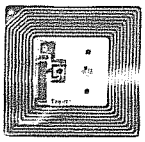


FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO IN FAKULTETA ZA MATEMATIKO IN FIZIKO

PROJEKT PRI PREDMETU KRIPTOGRAFIJA IN TEORIJA KODIRANJA 2

Identifikacija z RFID oznakami s pomočjo kriptografije

Avtor:
Slavko ŽITNIK

Mentor:
prof. dr. Aleksandar JURIŠIĆ

13. julij 2010

Kazalo

1 Uvod

Tehnologija RFID (*Radio Frequency IDentification*) nam omogoča identifikacijo preko radijskih valov. V zadnjih letih postaja zaradi vse nižje cene vse bolj popularna pri množični uporabi označevanja, sledenja predmetov. V prihodnosti se pričakuje, da bo popolnoma nadomestila identifikacijo s črtnimi kodami. Zaradi tega v tej seminarski opisujemo dva protokola, ki se lahko uporabljata v cenenih RFID oznakah.

Poleg svoje vsestranske uporabe pa se pojavljajo tudi varnostni problemi. Napadi lahko prihajajo od kogarkoli, tudi iz oddaljenosti več 10 metrov. V potrošniški uporabi je ena glavnih potreb zagotavljanje anonimnosti oziroma onemogočanje sledenja (*tracking*) Velikemu bratu. Prav tako je velik problem tudi, če bi o nas lahko kdo izdelal osebni profil (*profiling*) in nas nato identificiral glede na množico označenih predmetov (*hotlisting*).

V gospodarstvu pa želimo z novimi tehnologijami poceniti, pohitriti delovanje z enako varnostjo in nizkimi stroški vložka. Oznake želimo imeti čimcenejše in čimmanjše. Vsak RFID vsebuje mikročip, na katerem se lahko izvajajo računske operacije. Po občutku se cena oznake za vsakih dodanih 1000 logičnih vrat v čipu, poveča za 1 cent. Cilj te seminarske naloge je, da predstavimo protokola, ki nista računsko kompleksna, a vseeno omogočata varno identifikacijo.

Za nekatere storitve nekaj EUR za izdelavo RFID oznake ne predstavlja velikega stroška, saj jo lahko večkrat uporabljamo in si posledično lahko privoščimo boljšo zaščito. To pa ne pomeni, da pri masovni uporabi v knjižnicah, skladiščih ne potrebujemo varnosti.

2 Opis RFID oznake

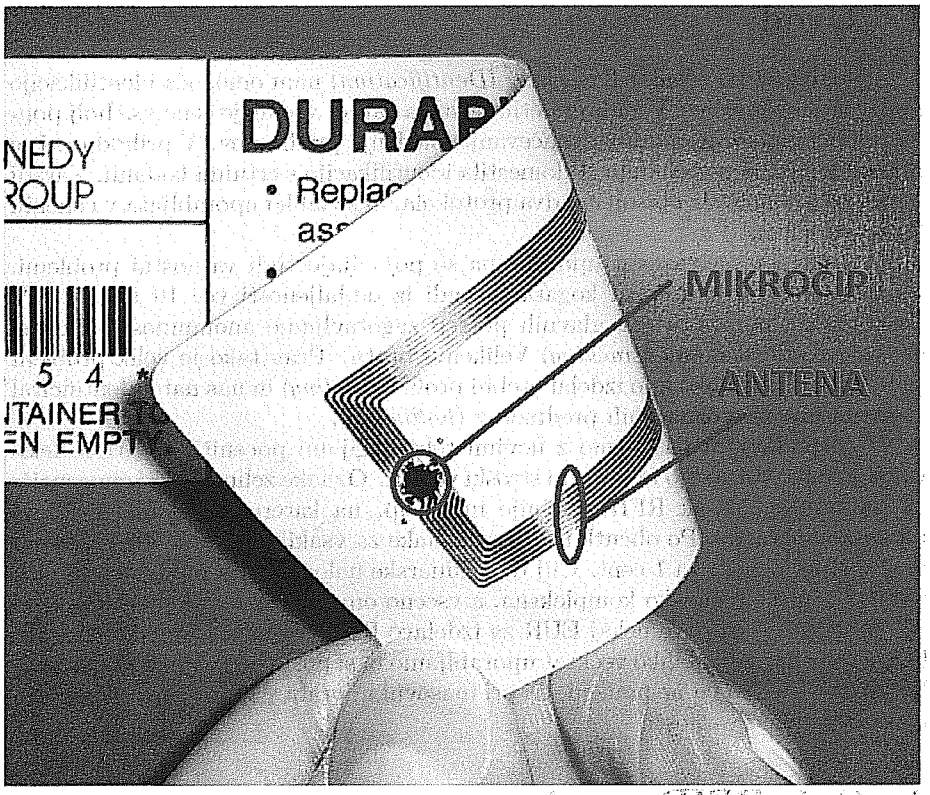
RFID je majhno elektronsko vezje, ki ga v sistemu imenujemo oddajnik ali oznaka. Sestavljen je iz integriranega vezja (čipa), ki hrani in procesira podatke, ter izvaja modulacijo in demodulacijo signalov. Drugi del oddajnika je antena, ki sprejema in oddaja radijske signale. Signale RFID oddajnikov sprejema RFID čitalec, kar nam omogoča identifikacijo predmetov oziroma bitij.

Na sliki 1 si lahko ogledate tipično sestavo RFID oznake.

2.1 Tipi RFID oznak

RFID sisteme delimo glede na napajanje (aktivni/pasivni) in način prenosa podatkov (induktivni/elektromagnetni). Več o tem si lahko preberete tudi na Wikipediji [12]:

- Tip napajanja:
  - *Aktivni RFID*: Oddajniki vsebujejo lastno napajanje (majhno baterijo). Zaradi tega se oznaki poveča cena, a imamo zaradi tega daljši domet in bolj zanesljivo delovanje v šumnih okoljih.
  - *Pasivni RFID*: Oddajniki ne vsebujejo lastnega napajanja, ampak potrebno energijo pridobijo iz induciranja signala v anteni. S tem se signal dovede do čipa, ki se "zbudi" in prične z delovanjem. Ker oddajniki nimajo baterije, so zato zelo poceni, a imajo krajši domet, a tudi manj nezanesljivo delovanje.

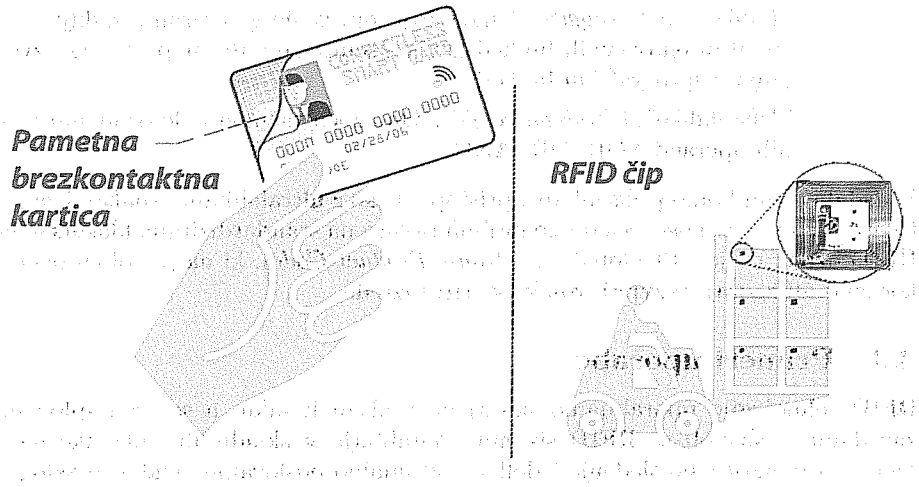


Slika 1: Prikaz RFID oznake

- Način prenosa podatkov:
  - *Induktivni RFID:* Oddajnik za prenos informacije uporablja princip magnetne indukcije. Preko dveh tuljav, ki sta priklopljeni na čip inducira napetost. Komunikacija s čitalcem deluje v bližnjem polju, zato je domet do nekaj 10cm. Oddajnik informacijo prenese čitalcu prek uporabe bremenske modulacije, ki pomeni spremeninjanje sklopnega faktorja v odvisnosti od poslanih podatkov.
  - *Elektromagnetni RFID:* Oddajnik komunicira preko elektromagnetnih valov, ki jih oznaki pošilja čitalec in se le ti odbijajo od oddajnika. Ta odboj se izkoristi, da se prenese informacija od oddajnika do čitalca. Ko se čip zbudi, začne spreminjati frekvenco signala glede na podatke, ki jih pošilja - modulacijski odboj.

2.2 RFID in pametne brezkontaktne kartice

Brezkontaktna pametna kartica (V svoji seminarski nalogi jih opisuje Moškon [7]) je pametna kartica, ki lahko preko radijskih valov komunicira z oddaljenim terminalom. Značilno sta čip in antena zapakirana v plastičnem ohišju velikosti bančne kartice. Glavna lastnost, ki jo loči od cenejših RFID čipov je večji spomin in možnost izvajanja internih kriptografskih operacij. Zaradi tega omogoča večjo



Slika 2: Prikaz razlike med pametno brezkontaktno kartico in RFID čipom

varnost in uporabo znanih kriptografskih protokolov. Ker je od RFID čipov večja in predvsem dražja, jo je neekonomsko uporabljati množično v skladiščih, trgovinah,... Kratko primerjavo značilnosti obeh si lahko ogledate v tabeli 1. V praksi za pametne kartice pogosto uporabljamo kar ime RFID čipi, zato si

	Pametna brezkontaktna kartica	RFID čip
Cena	0,5EUR ali več	manj od 0,05EUR
Fizična velikost	od 25x15mm debeline 1mm	od 5x5mm debeline papirja
Zmogljivost procesorja	možnost izvajanja kriptografskih funkcih	osnovne matematične operacije
Velikost pomnilnika	več KB	do 2Kb
Delovna razdalja	do 15cm	do nekaj 10m
Varnost podatkov	lahko zelo dobra	slabša

Tabela 1: Primerjava pametne brezkontaktne kartice in RFID čipa. Cene so bile pridobljene v spletni trgovini podjetja *1 Klik d.o.o.*

poglejmo naslednjo delitev, da se bomo še lažje odločili.

- Dražje *(High cost)* oznake: Te lahko obravnavamo kot pametne brezkontaktne kartice in so tudi dovolj zmogljive za uporabo javne kriptografije.
  - "Polno-kvalificirane" (*Full-fledged*): Imajo notranjo zmogljivost izvajanja simetrične, javne kriptografije, računanja enosmernih kriptografskih funkcij,...
  - "Enostavne" (*Simple*): Imajo zmožnost računanja naključnih števil in računanja zgoščevalnih funkcij.
- Cenejše (*Low cost*) oznake: Te lahko obravnavamo kot RFID oznake.

- "Lahke" (*Lightweight*): Omogočajo operacije generiranja naključnih števil in enostavnih funkcij kot je računanje CRC napak, a ne izvajanja kriptografskih funkcij.
- "Ultralahke" (*Ultralightweight*): Omogoča računanje le osnovnih bitnih operacij XOR, OR, AND, ...

V nadaljevanju bomo prikazali dva pristopa, kako z ultralahkimi oznakami zagotoviti čimvečjo varnost. Svetovno najbolj razširjena standardizirana klasifikacija RFID oznak pa je EPCglobal (*Electronic Product Code*), ki na podoben način, kot smo mi zgoraj, razvrsti oznake v štiri razrede.

2.3 Primeri uporabe

RFID tehnologijo uporabljamo skoraj na vsakem koraku, a se tega sploh ne zavedamo. Najbolj se RFID sistemi uporabljajo v skladiščih. Oznake nam omogočajo enostavno sledenje izdelkov, optimalno poslovanje, evidenco zalog v vsakem trenutku. Veliko takšnih sistemov predvideva le označevanje palet, vendar nekatera globalna podjetja kot so Airbus, Boeing, Wal-Mart, označujejo tudi vsak izdelek posebej. RFID sisteme uporabljamo tudi v knjižnicah, mehaničnih delavnicah, bolnišnicah, trgovinah, v igralništvu, za pregled lastnega inventarja opreme, plačilo mestnega prometa (Urbana[1]), beleženje delovnega časa, plačevanje cestnine (uporaba ABC v Sloveniji), ... RFID oznake bodo prisotne tudi v naših osebnih izkaznicah in potnih listih. Nekateri so hoteli tudi, da bi bil vsak evropski bankovec označen s svojo oznako.

Podjetje Nokia razvija tehnologijo NFC (*Near Fiel Communication*), ki bi združila to tehnologijo z mobilno in omogočila hitro in varno elektronsko poslovanje ali celo nadomestila uradni osebni dokument.

3 Napadi na RFID čipe

Raziskovalec Mitrokotsa s sodelavci so predlagali 4-nivojsko razdelitev napadov na RFID čipe [6], ki si jo lahko ogledamo na sliki 3. To je le ena izmed oblik klasifikacij, ki so jih predlagale tudi druge organizacije. Že na prvi pogled nam vzbudi podobnost s TCP/IP modelom. Predlagani nivoji si sledijo od fizičnega nivoja, do najvišjega strateškega, kjer lahko padejo prav vsi varnostni sistemi, saj je v neposredno v igro vključen tudi človeški faktor.

3.1 Napadi na fizičnem nivoju

Napadi na tem nivoju so povezani z naravo brezžičnega delovanja čipa in direktnega kontakta napadalca s čipom.

3.1.1 Trajno uničenje čipa

Odstranitev čipa. Ker so čipi ponavadi pritrjeni na izdelke kot nalepke, je zelo enostavno odstraniti čip in ga uničiti. Dražji izdelki, predvsem obleke v trgovinah so označene s čipi tako, da potrebujemo posebne klešče, da lahko odstranimo čip.

[1] Urbana je kartica za brezgotovinsko plačevanje mestnega potniškega prometa, parkirnin in ostalih storitev v Mestni občini Ljubljana, ki jo je le ta uvedla konec leta 2009.

Razmerje cena-uporabnost	Logistični faktorji	Omejitve okolja	Strateški nivo
CRM sistemi	ERP sistemi	Strežniška in vmesna programska oprema	Aplikacijski nivo
Sledenje ISO standardom	Sledenje EPC Gen standardom	Lastniški RFID protokoli	Omrežni/transportni nivo
Radijske frekvence	Strojna oprema čitalca	RFID oznaka	Fizični nivo

Slika 3: Razdelitev RFID tehnologije na nivoje, kjer lahko napadamo sistem

Uničenje čipa. Zaradi slabe zaščite lahko včasih čip uničimo tudi nenamerno, če z njim pretrdo ravnamo. Poleg tega so tudi slabo odporni na statično elektriko, ki tudi lahko povzroči njegovo nedelovanje.

KILL ukaz. Podjetji *Auto – IDcenter* in *EPCglobal* sta za RFID čipe predlagali ukaz *KILL*, ki bi za vedno onesposobil čip. Pravilo pravi, da naj bi vsak proizvajalec čipov določil unikatno geslo za vsak čip posebej, ki se ga uporabi ob klicu tega ukaza. Ukaz je bil razvit predvsem zaradi varnosti, saj takšno funkcionalnost potrebujejo trgovci, ki želijo ob prodaji izdelka zagotoviti, da pripadajočega čipa ne bi nihče več uporabljal dalje. Tu je bila predvsem na udaru zasebnost. Če si izberemo primer, da kupec kupi obleko in po nakupu RFID čip v obleki še vedno deluje, ga lahko trgovec enostavno spremlja, kar pa seveda ni sprejemljivo. Seveda pa lahko z namernim uničevanjem čipov tudi sabotiramo RFID sistem.

Zaščita
Proti takšnim napadom se tehnološko težko zaščitimo. Poskrbeti moramo predvsem za nadzor dostopa do izdelkov, varovanje prostorov. Če pa to ni izvedljivo, lahko čipe bolje pritrdimo ali jih celo vgradimo v izdelke, če je to mogoče. Proti napadu s KILL ukazom se lahko zaščitimo z učinkovitim upravljanjem gesel, saj standard predvideva 32-bitno geslo za izvedbo ukaza.

3.1.2 Začasna onesposobitev čipa

Napadalec lahko komprimitira sistem tudi samo za določen čas. Kot vemo, se RFID skener in čip sporazumevata preko elektromagnetnega valovanja. To lastnost lahko izkoristimo tako, da ko napadalec čip prekrijemo z npr. aluminijasto folijo in tako ustvarimo Faradayevo kletko. Iz fizikalnih lastnosti sledi, da če imamo nek objekt v celoti obdan s kovino in hočemo z objektom komunicirati preko elektromagnetnega valovanja, to valovanje do objekta ne bo prišlo. (Več o tem pojavu si lahko preberete na Wikipediji [11]). Podobno kot zgoraj se lahko zgodi tudi, da nenamerno zaradi okoliščin (npr. ledu) začasno onemogočimo delovanje. Glede na namen onesposobitve čipov zato ločimo:

Pasivna interferenca. Do pasivne interference - prepletanja dveh ali več valovanj lahko prihaja v okoljih s prekomernim elektromagnetnim šumom. Tega pojava ne štejemo med napade, saj se to dogaja nenamenoma. Močan šum v svoji okolici lahko povzročajo elektromotorji, radijski oddajniki,

električne napeljave... Zaradi takšnega okolja je moteno komuniciranje z RFID čipom.

Aktivno motenje. Aktivno motenje pa je napad s pomočjo zgoraj opisane pasivne interference. Cilj napada je, da napadalec ugotovi frekvenco, na kateri deluje RFID skener in nato namenoma moti komunikacijo z RFID čipi s svojim signalom.

Zaščita
Zunanje motenje lahko onemogočimo, če prostore, kjer želimo izvajati svojo dejavnost obdamo z aluminijasto folijo. Na trgu pa obstajajo tudi posebne zidne barve in nalepke za okna, ki preprečujejo prehajanje sevanja.

3.1.3 Napad srednjega napadalca (*Man-in-the-middle attack*)

Napad srednjega napadalca je značilen napad na komunikacijske sisteme. Včasih ga imenujemo tudi napad z vmesnim možem ali napad z vrivanjem. Osnovna ideja je v tem, da med RFID skener in ciljni RFID čip vstavimo napravo, ki deluje kot posrednik. Tako, da imata obe strani občutek, da se pogovarjata neposredno. V sofisticiranih napadih se lahko uporabljata celo dve napravi - ena za skener in ena za čip. V praksi bi to pomenilo, da bi lahko v sistemu za plačevanje, ki uporablja RFID tehnologijo, preusmerili signal na drug čip. Praktičen primer je npr. napad na plačevanje prometa na Nizozemskem. Napadalci so terminalu za plačevanje namesto prave kartice prislonili napravo, ki je poiskala legalno kartico v bližini in omogočila komunikacijo med legalno kartico in terminalom. S takšnim početjem so pridobili dovolj podatkov, da so lahko kasneje klonirali kartico in jo uporabljali. Bi lahko podobno storili pri kartici Urbana?
Zaščita
Najbolj učinkovita zaščita proti napadom srednjega napadalca je uporaba identifikacijskih shem, PIN kode, biometričnih podatkov, torej kriptiranje povezave med skenerjem in čipom.

3.2 Napadi na omrežnem/transportnem nivoju

Na tem nivoju nas zanimajo napadi na komunikacijo po prenosnem mediju v RFID sistemih.

3.2.1 Napadi na RFID čipe

Kloniranje. Osnovni RFID čip vsebuje le malo bralnega pomnilnika v katerem je zapisana naša številka. V praksi se je izkazalo da repliciranje takšnih čipov ni drago. Izdelati moramo le svoj RFID čip, ki vsebuje to številko. Ranljivost kloniranja RFID oznak je bila odkrita tudi novih pri nemških potnih listih.

Prevara - *Spoofing.* Prevara je način kloniranja, kjer fizično ne izdelamo kopije RFID oznake, ampak se le poskusimo predstaviti kot veljaven RFID čip. Ta napad lahko klasificiramo tudi kot napad z vmesnim možem, ko le vzpostavimo komunkacijski kanal do legalne oznake. Lahko pa poznamo protokol komuniciranja in skrivnosti, ki jih potrebujemo, in se lahko tudi v tem primeru enostavno predstavimo.

Zaščita
Tema napadoma se lahko lahko izognemo z uporabo varnih identifikacijskih shem, a pri tem smo omejeni z računsko sposobnostjo RFID čipov. Dva varna identifikacijska protokola bomo predstavili v poglavju 4. Kloniranju se včasih lahko izognemo če imamo popolno podatkovno bazo o čipih in njihovih lokacijah.

3.2.2 Napadi na RFID skenerje

Lažno predstavljanje Gre za podoben napad kot je prevara iz prejšnjega razdelka. Ker vemo, da RFID čip načeloma ne preverja ali komunicira z legitimnim skenerjem, lahko identiteto skenerja enostavno ponaredimo in dostopamo do informacije na čipu ali pa jo celo spreminjamo.

Prisluškovanje To je ena najbolj resnih nevarnosti, saj lahko vsakdo, ki je dovolj blizu, prisluškuje komunikaciji med čipom in skenerjem. Poleg tega lahko tudi sklepa kdaj oddaja skener in kdaj čip, saj skener oddaja s bistveno večjo močjo. Prejete informacije lahko služijo za razbitje protokola ali pridobitev potrebnih skrivnosti.

Zaščita
Če uporabljamo RFID čipe le za identifikacijo, je rešitev enaka kot v prejšnjem razdelku. Sicer lahko uporabimo močnejše širfirne algoritme, kar poveča ceno sistema.

3.2.3 Napadi na komunikacijski protokol

Uporabni RFID sistemi so zasnovani tako, da so čitalci črtnih kod povezani z zalednim računalniškim omrežjem s podatkovnimi in aplikacijskimi strežniki. Če uspešno izvedemo napad na tem nivoju, dejansko pridobimo nadzor nad celotnim sistemom.
Zaščita
Praviloma so čitalci črtnih kod in strežniki dovolj zmogljivi, da lahko med seboj komunicirajo s pomočjo javne kriptografije ali s simetrično, saj imamo zaprt sistem, zato je uporaba teh protokolov obvezna. Uporaba zadnjih popravkov programske opreme in pravilna nastavitev operacijskih sistemov ter omrežja. Nujno je stalno vzdrževanje sistema.

3.3 Napadi na aplikacijskem nivoju

Ti napadi se usmerjajo na aplikacije in na povezavo med uporabnikom in RFID čipom.

Neavtorizirano branje čipa. Če RFID čipi v sistemu ne uporabljajo nobenih avtentikacijskih protokolov, jih lahko preberemo brez sledi.

Modifikacija vsebine čipa. Če RFID čip ne uporablja nobenih varnostnih mehanizmov in vsebuje bralno-pisalni pomnilnik, ga lahko ravno tako brez sledi preberemo in tudi spremenimo.

Napad na vmesno programsko opremo - *middleware.* Ta tip napadov je povsem odvisen od uporabljane programske opreme. V zgodovini poznamo dva najbolj znana napada: "Buffer overflow attack" in "SQL injection". Predvsem slednji je zaradi malomarnosti še vedno ena najbolj pogostih ranljivosti v računalniških sistemih.

Zaščita
Tako strojno kot programsko opremo moramo graditi varno po uveljavljenih metodologijah in sprotnim testiranjem. Za boljšo varnost je pametno pregledovati izvorno kodo in pravilno implementirati varnostne algoritme.

3.4 Napadi na strateškem nivoju

Vzrok za uspešnost teh napadov je predvsem napaka v varnostni politiki organizacije in naivnosti zaposlenih. Zelo pogosto se pojavljajo kot del večnivojskih napadov, saj v na tem nivoju ponavadi enostavno pridemo do dela potrebnih informacij, ki nam olajšajo napad. Nekaj klasifikacij te vrste napadov bi bili **(konkurenčno) vohunstvo, socialni inženiring, grožnje zasebnosti, usmerjen varnostni napad na znano varnostno luknjo v sistemu, ki ga uporablja organizacija...**
Zaščita
Proti tem napadom se lahko upiramo s čimvečjo osveščenostjo ljudi, ki morajo natanko vedeti komu lahko zaupajo kakšno informacijo, da jih pravi administrator ne bo nikoli vprašal za geslo... Bolj, ko bodo ljudje vedeli za možnosti tovrstnih napadov, bolj bodo nanje pripravljeni.

3.5 Večnivojski napadi

Veliko napadov se ne omejuje samo na posamezen nivo. Našteli bomo nekaj napadov, ki uporabljajo več nivojev.

3.5.1 Prikrivanje kanala

Pri prikrivanju kanala napadalec zapiše želene podatke na nezaseden prostor na RFID oznaki. Če želimo zapisati veliko podatkov, lahko uporabimo nezaseden prostor na več oznakah. Praktičen primer uporabe bi bil, če na RFID v človeku, kamor shranimo zasebne informacije človeka in jih lahko kasneje tudi preberemo. Če bi imeli množico ljudi in bi naročili umor človeka, ki je v množici, bi ga lahko predhodno označili in nato hitro našli ter napadli. V oskrbovalnih verigah izdelki potujejo po vnaprej predpisanih poteh. To bi lahko izkoristile kriminalne združbe za prikrito komuniciranje.
Zaščita
Zaenkrat raziskovalci še niso odkrili, kako bi se zaščitili pred to nevarnostjo. Uporabno bi bilo neuporabljen pomnilnik čipa ob določenih intervalih brisati, a bi čipi morali imeti še napajanje. Najbolje pa je, da čipi vsebujejo ravno toliko pomnilnika, kot ga potrebujejo. Protokola, ki jih bomo predstavili na naslednjem poglavju ustrezata tej lastnosti.

3.5.2 Napad z onemogočanjem storitve (*DoS attack*)

Kot že ime pove, je cilj napada, da za določen čas onemogočimo delovanje sistema. Napadalec poskusi npr. uporabiti ukaz LOCK na RFID čipih, da začasno onemogoči komuniciranje s čipi ostalim uporabnikom.
Zaščita
Tudi tu je vprašanje zaščite še odprto, saj se problem te vrste napadov pojavlja v vseh vrstah omrežij. Npr. ukaza LOCK ne poznajo vsi čipi, lahko je tudi zaščiteno z geslom, kot KILL ukaz.

3.5.3 Analiza prometa

S pomočjo prisluškovanja beležimo pogovore med RFID čipi in skenerji ter jih nato poskušamo razbiti. Ko imamo več pogovorov, iščemo podobnosti z namenom, da odkrijemo skrivnosti na čipu ali skrivnostni kriptografski algoritem. Takšne pogovore lahko tudi uporabimo ob napadih prihodnje varnosti (*Forward security*), pri katerih iščemo kakšne skrivnosti je v določenem trenutku v preteklosti vsebovala RFID oznaka.

3.5.4 Napad stranskega kanala - *Side Channel Attack*

Napadalec meri spremembe toka na čipu, z namenom da ugotovi katere operacije se izvajajo glede na parametre. Več o tem napadu si lahko preberete v seminarski nalogi Moškona [7].
Zaščita
Poskusimo omejiti elektromagnetno sevanje, da ne doseže napadalca, a s tem hkrati uporabnikom skrajšamo razdaljo delovanja.

3.5.5 Ponovljeni *(Replay)* napadi

Napadalec poskusi izkoristiti slabosti identifikacijskega protokola izziv-odgovor z namenom, da posname enega od pogovorov s čipom in ga kasneje predvaja, da bi si pridobil dostop do njega.
Zaščita Uporaba časovnih žigov, enkratnih gesel, uporaba varnih identifikacijskih protokolov, ...

4 UMAP Protokoli

Leta 2006 je Peris-Lopez s sodelavci predstavil družino ultralahkih vzajemno-avtentikacijskih protokolov (UMAP - Ultralightweight Mutual Authentication Protocols), začenši z M^{2}AP [8], ki so mu isto leto sledili še EMAP in LMAP.

Vsaka oznaka vsebuje ID število. Predlagani protokoli predvidevajo, da so oznake sposobne izvajanja le enostavnih bitnih operacij AND, OR, XOR, krožne leve rotacije za y bitov in seštevanja po modulu 2^m.

Cilj tovrstnih protokolov je, da zagotavljajo varno avtentikacijo med računsko šibkim členom na eni strani in močnejšim na drugi strani. V ta okvir se lepo ujamejo RFID sistemi, saj želimo čimcenejše RFID oznake v sistemu. Poleg tega tudi zunanji prisluškovalci ne bi smeli ob komunikaciji vedeti s katero iznako komuniciramo, če so prisluškovali tudi prej. Anonimnost oznake se zagotavlja s shranjenim psevdonimom, ki se sčasoma spreminja. Poleg tega morajo pri javni komunikaciji prikriti ID število, ki ga vsebuje oznaka.

4.1 SASI protokol

Hung-Yu Chien je leta 2007 predlagal protokol SASI (Strong Authentication Strong Integrity) za cenene RFID oznake, ki ga je predstavil v [9]. Varnostna analiza protokola pa je predstavljena v [3] in [2], ki jo bomo tudi mi opisali. Ker je bil eden prvih resnih protokolov v tem segmentu, si bomo v nadaljevanju pogledali njegovo delovanje.

Protokol predvideva, da imamo v sistemu RFID oznake, RFID čitalce in zaledni sistem s podatkovno bazo. Privzeto je, da je komunikacija med čitalcem in bazo varna. To lahko omogočimo z uporabo močnih kriptografskih algoritmov, saj so čitalec in strežniki računsko dovolj zmogljivi. Vsaka oznaka hrani ID številko in dva zapisa trojk (IDS, $K1, K2$). V zalednem sistemu je shranjena ID številka in zadnja posodobljena trojka. Prvi zapis trojke predstavlja stare vrednosti, drugi pa potencialne nove vrednosti, ki se bodo uporabile ob naslednji komunikaciji. Besede so dolge 96 bitov, oznaka pa mora imeti dovolj prostora, da shrani 7 takšnih besed (1 beseda za ID in po 3 za vsako trojko).

Protokol sestoji iz treh faz:

1. Faza identifikacije oznake
2. Faza vzajemne avtentikacije
3. Faza posodabljanja ključev

Potek protokola si lahko ogledate tudi v tabeli 2.

1. Faza identifikacije oznake

Čitalec najprej nagovori oznako s sporočilom *hello*. Oznaka mu odgovori z naslednjim potencialnim IDS **psevdonimom**. Če ga čitalec ne najde v bazi, poskusi z nagovorom še enkrat in oznaka mu odgovori z IDS psevdonimom, ki je bil uporabljen nazadnje. Čitalec glede na posredovan psevdonim iz interne baze pridobi ID in ključa K1, K2.

2. Faza vzajemne avtentikacije

Čitalec zgenerira naključni števili $n1$ in $n2$ in s pomočjo njih izračuna števila A,B in C.

$$\begin{aligned} A &= \text{IDS} \oplus K1 \oplus n1 \\ B &= (\text{IDS} \vee K2) + n2 \\ \overline{K1} &= \text{Rot}(K1 \oplus n2, K1) \\ \overline{K2} &= \text{Rot}(K2 \oplus n1, K2) \\ C &= (K1 \oplus \overline{K2}) + (\overline{K1} \oplus K2) \end{aligned}$$

Vsa tri števila pošlje oznaki, ki iz njih izračuna naključni števili $n1$, $n2$.

$$\begin{aligned} n1 &= A \oplus \text{IDS} \oplus K1 \\ n2 &= B - (\text{IDS} \vee K2) \end{aligned}$$

Oznaka izračuna začasna ključa $\overline{K1}$, $\overline{K2}$ in $\overline{C}$.

$$\begin{aligned} \overline{K1} &= \text{Rot}(K1 \oplus n2, K1) \\ \overline{K2} &= \text{Rot}(K2 \oplus n1, K2) \\ \overline{C} &= (K1 \oplus \overline{K2}) + (\overline{K1} \oplus K2) \end{aligned} \tag{1}$$

Čitalec		Oznaka
1. Faza identifikacije oznake		
	hello →	
	← IDS	
2. Faza vzajemne avtentikacije		
$A = \text{IDS} \oplus K1 \oplus n1)$		
$B = (\text{IDS} \vee K2) + n2)$		
$\overline{K1} = \text{Rot}(K1 \oplus n2, K1))$		
$\overline{K2} = \text{Rot}(K2 \oplus n1, K2))$		
$C = (K1 \oplus \overline{K2}) + (\overline{K1} \oplus K2)$		
	$A\|\|B\|\|C$ →	
		$n1 = A \oplus \text{IDS} \oplus K1$
		$n2 = B - (\text{IDS} \vee K2)$
		$\overline{K1} = \text{Rot}(K1 \oplus n2, K1)$
		$\overline{K2} = \text{Rot}(K2 \oplus n1, K2)$
		$\overline{C} = (K1 \oplus \overline{K2}) + (\overline{K1} \oplus K2)$
		če $\overline{C} = C$, nadaljuj, sicer prekini.
		$D = (\overline{K2} + \text{ID}) \oplus ((K1 \oplus K2) \vee \overline{K1})$
3. Faza posodabljanja ključev		
		$\text{IDS}_{\text{old}} = \text{IDS}$
		$K1_{\text{old}} = K1$
		$K2_{\text{old}} = K2$
		$\text{IDS} = (\text{IDS} + \text{ID}) \oplus (n2 \oplus \overline{K1})$
		$K1 = \overline{K1}$
		$K2 = \overline{K2}$
	← D	
$\overline{D} = (\overline{K2} + \text{ID}) \oplus ((K1 \oplus K2) \vee \overline{K1})$		
če $\overline{D} = D$, nadaljuj, sicer prekini.		
$\text{IDS} = (\text{IDS} + \text{ID}) \oplus (n2 \oplus \overline{K1})$		
$K1 = \overline{K1}$		
$K2 = \overline{K2}$		

Tabela 2: Prikaz delovanja SASI protokola

Če se vrednost C ujema s $\overline{C}$, odgovori z D in nadaljuje z naslednjo fazo.

$$D = (\overline{K2} + \text{ID}) \oplus ((K1 \oplus K2) \vee \overline{K1})$$

3. Faza posodabljanja ključev

Ko čitalec prejme vrednost D, jo primerja z $\overline{D}$:

$$\overline{D} = (\overline{K2} + \text{ID}) \oplus ((K1 \oplus K2) \vee \overline{K1}) \quad (2)$$

Izračuna naslednji psevdonim IDS, ki mu ga bo ob prihodnji interakciji vrnila oznaka in posodobi ključa z novimi vrednostmi.

$$\begin{aligned} \text{IDS} &= (\text{IDS} + \text{ID}) \oplus (n2 \oplus \overline{K1}) \\ K1 &= \overline{K1} \\ K2 &= \overline{K2} \end{aligned}$$

Oznaka trojko, ki jo je uporabila v trenutni komunikaciji, označi kot staro in ustvari novo trojko z začasnima ključema in novoizračunanim psevdonimom. Psevdonim IDS te nove trojke se bo uporabil ob naslednji komunikaciji.

$$\begin{aligned} \text{IDS}_{\text{old}} &= \text{IDS} \\ K1_{\text{old}} &= K1 \\ K2_{\text{old}} &= K2 \end{aligned}$$

$$\begin{aligned} \text{IDS} &= (\text{IDS} + \text{ID}) \oplus (n2 \oplus \overline{K1}) \\ K1 &= \overline{K1} \\ K2 &= \overline{K2} \end{aligned}$$

Ob uspešni posodobitvi ključev nam shema zagotavlja tudi potrditev novih ključev $\overline{K1}$ in $\overline{K2}$, da s tem onemogočimo desinhronizacijske napade. Več o njih tudi v 4.1.4.

4.1.1 Pravilnost delovanja

Naloga protokola je, da se lahko RFID oznaka, enolično glede na ID številko, predstavi čitalcu. To je očitno, saj tako čitalec kot oznaka posedujeta iste ključe, iz katerih izračunata vrednosti C in D za primerjanje. Za kreiranje novih ključev, potrebujemo naključni števili $n1$ in $n2$, ki jih pozna čitalec, oznaka jih pa izračuna iz vrednosti A in B. Torej lahko trdimo, da če oznaka in čitalec posedujeta isto trojico skrivnosti, se lahko oznaka uspešno predstavi.

V nadaljevanju si bomo pogledali nekaj napadov in kako varen pred njimi je SASI protokol.

4.1.2 Napad s ponavljanjem (*Replay attack*)

Napad s ponavljanjem lahko naredimo na dveh mestih. V obeh primerih si lahko izberemo eno izmed obstoječih oznak v sistemu in se predstavimo z njenim starim IDS. To pomeni, da bo zelo verjetno moral čitalec še enkrat poslati ukaz *hello* oznaki, če predvidevamo, da je bila posodobitev ključev z originalno oznako uspešno izvedena.

- Lahko se predstavljamo kot oznaka in čitalcu pošljemo vrednost D, ki jo je nazadnje poslala oznaka. To ne bo uspelo, ker je vrednost D odvisna od novih ključev, ki jih izračunamo s pomočjo naključnih števil $n1$ in $n2$, ki si jih vsakič posebej izbere čitalec. Četudi bi čitalec deloval napačno in bi vsakokrat izbral isti naključni števili, nova ključa zaradi funkcije rotacije ne bi bila enaka starim in zato posledično tudi D ne. Verjetnost, da bi napad uspel je zelo majhna. Ena možnost, da bosta stara in nova ključa enaka, je, da sta $n1$ in $n2$ enaka 0 in $K1$ ter $K2$ enaka vrednosti $2^{97} - 1$.

Verjetnost, da to uspe, je zelo majhna, saj morajo biti vsa štiri 96-bitna števila pravilno izbrana. Lahko pa tudi popravimo generator naključnih števil, da nikoli ne vrne 0.

- Uspešen napad bi bil, da bi komprimitirali tako oznako kot čitalec. Poslali bi $A||BC||$ in odgovorili z D, vendar s tem ne bi niti spremenili notranjega stanja sistema niti izvedeli nobene skrivnosti.

4.1.3 Napad srednjega napadalca (*Man-in-the-middle*)

Napad "vmesnega moža" na tej shemi ne deluje. Tipično bi želeli iz poslanih $A||B||C$ ali D pridobiti skrivnosti, ki bi nam omogočale lažno predstavljanje.

- Če spreminjamo IDS, nas sistem gotovo ne bo zaznal kot oznako, s katero želi komunicirati ali pa bo hotel nadaljevati komunikacijo z idejo, da se mu je javila neka druga oznaka, ki se nahaja v sistemu in smo nakljućno uganili njen IDS. To nam nič ne pomaga, saj iz nadaljne komunikacije ne moremo ugotoviti na katero oznako se nanaša, niti ne moremo pravilno nadaljevati komunikacije, saj ne poznamo nobene skrivnosti oznake za računanje novih ključev.
- Če spreminjamo ABC, bo oznaka ugotovila neujemanje s C in končala protokol. Obstaja pa verjetnost, da nam bo uspelo. Takšne napade opisujemo v 4.1.4.
- Če spreminjamo število D, čitalec ne bo potrdil oznake.

Kljub spreminjanju teh števil, ne bomo mogli priti do nobene skrivnosti.

4.1.4 Napad z onemogočanjem storitve (*DoS, Disclosure Attack*)

Napad z onemogočanjem storitve je v našem primeru namenjen komprimitiranju sistema na način, da čitalec ne bo več prepoznal določenih oznak. Stanju, ko čitalec ne prepozna določene oznake, imenujemo desinhronizacijsko stanje in tudi napade imenujemo kar desinhronizacijski napadi (*de-synchronization attack*).

Pri SASI protokolu lahko poskusimo izvesti *DoS* napad na dva načina:

- Lahko prestrežemo D, da ga čitalec ne dobi. V tem primeru se bodo ključi na oznaki posodobili, pri čitalcu pa ne. Zaradi tega imamo na oznaki shranjeno dvoje ključev in se bodo ob naslednji komunikaciji uporabili starejši. Če D prestrežemo več kot enkrat, vseeno oznake ne bomo spravili v desinhronizacijsko stanje, saj bodo stari ključi, ki jih vsebuje tudi zaledni sistem vedno ostajali na drugem mestu, samo novi ključi se bodo menjavali.
- Lahko poskušamo pripraviti sistem do tega, da bosta oznaka in čitalec uporabila različne vrednosti $n1$ in $n2$ za posodobitev ključev, brez da bi se tega zavedala. Čitalec si bo števili sam zgeneriral, zato mu jih ne moremo spremeniti. Oznaki pa lahko pošljemo malenkost spremenjena sporočila, tako da se bodo spremembe po protokolu izničile, izračunani $n1$ in $n2$ pa bosta drugačni kot jih ima čitalec. Tako bosta imeli obe strani različen IDS in se naslednjič ne bosta mogli spoznati. Za popolno rešitev

tega problema predlagam, da tudi zaledni sistem hrani dve trojici kot tudi oznaka in se nato v najslabšem primeru poskusi avtenticirati 4x namesto 2x.

V nadaljevanju bomo prikazali tri napade iz druge točke.

Spreminjanje sporočil A, C in D. Napadalec najprej prisluškuje protokolu in nato zamenja $A||B||C$ z $A'||B'||C'$:

$$\begin{aligned} I_i &= 2^i \\ A' &= A \oplus I_0 \\ C' &= C \oplus I_0 \\ D' &= D \oplus I_0 \end{aligned}$$

Oznaka mora to novo trojico sprejeti. Recimo, da je $K2 \equiv 0 \pmod{n}$ (velja z verjetnostjo $1/n$) in da je najlažji bit $K2 \oplus \overline{K1}$ enak 0 (velja z verjetnostjo $1/2$). Torej bo veljalo $\mathrm{Rot}(K2 \oplus X, K2) = K2 \oplus X$. To velja z verjetnostjo vsaj $1/(2n)$, s katero bo oznaka trojico tudi sprejela.

$$\begin{aligned} C' &= [(K1 \oplus \overline{K2}) + (\overline{K1} \oplus K2)] \oplus I_0 \\ &= (K1 \oplus \overline{K2} \oplus I_0) + (\overline{K1} \oplus K2) \\ n1' &= A \oplus I_0 \oplus \mathrm{IDS} \oplus K1 \\ &= (\mathrm{IDS} \oplus K1 \oplus n1) \oplus I_0 \oplus \mathrm{IDS} \oplus K1 \\ &= n1 \oplus I_0 \\ n2' &= B - (\mathrm{IDS} \vee K2) \\ &= (\mathrm{IDS} \vee K2) + n2 - (\mathrm{IDS} \vee K2) \\ &= n2 \\ \overline{K1'} &= \mathrm{Rot}(K1 \oplus n2, K1) = \overline{K1} \\ \overline{K2'} &= \mathrm{Rot}(K2 \oplus n1', K2) = K2 \oplus n1' = \overline{K2} \oplus I_0 \\ \overline{C} &= (K1 \oplus \overline{K2} \oplus I_0) + (\overline{K1} \oplus K2) \\ &= C' \end{aligned}$$

Sedaj mora uspeti še, da čitalec sprejme D, ki mu ga bo poslala oznaka. Da bo to uspelo, mora biti najlažji bit ID enak 0, kar lahko trdimo z verjetnostjo $1/2$.

$$\begin{aligned} D' &= (\overline{K2'} + \mathrm{ID}) \oplus ((K1 \oplus K2) \vee \overline{K1}) \oplus I_0 \\ &= (\overline{K2} \oplus I_0 + \mathrm{ID}) \oplus ((K1 \oplus K2) \vee \overline{K1}) \oplus I_0 \\ &= ((\overline{K2} \oplus I_0 + \mathrm{ID}) \oplus I_0) \oplus ((K1 \oplus K2) \vee \overline{K1}) \\ &= (\overline{K2} + \mathrm{ID}) \oplus ((K1 \oplus K2) \vee \overline{K1}) \\ &= \overline{D} \end{aligned} \tag{3}$$

Ko čitalec sprejme vrednost, posodobi ključe z naključnimi vrednostmi $(n1, n2)$, oznaka pa z $(n1 \oplus I_0, n2)$. Očitno imata obe strani različni števili, zato bosta različno posodobili vrednosti in zato bo oznaka v desinhronizacijskem stanju. Uspešnost tega napada je $1/(4n) = 1/384$. Rešitev bi bila, da tudi zaledni sistem hrani zadnji dve trojici (IDS, $K1$, $K2$).

Spreminjanje sporočil B in C. Napadalec najprej prisluškuje protokolu in spremeni $A||B||C$ v $A||B'||C'$:

$$\begin{aligned} B' &= B+1 \\ C' &= C \oplus I_0 \end{aligned}$$

Predvidevamo, da je $K1 \equiv 0 \pmod{n}$ in najlažji bit vrednosti $K1 \oplus \overline{K2}$ ter $n2$ enak 0. Torej bo $\mathrm{Rot}(K1 \oplus X, K1) = K1 \oplus X$. Verjetnost tega dogodka je $1/(4n)$. Oznaka preveri veljavnost prejetih števil:

$$\begin{aligned} C' &= [(K1 \oplus \overline{K2}) + (\overline{K1} \oplus K2)] \oplus I_0 \\ &= (K1 \oplus \overline{K2}) + (\overline{K1} \oplus K2 \oplus I_0) \\ n1' &= A \oplus \mathrm{IDS} \oplus K1 \\ &= (\mathrm{IDS} \oplus K1 \oplus n1) \oplus \mathrm{IDS} \oplus K1 \\ &= n1 \\ n2' &= B+1-(\mathrm{IDS} \vee K2) \\ &= ((\mathrm{IDS} \vee K2) + n2) + 1 - (\mathrm{IDS} \vee K2) \\ &= ((\mathrm{IDS} \vee K2) + (n2 \oplus I_0)) - (\mathrm{IDS} \vee K2) \\ &= n2 \oplus I_0 \\ \overline{K1'} &= K1 \oplus n2' = \overline{K1} \oplus I_0 \\ \overline{K2'} &= \overline{K2} \\ \overline{C} &= (K1 \oplus \overline{K2'}) + (\overline{K1'} \oplus K2) \\ &= (K1 \oplus \overline{K2}) + (\overline{K1} \oplus I_0 \oplus K2) \\ &= C' \end{aligned}$$

Ob zgornjih predpostavkah, oznaka uspešno sprejme sporočilo. Sedaj mora veljati še, da je najlažji bit $K1 \oplus K2$ enak 1, saj bo čitalec le v tem primeru sprejel vrednost D. Verjetnost, da to velja, je enaka 1/2.

$$\begin{aligned} D &= (\overline{K2} + \mathrm{ID}) \oplus ((K1 \oplus K2) \vee (\overline{K1} \oplus I_0)) \\ &= (\overline{K2} + \mathrm{ID}) \oplus ((K1 \oplus K2) \vee \overline{K1}) \\ &= \overline{D} \end{aligned} \tag{4}$$

Enako kot v prejšnjem napadu, čitalec in oznaka posodobita ključe z različnimi vrednostmi. Čitalec z $(n1, n2)$, oznaka pa z $(n1, n2 \oplus I_0)$. Očitno zgenerirata različne nove skrivnosti. Verjetnost, da ta napad uspe je $1/(8n) = 1/768$.

Spreminjanje A in ugibanje C. Zgornja dva napada sta bila napada vmesnega moža. V tem primeru bomo skušali oznako pretentati, da imamo pravi čitalec. Če nam bo uspelo, bo oznaka v desinhronizacijskem stanju.

Najprej prisluškujemo prometu med oznako in pravim čitalcem. Zapomnimo si zadnje sporočilo $A||B||C$, ki ga je oznaki poslal legalen čitalec. Oznaka in čitalec bosta oba vsebovala iste nove ključe $K1, K2$. Oznaka bo hranila še starejši par ključev $K1_{\mathrm{old}}, K2_{\mathrm{old}}$. Cilj napada je, da oznaka misli, da v sistemu nimamo zadnjega IDS in zato s pomočjo starih ključev izračuna nove, ki se seveda ne ujemajo s pravim zalednim sistemom.

Napadalec spremeni A v A' in zgenerira vse možne vrednosti C_{1i} in C_{0i} za vsak $i \in \{0, 1, 2, ..., n-1\}$.

$$\begin{aligned} A' &= A \oplus I_0 \\ C_{0i} &= C + I_i \\ C_{1i} &= C - I_i \end{aligned}$$

Napadalec pošlje *hello* oznaki, ki odgovori z IDS. Napadalec pošlje naključno število, da oznaka odgovori ugotovi neujemanje in ko še enkrat pošlje *hello*, odgovori z IDS_{old}. Napadalec pošlje vrednosti $A'||B||C_{ji}$ in če oznaka odgovori z D, je napad uspel, sicer nadaljujemo. Pokažimo še, da napad zares deluje.

Algorithm 1 Algoritem napada za čitalec

```
for i = 0 to n-1 do
  for i = 0 to 1 do
    send hello
    recieve IDS
    send random value
    send hello
    recieve IDS_old
    send A'||B||C_ji
    if recieved_anything() then
      return success
    end if
  end for
end for
return unsuccess
```

Predpostavimo, da velja $i \equiv K2 \pmod{n}$, potem gotovo oznaka sprejme $A'||B||C_{0i}$ ali $A'||B||C_{1i}$. Če je i-ti najlažji bit $K1 \oplus \overline{K2}$ enak 0, potem je $\overline{C}$ enaka C_{0i}, sicer C_{1i}.

$$\begin{aligned} n1' &= n1 \oplus I_0 \\ n2' &= n2 \\ \overline{K1'} &= \text{Rot}(K1 \oplus n2, K1) = \overline{K1} \\ \overline{K2'} &= \text{Rot}(K2 \oplus n1 \oplus I_0, i) \\ &= \text{Rot}(K2 \oplus n1, i) \oplus \text{Rot}(I_0, i) \\ &= \overline{K2} \oplus I_i \\ \overline{C} &= (K1 \oplus \overline{K2} \oplus I_i) + (\overline{K1} \oplus K2) \\ C_{0i} &= [(K1 \oplus \overline{K2}) + (\overline{K1} \oplus K2)] + I_i \\ C_{1i} &= [(K1 \oplus \overline{K2}) + (\overline{K1} \oplus K2)] - I_i \end{aligned}$$

Ko bo oznaka sprejela novo vrednost, bo posodobila ključe z $(n1 \oplus I_0, n2)$, zaledni sistem pa bo imel drugačne ključe. Torej je oznaka desinhronizirana. Tudi v tem primeru bi lahko rešili sistem tako, da bi si še zaledni sistem zabeležil IDS_{old}. Da lahko napad izvedemo, potrebujemo maksimalno $2n$ ugibanj.

4.1.5 Odpornost sledenju in prihodnji varnosti (*Tracking, Forward security*)

Sistem zagotavlja prihodnjo varnost, če s trenutnimi skritimi podatki, ki jih vsebuje ne moremo priti do skritih podatkov, ki jih je vseboval v teku delovanja. Prikazali bomo napad na SASI protokol, ki dokazuje, da le ta ne omogoča te zaščite.

Napadalec lahko beleži vso komunikacijo med neko oznako in čitalcem in si gradi seznam zapisov (IDS_i, A_i, B_i, C_i, D_i). Recimo, da imamo N takšnih zapisov. Nad oznako moramo izvesti še fizični napad in s tem pridobimo vrednosti ID, IDS_m, $K1_m$, $K2_m$, IDS_{m+1}, $K1_{m+1}$ in $K2_{m+1}$.

Sedaj lahko napišemo algoritem, ki nam bo od stanja (ID, IDS_m, $K1_m$, $K2_m$) našel poljubno stanje (ID, IDS_{m-i}, $K1_{m-i}$, $K2_{m-i}$) za vsak i, ko velja $(m-i) > 0$. Tako bomo dobili skrivnost na poljubnem koraku in s tem dokazali, da SASI protokol ne zagotavlja prihodnje varnosti.

Algorithm 2 Algoritem za iskanje prejšnjih skrivnosti

```
for i = 0 to n-1 do
  get i-th record (IDS_i, A_i, B_i, C_i, D_i) from database
  n2 = (IDS_m ⊕ (IDS_i + ID)) ⊕ K1_m
  for j = 0 to n-1 do
    K1_{m-1} = Rot(K1_m, j) ⊕ n2
    if K1_m = Rot(K1_{m-1} ⊕ n2, K1_{m-1}) then
      K2_{m-1} = (C_i − (K1_{m-1} ⊕ K2_m)) ⊕ K1_m
      n1 = A_i ⊕ IDS_i ⊕ K1_{m-1}
      B' = (IDS_i ∨ K2_{m-1}) + n2
      D' = (K2_m + ID) ⊕ ((K1_{m-1} ⊕ K2_{m-1}) ∨ K1_m)
      if B' = B_i and D' = D_i then
        return (i, IDS_i, K1_{m-1}, K2_{m-1})
      end if
    end if
  end for
end for
```

Opis postopka

Za vsak zapis po vrsti, ki ga imamo v svoji bazi, poskusimo izračunati, če se ujema z m-tim. Najprej iz obeh izračunamo $n2$ in nato poskušamo za vse možne zamike izačunati K_{m-1}. Pravilnost izračunane vrednosti preverjamo, če lahko na podlagi K_{m-1} izračunamo pravi K_m. Če nam je uspelo, poskusimo sedaj izračunati še $K2_{m-1}$, $n1$, B', D', ki so vrednosti, s katerimi je oznaka računala v i-tem koraku. V zadnjem koraku preverimo, če se B' ujema z B_i in D' z D_i. Če se, smo uspešno odkrili skrivnosti prejšnje komunikacije, sicer ponovimo isti postopek za naslednji zapis v bazi.

4.1.6 Varnostna analiza

Pri analizi UMAP protokolov je bilo ugotovljeno, da ni varno v protokolu uporabljati le besedne funkcije. Zaradi tega so snovalci SASI protokola uporabili tudi levo rotacijo, ki ni besedna. Kljub temu smo ugotovili, da ima protokol še vedno nekaj pomanjkljivosti. Še več, nekateri raziskovalci so ugotovili, da

je možno izračunati ID le na podlagi prisluškovanja. Naštejmo nekaj glavnih slabosti:

- Drugi operand pri računanju IDS izračunamo kot $n2 \oplus \overline{K1}$. To poslabša statistične lastnosti, saj je tudi $\overline{K1}$ funkcija $n2$.
- Operacija za posodabljanje ključev je distributivna, kar lahko tudi izkoristimo pri napadu. Primer: $\overline{K1} = \text{Rot}(K1 \oplus n2, K1) = \text{Rot}(K1, K1) \oplus \text{Rot}(n2, K1)$
- Bitnih operacij AND in OR ne bi smeli uporabiti za izračunavanje javnih sporočil, saj vrnejo močno pristrano vrednost. Vemo, da četudi uporabimo operacijo AND (OR) nad naključnimi števili, je verjetnost, da je naš rezultat enak 0 (1) 0,75%. To je tudi ena izmed ključnih slabosti algoritma. Zaradi tega bi morala izmenjana sporočila z oznako izgledati čimbolj naključna. Posledica te slabosti je tudi, da lahko vrednost $n2$ dobro aproksimiramo že, če izračunamo $B - 1$.

4.2 Gossamer protokol

Zaradi varnostnih problemov SASI protokola, je leto kasneje Peris-Lopez Pedro objavil protokol Gossamer[2], ki je predstavljen v tabeli 3. Ta avtor je pred SASI protokolom že predlagal serijo UMAP protokolov. Gossamer protokol je bil grajen na podlagi sheme SASI protokola in je poskušal odpraviti njegove težave, da bi se lahko uporabljal v realnih RFID sistemih.

Enako kot SASI protokol, imamo v sistemu oznake in čitalce, ki so varno povezani z zaledním sistemom. Tako zaledni sistem kot oznaka vsebujeta ID, in trojko (IDS,$K1$,$K2$). Oznaka vsebuje dve trojki za preprečitev desinhronizacijskih napadov.

Operacije, ki jih protokol zahteva, so XOR, seštevanje po modulu 2^n, leva rotacija x za y mest in funkcijo *MixBits*, ki je opisana v algoritmu 3. Metoda *MixBits* naj bi signifikantno povečala varnost. Uporablja enostavne nelinearne funkcije kot je seštevanje in bitni zamik v desno.

Algorithm 3 MixBits(X,Y)

```
Z = X
for i = 0 to 31 do
    Z = (Z >> 1) + Z + Z + Y
end for
return Z
```

Protokol enako kot SASI sledi trem fazam: identifikacija oznake, vzajemna avtentikacija in posodabljanje ključev. V nadaljevanju bomo vse tri korake natančno spoznali.

1. Identifikacija oznake

Čitalec pozdravi oznako s sporočilom *hello*, ki mu odgovori z IDS. Če zaledni sistem najde IDS v svoji bazi, se prične naslednja faza, sicer čitalec še enkrat

[2]Gossamer iz ang.-slo. slovarja: 1. samostalnik: babje leto; zelo tenka tkanina; gaza; ameriško: tenek dežni plašč; 2. pridevnik: lahek, tenek, pajčevinast;

Čitalec		Oznaka
1. Faza identifikacije oznake		
	$\xrightarrow{hello}$	
	$\xleftarrow{\text{IDS}}$	
2. Faza vzajemne avtentikacije		
$A = \text{Rot}(\text{Rot}(\text{IDS} + K1 + \pi + n1, K2) + K1, K1)$		
$B = \text{Rot}(\text{Rot}(\text{IDS} + K2 + \pi + n2, K1) + K2, K2)$		
$n3 = MixBits(n1, n2)$		
$\overline{K1} = \text{Rot}(\text{Rot}(n2 + K1 + \pi + n3, n2) + K2 \oplus n3, n1) \oplus n3$		
$\overline{K2} = \text{Rot}(\text{Rot}(n1 + K2 + \pi + n3, n1) + K1 + n3, n2) + n3$		
$n1' = MixBits(n3, n2)$		
$C = \text{Rot}(\text{Rot}(n3 + \overline{K1} + \pi + n1', n3) + \overline{K2} \oplus n1', n2) \oplus n1'$		
	$\xrightarrow{A\|\|B\|\|C}$	
		$n1 = \text{Rot}(\text{Rot}(A, n - K1) - K1, n - K2) - \text{IDS} - K1 - \pi$
		$n2 = \text{Rot}(\text{Rot}(A, n - K2) - K2, n - K1) - \text{IDS} - K2 - \pi$
		$n3 = MixBits(n1, n2)$
		$\overline{K1} = \text{Rot}(\text{Rot}(n2 + K1 + \pi + n3, n2) + K2 \oplus n3, n1) \oplus n3$
		$\overline{K2} = \text{Rot}(\text{Rot}(n1 + K2 + \pi + n3, n1) + K1 + n3, n2) + n3$
		$n1' = MixBits(n3, n2)$
		$\overline{C} = \text{Rot}(\text{Rot}(n3 + \overline{K1} + \pi + n1', n3) + \overline{K2} \oplus n1', n2) \oplus n1'$
		če $\overline{C} = C$, nadaljuj, sicer prekini.
		$D = \text{Rot}(\text{Rot}(n2 + \overline{K2} + \text{ID} + n1', n2) + \overline{K1} + n1', n3) + n1'$
3. Faza posodabljanja ključev		
		$\text{IDS}_{\text{old}} = \text{IDS}$
		$K1_{\text{old}} = K1$
		$K2_{\text{old}} = K2$
		$n2' = MixBits(n1', n3)$
		$\text{IDS} = \text{Rot}(\text{Rot}(n1' + \overline{K1} + \text{IDS} + n2', n1') + \overline{K2} \oplus n2', n3) \oplus n2'$
		$K1 = \text{Rot}(\text{Rot}(n3 + \overline{K2} + \pi + n2', n3) + \overline{K1} + n2', n1') + n2'$
		$K2 = \text{Rot}(\text{Rot}(\text{IDS} + \overline{K2} + \pi + K1, \text{IDS}) + \overline{K1} + K1, n2') + K1$
	$\xleftarrow{D}$	
$D = \text{Rot}(\text{Rot}(n2 + \overline{K2} + \text{ID} + n1', n2) + \overline{K1} + n1', n3) + n1'$		
če $\overline{D} = D$, nadaljuj, sicer prekini.		
$n2' = MixBits(n1', n3)$		
$\text{IDS} = \text{Rot}(\text{Rot}(n1' + \overline{K1} + \text{IDS} + n2', n1') + \overline{K2} \oplus n2', n3) \oplus n2'$		
$K1 = \text{Rot}(\text{Rot}(n3 + \overline{K2} + \pi + n2', n3) + \overline{K1} + n2', n1') + n2'$		
$K2 = \text{Rot}(\text{Rot}(\text{IDS} + \overline{K2} + \pi + K1, \text{IDS}) + \overline{K1} + K1, n2') + K1$		
$\pi = 0x3243F6A8885A308D313198A2$		

Tabela 3: Prikaz delovanja Gossamer protokola.

pozdravi oznako, ki mu tedaj odgovori z IDS_{old}. Če tudi tega IDS ne najdemo v bazi, je oznaka v desinhronizacijskem stanju.

2. Vzajemna avtentikacija
Čitalec iz baze pridobi vrednosti ID, $K1$ in $K2$, zgenerira naključni števili $n1$ in $n2$ ter izračuna:

$$\begin{aligned} A &= \text{Rot}(\text{Rot}(\text{IDS} + K1 + \pi + n1, K2) + K1, K1) \\ B &= \text{Rot}(\text{Rot}(\text{IDS} + K2 + \pi + n2, K1) + K2, K2) \\ n3 &= MixBits(n1, n2) \\ \overline{K1} &= \text{Rot}(\text{Rot}(n2 + K1 + \pi + n3, n2) + K2 \oplus n3, n1) \oplus n3 \\ \overline{K2} &= \text{Rot}(\text{Rot}(n1 + K2 + \pi + n3, n1) + K1 + n3, n2) + n3 \\ n1' &= MixBits(n3, n2) \\ C &= \text{Rot}(\text{Rot}(n3 + \overline{K1} + \pi + n1', n3) + \overline{K2} \oplus n1', n2) \oplus n1' \end{aligned}$$

Čitalec oznaki pošlje sporočilo $A||B||C$ in ona iz A in B izračuna $n1$ in $n2$. Sedaj ima na voljo vse potrebne podatke, da izračuna oba nova ključa in C', ki ga primerja s prejetim C. Če sta enaka, nadaljuje s postopkom in izračuna D ter ga pošlje čitalcu. Oznaka potem nadaljuje s tretjo fazo - posodabljanjem ključev.

$$D = \text{Rot}(\text{Rot}(n2 + \overline{K2} + \text{ID} + n1', n2) + \overline{K1} + n1', n3) + n1'$$

Čitalec sprejme D in izračuna svoj D' na enak način kot oznaka. Če se ujemata, je bila faza skupne avtentikacije uspešna in sedaj s tretjo fazo nadaljuje tudi čitalec.

3. Posodabljanje ključev
Čitalec in oznaka posodobita skrivnosti na enak način, le da si oznaka hrani še predhodne skrivnosti. Oznaka najprej trenutne skrivnosti shrani kot predhodne:

$$\begin{aligned} \text{IDS}_{\text{old}} &= \text{IDS} \\ K1_{\text{old}} &= K1 \\ K2_{\text{old}} &= K2 \end{aligned}$$

Nadaljni postopek pa je skupen tako za čitalec kot za oznako:

$$\begin{aligned} n2' &= MixBits(n1', n3) \\ \text{IDS} &= \text{Rot}(\text{Rot}(n1' + \overline{K1} + \text{IDS} + n2', n1') + \overline{K2} \oplus n2', n3) \oplus n2' \\ K1 &= \text{Rot}(\text{Rot}(n3 + \overline{K2} + \pi + n2', n3) + \overline{K1} + n2', n1') + n2' \\ K2 &= \text{Rot}(\text{Rot}(\text{IDS} + \overline{K2} + \pi + K1, \text{IDS}) + \overline{K1} + K1, n2') + K1 \end{aligned}$$

Enako kot pri SASI protokolu sporočili C in D pomenita potrditev novih ključev z namenom preprečevanja enostavnih desinhronizacijskih napadov.

4.2.1 Pravilnost delovanja

Nova trojka (IDS,$K1$,$K2$) in vrednosti C ter D se izračunata iz predhodne trojke (IDS,$K1$,$K2$) in števil $n1$ ter $n2$. Predpostavimo, da oznaka ni v desinhronizacijskem stanju, torej tako oznaka kot čitalec vsebujeta enako trojko. Števili $n1$ ter $n2$ naključno izbere čitalec in jih v sporočilu preko A in B pošlje oznaki. Oznaka iz njih izračuna naključni števili in sedaj imata obe strani enake podatke. Ker uporabljata enak postopek za izračun C in D ter posodobitev vrednosti, lahko rečemo, da lahko oznako uspešno identificiramo.

4.2.2 Desinhronizacijski napad

Če bi hoteli izvajati napad na podoben način, kot smo ga pri SASI protokolu in bi zahtevali da so $K1$, $K2$, $n1$ in $n2$ enaki 0 po modulu n, ne bi uspeli. Problem nastane pri tem, da se vrednost $n3$ izračuna z $MixBits$ iz vrednosti $n1$ in $n2$, ki pa jih hočemo, da jih imata oznaka in čitalec različni, da lahko izračunata različna ključa, a ista C in D.

4.2.3 Napad prihodnje varnosti

Napad, kot smo ga pokazali pri SASI protokolu, na Gossamer ne deluje. Cilj napada je, da če posedujemo trenutne skrivnosti, da lahko iz njih dobimo predhodne skrivnosti, pri tem pa so nam lahko na voljo dosedanja sporočila. Na voljo imamo $(\mathrm{IDS}_i, A_i, B_i, C_i, D_i)$ za javno komunikacijo in ID, $(\mathrm{IDS}_m, K1_m, K2_m)$ ter $(\mathrm{IDS}_{m-1}, K1_{m-1}, K2_{m-1})$. Na podlagi tega bi morali izračunati npr. $n2'$, a ne bi šlo, saj bi potrebovali še $\overline{K1}$, $n1'$ in $n3$.

4.2.4 Pasivni napad za pridobitev ID

Podoben napad deluje tudi na SASI protokolu. Da bo napad uspešen, morata biti naključni vrednosti $n1$ in $n2$ enaki 0 po modulu n. Iz tega velja tudi, da je $MixBits(0 \pmod n, 0 \pmod n) \equiv 0 \pmod n$. Torej so $n1$, $n2$, $n3$, $n1'$ vsi enaki 0 po modulu n. Zaradi tega velja tudi:

$$\begin{aligned} C &= \mathrm{Rot}(\mathrm{Rot}(n3 + \overline{K1} + \pi + n1', n3) + \overline{K2} \oplus n1', n2) \oplus n1' \\ &= \overline{K1} + \pi + \overline{K2} \\ \overline{K1} + \overline{K2} &= C - \pi \end{aligned}$$

$$\begin{aligned} D &= \mathrm{Rot}(\mathrm{Rot}(n2 + \overline{K2} + \mathrm{ID} + n1', n2) + \overline{K1} + n1', n3) + n1' \\ &= \overline{K1} + \mathrm{ID} + \overline{K2} \\ \overline{K1} + \overline{K2} &= D - \mathrm{ID} \end{aligned}$$

$$\begin{aligned} \mathrm{IDS} &= \mathrm{Rot}(\mathrm{Rot}(n1' + \overline{K1} + \mathrm{IDS}_{\mathrm{old}} + n2', n1') + \overline{K2} \oplus n2', n3) \oplus n2' \\ &= \overline{K1} + \mathrm{IDS}_{\mathrm{old}} + \overline{K2} \\ \overline{K1} + \overline{K2} &= \mathrm{IDS} - \mathrm{IDS}_{\mathrm{old}} \end{aligned}$$

Iz zgornjih enačb lahko izpeljemo naslednje:

$$\begin{aligned} \mathrm{ID} &= D - C + \pi \\ &= D - \mathrm{IDS} + \mathrm{IDS}_{\mathrm{old}} \\ C - \pi &= \mathrm{IDS} - \mathrm{IDS}_{\mathrm{old}} \end{aligned} \tag{5}$$

Napad izvedemo tako, da beležimo promet med čitalcem in določeno oznako in preverjamo zgornje enačbe za vsaki dve zaporedni seji. Ko velja enačba 5, veljajo tudi predpostavke in lahko enostavno določimo ID. Odvisno od aplikacije, nam včasih npr. v trgovini ni pomembno, če napadalec izve ID, le da ne more izvedeti vrednosti ključev. Verjetnost, da ta napad uspe, je $1/n^4$.

Napad lahko preprečimo, če zagotovimo, da naključni števili nikoli ne zadoščata predpostavki. Napad bi lahko onemogočili tudi, če bi funkcijo $MixBits$ predelali tako, da izhod ni enak vhodu za 0 $(\mathrm{mod}\ n)$.

4.3 Primerjava protokolov

V tabeli 4 je prikazana primerjava protokolov. Očitno je Gossamer protokol močnejši kot SASI protokol. Ena izmed posledic tega je tudi to, da je identifikacija z njim daljša. V obremenjenih sistemih se lahko pokaže, da oznake potrebujejo veliko energije za računanje precej operacij po modulu in bitnega zamika. Zaradi tega so bile predlagane tudi izboljšave, ki namesto teh operacij, uporablja računanje s predznakom in logaritmi. Eden takšnih protokolov - SSL-MAP je predstavljen v [10].

	UMAP družina	SASI	Gossamer
Odpornost na desinhronizacijski napad	Ne	Ne	Da
Odpornost na napad s spreminjanjem sporočil	Ne	Ne	Da
Zasebnost in anonimnost	Ne	Ne	Ne
Prihodnja varnost	Ne	Ne	Da
Število javnih sporočil za avtentikacijo	4-5L	4L	4L
Spomin na oznaki	6L	7L	7L
Spomin v zalednem sistemu	6L	4L	4L
Operacije na oznaki	$\oplus, \vee, \wedge, +$	$\oplus, \vee, \wedge, +$, Rot	$\oplus$, +, Rot,*MixBits*

Tabela 4: Primerjava UMAP protokolov. L označuje dolžino ene besede.

Čeprav je iz tabele 4 razvidno, da je protokol Gossamer boljši, še ne pomeni, da SASI ni uporaben. SASI protokol je hitrejši, zato lahko posledično v določenem času preberemo več RFID oznak.

Recimo, da želimo SASI protokol uporabiti v knjižnici na RFID oznakah za označevanje knjig. Predpostavimo, da imajo knjige z istim naslovom in avtorjem isti ID. Začetne ključe in psevdonim IDS izberemo naključno in jih shranimo tudi v bazo v zaledni sistem.

- Pri napadih smo spoznali, da nam noben od obeh protokolov ne omogoča anonimnosti, torej lahko napadalec pridobi ID številko knjige. To nas ne skrbi, saj s tem ne razkrijemo nobene poslovne skrivnosti.
- Malo bolj zahteven je desinhronizacijski napad. Da lahko določeno knjigo uspešno desinhroniziramo glede na napad s spreminjanjem sporočil A,C in D, moramo pričakovano 384-krat poskusiti z napadom. Poleg tega moramo medtem, ko knjižničarka skenira naše knjige, s svojim čitalcem poskušati napad. Število poskusov je mnogo, zato se ne izplača. Naprava za komuniciranje z RFID oznakami je lahko zasnovana tako, da ne moremo prisluškovati, kar onemogoči ta napad. Če desinhroniziramo eno knjigo, nam to pri napadih na ostale nič ne pomaga. V primeru, da bi prišlo, do desinhronizacijskega napada, bi morala knjižničarka knjigo le še enkrat vnesti v bazo in težava bi bila odpravljena.
- Napad prihodnje varnosti nas ne skrbi, saj nam je vseeno, če uporabnik izve skrivnosti iz preteklosti, poleg tega mora tudi prisluškovati prometu.

Problem nastane le, če lahko nekdo profilira posameznika glede na to, katere knjige si izposoja. Da bi storil to, bi moral napadalec ves čas prisluškovati prometu v knjižnici za vse knjige in si za vsako knjigo graditi tabelo pogovorov. Poleg tega pa bi moral oznako na vsaki knjigi tudi fizično napasti, da bi prišel do skrivnosti. Takšen napad bi bil drag, poleg tega pa lahko čitalce za izposojo v knjižnici tudi dobro zaščitijo pred prisluškovanjem.

- Primer napada: Napadalec si izposodi eno drago in eno cenejšo knjigo. Nato doma zamenja RFID oznaki in vrne le poceni knjigo. V knjižnici pove, da je drugo knjigo izgubil in jo želi plačati. Če knjižničarka ni dovolj pazljiva in ne preveri, če se knjiga fizično ujema z bazo, lahko napadalec poceni pride do drage knjige.

Če bi uporabili Gossamer protokol, bi se le zaščitili proti desinhronizacijskem napadu. Proti fizičnem napadu se ne moremo zaščititi. Lahko bi le poostrili varnostno politiko v knjižnici. Pri vračanju knjig bi lahko z optičnim čitalcem preverjali izgled knjige in njeno težo.

Ali smo z uporabo enega od teh dveh protokolov na boljšem kot če bi uporabili navaden RFID čip, ki bi hranil le ID in bi imel le bralni pomnilnik? Da. V tem primeru, bi lahko uporabnik enostavno z majhnim čitalcem, ki bi ga enostavno prinesel med police za vsako knjigo pridobil ID.

- Najenostavneje, če bi izvedel ID poceni knjige ter oznako s takšnim ID nalepil na drago knjigo. S svojim čitalcem, ki bi ga imel s seboj, bi lahko kar v knjižnici vpisal ID v kupljeno RFID oznako in jo skupaj z aluminijasto folijo namestil nad originalno oznako v dragi knjigi. Od zunaj bi lahko izgledalo, kot da je napadalec v knjigo zataknil kartonček za branje. Na takšen način bi lahko zelo enostavno prišel do drage knjige z lažjim napadom.

5 Zaključek in ugotovitve

Predstavili smo dva protokola, ki jih lahko uporabljamo za identifikacijo v RFID sistemih. V zadnjih letih RFID tehnologija počasi izpodriva starejše črtne kode in princip razvoja tovrstnih protokolov, ki delujejo na poceni oznakah, lahko pomaga le k večji uporabi, saj znižuje ceno celotnega sistema. V organizacijah, kjer bi morali označiti tudi več kot 10.000 izdelkov, veliko vlogo igra vsak stotin.

Ugotovili smo, da samo bitne operacije niso dovolj dobre, saj oblikujejo pričakovan rezultat. To težavo so najprej hoteli odpraviti s SASI protokolom, ki pa ni bil povsem dobro premišljen, zato Gossamer uvaja še dodatno nebesedno funkcijo MixBits, ki javna sporočila naredi bolj randomizirana.

Vprašanje, ki se zastavja, je: Ali se sploh lahko dobro zaščitimo? Obstaja več odgovorov, ki bi jih omejil na kriptografske protokole, računsko moč naprav in ljudi. Vsi trije faktorji so tudi v medsebojni odvisnosti. Če imamo na voljo veliko računske moči, lahko uporabimo najboljše kriptograske protokole, vendar to poveča ceno. Za RFID sisteme bi lahko za oznake uporabili kar brezkontaktne pametne kartice, ki so malo večje in bi rešili problem z varnostjo. Najti moramo dobro razmerje. Če imamo v sistemu veliko sodelujočih, ki jih ponavadi v RFID sistemih imamo, moramo vzeti v zakup tudi ta faktor. Lahko naredimo še tako

varen sistem, a s trenutkom, ko ima nad njim popoln nadzor vsaj en človek, se ga da napasti preko njega.

Literatura

[1] E.G. Ahmed, E. Shaaban, and M. Hashem. Lightweight Mutual Authentication Protocol for Low Cost RFID Tags. *Arxiv preprint arXiv:1005.4499*, 2010.

[2] T. Cao, E. Bertino, and H. Lei. Security analysis of the SASI protocol. *IEEE Transactions on Dependable and Secure Computing*, pages 73–77, 2008.

[3] H.Y. Chien. SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. *IEEE Transactions on Dependable and Secure Computing*, 4(4):337–340, 2007.

[4] S. Karthikeyan and M. Nesterenko. RFID security without extensive cryptography. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, page 67. ACM, 2005.

[5] M. Langheinrich and R. Marti. Practical minimalist cryptography for RFID privacy. *IEEE Systems Journal*, 1(2):115–128, 2007.

[6] A. Mitrokotsa, M.R. Rieback, and A.S. Tanenbaum. Classification of RFID attacks. *Gen*, 15693:14443, 2009.

[7] S. Moškon. Brezkontaktne pametne kartice. *Seminarska naloga pri predmetu Kriptografija in teorija kodiranja 2*, 2009.

[8] P. Peris-Lopez, J. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda. M 2 AP: A minimalist mutual-authentication protocol for low-cost RFID tags. *Ubiquitous Intelligence and Computing*, pages 912–923, 2006.

[9] P. Peris-Lopez, J. Hernandez-Castro, J. Tapiador, and A. Ribagorda. Advances in Ultralightweight Cryptography for Low-Cost RFID Tags: Gossamer Protocol. *Information Security Applications*, pages 56–68, 2009.

[10] N. Rama and R. Suganya. SSL-MAP: A More Secure Gossamer-based Mutual Authentication Protocol for Passive RFID Tags.

[11] Wikipedia. Faradayeva kletka.

[12] Wikipedia. Radio frequency identification.