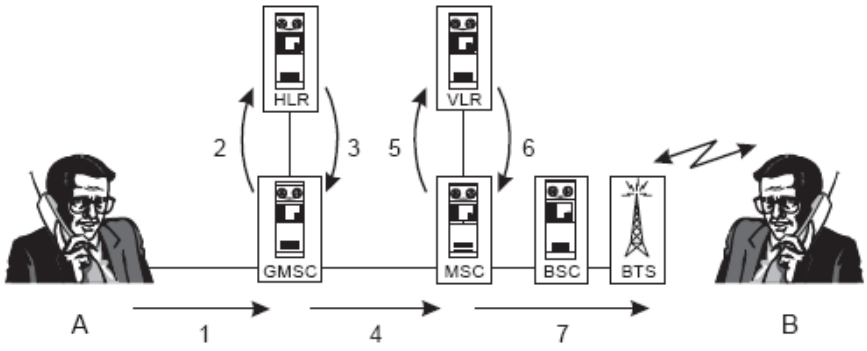
   (d) Mobilul începe criptarea şi decriptarea şi răspunde cu mesajul $CIPHMODCOM$ criptat. La cerere, el trimite şi $IMEI$-ul.

5. Reţeaua şi mobilul "comunică" pe canal. Reţeaua poate să schimbe canalul în funcţie de tipul apelului efectuat.

5.2.3 Rutarea apelului între două staţii mobile

Când *Alice* iniţiază un apel mobil către *Bob*, apelul va fi rutat către $GMSC$-ul reţelei. Dacă reţeaua are mai multe astfel de "porţi de intrare", apelul este rutat către $GMSC$-ul de care aparţine *Bob*, adică spre acel $GMSC$ al cărui HLR ataşat conţine o înregistrare

cu datele lui *Bob*. $GMSC$-ul va afla locaţia utilizatorului *Bob* prin lansarea unei cereri de localizare, la care răspunde un HLR. Acesta va transmite către $GMSC$ zona sau MSC-ul corespunzător lui *Bob*. Cu această informaţie, $GMSC$-ul poate ruta apelul către acel MSC care poate finaliza iniţierea apelului. MSC-ul – fiind acoperit de mai multe BSC şi BTS – va trebui să verifice locaţia exactă a lui *Bob* din VLR şi va primi un răspuns conţinând datele exacte.
Mai departe, MSC este capabil să trimită apelul spre *Bob*, folosind BTS-ul corect.



Rutarea apelului între două staţii mobile

Dacă unul din utilizatori doreşte încheierea apelului, staţia mobilă va trimite spre reţea un mesaj de deconectare. După eliberarea canalului de trafic şi după trimiterea unui mesaj de terminare către celelalte părţi implicate, toate resursele din reţea sunt eliberate şi apelul este încheiat.

5.3 Securitatea GSM

Conform cerinţelor, un sistem GSM trebuie să fie cel puţin la fel de sigur ca $PSTN$ (*Public Switched Telephone Network*) – reţeaua de telefonie publică bazată pe circuite de comutaţie.

5.3.1 Cerinţe de securitate

Securitatea GSM trebuie să ia în considerare atât pe operatorul de reţea cât şi pe client. Transmiterea informaţiilor prin unde radio (wireless) implică existenţa unor riscuri potenţiale care provin din interceptarea transmisiilor.

- Operatorii sistemului vor să se asigure că ei sunt cei care emit facturile către persoanele potrivite, să evite fraudele şi să aibă siguranţa serviciilor necompromise.

- Clienţii doresc să li se asigure confidenţialitatea convorbirilor.

Pentru asigurarea acestor cerinţe, sistemul este proiectat astfel încât să ofere reţelei wireless aceeaşi siguranţă pe care o are reţeaua fixă, siguranţă care înseamnă autentificare şi confidenţialitate împotriva ascultărilor nedorite în reţea.

Specificaţiile GSM identifică următoarele trei aspecte ale securităţii pe care le acoperă sistemul:

- **Autentificarea utilizatorilor**: telefonul mobil trebuie să-şi dovedească dreptul de acces la un anumit cont din reţeaua operatorului dorit.
- **Anonimitatea utilizatorilor**: identificarea unui utilizator în reţea devine o problemă dificilă pentru cineva din afara sistemului.
- **Confidenţialitatea**: informaţiile comunicate wireless trebuie să fie protejate, iar accesarea lor să fie posibilă doar utilizatorilor destinaţi.

5.3.2 Anonimitatea

În momentul in care utilizatorul îşi deschide echipamentul mobil pentru a-l folosi în reţeaua operatorului contractat, el trebuie să se identifice în reţea.

Identificarea se face cu ajutorul numărului unic $IMSI$ aflat pe SIM. Cum $IMSI$ este elementul cheie în rutarea apelurilor, reţeaua trebuie să ştie la orice moment unde se află telefonul identificat. Această funcţionalitate a reţelei se numeşte "gestionarea locaţiei" (*location management*.

Cum $IMSI$-ul identifică în mod unic un utilizator al reţelei – iar această informaţie este publică – este suficient ca cineva să-l intercepteze în aer, pentru a identifica utilizatorul corespunzător şi a-i afla locaţia.
Acest lucru este contracarat prin proprietatea de anonimitate, care protejează identitatea utilizatorului faţa de cineva care dispune de $IMSI$-ul său. Identitatea protejată în cadrul reţelei implică şi ascunderea locaţiei, precum şi a apelurilor efectuate.

GSM asigură anonimitatea folosind un identificator temporar pentru client – $TMSI$ (*Temporary Mobile Subscriber Identity*) care, spre deosebire de $IMSI$ (unic la nivel global), este valid numai local şi este asignat utilizatorului atunci când cartela SIM s-a autentificat în reţea. Pentru comunicarea cu acel SIM, reţeaua va folosi în continuare numai $TMSI$-ul alocat.

Atunci când este înregistrată o modificare a locaţiei, reţeaua asignează un nou $TMSI$ pentru staţia mobilă, care este stocat în reţea împreună cu $IMSI$. Staţia mobilă şi reţeaua vor folosi de acum pentru comunicare numai $TMSI$-ul. La închidere, staţia mobilă memorează $TMSI$-ul actual pe cartela SIM, pentru a fi accesibil atunci când este repornită.

Pe de altă parte, se poate vorbi – într-un schimb de informaţii între utilizatori – despre anonimitatea transmiţătorului şi a receptorului.
Anonimitatea transmiţătorului se referă la ascunderea identităţii expeditorului unui mesaj,

iar *anonimitatea receptorului* oferă posibilitea de a-l contacta pe destinatar chiar dacă acesta rămâne anonim.

Exemplul 5.4. *În [?] este prezentat un mecanism care să permită utilizatorului unei staţii GSM să efectueze/primească apeluri astfel încât furnizorul său de telefonie mobilă să nu poata determina identitatea celeilalte părţi implicate în convorbire.*
Pentru aceasta, autorii folosesc un Trusted Third Party Privacy Proxy iar soluţia propusă implică folosirea de alias-uri (identificatori unici asociaţi abonaţilor) astfel încât reţeaua să nu poată face legătura între acestea şi transmiţătorul/receptorul real decât pentru scopuri de facturare.

În sistemul GSM există o serie de probleme legate de stabilirea anonimităţii:

- Un abonat trebuie să poată fi accesat oricând: reţeaua trebuie să îi ştie mereu locaţia – pentru a putea ruta apelurile sosite către el. Chiar dacă receptorul nu cunoaşte identitatea apelantului, reţeaua trebuie să cunoască informaţiile ambelor părţi, pentru a putea stabili şi deschide canale de comunicare.
- Numărul de telefon al unei persoane – odată ştiut – poate fi considerat public: poate fi divulgat şi altor persoane, chiar fără consimţământul proprietarului.
- Apelurile GSM trebuie să se desfăşoare în timp real şi sunt facturate în funcţie de timp, de regulile stabilite în contract etc. Transmiţătorul trebuie să fie legat de o anume entitate ce poate fi identificată, astfel ca după un anumit interval de timp (de obicei lunar), operaţiunile efectuate de el în reţea să poată fi facturate.

5.3.3 Atacuri asupra anonimităţii utilizatorilor GSM

Atacurile asupra anonimităţii utilizatorilor GSM pot fi active sau pasive.

Atacatorul poate iniţia un atac pasiv când doar "ascultă" traficul, fără a efectua acţiuni; în schimb în monitorizarea activă, atacatorul se implică prin fabricarea şi inserarea unor mesaje, distrugerea altora etc.

Monitorizare pasivă

Ştim că la fiecare pornire a unei staţii mobile se ataşează un $IMSI$, pentru a informa reţeaua asupra faptului că $IMSI$-ul respectiv este activ din acel moment. Protocolul foloseşte procedura de actualizare a locaţiei din care staţia mobilă transmite un mesaj, împreună cu $IMSI$-ul său. Dacă acesta nu este înregistrat în reţea, atunci nu îi este asociată nici o cheie K_i, iar criptarea nu poate fi aplicată. Deci $IMSI$ trebuie transmis în clar, iar un atacator care ascultă traficul, îl poate extrage.

Aceasta nu este singura situaţie când poate fi capturat $IMSI$-ul. Există cazuri (de exemplu o disfuncţionalitate a bazei de date) când HLR-ul nu poate folosi $TMSI$ pentru a obţine anumiţi parametri ai utilizatorului şi prin urmare va cere în clar $IMSI$-ul.

Monitorizare activă

În acest caz *Oscar* poate comunica cu staţia mobilă, ceea ce presupune existenţa unui echipament mai sofisticat decât în cazul unei monitorizări pasive. El va folosi o procedură de identificare, în care reţeaua va efectua un IDENTITY REQUEST, cerând staţiei mobile, $IMSI$, $IMEI$ sau $TMSI$.

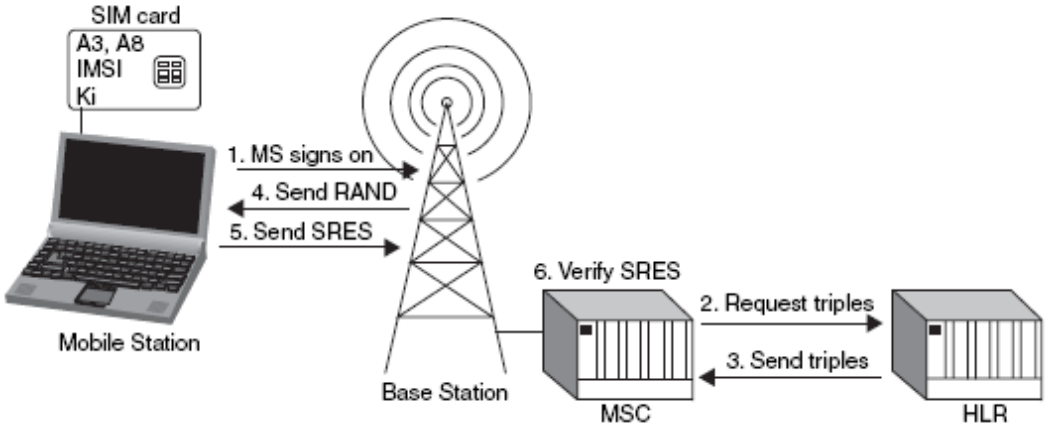
Deoarece GSM nu verifică autenticitatea unui mesaj, *Oscar* poate pretinde că este staţie de bază, obţinând astfel – prin intermediul unui astfel de mesaj – informaţia dorită. După ce deţine $IMSI$-ul, atacatorul îşi poate identifica victima.

Următorul pas este găsirea $TMSI$-ului pe care reţeaua îl alocă staţiei mobile, astfel ca *Oscar* să-l poată asocia $IMSI$-ului; aceasta îi permite ulterior să urmărească mişcările staţiei mobile.
$TMSI$-ul este criptat înainte de a fi transmis, deci *Oscar* va trebui întâi să-l decripteze. El va genera o situaţie în care cele două entităţi legitime care comunică să creadă că dispun de capabilităţi diferite de criptare. *Oscar* poate face acest lucru deoarece poate insera, distruge sau fabrica mesaje dupa bunul plac – lucru posibil datorită faptului că GSM nu asigură integritatea mesajelor şi nici autentificarea reţea-utilizator.

5.3.4 Autentificarea

Prin autentificare se evită situaţiile când persoane neautorizate pătrund în reţea pretinzând că sunt utilizatori acceptaţi ai reţelei. Înainte de a avea acces la serviciile unei reţele GSM, un utilizator trebuie să işi dovedească autenticitatea.

Vom descrie procedura de autentificare, reprezentată în figura următoare:



Procesul de autentificare în reţeaua GSM

1. La pornire, echipamentul mobil începe cu căutarea unei reţele wireless la care să se conecteze, scanând un anumit interval de frecvenţe. În momentul când a găsit

frecvenţa dorită, SIM-ul încearcă să se autentifice, trimiţând un mesaj la BTS prin care cere accesul la reţea.

2. BTS-ul comunică cu centrul de comutare pentru a decide permiterea accesului, care se face în urma unui protocol provocare - răspuns, bazat pe cheia secretă K_i partajată între telefon şi reţea.

3. Centrul de comutare obţine de la registrul de locaţii un triplet de forma

$$(RAND, SRES, K_c)$$

unde:

   (a) $RAND$ este un număr aleator pe 128 biţi,

   (b) $SRES$ este un răspuns pe 32 biţi de la $RAND$, semnat şi generat folosind cheia partajată K_i,

   (c) K_c este cheia de sesiune pentru criptare, generată tot cu ajutorul lui K_i.

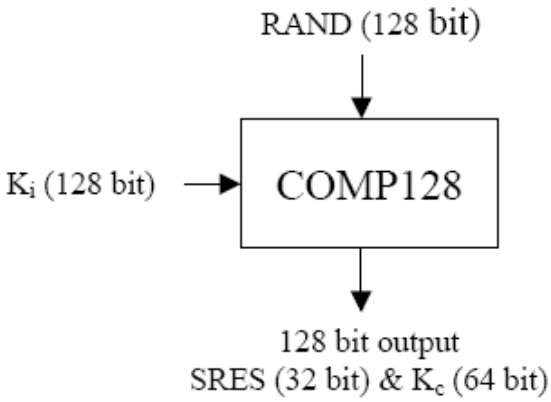
4. După obţinerea acestui triplet, $RAND$ este trimis (via BSC şi BTS) ca o provocare către staţia mobilă.

5. Ca răspuns la această provocare, cartela SIM a staţiei mobile va genera un $SRES$ – folosind algoritmul $A3$ şi acel K_i păstrat stocat ($SRES = A3(K_i, RAND)$).

6. Apoi, cartela SIM trimite $SRES$-ul calculat către MSC, care îl compară cu $SRES$-ul conţinut în tripletul primit de la HLR. Dacă cele două coincid, MSC deduce că echipamentul mobil deţine un SIM cu un K_i valid şi îi permite accesul în reţea; în caz contrar MSC va refuza accesul.

Există câteva aspecte importante de remarcat în procesul de autentificare:

- $A3$ şi $A8$ nu sunt algoritmi în sine, ci – mai degrabă – etichete pentru algoritmi. Funcţional, ei sunt funcţii one-way, ceea ce asigură imposibilitatea descoperirii cheii K_i. Furnizorii de telefonie mobilă sunt liberi să folosească orice algoritm doresc pentru a genera $SRES$ din K_i şi RAND. Specificaţia GSM foloseşte numele $A3$ ($A8$) numai pentru a se referi la un astfel de algoritm.
  Cele mai multe implementări GSM combină $A3$ cu $A8$ şi folosesc un singur algoritm pentru a servi ambele scopuri. Algoritmul $COMP128$ este de referinţă, fiind specificat de GSM şi folosit de majoritatea operatorilor. Acesta are la intrare cheia K_i şi $RAND$ şi generează $SRES$ pe 32 biti şi un alt număr pe 54 de biţi, căruia i se adaugă la sfârşit încă 10 biţi egali cu 0, pentru a forma cheia de sesiune K_c pe 64 biţi – folosită pentru asigurarea confidenţialiţaţii.



O implementare frecventă a lui A3/A8 (COMP128)

- Cheia secretă K_i, identificatorul $IMSI$, funcţiile $A3$ şi $A8$ sunt stocate şi implementate în cartela SIM a telefonului; K_i nu părăseşte niciodată cartela SIM.

- Este remarcabilă relaţia de încredere între componentele reţelei între care se transmit informaţiile confidenţiale (centrul de comutare - registrul de locaţii, BSC - MSC). Această încredere evidenţiază o caracteristică a reţelei GSM: încercarea de a securiza numai partea wireless. Motivul: GSM a evoluat din reţeaua de telefonie publică $PSTN$ şi îşi propune să fie cel puţin la fel de sigură. Partea centrală a $PSTN$ este securizată prin restricţionarea accesului fizic la reţea. Ideea a fost aplicată şi în GSM, unde componenta centrală din arhitectura sistemului se referă la reţeaua ce se găseşte dincolo de staţia de baza (BSC), considerată sigură – deoarece este controlată numai de furnizorii de telefonie mobilă, iar accesul la ea se află sub control foarte atent.
  Totuşi, există o relaţie între componente înafara reţelei centrale: BTS - BSC. Ea rămâne o cale de atacuri posibile; de altfel, GSM nu specifică nici o metodă de securizare a acesteia.

- În GSM, entitatea autentificată este cartela SIM iar nu utilizatorul în sine. Ce se întâmplă dacă echipamentul ce se doreşte a fi utilizat în reţea este unul furat ? Atunci când un client a pierdut echipamentul sau cartela SIM, el are responsabilitatea de a comunica aceasta către furnizorul său de telefonie mobilă. Cum reţeaua menţine o bază de date cu echipamentele valide – prin extrapolare – furnizorul de servicii ar putea avea şi o bază cu SIM-urile valide.

Un detaliu mai ascuns al procesului de autentificare în GSM este folosirea a cinci triplete de securitate pe care MSC le obţine de la HLR. Deşi unul singur este necesar pentru autentificarea unui utilizator în reţea, MSC cere mai multe triplete pentru a evita repetarea acestei acţiuni de fiecare dată când utilizatorul doreşte autentificarea, îmbunătăţind astfel performanţa sistemului prin deţinerea în "stoc" a încă patru triplete de rezervă pentru o folosire ulterioară.

5.3.5 Atacuri asupra algoritmului de autentificare prin acces fizic la cartela SIM

Ştim că fiecare operator de telefonie mobilă beneficiază de libertatea de a-şi alege un design precis pentru algoritmii $A3$ şi $A8$. Cel mai folosit este $COMP128$; deşi nu a fost niciodată publicat oficial, el s-a aflat prin inginerie inversă de către M. Briceno, I. Goldberg şi D. Wagner ([**?**]). Aceştia au efectuat şi o criptanaliză asupra lui $COMP128$, găsind cheia K_i partajată între telefonul mobil şi reţea. Având K_i, $A3$ şi $A8$, clonarea cartelei GSM este simplu de realizat.
Securitatea întregului sistem GSM se bazează pe cheia secretă K_i. Odată ce intrusul *Oscar* este capabil să extragă cheia, el poate nu numai să asculte apelurile abonatului urmărit, dar se poate substitui acestuia în cadrul reţelei, efectuând apeluri în contul lui.

Reţeaua GSM are un sistem de securitate care poate determina şi închide contul a două telefoane cu acelaşi ID folosite simultan în locaţii diferite. Dacă *oscar* este interesat numai să asculte convorbirile abonatului, el poate sta pasiv, fiind invizibil pentru reţeaua GSM.

Conform celor menţionate mai sus, $COMP128$ este compromis. *Oscar* poate observa intrarea şi ieşirea în cadrul algoritmului $A8$ şi – pe baza lor – poate calcula K_i (intrarea este o provocare aleatoare trimisă din reţeaua abonatului iar ieşirea este un răspuns de la $SRES$-ul telefonului mobil). Un element care facilitează calcularea K_i este trimiterea provocării şi a $SRES$ în clar pe calea aerului.

Fisura în procesul de autentificare a fost descoperită în 1998 ([**?**]) de către un grup de cercetatori $ISAAC$ (Internet Security, Applications, Authentication and Cryptography) împreună cu *Smartacrd Developer Association*; s-a constatat că un atacator cu acces fizic la telefonul ţintă poate face o clonă (duplicat identic cu originalul) şi poate efectua astfel apeluri frauduloase în contul abonatului.
O condiţie absolut necesară este accesul fizic la telefonul urmărit, ceea ce indică o fisură parţială şi nu una totală.

Atacul efectuat asupra $COMP128$ este bazat pe provocări alese. Atacatorii au ales un număr de provocări speciale şi au interogat SIM-ul pentru fiecare din ele. Acesta a aplicat algoritmul pentru fiecare provocare în parte, şi a returnat un răspuns; analiza răspunsurilor primite – exploatând slaba difuzie (anumite părţi ale funcţiei de dispersie depind numai de anumite părţi ale intrării în algoritm) – a putut determina valoarea secretă a cheii.

Implementarea acestui atac foloseşte un cititor de smartcard-uri şi un computer. Atacul presupune interogarea cardului de 150.000 de ori; cu un cititor de carduri capabil să efectueze aproximativ 6 invocări pe secundă, atacul durează 8 ore. Câteva calcule suplimentare sunt necesare pentru analiza răspunsurilor.
Aceasta este cea mai comună metodă de atac, şi o măsură imediată împotriva ei ar fi folosirea pentru autentificare a unei funcţii de dispersie criptografică mai puternice.

Versiunea $COMP128$ a devenit $COMP128-1$ şi două versiuni noi – $COMP128-2$ şi $COMP128-3$ – au fost propuse pentru corectarea acestor probleme de securitate. Aceste ultime versiuni sunt algoritmi care nu au fost supuşi criptanalizei publice. $COMP128-3$ rezolvă problema celor 10 biţi nuli din cheia de sesiune K_c. Operatorii de telefonie mobilă migrează spre aceşti algoritmi noi, dar – deoarece $A3$ şi $A8$ sunt stocaţi pe SIM – se impune schimbarea cartelelor abonaţilor. Interesant este faptul că, după ce în 1998 Briceno, Goldberg şi Wagner au publicat atacul asupro $COMP128$, majoritatea furnizorilor de telefonie mobilă nu au adoptat noua variantă $COMP128-2$, ci au păstrat-o pe prima, aplicând o restricţie asupra cartelei SIM: au fixat un număr maxim de invocări asupra cartelei (sub 150.000), după care aceasta se bloca.

Datorită faptului că algoritmul se află pe cartela SIM – care este un smartcard, orice descoperire asupra vulnerabilităţilor acestuia are repercusiuni asupra securităţii informaţiilor stocate pe SIM: $IMSI$ şi K_i.

5.3.6 Atac prin partiţionare

O metodă prin care cheia K_i poate fi extrasă fără a avea acces fizic la SIM este atacul printr-o metodă foarte eficientă, numită atac "*side channel*".

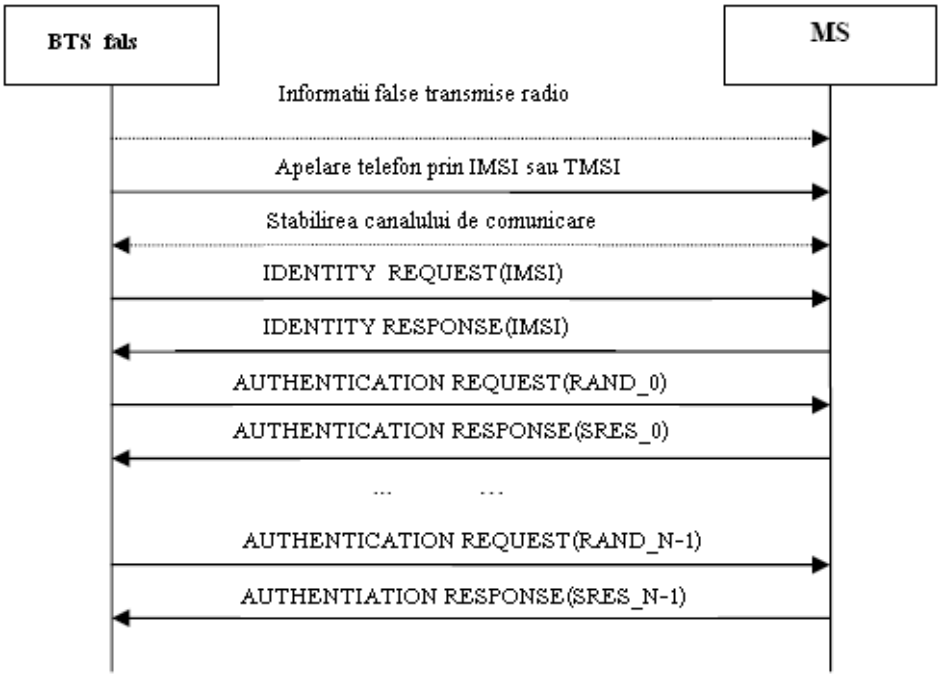
Atacurile side channel sunt atacuri criptanalitice care găsesc relaţia între informaţia de intrare şi cea de ieşire transmisă în timpul calculelor din canalele anexe; de exemplu sincronizarea operaţiilor, consumul de energie, emanaţii electromagnetice etc.
S-a constatat că prin analiza acestor caracteristici se pot obţine informaţii confidenţiale. Foarte multe atacuri bazate pe canalele anexe necesită cunoştinţe tehnice relativ la operaţiile interne ale sistemului care are integrată partea de criptografie.

Un algoritm puternic împotriva acestui tip de atac trebuie să aibă semnalul pe canalele anexe independent statistic de intrare, ieşire şi alte informaţii sensibile. O implementare adecvată poate asigura rezistenţă în faţa unui astfel de atac, dar pot rămâne unele relaţii statistice dependente care pot fi atacate uşor. În general, eliminarea unor astfel de atacuri constă în limitarea producerii de informaţii de tipul emanaţiilor electromagnetice sau sunet şi limitarea accesului la relaţiile între acestea.

"*Partitioning attacks*" este o clasă descoperită de cercetătorii de la IBM şi folosită pentru a ataca implementări de algoritmi care altfel ar rezista la atacurile side channel. Cu ajutorul lor, toată cheia de 128 biţi din $COMP128$ poate fi extrasă de pe un SIM folosind mai puţin de 1000 invocări cu intrări aleatoare, sau 255 intrări alese, sau doar 8 intrări special alese. Prin urmare, un adversar care se afla un posesia cartelei SIM poate extrage cheia secretă K_i în maxim un minut.
Atacul prin partiţionare poate fi folosit mai ales contra algoritmilor bazaţi pe căutari în tabele foarte mari. $COMP128$ este un astfel de exemplu; el utilizează căutări în 5 tabele de câte 512, 256, 128, 64 şi repectiv 32 biţi.

5.3.7 Atacuri wireless asupra algoritmului de autentificare

Faptul că nu întotdeauna este posibil să avem acces fizic la SIM a creat posibilitatea unui atac wireless (pe calea undelor radio). Deşi această abordare pare mai atractivă, ea introduce alt tip de obstacole. În primul rând, *Oscar* trebuie să poată simula BTS-ul ca unul legitim. Aceasta înseamnă că el are nevoie de o staţie de bază falsă, capabilă să genereze un semnal suficient de puternic aşa încât să depăşească semnalul BTS-ului legitim. Numai în acest caz ar fi posibilă comunicarea între BTS şi staţia mobilă. O posibilă rezolvare a acestei probleme ar fi lansarea atacului atunci când semnalul staţiei legitime este foarte slab (la metrou, în lift, în zone izolate etc).



Atac wireless asupra $COMP128$

Pe lângă aceasta, *Oscar* mai trebuie să ştie $IMSI$-ul sau $TMSI$-ul cartelei SIM pe care vrea să o cloneze. Când aceste resurse sunt disponibile, el va încerca să capteze staţia mobilă. Aceasta va efectua imediat o cerere de actualizare a locaţiei. Dacă pretinsa reţea a provocat telefonul mobil prin $TMSI$, $IMSI$-ul poate fi aflat uşor prin comanda *IDENTITY REQUEST*, la care telefonul trebuie să răspundă imediat.
În demersul pentru actualizarea locaţiei, intrusul va iniţia un proces de autentificare. Imediat după ce obţine o pereche provocare - răspuns, el lansează o nouă procedură de autentificare. Staţia mobilă trebuie să răspunda la fiecare provocare pe care o trimite reţeaua GSM. Procedeul continuă până când *Oscar* va obţine numărul de perechi necesare pentru a efectua clonarea.

Atacul funcţionează pe orice telefon mobil din GSM, fără nici un acces anterior la acesta (şi fără a cunoaşte vreun $IMSI$). Prin monitorizarea traficului, poate fi ales un $TMSI$ la întâmplare. Fiind un atac wireless, el se poate desfăşura de la distanţă.

Dacă *Oscar* obţine cheia K_i (prin clonarea cartelei) şi interceptează valoarea $RAND$ prin wireless în timpul stabilirii apelului, el poate calcula cheia K_c (dacă este folosit algoritmul $COMP128$ sau un alt algoritm cunoscut) şi poate asculta convorbirea în timp real.

După apariţia primelor atacuri, GSM a reacţionat imediat şi a creat alte două noi versiuni pentru COMP128. Dacă până atunci furnizorii foloseau în continuare $COMP128-1$ limitând doar numărul de provocări, în urmă apariţiei atacului ce necesită doar 8 provocări, această restricţie nu mai este posibilă: o cartelă SIM trebuie să răspundă la mai mult de 8 provocări pe parcursul unei zile.

Ultima versiune a lui $COMP128$ – a patra – este complet nouă şi se bazează pe un algoritm ce foloseşte standardul AES. Ea este implementată în reţelele $UMTS$, ceea ce presupune emiterea de noi cartele pentru abonaţii reţelei, o reactualizare a soft-ului HLR şi securitate contra clonării cartelei SIM.

Şi noul algoritm este pasibil de atacuri; chiar şi "*partitioning attacks*". Căutarea în tabele mari este folosită frecvent în algoritmi precum DES, AES şi $COMP128$, care – aplicate pe dispozitive limitate cum sunt smartcard-urile – rămân sensibile la atacuri side channel. Cercetătorii IBM au propus o metodologie pentru a crea rezistenţă în faţa unor astfel de atacuri. Ideea constă în înlocuirea unei căutări într-o tabelă cu o serie de căutări în locaţii complet aleatoare folosind un tabel auxiliar, ceea ce nu permite nici o scurgere de informaţii. Metoda poate fi implementată cu succes în dispozitivele cu memorie limitată cum sunt telefoanele mobile, pentru că foloseşte foarte puţină memorie RAM.

5.4 Confidenţialitate în GSM

Confidenţialitatea în GSM înseamnă protejarea informaţiei schimbate în timpul conversaţiei şi a informaţiilor de control asociate cu stabilirea unui apel.
Contextul de securitate este cheia de sesiune K_c pe 64 biţi, rezultată în urmă aplicării algoritmului $A8$ sau $COMP128$ – dacă acesta este folosit pentru a combina ambii algoritmi $A3$ şi $A8$. Această cheie oferă confidenţialitate pe interfaţa wireless BTS - ME, deci securizează comunicarea între echipamentul mobil şi staţia de bază.

Reamintim că GSM foloseşte tehnica de diviziune în timp pentru a face posibilă folosirea aceluiaşi canal radio pentru 8 utilizatori simultan. Fiecare utilizator va transmite şi primi informaţii numai pe durata unuia din cele 8 sloturi de timp disponibile în fiecare frame. Fiecare frame durează 4.6 milisecunde şi este indentificat printr-un număr asociat. O conversaţie GSM foloseşte 2 frame-uri: unul mergând de la staţia de bază spre staţia mobilă şi celălalt parcurgând aceeaşi cale în sens invers. Fiecare din aceste frame-uri conţine 114 biţi de informaţie.

Deci la fiecare 4.6 milisecunde, staţia mobilă primeşte 114 biţi de informaţie de la staţia de bază şi trimite alţi 114 biţi de informaţie către aceasta. Aceşti 228 biţi au nevoie de protecţie împotriva interceptării.

Algoritmul folosit pentru criptarea pachetelor de informaţii transmise wireless este $A5$. Acesta este un algoritm de criptare specificat de standardul GSM (spre deosebire de $A3$ şi $A8$ care sunt doar nişte nume) pentru a încuraja roaming-ul[4] între reţelele de telefonie oferite de diverşi operatori. Alegerea lui $A3$ şi $A8$ este la libera alegere a operatorului, datorită faptului că procesul de autentificare se desfăşoară între SIM şi registrul de locaţii al utilizatorilor (HLR) aferent furnizorului de telefonie mobilă. Pe de altă parte, procesul de criptare trebuie să se desfăşoare între staţia de bază (BTS) şi staţia mobilă (ME) (aceasta fiind prima rută pe care circulă pachetul de date ce conţine informaţiile trimise de utilizator); deci informaţia trimisă trebuie să fie deja criptată.

Algoritmul $A5$ este un sistem de criptare fluid ([**?**]) care generează o cheie unică pentru fiecare pachet de date folosind la intrare cheia de sesiune K_c pe 64 biţi şi numărul de secvenţă asociat frame-ului respectiv. Deoarece numărul de secvenţă al unui frame este uşor de aflat, confidenţialitatea datelor transmise se bazează numai pe menţinerea secretului cheii de sesiune K_c.
Cheia de sesiune poate fi schimbată la intervale regulate sau după cum considera furnizorul de servicii de telefonie.

După ce a fost obţinută cheia de sesiune, criptarea va începe imediat ce reţeaua GSM trimite echipamentului mobil o cerere de criptare. Spre deosebire de $A3$ şi $A8$ care sunt implementaţi în SIM, algoritmul $A5$ se află implementat hardware – direct în echipamentul mobil.

În prezent există trei versiuni dezvoltate pentru $A5$:

1. Prima, numită $A5/1$, este folosită în ţările membre ale $CEPT$ (organizaţie care cuprinde 48 ţări din Europa) şi în Statele Unite. A fost lansată în 1987 şi oferă cel mai înalt nivel de criptare wireless. Deşi oficial foloseşte 64 biţi, practic cheia nu depăşeşte 54 biţi, ultimii 10 biţi fiind setaţi pe zero.

2. Al doilea algoritm, $A5/2$, dezvoltat în 1989, este mai slab decât $A5/1$ şi este folosit preponderent în Asia.
   Schiţa celor doi algoritmi a fost ţinută secret, dar a fost descoperită prin inginerie inversă de către Briceno ([**?**]) în 1999 şi este disponibilă pe Internet.

3. În 2002 apare versiunea $A5/3$. Ca securitate, el este mai puternic decât primele două versiuni, dar este folosit numai în reţelele $UMTS$ pentru generaţia a treia ($3G$). Spre deosebire de versiunile precedente, schema lui $A5/3$ a fost făcută publică, construcţia lui fiind bazată pe sistemtul de criptare bloc KASUMI.

[4]Prin *roaming* se înţelege extinderea conectivităţii serviciilor în locaţii diferite de locaţiile de bază în care acestea au fost înregistrate.

Înafara acestora mai există o versiune ieftină – şi deci mai puţin complexă – numită $A5/0$, care nu presupune nici o criptare.

5.4.1 Algoritmul $A5/1$

Algoritmul fluid $A5/1$ acceptă la intrare o cheie de sesiune K_c pe 64 biţi şi numărul de frame f pe 22 de biţi care este public (fiecare frame are asociat un număr de frame, frame-urile consecutive având asociate numere consecutive).

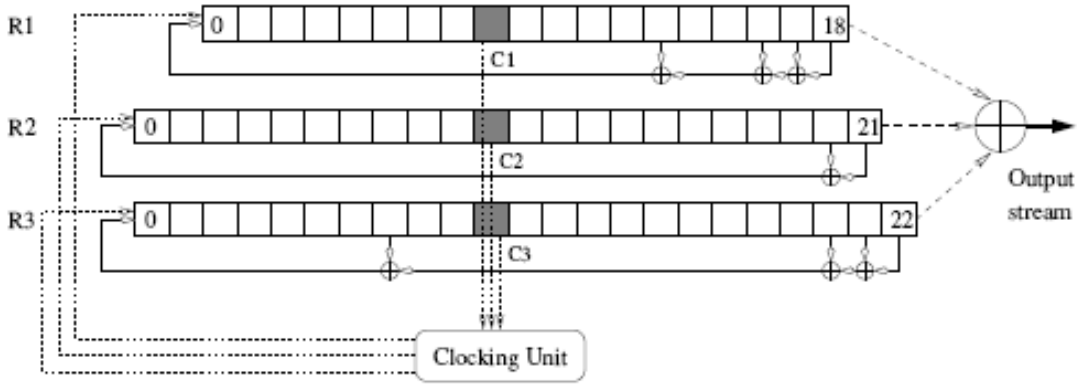
După cum am văzut, comunicarea în GSM se realizeaza prin *frame*-uri, unde un frame este transmis la fiecare 4.6 milisecunde. În fiecare frame, $A5/1$ este iniţializat cu cheia de sesiune şi cu numărul de frame. Cheia fluidă obţinută la ieşire ocupă 228 biţi şi este împărţită în două jumătaţi: prima parte (114 biţi) este folosită pentru criptarea datelor transmise de la reţea către telefonul mobil, iar a doua jumătate este folosită pentru criptarea datelor de la telefonul mobil către reţea.
Criptarea este realizată prin aplicarea operaţiei XOR asupra datelor transmise şi a jumătăţii corespunzătoare din cheia fluidă generată.

Trebuie menţionat că staţia de bază aplică acelaşi algoritm $A5/1$; deci fiecare din cele două părţi care comunică va folosi prima jumătate din cheia generată pentru a cripta datele pe care le transmite, iar a doua jumătate – pentru a decripta (decriptarea se face similar cu criptarea, prin aplicarea operaţiei XOR) datele primite.

$A5/1$ are o stare internă pe 64 biţi şi este construit din trei circuite $LFSR$ ([**?**]) de lungime 19, 22 şi respectiv 23 biţi fiecare, notate $R1$, $R2$ şi $R3$. La fiecare tact (*clocking*) fiecare registru calculează funcţia de întoarcere, după care este deplasat la dreapta cu o poziţie (cel mai din dreapta bit devine bit de ieşire) iar feedback-ul calculat este stocat în cea mai din stânga poziţie din registru.

$A5/1$ este iniţializat cu K_c şi f în trei paşi ($K_c[i]$ reprezintă al i-lea bit din K_c, iar $f[i]$ – al i-lea bit din f):

1. Setează $R1 = R2 = R3 = 0$.
2. For $i = 0$ to 63 do
   (a) Aplică un tact pentru toţi regiştrii
   (b) $R1[0] \leftarrow R1[0] \oplus K_c[i]; \quad R2[0] \leftarrow R2[0] \oplus K_c[i]; \quad R3[0] \leftarrow R3[0] \oplus K_c[i].$
3. For $i = 0$ to 21 do
   (a) Aplică un tact pentru toţi regiştrii
   (b) $R1[0] \leftarrow R1[0] \oplus f[i]; \quad R2[0] \leftarrow R2[0] \oplus f[i]; \quad R3[0] \leftarrow R3[0] \oplus f[i].$



Structura internă a algoritmului $A5/1$

Generarea cheii fluide se efectuează în 328 tacţi, la fiecare tact producându-se un bit de ieşire. La fiecare tact, ieşirea este un XOR între cei mai din dreapta biţi ai celor trei regiştri. Cheia fluidă de 228 biţi este generată astfel:

1. Execută iniţializarea cu cheia K_c şi numărul de frame f.
2. Execută $A5/1$ pentru 100 tacţi, fără a păstra ieşirea.
3. Execută $A5/1$ pentru 228 tacţi, biţii rezultaţi formând cheia fluidă.

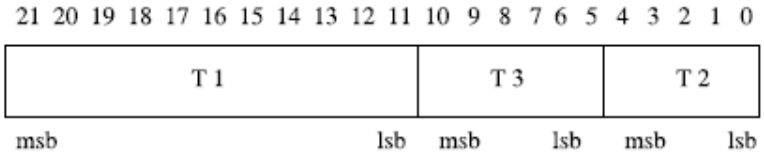
Mecanismul de calcul a bitului de ieşire la fiecare tact foloseşte regula majorităţii: fiecare registru are o poziţie – aproape de mijloc (locaţiile $R1[8]$, $R2[10]$ şi respectiv $R3[10]$) – marcată special. La fiecare tact se calculează valoarea majoritară din aceste trei poziţii, iar apoi, fiecare registru avansează o poziţie dacă şi numai dacă celula sa marcată (un D flip-flop) este identică cu valoarea majoritară calculată.

Deci, la fiecare tact se vor deplasa cel puţin doi regiştri; statistic, fiecare registru are probabilitatea 1/4 de a rămâne nemişcat şi probabilitatea 3/4 de a se deplasa ([**?**]).

5.4.2 Algoritmul A5/2

$A5/2$ este construit pornind de la arhitectura lui $A5/1$: se folosesc patru regiştri (de lungimi 19, 22, 23 şi 17 biţi), iar funcţiile de întoarcere ale primilor trei sunt identice ca pentru $A5/1$.

Algoritmul acceptă la intrare aceeaşi parametri ca şi $A5/1$: o cheie K_c pe 64 biţi şi o valoare publică f pe 22 biţi, derivată din numărul de frame (cunoscut public). Valoarea lui f este obţinută din numărul asociat frame-ului $TDMA$:

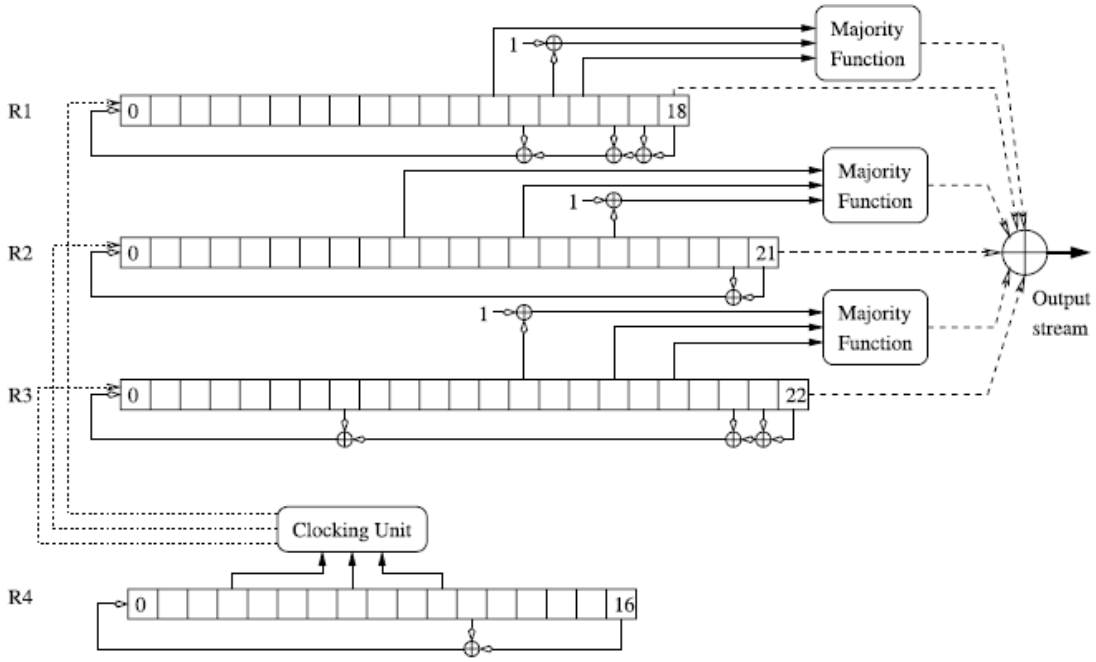


Obţinerea valorii f

unde $T1$ este câtul împărţirii numărului de frame la $51 \times 26 = 1326$, $T2$ este restul împărţirii numărului de frame la 26 şi $T3$ este restul împărţirii numărului de frame la 51.

$A5/2$ este iniţializat cu K_c şi f în patru paşi după cum urmează:

1. Setează $R1 = R2 = R3 = R4 = 0$.
2. For $i = 0$ to 63 do
   (a) Aplică un tact pentru cei patru regiştri.
   (b) $R1[0] \leftarrow R1[0] \oplus K_c[i]$; $R2[0] \leftarrow R2[0] \oplus K_c[i]$;
   $R3[0] \leftarrow R3[0] \oplus K_c[i]$; $R4[0] \leftarrow R4[0] \oplus K_c[i]$.
3. For $i = 0$ to 21 do
   (a) Aplică un tact pentru primii trei regiştri.
   (b) $R1[0] \leftarrow R1[0] \oplus f[i]$; $R2[0] \leftarrow R2[0] \oplus f[i]$;
   $R3[0] \leftarrow R3[0] \oplus f[i]$; $R4[0] \leftarrow R4[0] \oplus f[i]$.
4. Seteaza biţii $R1[15] \leftarrow 1$, $R2[16] \leftarrow 1$, $R3[18] \leftarrow 1$, $R4[10] \leftarrow 1$.



Şi în acest algoritm, la fiecare tact avansează cel puţin doi regiştri, în funcţie de valorile a trei biţi din $R4$. Apoi avansează $R4$ un tact.
La începutul fiecărui tact se calculează valoarea

$$x = maj\{R4[3], R4[7], R4[10]\}$$

Apoi, $R1$ avansează dacă $x = R4[10]$, $R2$ avansează dacă $x = R4[3]$ şi $R3$ avansează dacă $x = R4[7]$.

Procesul de generare a cheii poate fi sumarizat astfel:

1. Execută iniţializarea cu cheia K_c şi valoarea f.
2. Execută $A5/2$ pentru 99 tacţi, ignorând ieşirea.
3. Execută $A5/2$ pentru 228 tacţi, iar biţii rezultaţi formează cheia fluidă.

Procedura de criptare este identică cu cea a algoritmului $A5/1$.

5.4.3 Probleme legate de confidenţialitatea GSM

De-a lungul timpului s-au descoperit o serie de slăbiciuni ale arhitecturii de criptare a sistemului GSM, slăbiciuni care pot fi exploatate uşor.

- Partea de criptare a fost specificată ca o caracteristică opţională a reţelei. Prin urmare, operatorul de reţea poate alege dacă doreşte folosirea criptării sau nu. Unele telefoane mobile cum sunt cele din seria *Siemens S* afişează pe ecran un simbol de tipul *!* dacă opţiunea de criptare este dezactivată ([**?**]).

- Criptarea este utilizată numai pentru securizarea interfeţei dintre staţia mobilă şi BTS. Aceasta este singura legătură protejată criptografic, ceea ce expune restul reţelei la atacuri. Una dintre cele mai expuse interfeţe neprotejată prin criptare este BTS - BSC. Cum această legătură nu face parte din reţeaua "de bază" şi cum de obicei este o legătură fără fir (bazată pe microunde, pe satelit ş.a.m.d.), ea devine o ţintă atrăgătoare, interceptarea de apeluri fiind posibilă cu un echipament adecvat.

- Chiar şi algoritmul folosit pentru criptarea legăturii staţie mobilă - BTS nu este sigur, datorită puterii tot mai mare a hardware-lui. Folosind un simplu atac prin forţă brută, securitatea algoritmului poate fi compromisă în câteva ore.
  Principala problemă este lungimea mică a cheii de sesiune K_c: doar 54 biţi efectivi (plus alţi 10 biţi setaţi pe zero). Puterea hardware permite în prezent înregistrea pachetelor transmise între staţia mobilă şi BTS şi decriptarea lor la un moment ulterior.

- *GSM* foloseşte o procedură de autentificare într-un singur sens: numai reţeaua are dreptul să verifice identitatea abonatului (mai exact, a staţiei mobile). În schimb, staţia mobilă nu poate verifica autenticitatea reţelei. Aceasta permite punerea în practică a unui atac în care *BTS* este falsificat.

- Poate una dintre cele mai vizibile vulnerabilităţi ale sistemului *GSM* este faptul că nu asigură protecţia integrităţii pentru informaţiile transmise. Securitatea *GSM* vorbeşte despre autentificare şi confidenţialitate dar nu menţionează absolut nimic despre integritate. Absenţa unui mecanism care să o asigure înseamnă că la recepţie nu se poate verifica dacă mesajul primit a fost modificat sau nu. Se lasă astfel cale liberă pentru numeroase atacuri de tipul man-in-the-middle.

Algoritmii de criptare *GSM* nu sunt publicaţi în standardul *GSM*, ceea ce înseamnă că nu sunt disponibili pentru analiza comunităţii de securitate. Acest lucru a fost criticat pentru că încalcă principiul lui Kerckhoffs (securitatea sistemului rezidă numai în cheie). Se consideră indicat ca algoritmul de criptare să fie analizat public, aşa încât eventualele fisuri existente să fie descoperite şi publicate.

5.5 Atacuri active asupra reţelelor *GSM*

Să trecem în revistă câteva atacuri bazate pe faliile de securitate existente în protocolul de stabilire a apelului (prezentat în prima parte a capitolului); sunt atacuri active, deci este necesar ca *Oscar* să transmită (cu asumarea riscului de a fi detectat).

Faliile din protocol care pot fi exploatate sunt următoarele:

1. Autentificarea poate fi executată numai între mobil şi reţea, la începutul apelului, fiind numai la dispoziţia reţelei. Staţia mobilă nu poate solicita autentificarea. Dacă nu este efectuată nici o autentificare, K_c rămâne cel din conversaţia anterioară; în acest caz, reţeaua poate "autentifica" telefonul prin faptul că foloseşte acelaşi K_c pentru criptare.

2. Reţeaua alege algoritmul de criptare (sau poate alege să nu cripteze deloc). Telefonul doar raportează lista de algoritmi de criptare pe care îi suportă (printr-un mesaj numit *class-mark*).

3. Mesajul class-mark nu este protejat şi poate fi modificat de *Oscar*.

4. Faptul că doar telefonul se autentifică la reţea permite existenţa staţiilor de bază false.

5. Nu există o separare a cheilor: protocolul de stabilire a cheii este independent de algoritmul folosit – cheia K_c depinde numai de $RAND$ (ales de reţea), indiferent dacă pentru criptare se foloseşte $A5/1$, $A5/2$ sau $A5/3$.

6. Aceeaşi valoare $RAND$ poate fi folosită ori de câte ori doreşte reţeaua.

5.5.1 Atacul de tip class-mark

În cel mai simplu atac asupra protocolului, *Oscar* schimbă informaţia din mesajul class-mark pe care telefonul îl trimite reţelei la începutul conversaţiei; astfel, reţeaua crede că telefonul suportă numai $A5/2$. Deşi poae reţeaua preferă să lucreze cu $A5/1$, ea trebuie să folosească numai $A5/2$ sau $A5/0$ (fără criptare).

Intrusul are mai multe moduri de a modifica mesajul class-mark. El poate trimite – simultan cu utilizatorul – un mesaj alternativ class-mark, folosind însă un semnal radio mult mai puternic; astfel, la BTS semnalul lui *Oscar* depăşeşte semnalul mesajului original.
Sau, *Oscar* poate efectua un atac de tipul man-in-the-middle (se interpune între staţia mobilă şi BTS folosind o staţie mobilă falsă şi o staţie de bază falsă), aşa încât toate mesajele să treacă pe la el. Apoi poate pur şi simplu înlocui un mesaj class-mark cu un alt mesaj.

5.5.2 Recuperarea cheii K_c dintr-o conversaţie anterioară

Este un atac care recuperează cheia de criptare a unei conversaţii criptate înregistrată în trecut. Ea ar putea fi validă pentru convorbiri ulterioare dacă reţeaua alege să nu efectueaze protocolul de stabilire a unei noi chei.

Calea cea mai simplă de a decripta conversaţii înregistrate este când *Oscar* are acces la cardul SIM al utilizatorului. Atunci el îl poate alimenta cu $RAND$-ul folosit anterior în conversaţie. SIM-ul calculează şi întoarce spre intrus valoarea lui K_c (atacul este posibil pentru că GSM permite refolosirea valorii $RAND$).

Singura problemă este obţinerea accesului fizic la cardul SIM al utilizatorului. Există un atac man-in-the-middle care simulează un astfel de acces, prin folosirea unei staţii de bază false.

1. În faza de pregătire a atacului, *Oscar* înregistrează conversaţii criptate ale victimei.

2. La momentul atacului, el iniţiază o sesiune wireless cu telefonul victimei, prin staţia de bază falsă, iar apoi o procedură de autentificare, folosind valoarea $RAND$ din timpul conversaţiei criptate. Telefonul va returna un $SRES$ cu aceeaşi valoare ca $SRES$-ul conversaţiei înregistrate.

3. *Oscar* cere telefonului să înceapă criptarea cu $A5/2$.
   Aparatul trimite o certificare (criptată cu $A5/2$) cu acelaşi K_c din conversaţia înregistrată (K_c depinde de $RAND$, care este identic cu cel din conversaţia înregistrată).

4. În final, intrusul poate folosi un atac cu text criptat pentru $A5/2$ (un astfel de atac este prezentat mai târziu), care poate fi repetat de mai multe ori pentru toate valorile $RAND$ existente în înregistrare.

Atacul prezentat lasă urme, pentru că telefonul reţine ultimul K_c pentru a-l folosi în următoarea conversaţie.

Oscar poate aduce telefonul la starea anterioară atacului, prin efectuarea unei noi proceduri de autentificare folosind ultima valoare (legitimă) $RAND$ emisă telefonului. Sau, *Oscar* poate recupera valoarea curentă K_c stocată pe telefon prin efectuarea atacului, dar omiţând procedura de autentificare.

El poate folosi cheia K_c pentru a pătrunde în conversaţii ulterioare, până când reţeaua iniţiază o nouă procedură de autentificare.

5.5.3 Atac de tipul man-in-the-middle

Oscar poate intercepta conversaţiile în timp real prin lansarea unui atac man-in-the-middle.

Pentru aceasta, el foloseşte o staţie de bază falsă pentru a comunica cu telefonul ţintă, pretinzând reţelei că este staţia mobilă .

1. Când reţeaua iniţiază protocolul de autentificare, ea trimite lui *Oscar* o cerere de autentificare, pe care acesta – sub chipul staţiei de bază – o trimite mai departe utilizatorului.

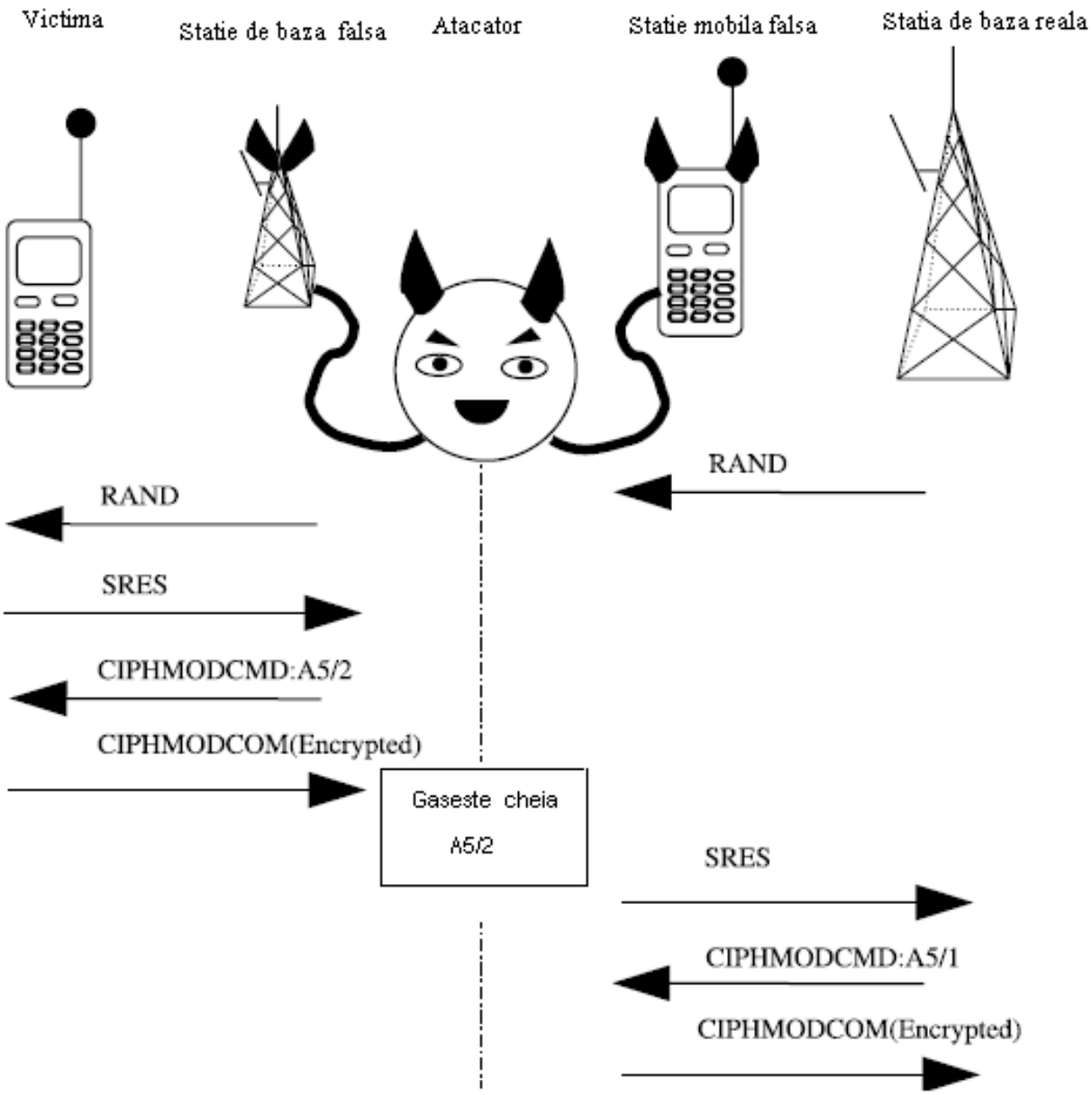
2. Utilizatorul calculează $SRES$ pe care îl returnează lui *Oscar*, care îl va reţine temporar, fără a-l retrimite reţelei.

3. *Oscar* cere telefonului să înceapă criptarea folosind $A5/2$.
   Această cerere pare legitimă telefonului, întrucât intrusul joacă rolul reţelei. Telefonul începe criptarea folosind $A5/2$ şi trimite o certificare (acknowledgment numit $CIPHMODCOM$ - *Cipher Mode Complete*, care confirmă că a început criptarea) criptată.

4. *Oscar* foloseşte atacul cu text criptat din secţiunea **??** pentru a găsi K_c (în mai puţin de o secundă).

5. Abia acum *Oscar* trimite valoarea $SRES$ spre reţeaua reală.

6. Acum *Oscar* este "autentificat" în reţea, care îi cere să înceapă criptarea folosind $A5/1$.
   Intrusul cunoaşte deja cheia K_c şi poate trimite răspunsul criptat cu $A5/1$.



Atac man-in-the-middle pentru A5/2

Din acest moment, reţeaua îl vede pe *Oscar* ca staţie mobilă, iar *Oscar* poate continua conversaţia, sau o poate încredinţa telefonului.

5.6 Criptanaliza algoritmului $A5/2$

5.6.1 Primul atac cu text clar cunoscut

Prima criptanaliză a algoritmului $A5/2$ a fost realizată în august 1999 de cercetătorii americani Goldberg, Wagner şi Green, fiind un atac cu text clar cunoscut. Atacul necesită două frame-uri de text criptat, al căror text clar corespunzător are o diferenţă XOR cunoscută.

Observaţia 5.1. *Datorită faptului că* $R4[10]$ *este setat pe valoarea 1 în timpul iniţializarii,* $R4$ *va avea aceeaşi valoare după iniţializare indiferent dacă bitul* $f[10]$ *este zero sau unu.*

Iniţial, $Oscar$ încearcă să găsească două frame-uri diferite care au aceeaşi valoare f până la $f[10]$. După etapa de iniţializare, cele două frame-uri vor avea aceeaşi valoare pentru $R4$. Având în vedere construcţia lui f din numărul de frame, practic se caută două frame-uri aflate la distanţa $26 \times 51 = 1326$ frame-uri $TDMA$ (aproximativ 6 secunde) de primul frame cu $f[10] = 0$.
Altfel, intrusul trebuie să mai aştepte 6 secunde pentru un frame cu $f[10] = 0$.

Prin urmare, $Oscar$ este obligat să aştepte între 6 şi 12 secunde pentru a obţine datele necesare pentru atac.

Atacul se desfăşoară astfel:

1. Fie f_1 şi f_2 valorile f pentru cele două frame-uri iar k_1 şi k_2 cheile fluide corespunzătoare. Notăm valoarea regiştrilor $R1 \dots R4$ din primul frame, imediat după iniţializare cu $R1_1 \dots R4_1$ şi similar, în al doilea frame $R1_2 \dots R4_2$.

   Alegerea lui f_1 şi f_2 conform observaţiilor de mai sus asigură $R4_1 = R4_2$. Ceilalţi regiştri nu sunt egali, dar datorită liniarităţii în biţii lui f_1 şi f_2 a procesului de iniţializare, diferenţa dintre $R1_1, R2_1, R3_1$ şi $R1_2, R2_2, R3_2$ este liniară în diferenţa dintre f_1 şi f_2.

   Deci, se poate scrie $R1_1 = R1_2 \oplus d_1$, $R2_1 = R2_2 \oplus d_2$, $R3_1 = R3_2 \oplus d_3$ cu d_1, d_2, d_3 constante.

2. Fiind dată valoarea lui $R4$, diferenţa $k_1 \oplus k_2$ este liniară în $R1_1, R2_1, R3_1$. Deci, fiind dat $R4$, se cunoaşte comportarea tuturor regiştrilor.

   Fie l_1, l_2 şi l_3 valori reprezentând câţi tacţi au avansat regiştrii $R1, R2$ şi $R3$ până la sfârşitul ciclului i.

   Valorile celor trei regiştri la încheierea ciclului i din primul frame sunt $L1^{l_1}R1_1$, $L2^{l_2}R2_2$ şi $L3^{l_3}R3_3$ cu $L1, L2$ şi $L3$ matrici care reprezintă un tact pentru registrul corespunzător.

   Analog, pentru frame-ul al doilea, valorile celor trei regiştri la sfârşitul ciclului i sunt $L1^{l_1}(R1_1 \oplus d_1)$, $L2^{l_2}(R2_2 \oplus d_2)$ şi $L3^{l_3}(R3_3 \oplus d_3)$.

Notând cu $g_1(R1) \oplus g_2(R2) \oplus g_3(R3)$ bitul de ieşire din $A5/2$, se observă că funcţiile $g_1(\cdot), g_2(\cdot)$ şi $g_3(\cdot)$ sunt pătratice (obţinute prin aplicarea funcţiei maj).

3. Goldberg, Wagner şi Green au mai observat că XOR-ul dintre biţii de ieşire poate fi exprimat ca funcţie liniară între biţii stării interne din primul frame.

   XOR-ul dintre biţii de ieşire (la ciclul i) ai celor două frame - uri este:

$$g_1(L1^{l_1} \cdot R1_1) \oplus g_1(L1^{l_1} \cdot R1_1 \oplus d_1) \oplus g_2(L2^{l_2} \cdot R2_1) \oplus g_2(L2^{l_2} \cdot R2_1 \oplus d_2)$$
$$\oplus g_3(L3^{l_3} \cdot R3_1) \oplus g_3(L3^{l_3} \cdot R3_1 \oplus d_3) = g_{d_1}(L1^{l_1} \cdot R1_1) \oplus g_{d_2}(L2^{l_2} \cdot R2_1) \oplus g_{d_3}(L3^{l_3} \cdot R3_1)$$

   unde $g_{d_1}(\cdot), g_{d_2}(\cdot), g_{d_1}(\cdot)$ se pot defini ca funcţii liniare.

   Deci XOR-ul de ieşire este liniar în $R1_1, R2_2, R3_3$. Mai rămâne de văzut dacă funcţiile $g_{d_1}(\cdot), g_{d_2}(\cdot), g_{d_1}(\cdot)$ sunt liniare.

   Mai general, rămâne de arătat că pentru o funcţie pătratică $g(x_1, ..., x_n)$ şi $d = d_1, ..., d_n$ cu $x_i, d_i \in \{0, 1\}$, funcţia

$$g_d = g(x_1, ..., x_n) \oplus g(x_1 \oplus d_1, x_2 \oplus d_2, ..., x_n \oplus d_n)$$

   este liniară în $x_1, ..., x_n$.

4. Deoarece g este pătratică şi $x_i x_i = x_i$, g poate fi scris astfel:

$$g(x_1, ...x_n) = \sum_{1 \le i,j \le n} a_{i,j} x_i x_j \oplus a_{0,0}$$

   unde $a_{i,j} \in \{0, 1\}$ sunt fixate pentru un anumit g.

   Deci,

$$g_d = \sum_{1 \le i,j \le n} a_{i,j}(x_i x_j \oplus (x_i \oplus d_i) \oplus (x_j \oplus d_j)) =$$
$$= \sum_{1 \le i,j \le n} a_{i,j}(x_i x_j \oplus x_i x_j \oplus x_i d_j \oplus d_i x_j \oplus d_i d_j)) = \sum_{1 \le i,j \le n} a_{i,j}(\oplus x_i d_j \oplus d_i x_j \oplus d_i d_j))$$

   Fiind date $d_1, ..., d_n$, ultima expresie este liniară în $x_1, ..., x_n$.

   Prin urmare, ştiind $R4$ şi $k_1 \oplus k_2$, stările interne $R1_1, R2_1$ şi $R3_1$ pot fi aflate prin rezolvarea unui sistem de ecuaţii liniare.

   K_c poate fi obţinut din starea internă iniţială şi f_1 prin inversarea paşilor de iniţializare din $A5/2$.

   Cât despre $R4$, acesta nu se dă şi deci $Oscar$ trebuie să testeze pe rând toate cele 2^{16} valori posibile pentru $R4$ (lungimea lui este de 17 biţi, dar $R4[10]$ este setat pe 1) şi pentru fiecare valoare să rezolve sistemul de ecuaţii liniare rezultat, până găseşte o soluţie consistentă.

O soluţie mai rapidă este posibilă dacă se filtrează valorile corecte pentru $R4$.

Starea internă iniţială a lui $R1, R2$ şi $R3$ este pe 61 biţi (trei biţi din $R1, R2, R3$ sunt setaţi pe 1).
Prin urmare, sunt necesari 61 biţi din $k_1 \oplus k_2$ pentru a construi K_c, în timp ce $k_1 \oplus k_2$ are o lungime de 114 biţi.

Este deci posibilă construirea unui sistem liniar unic determinat, a cărui soluţie este starea internă. Cele $114 - 61 = 53$ ecuaţii dependente se reduc prin eliminare gaussiană. Ele depind de valoarea lui $R4$; deci, pentru fiecare valoare a lui $R4$ se pot scrie 53 ecuaţii $V_{R4} \cdot (k_1 \oplus k_2) = 0$ unde V_{R4} este o matrice de 53×114 biţi iar 0 este un vector cu 53 valori nule.

Redundanţa se foloseşte pentru filtrarea valorilor greşite pentru care $V_{R4} \cdot (k_1 \oplus k_2) \neq 0$.

O implementare directă a atacului pe un calculator de $2GHz$ pe 32 biţi în care toate matricile posibile V_{R4} sunt preîncărcate, consumă aproximativ 15 MBs de memorie RAM şi necesită câteva milisecunde CPU pentru filtrarea valorilor corecte pentru $R4$.

Odată ce s-a aflat $R4$, se pot rezolva ecuaţiile liniare pentru valoarea sa, obţinând astfel valorile pentru $R1_1, R2_1$ şi $R3_1$.

Stocarea acestor sisteme de ecuaţii după eliminarea Gauss necesită cam 60 MBs de memorie.

Timpul de preprocesare necesar (pentru calculul ecuaţiilor şi al matricilor V_{R4}) pe un PC este de câteva minute.

Rămâne o problemă a acestui atac obţinerea acelor frame-uri care se află exact la distanţa cerută, cunoscând şi XOR-ul dintre cheile fluide calculate pentru criptarea lor.

Atacul Petrovici - Fuster - Sabater

Un alt atac cu text clar cunoscut este propus de Petrovic şi Fuster-Sabater ([**?**]) în anul 2000.

El necesită patru frame-uri (nu neapărat separate de o anumită distanţă) din secvenţa de ieşire pentru reconstrucţia următorilor biţi de ieşire. Deşi atacul nu descoperă starea internă a lui $A5/2$, este totuşi capabil să recupereze restul conversaţiei.

Atacul mizează pe reiniţializările dese care au loc în cadrul convorbirii (pentru fiecare frame transmis) şi distribuţia proastă a valorilor speciale marcate din $LFSR$ pentru a reconstrui relaţiile liniare dintre biţii de ieşire.

După ce este realizată iniţializarea algoritmului şi cele 100 de cicluri care se ignoră, se scrie sistemul de ecuaţii neliniare având ca necunoscute stările regiştrilor care au avansat un tact. Pentru fiecare tact cu deplasare al algoritmului, sistemului i se adaugă o nouă ecuaţie. Probabilitatea ca rangul matricii să fie apropiat de numărul de variabile este destul de mică. Dar – chiar şi aşa – este posibilă stabilirea relaţiilor liniare între biţii necunoscuţi care urmează a fi aflaţi şi mulţimea de biţi cunoscuţi.

Atacul propus se bazează pe analiza sistemului de ecuaţii asociat fiecărei stări iniţiale a registrului $R4$.
După primul set de operaţii din partea de iniţializare (cel în care se aplică XOR între primii biţi din registre şi cheia secretă), se poate scrie un sistem de ecuaţii neliniare având ca variabile stările interne ale primilor trei regiştri la acel moment. Aceste variabile se notează $x_1, .., x_{64}$ unde primele 19 variabile corespund stării din $R1$, următoarele 22 corespund stării din $R2$ iar ultimele 23 corespund stării din $R3$. Ecuaţiile obţinute sunt pătratice. Numărul maxim de variabile din sistemul liniarizat asociat este:

$$n = 64 + \begin{pmatrix} 19 \\ 2 \end{pmatrix} + \begin{pmatrix} 22 \\ 2 \end{pmatrix} + \begin{pmatrix} 23 \\ 2 \end{pmatrix}$$

Sistemul de ecuaţii obţinut este liniarizat prin înlocuirea termenilor neliniari cu noi variabile compuse. *Oscar* poate folosi ecuaţiile liniar dependente pentru a reconstrui biţii necunoscuţi de la ieşire care apar la foarte scurt timp după biţii cunoscuţi.

5.6.2 Atacurile Elad Barkan, Eli Biham şi Nathan Keller

Atac neoptimizat cu text clar

Un atac cu text clar cunoscut care îmbunătăţeşte primul atac descris anterior a fost realizat în 2003 de o echipă de cercetători din Israel: Elad Barkan, Eli Biham şi Nathan Keller ([**?**]). Ca şi atacul Petrovici - Fuster - Sabater, el necesită patru frame-uri; dar în final recuperează starea internă dată de $R1, R2, R3, R4$ şi găseşte şi cheia de sesiune.

Atacul determină valoarea lui $R4$ şi scrie fiecare bit de ieşire ca un termen pătratic depinzând de $R1, R2$ şi $R3$. Autorii descriu o metodă de reprezentare a fiecărui bit de ieşire ca un termen pătratic în stările $R1, R2, R3$ din primul frame.
Având astfel biţii de ieşire din patru frame-uri, ei construiesc un sistem de ecuaţii pătratice pe care îl rezolvă prin liniarizare, recuperând valorile iniţiale pentru $R1, R2$ şi $R3$.

În continuare vom detalia modul de construire a sistemului, precum şi rezolvarea lui.

Fie k_1, k_2, k_3, k_4 cheile fluide din cele patru frame-uri f_1, f_2, f_3, f_4. Fiecare k_j reprezintă ieşirea unui frame complet, deci are 114 biţi lungime. Starea iniţială internă din registrul Ri din frame-ul f_j (după liniarizare, dar înainte de cei 99 tacţi) se notează cu Ri_j.

Fiecare bit de ieşire poate fi scris ca o funcţie pătratică depinzând de starea iniţială internă a regiştrilor $R1, R2$ şi $R3$. Scopul urmărit este construirea unui sistem de ecuaţii pătratice care exprimă egalitatea între termenii pătratici pentru fiecare bit de ieşire, şi valoarea actuală a bitului din cheia fluidă cunoscută. Soluţia unui astfel de sistem ar face cunoscută starea internă iniţială.
Impedimentul principal este faptul că rezolvarea unui sistem de ecuaţii pătratice este o problemă NP - completă. Totuşi, există facilităţi când sistemul este supradefinit (sunt 61 variabile şi 114 ecuaţii), iar complexitatea lui scade considerabil pe măsură ce ecartul dintre numărul de ecuaţii şi numărul de variabile creşte. Prin urmare, vom adăuga ecuaţii

din alte frame-uri, cu condiţia ca ele să fie definite peste aceleaşi variabile: valorile iniţiale ale lui $R1, R2, R3$ din frame-ul f_1.

Odată combinate ecuaţiile din patru frame-uri, sistemul se rezolvă prin liniarizare. Se construiesc astfel de sisteme pentru fiecare din cele 2^{16} valori posibile pentru $R4_1$ şi se rezolvă până se găseşte o soluţie consistentă.
Soluţia este starea iniţială internă a frame-ului f_1.

Din modul de calcul, funcţia majoritară operează pe biţii aceluiaşi registru. Deci, termenii pătratici constau din perechi de variabile din acelaşi registru. $R1$ contribuie cu 18 variabile şi toate cele $\frac{17 \cdot 18}{2} = 153$ produse ale lor, $R2$ contribuie cu $21+\frac{21 \cdot 20}{2} = 21+210$ variabile, iar $R3$ contribuie cu $22+\frac{22 \cdot 21}{2} = 22+231$ variabile; deci după liniarizare rezultă 655 variabile. La acestea se adaugă constanta 1 care reprezintă partea afină a ecuaţiilor. Mulţimea acestor 656 variabile pentru frame-ul f_i se notează cu S_i.

Rămâne de arătat cum se pot descrie biţii de ieşire din frame-urile f_2, f_3, f_4 drept combinaţii liniare de variabile din mulţimea S_1. Se presupune cunoscută valoarea lui $R4_1$ şi faptul că iniţializarea este liniară în valoarea publică f.

Cum $R1_1, R2_1, R3_1$ sunt necunoscute, ştim numai XOR-ul dintre $R1_1, R2_1, R3_1$ şi respectiv $R1_2, R2_2, R3_2$.

Fiecare valoare din S_2 va fi translatată în S_1 astfel:
Fie α_1 valoarea concatenată a variabilelor liniare din S_1 şi g o funcţie pătratică astfel încât $S = g(\alpha_1)$. Stim că valoarea concatenată a variabilelor liniare din S_2 poate fi scrisă ca $\alpha_2 = \alpha_1 \oplus d_{1,2}$ şi deci $S_2 = g(\alpha_2)$.

La fel ca în primul atac, XOR-ul dintre S_2 şi S_1 este liniar în biţii lui α_1, deci S_2 poate fi scris în termeni liniari în variabilele lui S_1.
Se poate construi deci un sistem de ecuaţii pătratice folosind cheile fluide din cele patru frame-uri, cu variabile luate numai din S_1. Se va crea un sistem de ecuaţii de forma $S_{R4_1} \cdot S_1 = k$ unde S_{R4_1} este matricea sistemului de dimensiune 456×656, iar $k = k_1 \| k_2 \| k_3 \| k_4$ (de 456 biţi).

Odată obţinute 656 ecuaţii liniar independente, sistemul poate fi rezolvat prin eliminare gaussiană. Se observă însă că – practic – este dificil să colecţionăm 656 ecuaţii liniar independente (datorită frecventelor iniţializări ale algoritmului $A5/2$ – odată la fiecare 228 biţi). Nu trebuie aflate însă toate variabilele, ci numai cele liniare. Autorii au ajuns experimental la concluzia că aproximativ 450 ecuaţii liniar independente sunt suficiente pentru a determina valorile variabilelor liniare originale din S_1.

De asemenea se mai pot obţine 13 ecuaţii liniare adiţionale, datorită cunoaşterii lui $R4_1$ şi a numărului de frame. Fie $R1234_1 = R1_1 \| R2_1 \| R3_1 \| R4_1$ un vector de 77 biţi (neluând în considerare cei patru biţi setaţi pe 1 în cadrul procedurii de iniţializare). Cum $R1234_1$ este liniar în biţii lui K_c şi f_1, putem scrie:

$$R1234_1 = N_K \cdot K_c \oplus N_f \cdot f_1 \tag{1}$$

unde N_K este o matrice de 77×64 şi N_f este o matrice de 77×22 care reprezintă

componenta de iniţializare din $A5/2$. Spaţiul liniar generat de coloanele lui N_K este de dimensiune 64, dar cum fiecare vector are 77 de biţi, rămân 13 ecuaţii liniare în $N_K \cdot K_c$. Fie H_K matricea acestor ecuaţii, de dimensiune 13×77:

$$H_K \cdot N_k = 0$$

unde 0 este matricea nulă de dimensiuni 13×64.
Prin multiplicarea relaţiei (1) la stânga cu H_K se obţine:

$$H_K \cdot R1234_1 = H_K \cdot N_K \cdot K_c \oplus H_K \cdot N_f \cdot f_1 = H_K \cdot N_f \cdot f_1$$

H_K se poate împărţi în H_K^L şi H_K^R astfel încât:

$$H_K \cdot R1234_1 = H_K^L \cdot R123_1 \oplus H_K^R \cdot R4_1$$

unde $H_K = H_K^L \| H_K^R$, cu H_K^L de dimensiune 13×61, H_K^R de dimensiune 13×16 şi $R123_1 = R1_1 \| R2_1 \| R3_1$.

Rezultă:

$$H_K \cdot N_f \cdot f_1 = H_K \cdot R1234_1 = H_K^L \cdot R123_1 \oplus H_K^R \cdot R4_1$$

care poate fi scris:

$$H_K^L \cdot R123_1 = H_K \cdot N_f \cdot f_1 \oplus H_K^R \cdot R4_1.$$

Deci, pe baza lui $R4_1$ şi a lui f_1 se pot obţine 13 ecuaţii liniare peste biţii din regiştrii $R1, R2, R3$.

Recapitulând, atacul constă în:

1. Se încearcă toate cele 2^{16} valori;

2. Pentru fiecare valoare se rezolvă sistemul liniarizat de ecuaţii care descriu biţii de ieşire pentru patru frame-uri.

3. Soluţia obţinută plus valoarea lui $R4_1$ reprezintă o sugestie pentru starea internă. Majoritatea valorilor lui $R4_1$ pot fi identificate uşor datorită inconsistenţelor apărute la eliminarea gaussiană.

Complexitatea timp a atacului se poate calcula astfel: sunt 2^{16} valori aleatoare pentru biţii din $R4_1$. Pentru fiecare valoare se rezolvă un sistem liniar binar de 656 variabile, care presupune $656^3 \approx 2^{28}$ operaţii XOR.
Deci, complexitatea totală este de aproximativ 2^{44} operaţii XOR.

Autorii spun că implementarea algoritmului pe un sistem Linux 800 MHz Pentium III găseşte starea internă în aproximativ 40 de minute (estimare 2003) şi necesită puţină memorie (sistemul liniarizat ocupă aproximativ 54 KB).

Un atac optimizat asupra lui $A5/2$

După publicarea atacului anterior, aceeaşi autori revin cu o versiune optimizată, care găseşte K_c în câteva milisecunde, folosind tabele precalculate şi stocate în memorie.

Ideea de bază este următoarea: într-o fază de precalcul, pentru fiecare valoare $R4_1$ se determină dependenţele care apar în timpul eliminării gaussiene pentru sistemul de ecuaţii. Apoi – în faza de atac – se filtrează valorile corecte pentru $R4_1$ prin aplicarea verificărilor de consistenţă pe cheile fluide cunoscute.

Deci, într-o primă fază de precalcul, se stabilesc în avans sistemele de ecuaţii pentru toate valorile $R4_1$. Tot în avans se rezolvă fiecare astfel de sistem; în particular, fiind dat un sistem de ecuaţii $S_{R4_1} \cdot S_1 = k$, se calculează o matrice T_{R4_1} pentru care $T_{R4_1} \cdot S_{R4_1}$ este rezultatul eliminării Gauss a lui S_{R4_1}. Cum S_{R4_1} nu depinde numai de $R4_1$ dar şi de XOR-ul dintre valorile f ale frame-urilor, trebuie efectuat un precalcul pentru mai multe XOR-uri de valori f. Similar atacului anterior, XOR-ul între valorile f este folosit la translatarea mulţimilor de variabile S_1, S_2, S_3 în S_1.

Fiind necesară cunoaşterea în avans a acestor diferenţe XOR, se efectuează precalcule pentru diverse valori posibile pentru diferenţe, iar rezultatele se păstrează în tabele. Apoi, în faza de atac, se folosesc tabelele asociate valorilor f din frame-uri.
Dacă se dau valori pentru cheile fluide corespunzătoare unui frame având valoarea f neacoperită de precalculele făcute, aceste chei se vor abandona, aşteptându-se altele, care îndeplinesc cerinţa menţionată.

Revenind la faza de atac, se calculează $t = T_{R4_1} \cdot k$ pentru fiecare valoare a lui $R4_1$. Primele elemente ale vectorului t sunt variabilele din S_1 parţial rezolvate; dar – deoarece unele ecuaţii sunt liniar dependente – restul elementelor lui t ar trebui să fie zero (reprezentând ecuaţiile dependente). Prin urmare, se verifică dacă ultimele elemente din t sunt într-adevăr nule (cheia k este consistentă cu valoarea testată pentru $R4_1$). Odată găsită o valoare consistentă pentru $R4_1$, ea poate fi verificată prin calcularea cheii şi efectuarea de teste prin criptare.
O implementare mai rapidă nu reţine în memorie matricea T_{R4_1}, ci doar ultimele linii $T^0_{R4_1}$: cele care corespund elementelor nule din t. Apoi, pentru verificarea consistenţei valorii $R4_1$, se poate testa numai dacă $t' = T^0_{R4_1} \cdot k$ este un vector de zero-uri.

Analiza complexităţii timp şi spaţiu a atacului folosind un singur tabel precalculat (pentru un singur XOR între valorile f ale frame-urilor): autorii spun că timpul necesar pentru precalcule este comparabil cu timpul necesar pentru efectuarea atacului în varianta neoptimizată, adică aproximativ 40 minute. În faza de atac, trebuie păstrate în memorie matricile (pentru operaţii rapide). O singură matrice de sistem are cam $456 \cdot 16$ biţi, deci sunt necesari cam 60 MBs pentru reţinerea tabelului corespunzător a 2^{16} valori posibile ale lui $R4_1$. Alţi $64 \cdot 456 \cdot 216 \approx 240$MBs sunt necesari pentru păstrarea matricilor folosite la aflarea stării interne având $R4_1$ şi cheia fluidă respectivă.
Timpul atacului este de 250 cicluri CPU pentru înmulţirea şi verificarea unei matrici, sau aproximativ câteva milisecunde în total pe un PC (date 2006). După aflarea unui

candidat pentru $R4_1$, încărcarea matricei soluţie relevante de pe disc durează câteva zeci de milisecunde.

În implementarea autorilor, atacul durează mai puţin de o secundă pe un PC.

Atac cu text criptat

În această secţiune este prezentat un atac care transformă atacurile anterioare într-unul cu text criptat cunoscut, şi care verifică anumite combinaţii liniare de biţi dinainte de criptare, bazându-se pe teoria codurilor corectoare de erori.

GSM foloseşte corectarea erorilor pentru a rezista la erorile de canal. În timpul transmisiei, un mesaj este întâi transformat cu un cod corector de erori, ceea ce măreşte considerabil lungimea mesajului. În faza a doua, mesajul este criptat şi ulterior transmis. Această procedură inversată contrazice practica obişnuită: de a cripta întâi un mesaj şi apoi de a-l codifica cu un cod corector de erori. Tocmai această trecere a mesajului prin codurile corectoare de erori înainte de criptare introduce o redondanţă exploatată în atacul curent.

Există diverse scheme de corectare a erorilor în GSM, pentru diferite canale. Autorii atacului s-au concentrat asupra codurilor corectoare de erori aplicate canalului $SACCh$, folosite de asemenea şi pentru canalul $SDCCh$. Ambele canale sunt utilizate la începutul apelului. Alte canale sunt accesate în alte etape ale conversaţiei, iar atacul propus poate fi adaptat şi la acestea (deşi este suficient să găsim cheia pe canalul $SDCCh$ la începutul apelului, pentru că aceasta nu se schimbă în timpul conversaţiei).

În $SACCh$, mesajul sursă (care se codifică) are o lungime fixă de 184 biţi; după codificare, va avea 456 biţi. Acestora li se aplică procedura de interleaving[5], după care sunt împărţiţi în patru frame-uri, ulterior criptate şi transmise.

Operaţiile de codificare şi interleaving pot fi modelate împreună ca o multiplicare a mesajului sursă (reprezentat ca un vector binar P de 184 biţi) cu o matrice G de dimensiune 456×184 peste $GF(2)$ şi apoi XOR-area rezultatului cu un vector constant g:

$$M = (G \cdot P) \oplus g \tag{2}$$

Vectorul M este împărţit apoi în patru frame-uri, iar fiecare este criptat conform procedeului corespunzător algoritmului $A5/2$.

Cum G este o matrice binară, pentru orice vector M care verifică (2) există $456-184 = 272$ ecuaţii liniar independente.

Fie H o matrice definită prin relaţia

$$H \cdot (M \oplus g) = 0$$

unde M este dat de (2) (în teoria codurilor, H se numeşte *matrice de control*).

[5]Pentru detalii privind tehnicile de codificare, a se vedea [**?**].

Să arătăm cum se foloseşte redondanţa mesajului pentru a lansa atacul.
Mesajul criptat este calculat după formula

$$C = M \oplus k$$

unde $k = k_1 \| k_2 \| k_3 \| k_4$ este cheia fluidă pentru cele patru frame-uri. Se folosesc aceleaşi ecuaţii pentru $C \oplus g$ adică

$$H \cdot (C \oplus g) = H \cdot (M \oplus k \oplus g) = H \cdot (M \oplus k) \oplus H \cdot k = 0 \oplus H \cdot k = H \cdot k$$

Cum valoarea textului criptat C este cunoscută (g este fixat şi cunoscut), ceea ce rămâne sunt ecuaţii liniare peste biţii lui k.
Pentru fiecare ghicire a lui $R4_1$ se înlocuieşte fiecare bit al lui k în acest sistem de ecuaţii, cu descrierea sa ca termen liniar peste S_1; se ajunge la un sistem de ecuaţii peste cele 656 variabile din S_1. Fiecare bloc de 456 biţi care se codifică furnizează cele 272 ecuaţii mentionate anterior; deci, după două astfel de blocuri se obţin peste 450 ecuaţii.
Într-o manieră asemănătoare atacului neoptimizat, se efectuează apoi eliminări gaussiene, cele 450 ecuaţii fiind suficiente pentru a găsi valorile variabilelor liniare originale din S_1.

În final, K_c este determinat prin inversarea etapei de iniţializare din $A5/2$.

Restul atacului şi complexitatea timp sunt similare cu cazul precedent. Principala diferenţă este aceea că în atacurile precedente se cunosc biţii cheilor fluide, în timp ce în acest atac se cunoasc valoarile combinaţiilor liniare ale biţilor cheii fluide.
Prin urmare, ecuaţiile rezultate aici sunt combinaţii liniare ale ecuaţiilor din atacurile precedente.

Exemplul 5.5. *Fie $S_{R4_1} \cdot S_1 = k$ un sistem de ecuaţii din atacul optimizat, unde S_{R4_1} este matricea sistemului.*

Înmulţim acest sistem la stânga cu H şi obţinem

$$(H \cdot S_{R4_1}) \cdot S_1 = (H \cdot k).$$

H este matricea fixată prezentată anterior iar $H \cdot k$ este calculat din textul criptat. Prin urmare, acest sistem se poate rezolva şi atacul continuă ca în secţiunea anterioară.

În atacul cu text clar cunoscut, se încearcă toate cele 2^{16} sisteme de ecuaţii posibile S.

În atacul cu text criptat cunoscut, se încearcă toate cele 2^{16} sisteme de ecuaţii posibile $H \cdot S_{R4_1}$. În faza sa de precalcul, pentru fiecare astfel de sistem se găsesc dependenţele liniare ale liniilor (prin eliminare gaussiană). Când se trece la faza de atac, se elimină valorile greşite ale lui $R4_1$ verificând dacă dependenţele liniare găsite în etapa de precalcul se menţin pe biţii lui $H \cdot k$.

O diferenţă tehnică între atacul cu text criptat şi atacul cu text clar cunoscut este aceea că – în timp ce pentru cel din urmă atac, patru frame-uri de text clar furnizează suficiente ecuaţii, pentru primul atac sunt necesare opt frame-uri.
Motivul este că în atacul cu text criptat, din 456 biţi de text criptat se extrag numai 272 de ecuaţii.

Complexitatea timp a atacului cu text criptat optimizat este aceeaşi ca în cazul atacului cu text clar optimizat. Autorii au implementat o simulare a atacului şi au certificat experimental aceste rezultate.

5.6.3 Atac hardware asupra lui $A5/2$

Atacul prezentat în această secţiune a fost publicat într-un articol din 2007 ([**?**]). Autorii pornesc de la atacul cu text criptat cunoscut şi arată că atacul bazat pe hardware aduce îmbunătăţiri în termeni de timp, memorie şi flexibilitate.

Abordarea este similară cu cea descrisă în atacul anterior, dar fără a folosi precalcule şi fără a impune condiţii asupra frame-urilor de text criptat. Mai mult, autorii au creat o arhitectură care să atace direct canalul de vorbire. În particular, pentru atac se folosesc frame-uri de text criptat din canalele de trafic pentru vorbire (TCh) în loc de canale de control specifice ($SDCCh$). Avantajul este că monitorizarea (pentru recepţionarea de frame-uri) poate începe oricând în timpul apelului, fără a fi nevoie să se aştepte trimiterea pachetelor de date printr-un canal de control specific.

Atacul asupra canalului de vorbire necesită 16 frame-uri de text criptat la intrare, iar la ieşire va găsi starea iniţială (secretă). Principalele părţi ale arhitecturii hard constau în 3 generatori de ecuaţii şi un bloc care se ocupă de rezolvarea sistemelor liniare. Componenta hardware propusă pentru cel din urma bloc se numeşte $SMITH$ şi poate rezolva orice sistem supra-determinat de ecuaţii liniare, efectuând extrem de rapid eliminări gaussiene. Ea a fost construită pe un $FPGA$ (reţea de porţi logice reconfigurabile) de cost redus iar ideea matematică este următoarea: un sistem liniar de ecuaţii de forma $A \cdot x = b$ se transformă (prin aplicarea operaţiilor elementare asupra liniilor) în sistemul echivalent

$$U \cdot x = b' \tag{3}$$

unde U este o matrice superior triunghiulară.
Sistemul (3) poate fi rezolvat prin substituţie inversă. Folosindu-se de paralelizare hardware, complexitatea timp este cel mult pătratică, iar cea medie este liniară.
La o iteraţie, fiecare generator de ecuaţii produce o ecuaţie liniară având ca variabile biţii stării interne din $A5/2$. După 185 iteraţii (când au fost încărcate 555 ecuaţii), blocul $SMITH$ efectuează eliminări Gauss - Jordan paralelizate. Ieşirea sugerează starea secretă candidată – cea care trebuie verificată. Candidatul corect este astfel găsit după aproximativ 228 tacţi.

În această formă, atacul se poate realiza într-o secundă, la o viteză de operare de 256 MHz pentru principala componentă chip şi 512 MHz pentru celelalte, arhitectura consumând numai $12,8$ waţi.

Abordarea atacului este similară cu cea propusă în atacul cu text criptat; el necesită l frame-uri criptate cu aceeaşi cheie de sesiune K.

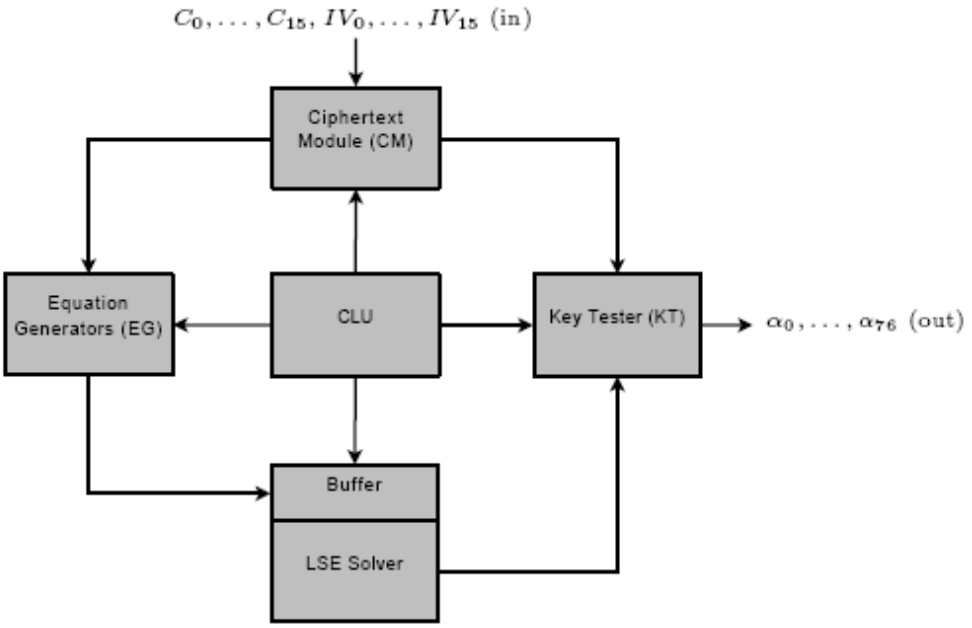
Parametrul l depinde de canalul ce va fi atacat. De pildă, sunt necesare 16 frame-uri pentru atacul asupra canalului de vorbire şi în jur de 8 frame-uri pentru atacul canalului

SDCCh.

Ideea generală este de a ghici starea internă a registrului $R4$ chiar după iniţializare, şi de a scrie fiecare bit al cheii generate – folosită pentru criptarea celor l frame-uri de text criptat – în termeni de stările iniţiale ale regiştrilor $R1, R2, R3$.
Folosind anumite informaţii despre biţii cheii, se construieşte un sistem supradeterminat pătratic de ecuaţii, care apoi este liniarizat şi rezolvat prin eliminări gaussiene.

Procedura se repetă pentru diverse valori ale lui $R4$ până este găsită soluţia corectă. Cu aceasta se poate construi uşor starea internă a lui $A5/2$ după iniţializare, pentru frame-uri arbirare care au fost criptate folosind cheia K.
Apoi se pot decripta toate celelalte frame-uri şi găsi, în final, cheia de sesiune.

Arhitectura necesară pentru realizarea atacului este schiţată în figura de mai jos; ea acceptă la intrare 16 frame-uri de text criptat şi cele 16 numere f de frame corespunzătoare; ieşirea conţine starea iniţială internă a celor 4 regiştri, formată din 77 biţi $\alpha_0, ...\alpha_{77}$ (s-au exclus cei 4 biţi setaţi pe valoarea 1).



Arhitectură propusă pentru atacul $A5/2$

Cele 16 frame-uri – împreună cu numerele de frame corespunzătoare, notate IV – sunt stocate în Ciphertext Module (CM). Fiecare din cei trei generatori de ecuaţii (EG) generează 185 ecuaţii liniare având biţii α_i $(0 \leq i \leq 60)$ ca variabile.
EG primesc biţii de text criptat şi numerele de frame IV de la CM. Fiecare ecuaţie generată este trecută în bufferul din blocul care rezolvă ecuaţiile (LSE - Solver). Bufferul este necesar pentru că $LSE-Solver$-ul acceptă o ecuaţie la fiecare tact, iar cele trei EG-uri produc simultan câte o ecuaţie. După ce cele 555 ecuaţii se acumulează în $LSE-Solver$, acesta lansează etapa de rezolvare, producând un candidat pentru starea secretă.

Candidatul este trimis din $LSE-Solver$ către *Key Tester*, care verifică dacă a fost găsită o stare corectă.
Procesul de verificare se desfăşoară în paralel cu determinarea unui nou candidat. Mai precis, în timp ce ecuaţiile pentru al j-lea candidat sunt generate de EG, al $(j-1)$-lea candidat este testat de către KT. Toate procesele sunt controlate de *Unitatea Logică de Control*, care efectuează sincronizarea şi clocking-ul pentru $CM, EG, LSE-Solver$ şi KT. Principala sa sarcina este de a verifica dacă ecuaţiile conţin combinaţii corecte de biţi de text criptat şi cheie.

5.7 Concluzii

Modelul de securitate propus de standardul GSM este compromis pe mai multe nivele şi este deci vulnerabil la atacuri care ţintesc anumite părţi ale reţelei operatorului. Algoritmii de securitate (nepublicaţi) încorporaţi în sistem s-au dovedit nesiguri. Algoritmul $A5/2$ folosit pentru criptare este vulnerabil la atacuri cu text clar şi chiar cu text criptat, iar dimensiunea redusă a spaţiului cheii şi a regiştrilor $LFSR$ din arhitectura algoritmilor de criptare permit chiar un atac prin forţă brută.

Există mai multe atacuri publicate pentru ambii algoritmi $A5/1$ şi $A5/2$.

Atacul pentru algoritmul $A5/2$ – care necesită numai 4 frame-uri de text clar cunoscut şi se bazează pe o etapă masivă de precalculeori – reuşeşte obţinerea cheii şi decriptarea într-un timp foarte scurt, de numai o secundă. Etapa de precalcul poate fi efectuată pe majoritatea computerelor actuale.
Pe de altă parte, atacul bazat pe hardware aduce îmbunătăţiri în complexitatea timp; el se bazează pe o arhitectură hardware masiv paralelă, care elimină necesitatea unei etape premergătoare de calcule.

Atacurile asupra algoritmului $A5/1$ sunt şi mai numeroase, ridică probleme mai mari, dar chiar şi în acest caz progresele tehnice simplifică mult lucrurile. Cea mai recentă propunere de atac pentru $A5/1$ ([**?**]) este de asemenea bazată pe hardware şi este remarcabilă pentru că este complet pasivă, costul total pentru harware se situează în jurul sumei de numai 1000 dolari, iar timpul total de spargere a cheii este de 30 de minute (putând ajunge şi la câteva secunde, dacă se măreşte puterea hardware).

Slăbiciunile dovedite ale algoritmilor de criptare GSM nu sunt singurele existente în sistem. Anonimitatea poate fi atacată activ, folosind echipamente care pot fi achiziţionate la preţuri relativ modice (de pildă, dacă se foloseşte o staţie de bază uzată).

Ameninţările legate de clonarea SIM par să fie destul de reale. Este adevărat că metoda cea mai uşoară este prin acces fizic la cartela ţintă, astfel încât nu poate fi o ameninţare foarte serioasă. Pe de altă parte, pentru a clona o cartelă SIM fără a avea acces fizic la ea devine mult mai costisitor. În principiu, atacatorul are nevoie de o staţie de bază cu semnal puternic, costul acesteia putând fi astăzi în jurul a câteva zeci de mii de dolari.

Bibliografie

[1] A. Atanasiu – *Securitatea informaţiei, vol. 1 (Criptografie)*, Ed. Infodata, Cluj (2007)

[2] A. Atanasiu – *Teoria codurilor corectoare de erori*, Ed. Universităţii Bucureşti (2001)

[3] P. Chandra – *BULLETPROOF WIRELESS SECURITY: GSM, UMTS, 802.11, and Ad Hoc Security (Communications Engineering)*, Elsevier (2005).

[4] A. Mihăiţă – *Securitatea sistemelor GSM*, Lucrare dizertaţie, 2009.

[5] S. Martin – *Communication Systems for the Mobile Information Society*, Wiley (2006).

[6] S. Redl, M. Weber, K. Matthias, M. Oliphant – *GSM and Personal Communications Handbook*, Artech House, 1998.

[7] E. Barkan, E. Biham, N. Keller – *Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication* , Journal of Cryptology, vol. 21, nr.3 (2008).

[8] E. P. Barkan – *Cryptanalysis of Ciphers and Protocols*, PhD Thesis, Technion Israel Institute of Technology (2006)

[9] P. Yousef – *GSM Security: a Survey and Evaluation of the Current Situation*, Master Thesis, Linkping, 2004

[10] A. Bogdanov, T. Eisenbarth, A. Rupp – *A Hardware-Assisted Realtime Attack on A5/2 without Precomputations*, Cryptographic Hardware and Embedded Systems - CHES 2007, LNCS, Sprnger Verlag (2007).

[11] P.T. Ngarm, P. Poocharoen – *GSM Security Vulnerability*, http://islab. oregonstate.edu/koc/ece478/03Report/toparkngarm-poocharoen-project578.pdf

[12] M. Briceno, I. Goldberg, D. Wagner – *A pedagogical implementation of the GSM A5/1 and A5/2 "voice privacy" encrypted algorithms*, http://cryptome.org/*GSM*-a512.htm, 1999

[13] N. Courtois, W. Meier – *Algebraic Attacks on Stream Ciphers with Linear Feedback*. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 346359, Springer, Heidelberg (2003).

[14] J. Neil Croft, S. O. Martin – *The use of a Third Party Proxy in Achieving GSM Anonymity*, Southern African Telecommunication Networks and Applications Conference 2004 (SATNAC 2004), D Browne (ed), Stellenbosch, Africa de Sud (2004).

[15] S. Petrovic, A. Fuster-Sabater – *Cryptanalysis of the A5/2 Algorithm*, IACR ePrint, Report 200/52 (2000)

[16] B. Schneier – http://www.schneier.com/blog/