Varna shramba uporabniških overitvenih podatkov v jedru 5G

Miha Rihtaršič, Uroš Brovč, Urban Zaletel

Kontron d.o.o, Kranj, Slovenija
miha.rihtarsic@kontron.com, uros.brovc@kontron.com, urban.zaletel@kontron.com

Varna hramba, obdelava in administracija občutljivih podatkov je zanimiv izziv, ki ima lahko različne rešitve glede na način obdelave, uporabniški namen ter stopnjo zahtevane varnosti. V 5G sistemih varnostna zahteva predpisuje zaščito občutljivih overitvenih podatkov med hranjenjem ter njihovo varno obdelavo med uporabo. V članku predstavljamo tehnologijo zaupnega procesiranja Intel® SGX, ki omogoča splošno obdelavo podatkov z zaščito pred okolico. Opisujemo način, kako z uporabo SGX zagotovimo močno varnost overitvenih podatkov in kako je potrebno prilagoditi administrativne naloge, da sovpadajo s tem načinom delovanja. Tak pristop je še posebej primeren za uporabo v mobilnih postavitvah 5G jeder, kjer obstaja večja nevarnost fizičnega dostopa do sistema — na primer pri jedru, nameščenem v vojaškem vozilu ali nahrbtniku za hitro taktično postavitev omrežja na terenu. S tem pristopom lahko tudi takšne postavitve z visoko izpostavljenostjo zadostijo strogim varnostnim zahtevam sodobnih 5G omrežij.

Ključne besede:
zaupno procesiranje,
5G,
overitev,
šifriranje,
varnost.

Univerzitetna založba Univerze v Mariboru
ISBN 978-961-299-037-4
DOI https://doi.org/10.18690/um.feri.7.2025.12

1 Uvod

Sistem 5G jedro za delovanje potrebuje neprekinjen dostop do baze naročniških overitvenih podatkov, ki so tipično občutljive narave. Izpostavljenost teh podatkov — bodisi zaradi varnostne ranljivosti, bodisi zaradi zlonamernih dejanj — lahko vodi do resnih posledic, glede na obseg, kritičnost in javno vidnost 5G infrastrukture.

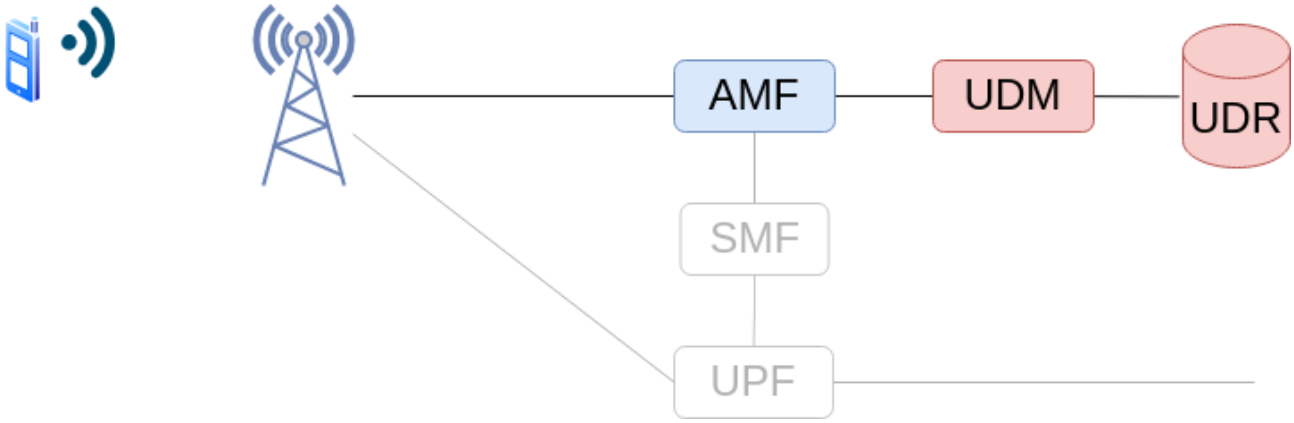
Posebej izpostavljen primer je uporaba prenosnih 5G jeder, ki delujejo zunaj klasičnih podatkovnih centrov – na primer jedro vgrajeno v vojaško vozilo ali nahrbtnik, namenjeno hitri postavitvi zasebnega omrežja v kriznih situacijah. V takšnih okoljih so naprave pogosto fizično dostopne napadalcem, kar še dodatno poveča tveganje za kompromitacijo zaupnih podatkov.

S pomočjo Intel® Software Guard Extensions in podobnimi tehnologijami smo razvili pristop, kjer lahko samo ena komponenta dostopa do nešifrirane verzije overitvenih podatkov in jim dodatno utrdili varnost pred več tipi napadov, zlasti pred napadalci s fizičnim dostopom do sistema.

2 Overitev v 5G

Vsaka uporabniška naprava (UE) mora 5G jedru dokazati verodostojnost svoje identitete, preden od 5G sistema zahteva omrežne storitve. Ta postopek overitve temelji na tem, da si USIM modul znotraj naprave in 5G jedro delita skupno skrivnost. Te skrivnosti hranita pri sebi in jih ne pošiljata neobdelane. 5G jedro jo uporabi pri ustvarjanju kriptografskega izziva, ki ga je mogoče rešiti samo lastnikom te iste skrivnosti. Izziv pošlje UE-ju, ki pa je praviloma edina naprava poleg 5G jedra, ki ima to skrivnost. Če reši izziv in pošlje pravilni odgovor, ga 5G sistem smatra kot dokaz identitete. Hkrati je en izmed rezultatov tega izziva simetričen ključ, ki si ga po overitvi lastita samo 5G omrežje in UE, in ga uporabljata za šifriranje nadaljnje komunikacije [1].

2.1. Sistem 5G



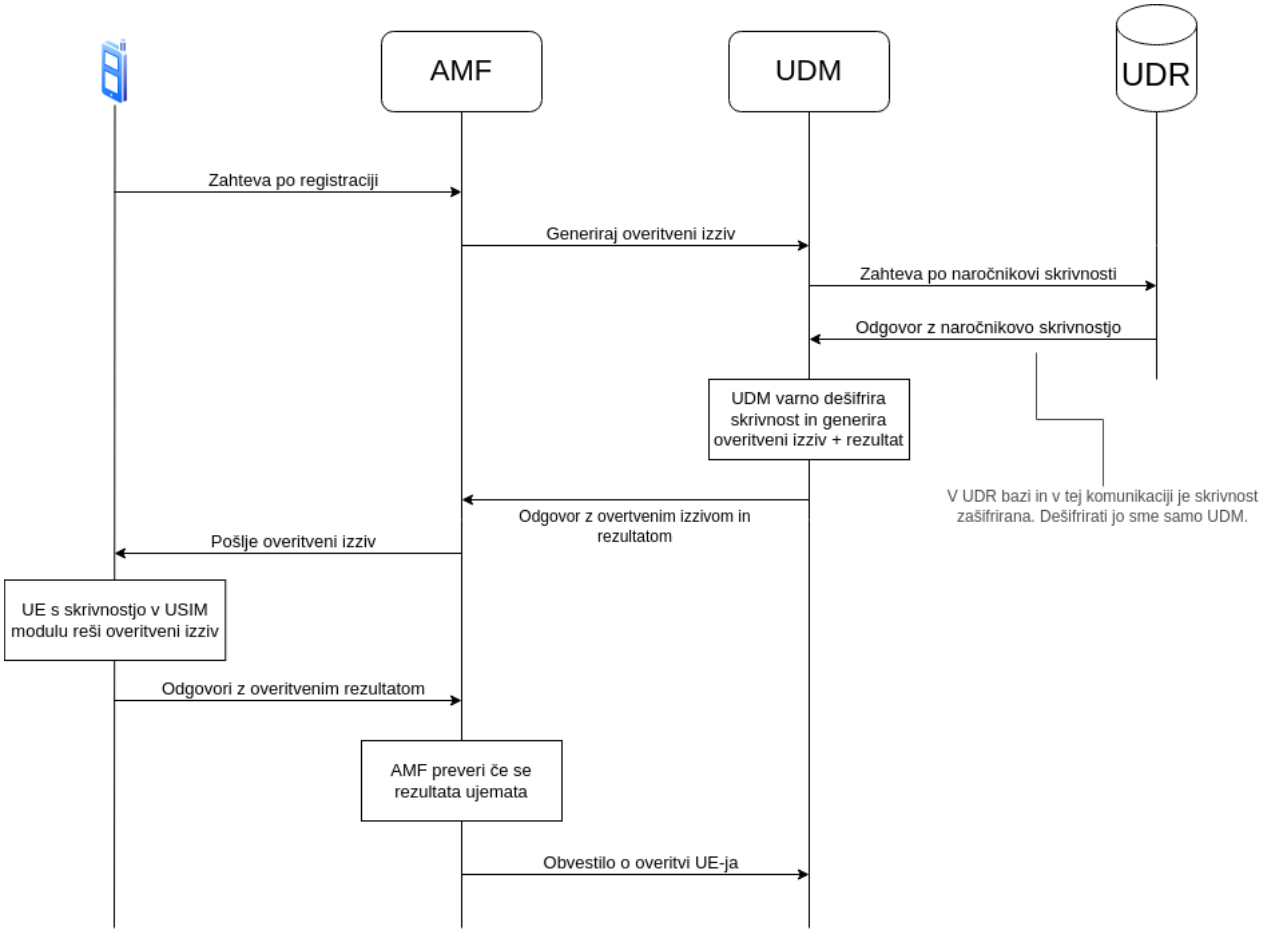
Slika 1: 5G omrežje.

5G sistem je sestavljen iz več elementov, kot je prikazano na sliki 1. Na levi sta ponazorjena UE in gNB antena, na desni so pa prikazani elementi 5G jedra. Ti tipično delujejo kot strežniki in med seboj komunicirajo preko mrežnih protokolov, največkrat preko HTTP/2. Osredotočamo se zlasti na AMF, UDM in UDR, ki sodelujejo pri overitvi UE. AMF se smatra kot vstopna točka v 5G jedro. UE-ji preko njega pridobivajo dostop do omrežnih storitev. UDM je pa med drugim zadolžen za ustvarjanje overitvenih izzivov iz podatkov, ki jih pridobi od UDR-ja. AUSF-ja, ki overitveni izziv priredi v interesu ohranjanja UE-jeve zasebnosti in overitven izziv tudi preveri, tukaj posebej ne izpostavljamo. Izziv (RAND) ter ostali avtentikacijski podatki se izračunajo v okviru komponente

ARPF, ki je del UDM. Namen poenostavitve je, da se osredotočimo na vidike, povezane z upravljanjem in zaščito ključev, kar spada v domeno UDM, medtem ko je AUSF v tem kontekstu obravnavan posredno.

Poleg AMF, UDM in UDR so v 5G jedru prisotne še ostale omrežne funkcije, ki omogočajo razne druge storitve. Kot primer sta na sliki 1 narisana SMF in UPF, ki overjenim UE-jem na AMF-jevo zahtevo nudita omrežno storitev.

2.2. Potek overitve



Slika 2: Potek overitve.

Na sliki 2 je na visokem nivoju prikazano, kako poteka overitev UE-ja v 5G omrežje [2]. Ob registracijski zahtevi UE-ja AMF pošlje UDM-ju zahtevo za pridobitev overitvenega izziva za tega uporabnika. UDM od UDR-ja pridobi zašifrirano skrivnost tega UE-ja, jo dešifrira ter na njeni podlagi ustvari izziv in pripadajoči pričakovani rezultat. Oboje posreduje AMF-ju, ki nato UE-ju pošlje le izziv. UE uporabi svojo lokalno skrivnost, shranjeno v USIM-modulu, za izračun odgovora na izziv in ga pošlje AMF-ju. AMF preveri, ali se prejeti odgovor ujema s pričakovanim rezultatom, ki ga je prejel od UDM-ja. Če se vrednosti ujemajo, to potrjuje, da UE razpolaga s pravilno skrivnostjo, kar pomeni, da je uspešno overjen.

5G sistem je fleksibilen in omogoča tudi gostovanje. V tem primeru UE komunicira z AMF-jem iz gostujoče mreže, AMF pa z UDM-jem iz UE-jeve domače mreže. Edino ta UDM ima namreč dostop do UE-jevih podatkov.

2.3. Varnost overitvenih podatkov

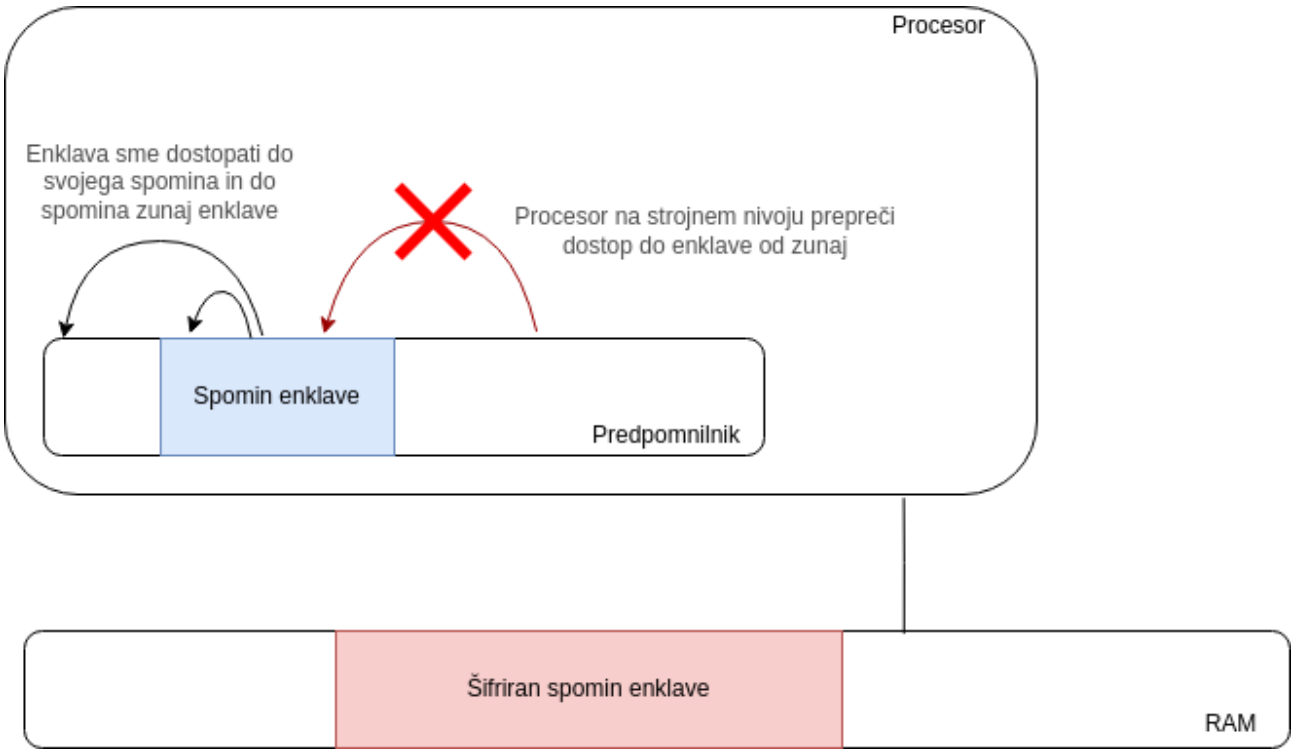
UDR-jeva baza UE-jevih skrivnosti je zašifrirana tako, da lahko podatke dešifrira samo UDM. Zato UDR-jeva baza sama po sebi ni občutljiva. UDM iz dešifrirane verzije skrivnosti ustvari overitveni izziv in rezultat, nato pa skrivnost izbriše. Velja tudi, da se iz izziva in rezultata ne da rekonstruirati skrivnosti, zato tudi nista sama po sebi občutljiva podatka. Najbolj občutljiv podatek je dešifrirana skrivnost, s katero ima pa opravka samo UDM, zato se osredotočamo zlasti na varno delovanje UDM-ja.

3 Predstavitev Intel® Software Guard Extensions (SGX)

Intel® Software Guard Extensions (SGX) je funkcionalnost na nekaterih Intelovih procesorskih modelih, ki omogoča zaščiteno zaupno procesiranje. Z njo definiramo enklave, to so območja računalniškega spomina, ki so na nivoju strojne opreme zaščitene pred okolico, vključno z napadalcem s fizičnim dostopom in ostalo programsko opremo na istem procesorju [3].

3.1. Enklave

Navaden proces v sodelovanju z operacijskim sistemom definira enklavo. Vanjo vpiše programsko kodo in podatke, ki so potrebni za varno procesiranje. Nato s SGX ukazom inicializira enklavo, po tem pa lahko vanjo večkrat vstopi in izstopi, kar pomeni, da začne poganjati kodo znotraj enklave.



Slika 3: Vizualizacija enklave.

Po inicializaciji je enklava zaščitena pred branjem in pisanjem iz okolice, kot je ponazorjeno na sliki 3. Procesor prepreči posege ostalim procesom na strojnem nivoju. S tem je izboljšana zaščita pred programskimi vdori v katerikoli proces zunaj enklave. Pomembno pa je tudi, da je v veliki meri poseg preprečen tudi napadalcem s fizičnim dostopom do sistema. Spomin v enklavi nikoli ne zapusti procesorja nešifriran. Preden se podatki zapišejo v RAM, se znotraj procesorja zašifrirajo in njihova celovitost se zaščiti. Napadalec z dostopom do fizičnega RAM-

a ali vodil med RAM-om in procesorjem ne more brati podatkov. In tudi če podatke skuša prirediti, CPU to zanesljivo zazna kot okvaro celovitosti in enklavo podre.

Če želi napadalec s fizičnim dostopom brati ali prirejati podatke, mora posegati v sam procesor. Zaradi majhnosti vezji na procesorju se to smatra kot zelo drag poseg [5]. Cenejši posegi so destruktivni in jih ni težko zaznati, ker mora napadalec za dalj časa vzeti procesor v analizo, preden ga prebere in nadomesti z delujočo kopijo.

3.2. Funkcionalnosti enklave

Koda, ki se izvaja znotraj enklave, je skoraj popolnoma poljubna in ima samo nekaj specifičnih omejitev zaradi tehničnih razlogov [4]. Dodatni stroški vstopov v enklavo in poganjanja enklav so tudi majhni. To nam omogoča fleksibilnost pri načrtovanju varnostnih sistemov s pomočjo SGX. Na primer, programiramo lahko poljubne algoritme za šifriranje in izpeljavo simetričnih ali asimetričnih ključev, operiramo lahko s poljubnim številom ključev glede na podatke in ustvarjamo kriptografske izzive po poljubnih standardih.

Poleg tega nam SGX v okviru enklave omogoča razne zanimive dodatne funkcionalnosti. Pomemben primer je ustvarjanje tako imenovanega pečatnega ključa. Znotraj enklave koda s SGX ukazom pridobi pečatni ključ, za katerega SGX zagotavlja, da ga je možno ustvariti samo v tej specifični enklavi na tem specifičnem procesorju. Z njim lahko enklava zašifrira občutljive podatke ter jih zapiše na nezaščiteno mesto zunaj enklave, na primer na disk z namenom, da po ponovnem zagonu sistema podatke prebere ter dešifrira s pečatnim ključem.

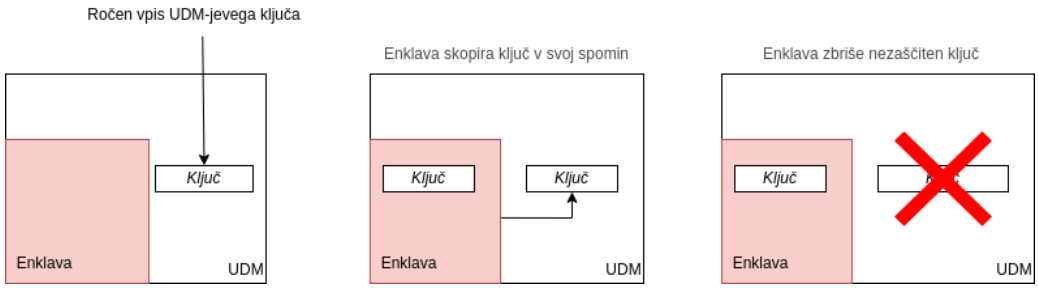
4 Zaščita overitvenih podatkov s pomočjo SGX

5G jedro hrani overitvene podatke svojih UE-jev v UDR-jevi naročniški bazi, ki mora biti ves čas na voljo za neprekinjeno zagotavljanje storitev. Baza vsebuje naročnikove skrivnosti, ki so eden najbolj občutljivih podatkov v 5G sistemu. Napadalec, ki mu uspe pridobiti te podatke, lahko namreč pooseblja tuje UE-je in s tem pride do nepooblaščenih storitev in podatkov ali povzroča različne vrste škode drugim UE-jem in celotnemu omrežju. Zato imamo v interesu čim močnejšo zaščito teh podatkov.

5G standard predpisuje, da so skrivnosti v UDR-ju šifrirani in da ima UDM možnost te skrivnosti na varen način dešifrirati. S tem je UDR-jeva baza overitvenih podatkov zaščitena pred branjem. Bazo lahko tudi brez problema varnostno kopiramo in po želji repliciramo.

4.1. Integracije SGX na UDM

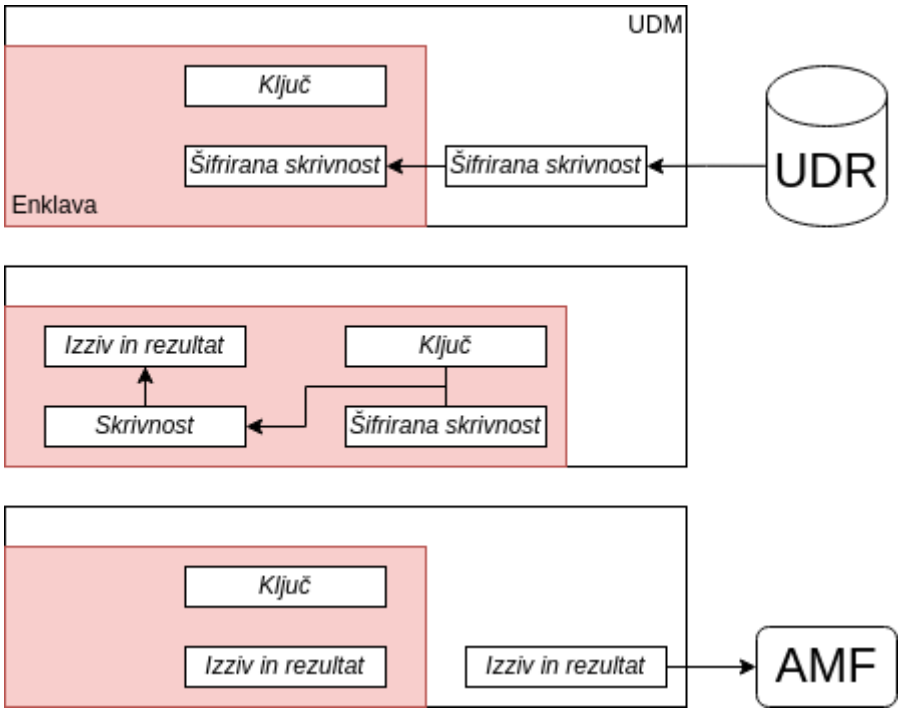
Naš pristop deluje tako, da najprej ustvarimo simetričen ključ, ki ga kličemo UDM-jev ključ. Pri prvem zagonu UDM-ja ga na roke nezaščiteno vnesemo v UDM aplikacijo. Postopek je orisan na sliki 4. UDM nato zažene enklavo, ki ključ skopira v svoj zaščiten spomin, njegov nezaščiten original pa zbriše iz spomina. Predpostavljamo, da imamo v tem časovnem oknu, ko je ključ še nezaščiten v UDM-jevem spominu, dovolj nadzora nad sistemom, da napadalec ne more prebrati ključa. Na primer, to prvo oskrbo izvajamo z izklopljenim mrežnim kablom.



Slika 4: Zagon UDM-ja.

UDM-jeva enklava nato zapečati simetričen ključ, kar pomeni, da ga zašifrira s svojim pečatnim ključem, ter ga shrani na disk. SGX zagotavlja, da je UDM-jeva enklava edina, ki pozna pečatni ključ, zato je z njim zašifriran simetričen ključ varen pred okolico. Če se strežnik ponovno zažene, ali če UDM ponovno zaženemo, na primer zaradi posodobitve, potem ponovni ročni simetričnega ključa ni potreben. UDM po ponovnem zagonu samodejno prebere zapečateni ključ iz diska, enklava ga pa odpečati.

Kot je prikazano na sliki 5, pošlje UDR za vsako overitev UDM-ju šifrirano skrivnost. UDM-jeva enklava jo prebere v zaščiten spomin in jo s simetričnim ključem dešifrira, nato pa iz nje izračuna overitveni izziv. Le tega skopira v spomin izven zaščitenega območja, da ga nato UDM pošlje naprej AMF-ju. Na tak način dešifrirane skrivnosti nikoli ne zapustijo zaščitenega spomina enklave.



Slika 5: UDM dešifrira in uporabi UE-jevo skrivnost.

Ta pristop ima tudi to prednost, da nam bistveno ne oteži replikacije UDM-jev za namen visoke razpoložljivosti. Z enakim postopkom zaženemo UDM na več strežnikih in enklava bo na vsakem v disk zapisala zapečaten simetričen ključ. Avtomatizacija nato poljubno zaganja in ugaša replike UDM-ja na teh strežnikih brez potrebe ročnega vnosa simetričnega ključa.

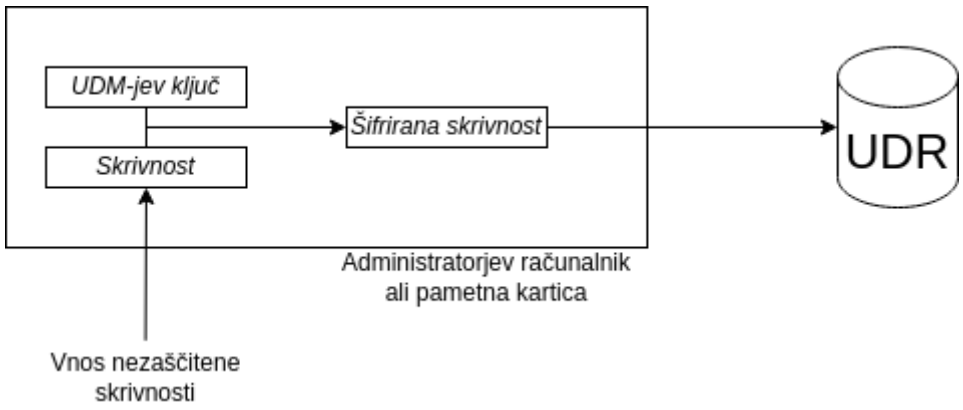
4.2. Varnostna kopija simetričnega ključa

Varnostna kopija UDR-jeve baze same ni dovolj, ker je šifrirana in bi ob izgubi UDM-jevega simetričnega ključa izgubili možnost dešifriranja podatkov. Zato je potrebno varnostno kopirati tudi simetričen ključ, za kar tipično uporabimo eno izmed shem delitve skrivnosti [6]. Tako ima več lastnikov pri sebi shranjene koščke ključa na tak način, da potrebujemo podatke od določene podmnožice lastnikov za rekonstruiranje ključa.

4.3. Vnos novih podatkov v UDR bazo

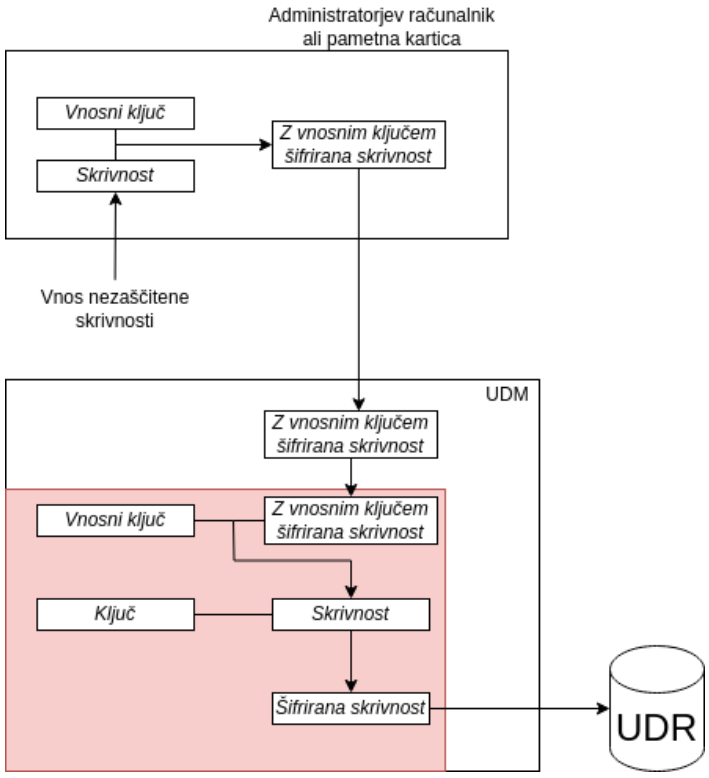
Za vnos novih naročnikov v UDR bazo potrebuje administrator UDM-jev simetričen ključ, da šifrira naročnikove skrivnosti pred vpisom. Postopek je prikazan na sliki 6. Za varnejše upravljanje ima možnost, da podatke šifrira na pametni kartici ali podobnem HSM-ju, ki zagotavlja varno shrambo ključa. V primeru delitve skrivnosti med več

lastnikov je pa za vnos novega naročnika potrebna prisotnost več lastnikov hkrati, kar na račun oteženi priročnosti poveča varnost.



Slika 6: Vnos skrivnosti novega naročnika v UDR-jevo bazo.

Druga možnost je uporaba dodatnega simetričnega ključa, imenovanega vnosni ključ, ki se uporablja izključno za namen vnosa novih naročnikov. Tega hranijo administratorji, ki želijo v UDR bazo vnašati nove naročnike. UDM-jevi enklavi ob zagonu poleg UDM-jevega simetričnega ključa priskrbimo še vnosni ključ. Na sliki 7 je ponazorjeno, kako administrator z vnosnim ključem zašifrira skrivnost novega naročnika in jo pošlje UDM-ju, po možnosti na daljavo.



Slika 7: Vnos skrivnosti novega naročnika preko vnosnega ključa.

UDM-jeva enklava to skrivnost z vnosnim ključem dešifrira in nazaj zašifrira z UDM-jevim simetričnim ključem, nato pa jo vpiše v UDR-jevo bazo. Enklava mora biti programirana tako, da vnosni ključ uporabi samo za namene šifriranja in nikoli dešifriranja. Zagotoviti moramo tudi, da se operacije dešifriranja kriptografsko ne da zlorabiti za pomoč pri šifriranju. Na primer, v shemi AES-CTR se IV nikoli ne sme ponoviti [7]. Na tak način zagotovimo, da vnosni ključ ne more omogočiti dostopa do skrivnosti že vpisanih naročnikov in je zato manj občutljiv od UDM-

jevega ključa. To je kar zaželena lastnost, saj se vnosni ključ tipično potrebuje bolj pogosto od UDM-jevega ključa, ki je potreben zgolj za postavitev UDM-ja na nov strežnik. Omogočena je tudi fleksibilnejša razdelitev nalog med administratorji, kateri bodo hranili UDM-jev ključ in kako, ter kateri bodo hranili vnosni ključ.

5 Alternativne rešitve za zaščito overitvenih skrivnosti

Čeprav Intel® SGX ponuja učinkovito rešitev za zaščito overitvenih skrivnosti znotraj procesorskih enklav, obstajajo tudi druge arhitekture, ki omogočajo primerljivo varnost z drugačnimi pristopi.

5.1. Uporaba strojnih varnostnih modulov (HSM)

Strojni varnostni moduli (HSM – Hardware Security Modules) so namensko grajene naprave, zasnovane za varno upravljanje s kriptografskimi ključi in občutljivimi operacijami. Podobno kot SGX omogočajo tudi HSM-ji, da se skrivnosti obdelujejo v izoliranem, strojno zaščitenem okolju. V kontekstu 5G overitve lahko arhitektura deluje tako, da:

- UDM podatke iz UDR-ja posreduje HSM-ju;
- HSM izvede dešifriranje skrivnosti in generacijo overitvenega izziva;
- izziv se preko UDM-ja posreduje AMF-ju, brez da bi dešifrirane skrivnosti zapustile HSM.

HSM pa med tem zagotavlja varnost simetričnega ključa in skrivnosti po podobnem principu kot opisana SGX enklava v poglavju 4.1. HSM-ji zagotavljajo trajno zaščito ključa, možnost varnostnega revizorstva in so primerni za okolja, kjer je zanesljivost preverjene strojne opreme ključna (npr. bančništvo, vojska, kritične infrastrukture).

5.2. Programske rešitve z uporabo sistemov za upravljanje tajnosti

Poleg rešitev, ki temeljijo na strojni izolaciji (npr. SGX ali HSM), lahko podobno arhitekturo za zaščito overitvenih skrivnosti uresničimo tudi povsem programsko z uporabo sistemov za upravljanje tajnosti. Za okolja, kjer strojna zaščita ni izvedljiva ali ekonomsko upravičena, so primerna orodja za programsko zaščito skrivnosti, kot je HashiCorp Vault.

Vault sistem omogoča, da se skrivnosti hranijo centralizirano in šifrirano, dostop pa je omogočen le avtoriziranim komponentam preko varnostnih politik (RBAC). Dodatno Vault podpira tudi uporabo t. i. "transit engine", s katerim se kriptografske operacije, kot je generacija overitvenih izzivov, izvajajo neposredno v Vaultu, ne da bi bila skrivnost kadarkoli izpostavljena zunanjemu okolju.

V 5G arhitekturi lahko tako UDM prevzame vlogo posredovalca: pridobi podatke od UDR, jih posreduje Vaultu, ta izvede potrebne operacije in vrne rezultat (npr. overitveni izziv), ki ga UDM nato posreduje naprej proti AMF. Ta pristop je z vidika integracije zelo fleksibilen in primeren za okolja, kjer fizična strojna izolacija ni mogoča ali je neučinkovita in se uporablja npr. za oblačna in porazdeljena okolja.

5.3. Primerjava pristopov

Lastnost	Intel SGX	HSM	SW
Tip zaščite	Procesorska enklava	Namenska strojna oprema	Programska rešitev
Fizična varnost	Srednja–visoka	Zelo visoka	Odvisno od okolja
Integracijska zahtevnost	Nizka–srednja	Visoka	Nizka
Primernost za mobilna 5G jedra	Da	Omejena	Da
Razširljivost	Srednja	Omejena	Visoka
Stroškovna učinkovitost	Srednja	Nizka	Visoka

6 Nadaljnja raziskava - SGX-ova oddaljena atestacija

SGX omogoča tako imenovano oddaljeno atestacijo [8]. To je proces, s katerim enklava nekemu drugemu sistemu dokaže, da vsebuje avtentično kodo in podate, in da se resnično izvaja na SGX zaščiteni strojni opremi. Omogoča tudi delitev podatka med oddaljenim sistemom in enklavo z zagotovitvijo, da je ta podatek res enak in da ni nihče mogel posegati vanj in ga spremeniti.

To je precej močno orodje, ki odpira možnosti načrtovanja zanimivih novih arhitektur ali za izboljšavo obstoječe rešitve. Na primer, začetni vnos UDM-jevega in vnosnega ključa bi lahko elegantneje potekal preko varnega komunikacijskega kanala med oddaljenim sistemom in UDM-jevo enklavo, vzpostavljenega s pomočjo atestacije.

7 Zaključek

Overitev v 5G sistemu zahteva varno upravljanje z občutljivimi uporabniškimi podatki. Tehnologija Intel SGX nudi možnost varnega obdelovanja zaupnih podatkov in nam omogoča implementacijo sistema, ki ustreza tej 5G varnostni zahtevi. Predstavljeni pristop uporabi SGX tako, da noben občutljiv podatek nikoli ne zapusti zaščitenega območja za katerega je zagotovljena močna varnost pred zunanjimi napadalci. Hkrati načrtuje način, kako upravljati z varnostnimi kopijami podatkov, ter kako zaščititi postopek vpisa novih uporabniških podatkov v bazo znotraj 5G sistema.

Poleg opisanega pristopa so predstavljene tudi alternative za upravljanje z občutljivimi podatki, skupaj z njihovimi ključnimi značilnostmi, prednostmi in slabostmi. Končna izbira rešitve je odvisna predvsem od konkretnega primera uporabe ter stopnje zahtevanih varnostnih mehanizmov.

Literatura

[1] David Basin, Jannik Dreier, Lucca Hirschi, Saša Radomirović, Ralf Sasse, Vincent Stettler, "A Formal Analysis of 5G Authentication", arXiv:1806.10360.

[2] 3rd Generation Partnership Project; Technical Specification Group, "Security architecture and procedures for 5G system.", TS 33.501, v17.5.0.

[3] https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/linux-overview.html, Intel® SGX SDK for Linux* OS, obiskano 10. 7. 2025.

[4] Intel®, Intel® Software Guard Extensions Programming Reference, 2014.

[5] Victor Costan, Srinivas Devadas, "Intel SGX Explained", Cryptology {ePrint} Archive, Paper 2016/086, 2016, https://eprint.iacr.org/2016/086.

[6] https://en.wikipedia.org/wiki/Secret_sharing, Secret sharing – Wikipedia, obiskano 2. 4. 2025.

[7] https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#CTR, Block cipher mode of operation – Wikipedia, obiskano 22. 7. 2025.

[8] https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/attestation-services.html, Attestation Services for Intel® Software Guard Extensions, obiskano 22. 7. 2025.