

# Automatizirana analiza autentifikacijskih protokola pomoću alata Tamarin prover

---

Lovei, Andrej

Undergraduate thesis / Završni rad

2024

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:168:728366>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-20**



*Repository / Repozitorij:*

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1517

**AUTOMATIZIRANA ANALIZA AUTENTIFIKACIJSKIH  
PROTOKOLA POMOĆU ALATA TAMARIN PROVER**

Andrej Lovei

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1517

**AUTOMATIZIRANA ANALIZA AUTENTIFIKACIJSKIH  
PROTOKOLA POMOĆU ALATA TAMARIN PROVER**

Andrej Lovei

Zagreb, lipanj 2024.

## ZAVRŠNI ZADATAK br. 1517

Pristupnik: **Andrej Lovei (0036540822)**  
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo  
Modul: Računarstvo  
Mentor: izv. prof. dr. sc. Ante Đerek

Zadatak: **Automatizirana analiza autentifikacijskih protokola pomoću alata Tamarin prover**

### Opis zadatka:

U sklopu završnog rada potrebno je istražiti teorijske osnove te način rada alata za analizu sigurnosnih protokola Tamarin prover. U sklopu praktičnog rada potrebno je ostvariti formalni nekog jednostavnog protokola za jednostranu ili obostranu autentifikaciju. Dodatno, potrebno je specificirati željena sigurnosna svojstva koristeći hierarhiju autentifikacijskih svojstava, te pokušati formalno verificirati ili opovrgnuti navedena svojstva. Radu je potrebno priložiti izvorni kôd razvijenih i korištenih programa, citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 14. lipnja 2024.

Zahvaljujem mentoru izv. prof. dr. sc. Anti Đereku na pomoći  
i savjetima prilikom izrade završnog rada.

## Sadržaj

1. Uvod .....	1
2. Mrežni protokoli .....	2
3. Tamarin prover .....	4
3.1. Funkcijski simboli .....	4
3.2. Stvaranje modela .....	4
3.2.1. Pravila, činjenice, izrazi.....	4
3.3. Dokazivanje sigurnosnih svojstava .....	6
3.3.1. Anotacije lema .....	6
3.3.2. Predizračun .....	6
3.3.3. Dokazivanje lema .....	7
3.4. Primjer modeliranja .....	7
3.5. Interaktivan prikaz u pregledniku.....	9
4. Analizirani sigurnosni protokol .....	10
4.1. Opis protokola .....	10
4.1.1. Inicijalna faza postavljanja za novi IoT uređaj.....	10
4.1.2. Normalna razmjena poruka između gateway-a i IoT uređaja.....	12
4.1.3. Ažuriranje sesijskog ključa.....	12
4.1.4. Neuspjeh u sinkronizaciji Ks.....	13
Zaključak .....	14
Literatura .....	15
Sažetak.....	16
Summary.....	17

# 1. Uvod

Sigurnosni protokoli su od ključne važnosti za zaštitu osjetljivih informacija tijekom komunikacije na Internetu. Autentifikacijski protokoli su prvi korak u komunikaciji između dva entiteta koji žele uspostaviti sigurnosnu komunikaciju. Omogućavaju da primajući entitet (npr. server) autentificira povezujućeg entiteta (npr. klijent) i obrnuto, sprečavajući neovlašteni pristup podacima i lažno predstavljanje (svaka strana je sigurna u identitet one druge).

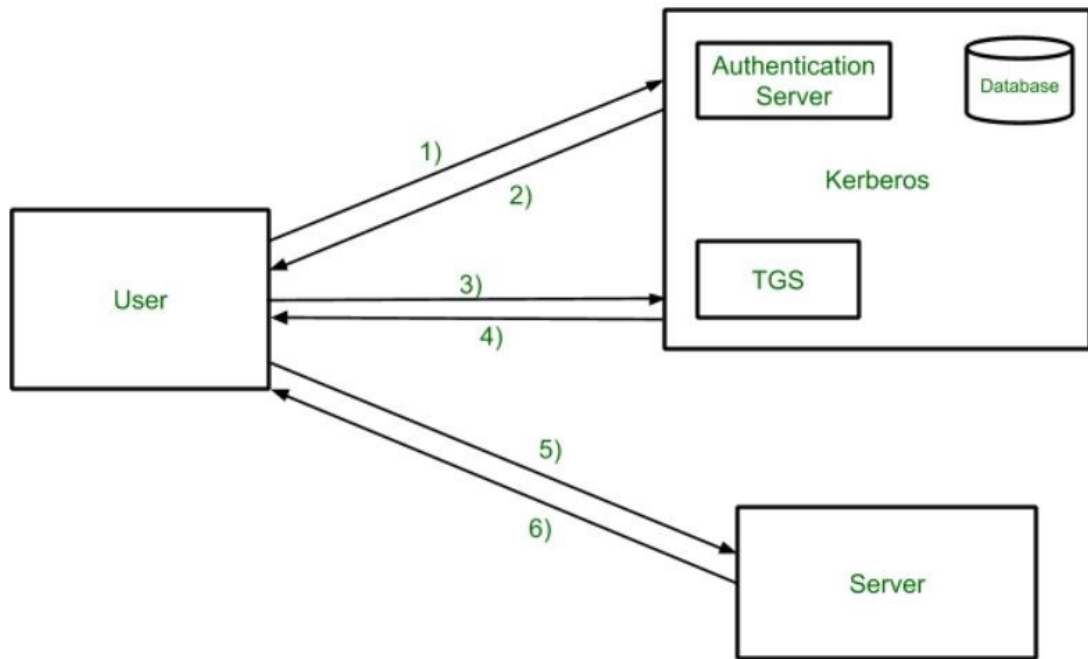
Tamarin prover je moćan alat za simboličko modeliranje i analizu sigurnosnih protokola kroz matematičke modele i logičke dokaze. Model sigurnosnog protokola se definira pravilima koja specificiraju akcije poduzete od strane agenata prisutnih u radu protokola te lemmama koje iskazuju svojstvo koje želimo dokazati da protokol posjeduje. Tamarin potom može dati dokaz leme ili primjer napada.

Nakon proučavanja Tamarin dokumentacije i određenih modela pruženih od Tamarina, izmodelirao sam sigurnosni protokol za obostranu autentifikaciju i razmjenu ključeva te probao ispitati neka njegova svojstva. Protokol uključuje IoT uređaj i gateway pa se zbog ograničenja u snazi i memoriji koristila simetrična kriptografija. Ispitao sam svojstva autentifikacije (ako je uspostavljena sesija između dva uređaja, jedan od sudionika ne može biti napadač) i tajnost ključeva. Sva ta svojstva su dokazana tako da je protokol siguran u tom pogledu.

## 2. Mrežni protokoli

Mrežni protokol određuje kako će se podaci prenositi preko mreže. U to spadaju komunikacijski mrežni protokoli (UDP, TCP...), protokoli za upravljanje (SNMP...) te sigurnosni protokoli koji koriste kriptografiju.[7] Komunikacijski protokoli određuju pravila za putovanje podataka mrežom tako da ne budu oštećeni. Upravljački protokoli služe za upravljanje i nadzor mrežnih uređaja.[8] Sigurnosni protokoli osiguravaju da podaci i informacije budu sigurne od napadača i mogu se analizirati Tamarinom, a postupak će se razlikovati od protokola do protokola u tome što je važno za ispitati za specifičan protokol i što protokol obećava. Sustavi umjetne inteligencije i strojno učenje koriste podatke sa različitih izvora i potrebno je da su ti podatci sigurno došli do odredišta gdje se učenje odvija. Također, važno područje gdje su sigurnosni mrežni protokoli ključni je bežično komuniciranje IoT uređaja. Jedan od tih sam modelirao u Tamarinu. Kerberos je protokol za obostranu autentifikaciju između klijenta i servera koji se, za razliku od protokola koji sam analizirao, oslanja na pouzdanu treću stranu. Klijent se svojim ID-em autentificira autentifikacijskom serveru koji je dio centra distribucije ključeva (pouzdana treća strana, KDC). KDC mu vraća ulaznicu za izdavanje ulaznice koja vrijedi samo određeni vremenski period uz sesijski ključ, kriptirano klijentovom lozinkom. Klijent dekriptira poruku i šalje je serveru za izdavanje ulaznica. Ulaznica sadrži klijentovo ime i mrežnu adresu. Server za izdavanje ulaznica potom stvara ulaznicu za zahtijevanje usluga od servera. Klijent šalje ulaznicu i autentifikator serveru koji sve to verificira te mu daje pristup usluzi. Uz jake korisničke lozinke, Kerberos je vrlo siguran pa ga koriste Microsoft Windows i razni Unix operacijski sustavi. Na slici Sl. 2.1 je prikazan Kerberos protokol.





Sl. 2.1 Kerberos protokol [10]

## 3. Tamarin prover

Tamarin prover specifikira protokole koristeći pravila. Stanje protokola je definirano znanjem napadača, porukama na mreži, informaciji o svježe generiranim vrijednostima te stanjem protokola. Interakcija napadača i protokola se događa ažuriranjem mrežnih poruka i generiranjem novih poruka.

### 3.1. Funkcijski simboli

Tamarin sadrži skup ugrađenih funkcijskih simbola i dopušta korisniku definiranje dodatnih. Funkcijski simbol pair (sintaktički je dozvoljeno i  $\langle \text{message1}, \text{message2} \rangle$ ) predstavlja par dviju poruka.  $\langle m_1, \langle m_2, \dots, \langle m_{n-1}, m_n \rangle \dots \rangle$  možemo lakše označavati samo kao  $\langle m_1, m_2, \dots, m_{n-1}, m_n \rangle$ . Kriptografski funkcijski simboli se aktiviraju uključivanjem teorija poruke, od kojih sam koristio hashing i symmetric-encryption. Također sam definirao vlastite funkcijske simbole mac/2 (Message Authentication Code) i hkdf/3 (Hashed Message Authentication Code (HMAC)-based Key Derivation Function). Broj uz funkcijski simbol je broj argumenata funkcije.

### 3.2. Stvaranje modela

Modeliramo protokol opisujući njegovo ponašanje pravilima te njegova sigurnosna svojstva lemmama. Koristi se Dolev-Yao model za napadača, može presresti, modificirati i poslati bilo koju poruku.

#### 3.2.1. Pravila, činjenice, izrazi

Kriptografske poruke su reprezentirane kao izrazi, Tamarin apstrahira poruke kako bi model bio jednostavniji. Kriptografska poruka je ili konstanta ili  $n$  kriptografskih poruka provučenih kroz funkcijski simbol arnosti  $n$ .

Multiset je set elemenata koji može sadržavati više istih elemenata. U Tamarinu, stanje sistema je multiset činjenica. Inicijalno stanje je prazan multiset. Pravila su sastavljena od niza činjenica: na lijevoj strani (moraju biti već stvorene da bi se pravilo

moglo primijeniti), akcijskih činjenica koje se koriste u lemmama te na desnoj strani pravila (činjenice koje nastaju kad se pravilo izvrši). Pravila su multiset rewriting tipa da se omogući usporedno djelovanje protokola i napadača, pravila primjenjujemo na činjenice u multisetovima tako da se više pravila može izvoditi u isto vrijeme. U slučaju da ne želimo da se činjenica poništi kad se izvrši pravilo (linearne činjenice), definiramo stalne činjenice tako da ispred njih dodamo uskličnik !.

Činjenice imaju izraze kao argumente, forma činjenica je  $F(i_1, \dots, i_n)$  gdje su  $i_k$  izrazi. Imaju fiksiran broj argumenata u svakoj instanci jedne činjenice. Tri činjenice su ugrađene:

- In – agent prima poruku iz nesigurne mreže, uvijek na lijevoj strani pravila
- Out – agent šalje poruku u nesigurnu mrežu, samo na desnoj strani pravila, napadač dobiva informacije i podatke konzumiranjem ove činjenice
- Fr – generira nasumičnu vrijednost, također uvijek na lijevoj strani pravila.

Posebna vrsta činjenica je akcijske činjenice koje se koriste u dokazivanju lema, ne pojavljuju se u pojedinom stanju sustava, nego opisuju što se dogodilo u prijelazu stanja iz jednog u drugo.

Varijable mogu biti:

- fresh – svježe generirana slučajna vrijednost ( $\sim v1$ ),
- pub – javna varijabla ( $\$v2$ ),
- nat – prirodni broj ( $\%v3$ )
- temporal – vremenska varijabla ( $\#v4$ ).

Generičko pravilo u Tamarinu izgleda ovako:

rule ruleName:

$$[ \text{Fact}_1(v_1^1, \dots, v_n^1), \dots, \text{Fact}_k(v_1^k, \dots, v_m^k) ]$$

$$- [ \text{ActionFact}_1(v_1^1, \dots, v_r^1), \dots, \text{ActionFact}_t(v_1^t, \dots, v_q^t) ] \rightarrow$$

$$[ \text{Fact}_1(v_1^1, \dots, v_z^1), \dots, \text{Fact}_p(v_1^p, \dots, v_y^p) ]$$

Ako se premisa izostavi ili je isključivo generiranje svježih varijabli ( $\text{Fr}(\sim v1)$ ), to pravilo se može izvršiti neograničeno puta.

### 3.3. Dokazivanje sigurnosnih svojstava

Dokazivanje sigurnosnih svojstava protokola se odvija lemama. Leme se pišu manipulacijom akcijskih činjenica matematičkim izrazima. Ako su sve činjenice iz premise pravila u trenutnom stanju, Tamarin koristi odgovarajuće pravilo da dođe u sljedeće stanje u kojem će biti činjenice iz zaključka. Tamarin će dodati akcijsku činjenicu iz tog pravila u trag. Trag je jedno moguće izvršavanje protokola sa svim pravilima koja se apliciraju od početka do kraja. Specifična sintaksa za Tamarin:

- $\text{ActionFact}() @ i$  – akcijska činjenica se dogodila u vremenskom trenutku  $\#i$
- $i < j$  – vremenski trenutak  $\#i$  je prije  $\#j$
- $\#i = \#j, x = y$  –  $i$  je isti vremenski trenutak kao  $j$ ,  $x$  je ista poruka kao  $y$ .

Implicitno se lema dokazuje tako da vrijedi za sve moguće tragove protokola. Ako želimo provjeriti postoji li trag za koji nešto vrijedi dodajemo ključnu riječ *exists-trace* prije izraza svojstva.

#### 3.3.1. Anotacije lema

Anotacije lema se dodaju u zagradama uz lemu. Anotacija *use\_induction* uvijek preferira dokaz indukcijom pred pojednostavljenjem. *Reuse* lema se koristi u svim lemama koje ju sintaktički slijede. *Source* lema se dokazuje indukcijom i isključivo sirovim izvorima (*raw sources*), a stvaraju rafinirane izvore (*refined sources*) pomoću kojih se onda dokazuje sve ostale leme. One su jedine leme koje ne koriste *reuse* leme.

#### 3.3.2. Predizračun

U fazi predizračuna Tamarin prolazi kroz sva pravila i za sve činjenice u premisama, odredit će njihove moguće izvore. Prikazat će sva pravila koja dovode do određene činjenice. Za neke činjenice Tamarin možda ne nađe izvor pa će prikazati da su ostale parcijalne dekonstrukcije u sirovim izvorima. U tom slučaju može se dogoditi da automatsko dokazivanje ne završi. Pošto Tamarin ne zna otkud je došla činjenica, moguće

je da koristi jedno pravilo rekurzivno zbog svoje heuristike, a da to pravilo nije izvor činjenice.

Parcijalne dekonstrukcije se mogu riješiti sources lemmama. Tamarin prvo konstruira sirove izvore, potom automatski dokaže sve sources leme koje su onda primijenjene na sirove izvore da bi iz njih dobili rafinirane izvore. Ostale leme se potom dokazuju pomoću rafiniranih izvora.

### 3.3.3. Dokazivanje lema

Lemu možemo dokazivati automatski ili ručno. Želimo li dokazivati automatski, Tamarin sve radi sam, odabire 'smart' heuristiku i provodi dokaz odabirući najbolji izraz po toj heuristici. Ručno mi biramo za svaki korak u kojem smjeru će dokaz ići. U svakom trenutku vidimo stanje sustava i graf pravila koji je konstruiran. Na kraju, ako uspije završiti, Tamarin zazeleni cijelu lemu ako vrijedi, a ako nađe protuprimjer, zacrveni lemu i pokaze vizualno kako napadač može napasti sustav.

## 3.4. Primjer modeliranja

U ovom potpoglavlju ću dati primjer modeliranja u Tamarinu. Uzmimo, na primjer, uspostavljanje najjednostavnije sesije između klijenta i servera. Koristit ćemo Diffie-Hellman razmjenu ključeva. Prvo nam treba inicijalno pravilo koje Tamarin može izvesti bez postojećih činjenica. To možemo napraviti tako da je premisa pravila generiranje slučajne vrijednosti, u našem slučaju privatnog ključa klijenta ili servera (kod Diffie-Hellmana nema razlike):

```
rule Generate_private_keys_and_create_entities:
  [ Fr(~privateKey) ]
  -->
  [ !EntityAndKey($Entity, ~privateKey),
    Out(g^(~privateKey)) ]
```

Za korištenje  $g^{(\sim\text{priv})}$ , čime dobivamo javni ključ od privatnog, potrebna nam je ugrađena teorija diffie-hellman koju uključujemo u naš model sintaksom: `builtins: diffie-hellman`. Iz pravila dobivamo činjenicu `EntityAndKey` koja povezuje javno ime entiteta (klijenta ili servera) sa privatnim ključem te je u mrežu poslan javni ključ. Pošto se pravilo može pozvati neograničen broj puta, pravilo se koristi i za kreiranje

klijenta sa privatnim ključem i servera sa različitim privatnim ključem. Nakon toga nam treba pravilo u kojem klijent šalje prvu poruku serveru:

```
rule Client_sends_message:
  let
    publicKeyClient = g^(~privateKeyClient)
  in
    [ !EntityAndKey($Client, ~privateKeyClient),
    !EntityAndKey($Server, ~privateKeyServer) ]
  -->
    [ Out(<'initialize_session', $Client, $Server,
    publicKeyClient>) ]
```

Let a = b in je sintaksa kojom možemo olakšati razumijevanje i vrijedi za cijelo pravilo. Trebaju postojati i klijent i server pa je to označeno u premisi. Sesija se uspostavlja tako da se javni ključ drugog entiteta potencira sa privatnim ključem prvog. U sljedećem pravilu bi u zaključku bila činjenica `Session($server, $client, publicKeyClient^~privateKeyServer)`:

```
rule Server_receives_message:
  let
    publicKeyClient = g^(~privateKeyClient)
    publicKeyServer = g^(~privateKeyServer)
    key = publicKeyClient^~privateKeyServer
  in
    [ !EntityAndKey($Server, ~privateKeyServer),
    !EntityAndKey($Client, ~privateKeyClient),
    In(<'initialize_session', $Client, $Server,
    publicKeyClient>)]
  --[ServerWithClientSession($Server, $Client, key)]->
    [ !Session($Server, $Client, key),
    Out(<'session_established', $Client, $Server,
    publicKeyServer>) ]
```

Nakon što server pošalje svoj javni ključ, klijent uspostavlja sesiju sa serverom na isti način, `Session($client, $server, publicKeyServer^~privateKeyClient)`. Akcijska činjenica `ServerWithClientSession($Server, $Client, key)` označuje da se u ovom pravilu uspostavila sesija između servera i klijenta i to se može koristiti u lemmama. Pretpostavimo da je u sljedećem pravilu uspostavljena sesija klijenta sa serverom i da je definirana akcijska činjenica `ClientWithServerSession($Client, $Server, key)`. Sada možemo probati dokazati jednu jednostavnu lemu:

```
lemma Server_sends_message_after_client:
```

```

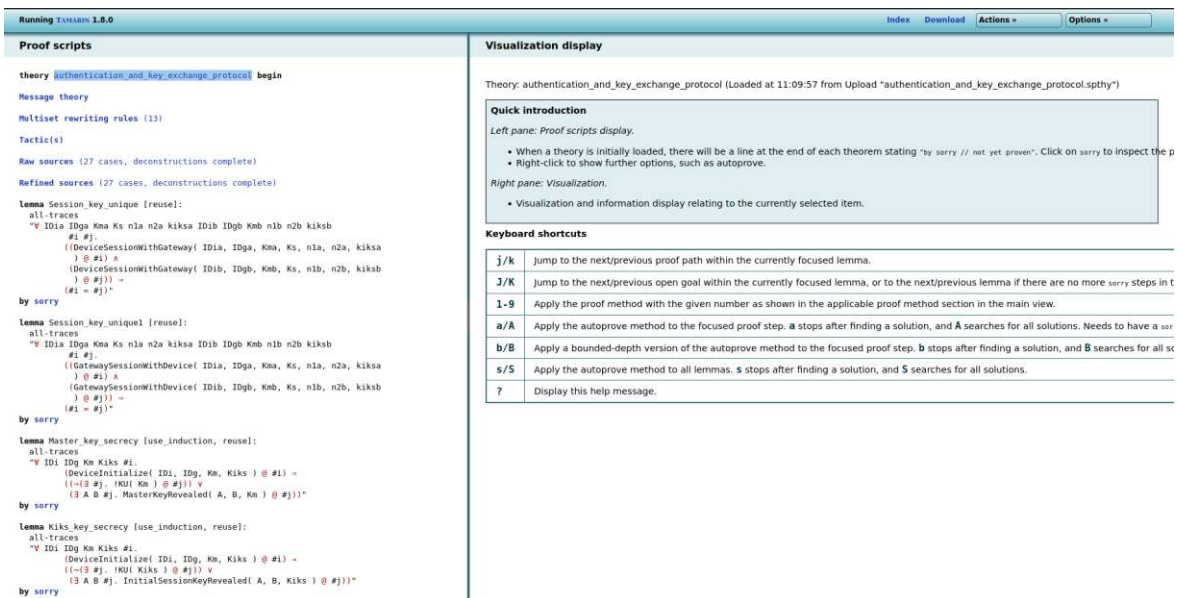
"
  (All Client Server SessionKey #i.
  ClientWithServerSession(Client, Server, SessionKey) @ i
  ==>
  (Ex #j. ServerWithClientSession(Server, Client, SessionKey) @
  j & #j < #i)
)"

```

Ovdje hoćemo dokazati da za sve klijente, servere, sesijske ključeve i vremenske trenutke #i vrijedi da ako je klijent uspostavio sesiju sa serverom, postoji trenutak j koji se dogodio prije u kojem je server uspostavio sesiju sa klijentom. Sa K(SessionKey) možemo označiti da napadač zna vrijednost SessionKey.

### 3.5. Interaktivan prikaz u pregledniku

Tamarin možemo pozvati uz argument 'interactive' u terminalu. Tada možemo otići na <http://127.0.0.1:3001> i tamo će biti učitane sve sphy datoteke (security protocol theory, datoteke koje Tamarin prihvaća) iz direktorija kojeg smo napisali u naredbi za pokretanje. Kad odaberemo jednu od njih, vidimo multiset rewriting pravila, sirove izvore, rafinirane izvore i sve leme, što je prikazano slikom Sl. 3.1. Klikom na sorry kod određene leme, prikazuju nam se trenutne opcije za dokazivanje (pojednostavi, indukcija, automatski dokaži).



Sl. 3.1 Interaktivan prikaz [2]

## 4. Analizirani sigurnosni protokol

Povećanjem praćenja zdravstvenog stanja pacijenata i videonadzora, korištenje IoT uređaja raste. Za pojedine slučajeve, potrebno je osigurati podatke koji se šalju i autentificirati tko šalje, no uređaji su ograničeni resursima (procesuiranje, količina memorije i energije potrebne za komuniciranje) što predloženi protokol učinkovito koristi. Korištena je simetrična kriptografija jer asimetrična kriptografija zahtjeva veliku računalnu moć i puno memorije.

### 4.1. Opis protokola

Protokol koristi 2 dijeljena ključa između IoT uređaja i gateway-a, dugoročni master ključ i kratkoročni ključ sesije. Master ključ i inicijalni sesijski ključ su umetnuti ručno na početku u IoT uređaj i gateway. Tajni ključevi nikad nisu razmijenjeni preko mreže, a uređaj i gateway se međusobno autentificiraju kriptirajući poruke master ključem i hashiranjem poruka sesijskim ključem. Savršena tajnost naprijed (perfect forward secrecy) je važno svojstvo sigurnosnog protokola koje osigurava da čak i ako se dugoročni tajni ključ otkrije, sve prijašnje sesije moraju biti sigurne (ključevi prijašnjih sesija ne smiju biti kompromitirani). Ovaj protokol ima ovo svojstvo jer često ažurira sesijske ključeve sa HKDF uz korištenje nasumičnih okvira poslanih u prethodnoj sesiji i ključa prethodne sesije pa je svaki sljedeći ključ ovisan o svim prijašnjim sesijskim ključevima.

#### 4.1.1. Inicijalna faza postavljanja za novi IoT uređaj

Kada se novi IoT uređaj pridruži mreži, potrebno je dvostrana autentifikacija između njega i gateway-a te moraju uspostaviti prvi sesijski ključ (inicijalni sesijski ključ koji je manualno upisan u oba uređaja služi samo za deriviranje prvog sesijskog ključa).

IoT uređaj šalje svoj ID i niz znakova 'initialization' te nasumičnu vrijednost (Nonce1) kriptiranu master ključem (Km). Također šalje i MAC cijele poruke kombinirane s inicijalnim sesijskim ključem (Kiks), kriptiran s Km. Integritet podataka se potvrđuje tako što primatelj poruke sa svojim identičnim ključevima Kiks i Km izračuna MAC cijele dobivene poruke i usporedi s onim što je dobio.

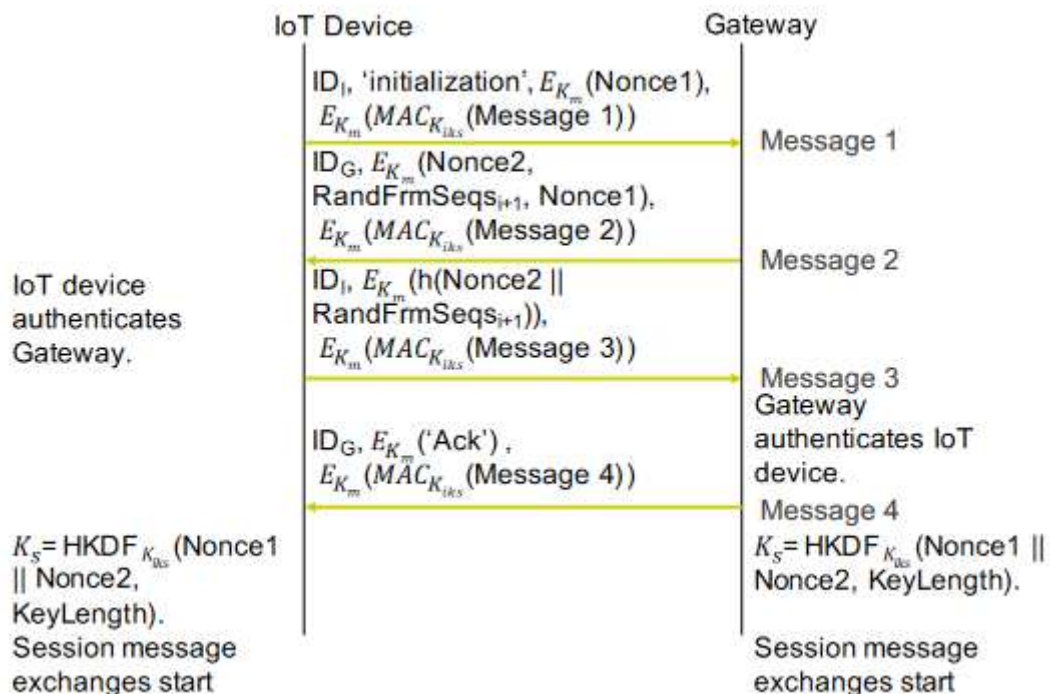


Gateway potom odabire slučajan set nizova brojeva budućih okvira (RandFrmSeqs) za ažuriranje sljedećeg sesijskog ključa (Ks). Kriptira novogeneriranu nasumičnu vrijednost (Nonce2), Nonce1 te RandFrmSeqs master ključem i to šalje zajedno sa svojim ID-em. Nonce1 se isto šalje da bi IoT bio siguran da je gateway dobio dobru vrijednost. Za svaku se poruku koristi isti mehanizam izračuna MAC-a poruke i šalje se zajedno s njom te primatelj računa MAC i uspoređuje sa dekriptiranom porukom.

IoT uređaj je autentificirao gateway. Uz to, potvrđuje da gateway ima ispravan Kiks i da je primio dobar Nonce1. Da bi uređaj potvrdi da je dobio dobar RandFrmSeqs i da ga gateway autentificira, šalje hash od Nonce2 i RandFrmSeqs te MAC poruke.

Primanjem dobrog MAC-a, gateway autentificira IoT uređaj i zna da je primio dobar RandFrmSeqs i Nonce2. Na kraju samo šalje poruku potvrde (svoj ID, kriptiran niz znakova 'Ack' i kriptiran MAC poruke). Gateway, kao i IoT uređaj kad potvrdi MAC zadnje poruke, generira sesijski ključ (Ks) pomoću funkcije derivacije ključa bazirane na HMAC-u koja uzima Kiks i zajedno sa Nonce1 i Nonce2 kao salt (slučajne vrijednosti jer je hkdf deterministička funkcija).

Postupak je prikazan na slici Sl. 4.1.

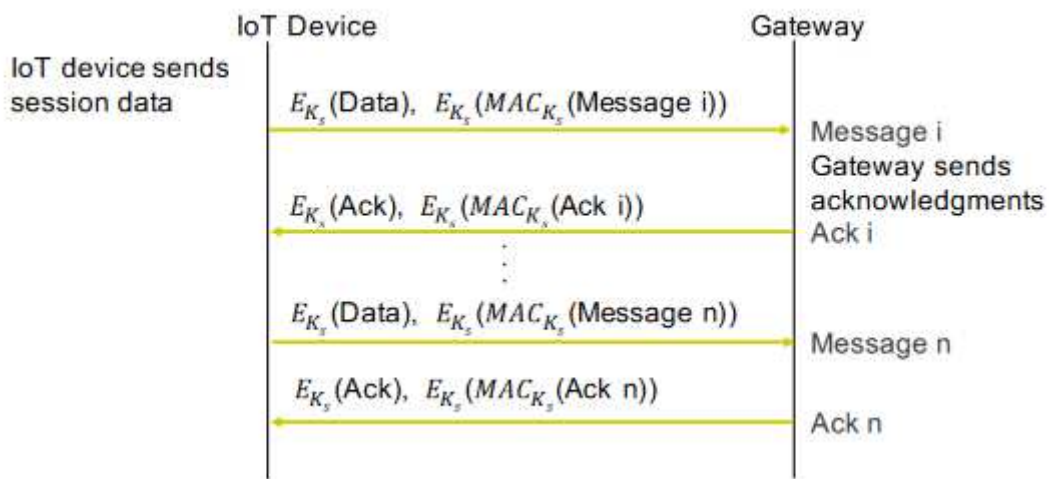


Sl. 4.1 Autentifikacija novog IoT uređaja i deriviranje početnog sesijskog ključa [1]

### 4.1.2. Normalna razmjena poruka između gateway-a i IoT uređaja

Nakon što su oba izgenerirala sesijski ključ, za normalnu komunikaciju koriste samo njega. Oba uređaja čuvaju okvire koji su određeni u RandFrmSeqs. Poruke su kriptirane sesijskim ključem i poslane zajedno sa kriptiranim MAC-om poruke. Gateway potvrđuje primitak okvira. Ažuriranje sesijskom ključa se događa kad se dostigne prag razmijenjenih okvira.

Normalna komunikacija je prikazana slikom Sl. 4.2.

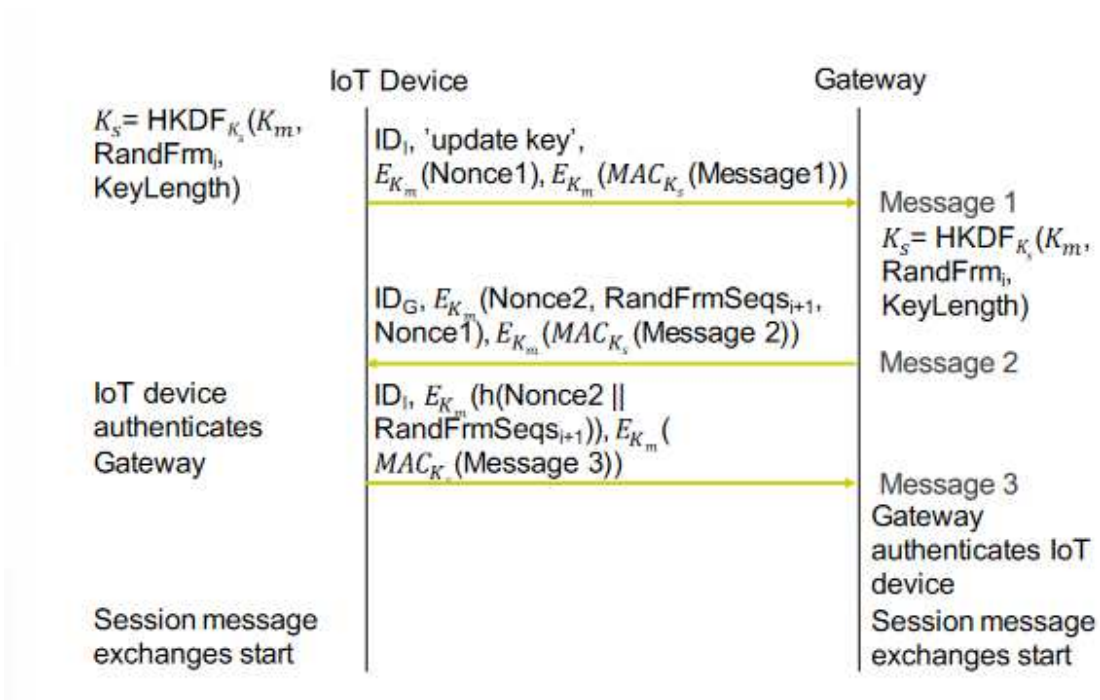


Sl. 4.2 Normalna komunikacija između gateway-a i IoT uređaja [1]

### 4.1.3. Ažuriranje sesijskog ključa

IoT uređaj izračuna HKDF od RandFrmSeqs (dobiven iz prošle sesije) koji će se koristiti kao informacijski ulaz i master ključa ( $K_m$ ) kao salt, koristeći trenutni sesijski ključ ( $K_s$ ). To će biti sljedeći  $K_s$ . Uređaj šalje 'update' poruku gateway-u. Gateway također izračuna novi  $K_s$  uz generiranje novog slučajnog seta okvira za sljedeću sesiju i šalje poruku uređaju koji ga potom autentificira. Na kraju, uređaj šalje poruku gateway-u kako bi ga gateway autentificirao. IoT uređaj i gateway su se međusobno identificirali i opet imaju zajednički  $K_s$ .

Faza ažuriranja je prikazana na slici Sl. 4.3.



Sl. 4.3 Ažuriranje sesijskog ključa [1]

#### 4.1.4. Neuspjeh u sinkronizaciji $K_s$

Kad se dogodi neuspjeh u sinkronizaciji sesijskog ključa  $K_s$  (npr. neki od uređaja se mora ponovno pokrenuti), može se uspostaviti  $K_s$  ponovno na siguran način.  $K_m$  i  $K_{ks}$  su u memoriji oba uređaja tako da se može provesti gotovo identičan proces kao i kod prve faze inicijalizacije i autentifikacije uređaja, samo se šalje 'reset' ključna riječ umjesto 'initialization'. Uz to, iz memorije oba uređaja se brišu svi okviri kako bi imali iste okvire za sljedeće ažuriranje  $K_s$ -a.

## Zaključak

U današnjem digitalnom svijetu od posebne su važnosti sigurnosni mrežni protokoli jer se gotovo sve odvija preko Interneta. Autentifikacija uređaja na mreži je potrebna kako podaci ne bi bili poslani nekome tko do njih ne bi trebao moći doći. Kada se uređaji autentificiraju, mogu biti sigurni s kim pričaju i koliko i koje podatke mu mogu poslati.

Analiziran je protokol za obostranu autentifikaciju i razmjenu ključeva. Korištena je simetrična kriptografija uz kod autentifikacije poruke (MAC). Cilj protokola je uspostava sigurne komunikacije između IoT uređaja i gateway-a. Na početku se u oba uređaja ručno unesu dva ključa koja će biti korištena za deriviranje sesijskog ključa, a nakon toga se za svaki novi ključ koristi prijašnji sesijski ključ.

U ovom radu sam analizu protokola proveo alatom Tamarin prover. U Tamarinu se protokol opisuje pravilima koja imaju premise i zaključke. Premise su činjenice koje uzimaju varijable kao argumente. Kad su sve premise prisutne, Tamarin ih konzumira i dodaje nove činjenice koje su navedene u zaključku u trenutno stanje sustava te dodaje akcijske činjenice na kraj traga. Trag je jedno moguće izvršavanje protokola. Lemama se matematičkim jezikom dokazuju svojstva protokola uz pomoć akcijskih činjenica.

## Literatura

- [1] [https://www.ndss-symposium.org/wp-content/uploads/2018/07/diss2018\\_4\\_Bin-Rabiah\\_paper.pdf](https://www.ndss-symposium.org/wp-content/uploads/2018/07/diss2018_4_Bin-Rabiah_paper.pdf)
- [2] <https://tamarin-prover.com/manual/master/tex/tamarin-manual.pdf>
- [3] <https://youtu.be/XptJG19hDcQ>
- [4] <https://github.com/tamarin-prover/tamarin-prover/tree/develop/examples>
- [5] <https://pages.di.unipi.it/milazzo/teaching/AA1819-CMCS/slides/09-MultisetRewriting.pdf>
- [6] [https://en.wikipedia.org/wiki/Dolev%E2%80%93Yao\\_model](https://en.wikipedia.org/wiki/Dolev%E2%80%93Yao_model)
- [7] <https://www.forbes.com/advisor/business/types-network-protocols/>
- [8] <https://www.infosecinstitute.com/resources/network-security-101/a-deep-dive-into-network-security-protocols-safeguarding-digital-infrastructure-2024/>
- [9] [https://en.wikipedia.org/wiki/Kerberos\\_\(protocol\)](https://en.wikipedia.org/wiki/Kerberos_(protocol))
- [10] <https://www.geeksforgeeks.org/kerberos/>
- [11] [https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman\\_key\\_exchange](https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange)

# Sažetak

## **Automatizirana analiza autentifikacijskih protokola pomoću alata Tamarin prover**

Sigurnosni protokoli su važan dio u mrežnoj komunikaciji jer se često prenose osjetljive informacije. U ovom radu sam analizirao protokol za obostranu autentifikaciju između IoT uređaja i gateway-a pomoću alata Tamarin prover. Tamarin prover je alat za simboličko modeliranje i analizu protokola. Ispitao sam svojstva autentifikacije i tajnost ključeva.

Ključne riječi: Tamarin prover, sigurnosni mrežni protokol, simboličko modeliranje protokola, autentifikacija

# Summary

## **Automated analysis of authentication protocols using the Tamarin prover tool**

Security protocols are crucial part of network communication as sensitive information is often transmitted. In this paper, I analyzed a mutual authentication protocol between IoT devices and a gateway using the Tamarin prover tool. Tamarin prover is a tool for symbolic modeling and protocol analysis. I tested properties of authentication and key secrecy.

Keywords: Tamarin prover, security network protocol, symbolic protocol modeling, authentication