O praktičnoj primeni vremenskih žigova upotrebom ulančane šeme

Dragan Spasić
Javno preduzeće PTT saobraćaja "Srbija"
Beograd, Srbija
dspasic@ptt.rs

Branislav Milojković
Računarski fakultet Univerziteta Union
Beograd, Srbija
bmilojkovic07@raf.edu.rs

Stevan Milinković
Računarski fakultet Univerziteta Union
Beograd, Srbija
smilinkovic@raf.edu.rs

***Sadržaj* - U ovom radu dat je kratak opis praktičnih šema sistema za izdavanje vremenskih žigova, uz objašnjenje njihovih prednosti i nedostataka. Navedena su dva tipa ulančanih šema i pokazano je da su obe primenljive u praksi. U skladu sa tim, opisano je funkcionisanje jednog softverskog rešenja baziranog na delimično uređenoj ulančanoj šemi. Ukazano je na problem ispravnog projektovanja sistema, posebno u odnosu na obnavljanje vremenskih žigova. Istaknuto je da je za širu praktičnu primenu ovih šema glavna prepreka zakonska regulativa, koja se i dalje zasniva na primeni proste šeme.**

Ključne reči - vremenski žig; TSA; ulančana šema

I. UVOD

Elektronski dokumenti su vrlo osteljivi u smislu da se lako menjaju ili brišu, tako da je od izuzetne važnosti mogućnost dokazivanja da je elektronski dokument kreiran određenog datuma u određeno vreme. Nezamenljivu ulogu u tome ima upotreba vremenskih žigova kao sredstava za vremensku autentifikaciju, tj. njihova mogućnost da dokažu da je određeni dokument nastao pre nekog vremenskog trenutka. Ovo je obavezno svojstvo u primenama kao što su podnošenje zahteva za patente, elektronsko glasanje ili elektronsko poslovanje, gde su moguće prevare direktno vezane sa gubitkom novca. Vremenski žigovi mogu da se koriste i kao sredstvo za sprečavanje poricanja. Digitalni potpis je zakonski validan samo ako je napravljen kada je i potpisnikov sertifikat takođe bio validan. Iako je infrastruktura javnog ključa robustan mehanizam, verifikacija potpisa zahteva pristup bazi podataka sertifikacionog tela koja sadrži kopiju relevantnih sertifikata. Imajući u vidu da je sertifikacionom telu nepraktično da dozvoli ovakav pristup isteklim sertifikatima, validacija dokumenata će biti znatno otežana i skupa. S druge strane, kako tehnike za razbijanje kodova sve više napreduju, dužina kriptografkih ključeva povremeno mora da se povećava. Međutim, ova promena dužine ključeva ne može da se primeni retroaktivno, na već postojeće potpisane dokumente. Poseban izazov predstavlja činjenica da nismo sigurni kako u dalekoj budućnosti da verifikujemo dugoživeće dokumente. Treba imati na umu da, radi sprečavanja kompromitovanja privatnog ključa, parovi ključeva u kriptosistemu javnog ključa imaju ograničen vek trajanja, koji može da bude kraći nego što je trajanje dokumenta. Na žalost, digitalni potpis nije dovoljan da se obezbedi neporicanje. Ukoliko želimo da poreknemo svoj potpis, dovoljno je da tvrdimo da su podaci već bili kompromitovani u trenutku upotrebe potpisa. Međutim vremenski žig koji je povezan sa potpisanim dokumentom može da spreči takvo poricanje.

II. ŠEME IZDAVANJA VREMENSKIH ŽIGOVA

Šeme izdavanja vremenski žigova su u literaturi klasifikovane u tri različita tipa. Proste i ulančane šeme su primenljive u praksi, dok su distriburine šeme još uvek u fazi teorijskih razmatranja i ekperimentalnih ispitivanja.

Kod proste šeme, za vremenski žig zaduženo je telo u koje imamo poverenja (Time Stamping Authority, TSA) i to na takav način da ne uključuje podatke vezane za ostale vremenske žigove. Prostu šemu jednostavno je implementirati, a vremenski žigovi od različitih TSA mogu da se porede.

Vremenski žig po prostoj šemi se izdaje na sledeći način:

- Klijent šalje dokument X (ili njegovu hash vrednost) TSA serveru;
- TSA pridružuje tekuće vreme t i TSA identifikator ID dokumentu, a zatim elektronski potpisuje ovako dobijeni kompozitni dokument (ID, t, X);
- TSA vraća klijentu vrednosti: t i $s = sig_{TSA}(ID, t, X)$.

Zakonska regulativa velikog broja zemalja zahteva da izdavaoci vremenskih žigova rade u skladu sa protokolom RFC3161 [1] koji je zasnovan je na prostoj šemi. Na osnovu toga razvijena je znatna količina komercijalnog softvera, kao i softvera otvorenog koda [2]. Međutim, glavni problem kod proste šeme je da moramo da imamo neograničeno poverenje u TSA. Ukoliko bi TSA iz bilo kod razloga izmenio vremenski parametar nekog vremenskog žiga, to niko ne bi mogao da otkrije. Takođe, ako bi došlo do krađe TSA privatnog ključa, neko drugi bi mogao da bez ikakvih problema izdaje lažne vremenske žigove.

Neophodnost ovako velikog poverenja u TSA može da se znatno umanji upotrebom ulančanih šema. Najpoznatiju šemu ovog tipa objavili su Haber i Stornetta [3]. Osnovna ideja njihove ulančane šeme je generisanje vremenkog žiga koji obuhvata podatke i drugih vremenskih žigova. Na taj način izgrađuje se lanac koji je povezan upotrebom jednosmernih hash funkcija. Ukoliko bi neko namerno pokušao da promeni vremenski žig, on bi to morao da uradi i sa ostalim žigovima u lancu. Ovo je glavna prednost u odnosu na prostu šemu, ali s duge strane imamo komplikovaniju verifikacionu proceduru, jer je za nju potrebno neposredno učešće TSA. Ulančana šema tipično se sastoji od tri faze: agregacija, ulančavanje i objavljivanje.

Agregacija: U ovoj fazi, svi zahtevi koje je primio TSA u okviru kratkog vremenskog intervala (agregaciona runda) tretiraju se kao istovremeni. Rezultat agregacione runde je binarni niz čiji sadržaj zavisi od svih dokumenata u toj rundi. Osnovni cilj agregacije je da se smanji opterećenje TSA, ukoliko je ulančavanje skupa operacija. U suprotnom, agregacija može da se svede i na specijalan slučaj, kada u jednoj rundi imamo samo jedan zahtev.

Ulančavanje: Rezultat agregacije se ulančava sa agregacionim vrednostima dobijenim u prethodnim rundama, pri čemu se ulančavanje ne može izvršiti bez prethodnih agregacionih vrednosti. Ovo uspostavlja jednosmerno uređenje između agregacionih vrednosti pojedinih rundi, kada se dobija tzv. relativna vremenska autentifikacija: vremenski žigovi različitih agregacionih rundi mogu da se porede. Ovo dalje ukazuje i na to da vremenski parametar nije nužno sastavni deo ulančane šeme.

Objavljivanje: S vremena na vreme, TSA objavljuje u nekom od široko dostupnih medijuma (npr. u dnevnim novinama) varijantu poslednjeg vremenskog žiga, čime sebe vezuje za sve prethodno izdate vremenske žigove. Objavljene vrednosti se koriste za verifikaciju vremenskih žigova, kao i za proveru da li TSA ispravno funkcioniše.

A. Totalno uređena ulančana šema

Neka je $Y = (y_1,\ldots, y_N)$ niz k-bitnih brojeva. Pod pojmom *uređenog akumulatora* podrazumevamo triplet algoritama (A, P,V) gde su:

- A akumulacioni algoritam koji, uz zadato $n\in\{1,\ldots,N\}$ i niz $(y_1,\ldots,y_n)$, na svom izlazu daje k-bitni broj $A(n,(y_1,\ldots,y_n))$ koji se naziva n-ti akumulirani sažetak (digest) poruke.
- P algoritam za generisanje dokaza koji, uz date Y i n kao argumente, na svom izlazu daje sertifikat n-tog reda $c(n)=C(n,Y)$ koji sadrži informaciju o relativnom položaju y_n unutar Y.
- V verifikacioni algoritam koji, uz dati broj y i par sertifikata (c,c'), sažetak poruke $a=A(n, y_1,\ldots,y_{n-1})$ i indeks $n\in\{1,\ldots,N\}$, daje odgovor 'tačan' ako c i c' zajedno sačinjavaju korektan dokaz da $y_m\in\{y_1,\ldots,y_{n-1}\}$. Pretpostavlja se da je za $m<n$:

$$V(n,y_m,C(m,y_1,\ldots,y_N),C(n,y_1,\ldots,y_N), A(n, y_1,\ldots,y_n)) = \texttt{true}.$$

Potencijalnom napadaču E bilo bi veoma teško da pronađe korektan dokaz za brojeve $y\in\{y_1,\ldots,y_{n-1}\}$, tj. verovatnoća

$$P[(n,y,y_1,\ldots,y_N,c,c')\leftarrow E,\ a\leftarrow A(n, y_1,\ldots,y_n)\colon y\notin\{y_1,\ldots,y_{n-1}\},\ V(n,y,c,c',a) = \texttt{true}]$$

je zanemarljiva.

Primer ovakvog pristupa je linearna ulančana šema [3].

B. Delimično uređena ulančana šema

Neka je $Y = (y_1,\ldots, y_N)$ niz k-bitnih brojeva. Pod pojmom *jednosmernog akumulatora* podrazumevamo triplet algoritama (A, P,V) gde su:

- A akumulacioni algoritam koji preslikava Y u broj $A(Y)$ koji se naziva akumulirani sažetak poruke.
- P algoritam za generisanje dokaza koji, uz date Y i n kao argumente, na svom izlazu daje dokaz $p(n) = P(n,Y)$ koji potvrđuje da je y_n upotrebljen prilikom izračunavanja akumuliranog sažetka poruke $A(Y)$.
- V verifikacioni algoritam koji, uz dati triplet $(y, A(Y), p)$, daje odgovor 'tačan' ako i samo ako je p korektan dokaz da je y korišćen prilikom izračunavanja $A(Y)$.

Potencijalnom napadaču E bilo bi veoma teško da pronađe korektan dokaz za brojeve koji nisu u Y (čak i u slučaju da mu je poznat niz Y), tj. verovatnoća

$$P\ [\ (y, Y, p) \leftarrow E : y \notin Y,\ V(y, A(Y), p) = \texttt{true}\ ]$$

je zanemarljiva.

Primer ovog pristupa je upotreba Merkleovog stabla [4].

Delimično uređena ulančana šema je znatno efikasnija i jednostavnija za implementaciju jer funkcioniše na jednom skupu zahteva (jednoj agregacionoj rundi), bez obzira na broj zahteva koji se obrađuju. Totalno uređena ulančana šema zahteva jednu obradu po svakom zahtevu. S druge strane, totalno uređena ulančana šema omogućuje upoređivanje bilo kojih vremenskih žigova, dok je kod delimično uređene ulančane šeme to moguće samo za žigove iz različitih rundi.

U Tabeli 1 navedene su karakteristike četiri različita rešenja za zaštitu integriteta elektronskog dokumenta, prema funkcionalnostima koje su numerisane na sledeći način:

1. Rešenje se bazira na standardu.
2. Mogućnost detektovanja narušavanja integriteta e-dokumenta.
3. Mogućnost uključivanja identiteta pri zaštiti integriteta e-dokumenta.
4. Evidentiranje tačnog vremena na siguran način pri zaštiti integriteta e-dokumenta.
5. Sprečavanje promene dokumenta od strane vlasnika dokumenta posle zaštite integriteta.
6. Nemogućnost kompromitacije.
7. Obezbeđivanje integriteta tokom dugog vremenskog perioda.
8. Mogućnost nezavisne provere integriteta.

TABELA I. POREĐENJE REŠENJA ZA ZAŠTITU INTEGRITETA ELEKTRONSKOG DOKUMENTA

Funkc.	Žigosanje upotrebom ulančane liste	RFC 3161	E-potpis	Hash
1	da	da	da	da
2	da	da	da	da
3	da	da	da	ne
4	da	da	ne	ne
5	da	da	ne	ne
6	da	ne	ne	ne
7	da	ne	ne	ne
8	da	ne	ne	ne

U odnosu na rasprostranjenu prostu šemu, vremensko žigosanje upotrebom ulančane šeme je superiornije po pitanju tačaka 6, 7 i 8. Pri tome su zadržane sve funkcionalnosti koje poseduje RFC 3161.

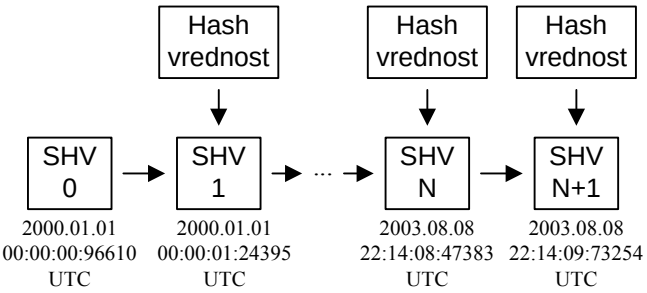
Još jedna važna prednost vremenskog žigosanja sa ulančanom šemom je u tome što ne mora da se primenjuje elektronsko potpisivanje vremenskih žigova, odnosno što ne moraju da se koriste kriptografski ključevi za potpisivanje. To u praksi znači da izdavalac vremenskih žigova ne mora da poseduje složenu PKI infrastrukturu, pa shodno tome ne postoje inherentni PKI rizici (kompromitovanje privatnog ključa ili isticanje važenja TSA sertifikata). Umesto elektronskog potpisivanja, primenjuje se izračunavanje hash vrednosti žigosanog dokumenta kome se pridružuje tačno vreme i tako kreira vremenski žig, a hash vrednosti svih žigosanih dokumenata se ulančavaju u agregatnu hash vrednost (Aggregate Hash Value - AHV), koja se periodično javno objavljuje. S obzirom na to da AHV može da se proveri na osnovu objavljenih podataka, nemoguće je izdati vremenski žig sa netačnim vremenom ili uspešno verifikovati elektronski dokument koji prethodno nije vremenski žigosan, jer bi se to evidentiralo preko promene AHV vrednosti. Svi kreirani vremenski žigovi smeštaju se u bazu podataka, koja se naziva Univerzalni registar.

III. PRAKTIČNO VREMENSKO ŽIGOSANJE UPOTREBOM ULANČANE ŠEME

Pojavom standarda ISO/IEC 18014-3 [5] i ANSI X9.95 [6] stvorili su se uslovi za primenu ulančane šeme u praksi, pa je u skladu sa tim započeo i razvoj odgovarajućih softverskih rešenja. U nastavku je opisan softver AbsoluteProof kompanije Surety [7] koji predstavlja tipičnu praktičnu implementaciju delimično uređene ulančane šeme.

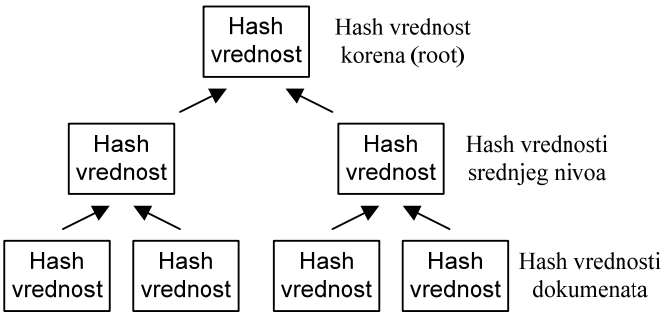
A. Univerzalni registar i ulančavanje hash vrednosti

Ispravnost rada sistema zasniva se na zaštiti integriteta Univerzalnog registra. Ako bi se izmenio sadržaj Univerzalnog registra, to bi značilo da su svi izdati vremenski žigovi kompromitovani tj. neispravni. Da bi se obezbedio integritet izdatih vremenskih žigova, server kriptografski ulančava hash vrednosti iz vremenskih žigova. To se izvršava izračunavanjem zbirne hash vrednosti (Summary Hash Value - SHV), a izračunavanje se sprovodi nad svim hash vrednostima izdatih vremenskih žigova u datom trenutku. Proces započinje od SHV0 kome se pridružuje hash vrednost dokumenta koji se vremenski žigoše. Zatim, izračunavanjem njihove hash vrednosti dobija se SHV1, itd. (Sl. 1). Za svaku hash vrednost koja je ulančana u lanac, u Univerzalnom registru se čuvaju hash vrednost, vremenski žig i sledeća sumarna hash vrednost.



Slika 1. Lanac zbirnih hash vrednosti (SHV)

Kada serveru u istom trenutku stigne više zahteva za izdavanje vremenskih žigova (jedna runda), hash vrednosti dokumenata se povezuju u strukturu binarnog stabla (Sl. 2). Listovi tog stabla su hash vrednosti dokumenata koji se vremenski žigošu. Ukoliko je potrebno, ubacuju se slučajne ili standardne vrednosti pojedinih listova, da bi njihov broj uvek bio paran. Na osnovu uparenih hash vrednosti dokumenata, primenom hash algoritama izračunavaju se hash vrednosti srednjeg nivoa, a zatim se izračunava hash vrednost korena stabla. Hash vrednost korena se povezuje sa poslednjim SHV u lanac zbirnih hash vrednosti. U praktičnom radu, server na opisan način obrađuje sve hash vrednosti dokumenata u okviru određenog vremenskog perioda. Na taj način se obezbeđuju bolje performanse sistema i značajno smanjuje obim podataka koje je potrebno uskladištiti.



Slika 2. Primer Merkleovog binarnog stabla

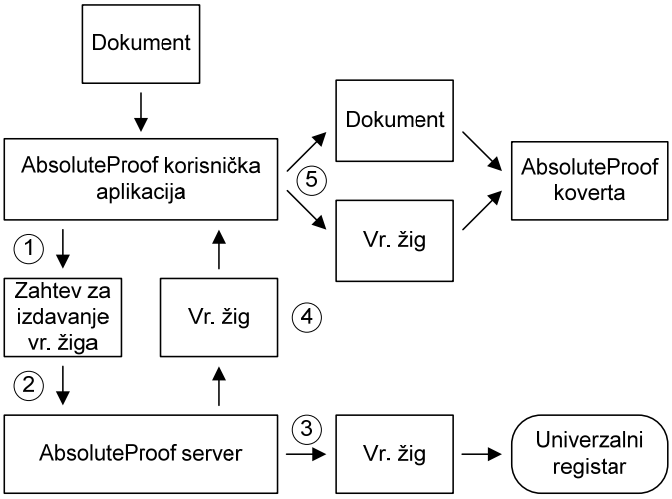
AHV se takođe izračunava formiranjem Merkleovog stabla od SHV u lancu. Listovi stabla su SHV, a koren je AHV [8].

B. Postupak vremenskog žigosanja dokumenta

Postupak vremenskog žigosanja elektronskog dokumenta ulančanom šemom sastoji se od sledećih koraka (Sl. 3):

1. Korisnička aplikacija izračunava hash vrednost dokumenta. Ova vrednost se izračunava primenom dva hash algoritma, SHA-256 i RIPEMD-160, a dobijene hash vrednosti se objedinjuju.
2. Korisnička aplikacija kreira zahtev za izdavanje vremenskog žiga u kome se nalazi hash vrednost dokumenta, a zatim zahtev šalje serveru.

3. Server dobijenoj hash vrednosti dokumenta pridružuje tačno vreme i tako kreira vremenski žig koji smešta u Univerzalni registar.
4. Server šalje vremenski žig korisničkoj aplikaciji.
5. Korisnička aplikacija, nakon prijema vremenskog žiga, kreira kovertu koja sadrži žigosan dokument i vremenski žig.



Slika 3. Vremensko žigosanje dokumenta

Dobijeni vremenski žig sastoji se od sledećih podataka:

- Naziv standarda po kojem je kreiran vremenski žig.
- Vreme kreiranja vremenskog žiga.
- Broj zone.
- Naziv hash algoritma čijom primenom je izračunata hash vrednost žigosanog dokumenta (nekada su se koristili sada već napušteni algoritmi MD5 i SHA-1).
- Hash vrednost žigosanog dokumenta.
- Serijski broj vremenskog žiga.
- Naziv izdavaoca vremenskog žiga.

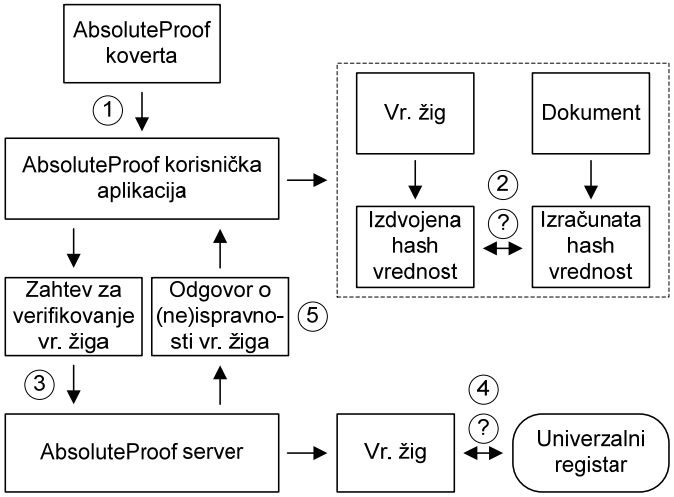
Vremenskom žigu mogu naknadno da se pridruže:

- Podaci potrebni za verifikovanje žigosanog dokumenta bez konsultovanja servera, tj. podaci za nezavisno vertifikovanje. Tu pre svega spada zbirna hash vrednost, koja se periodično javno objavljuje. S obzirom da zbirna hash vrednost koja odgovara jednom vremenskom žigu ne postoji u trenutku kreiranja tog vremenskog žiga, ona prilikom kreiranja vremenskog žiga ne može da mu se doda. Dodavanje zbirne hash vrednosti vremenskom žigu može da se uradi tek posle njenog objavljivanja. Zbirna hash vrednost može da se koristi ne samo kao dodatak vremenskog žiga za potrebe nezavisnog verifikovanja od strane servera, već i za proveru integriteta Univerzalnog registra.
- Novi vremenski žig. On se dodaje tokom operacije obnavljanja vremenskog žiga. Korisnici bi trebalo da obnove vremenski žig kada procene da hash algoritam kojim je izračunata hash vrednost žigosanog dokumenta nije dovoljno sigurna i da joj preti kompromitacija. Obnavljanje vremenskog žiga se sprovodi uz primenu novog i sigurnijeg hash algoritma, uz kreiranje novog vremenskog žiga. Na taj način se obezbeđuje da poslednji kreiran vremenski žig bude uvek ispravan, a time se stvara i mogućnost uspešnog verifikovanja žigosanog dokumenta tokom dugog vremenskog perioda.

C. Verifikovanje vremenski žigosanog dokumenta

Verifikovanjem vremenski žigosanog dokumenta može da se utvrdi da li je očuvan integritet dokumenta od trenutka vremenskog žigosanja. Bilo kakva promena žigosanog dokumenta ili vremenskog žiga dovešće do neuspešne verifikacije. Postupak verifikovanja vremenski žigosanog dokumenta sastoji se od sledećih pet koraka (Sl. 4):

1. U korisničku aplikaciju učita se koverta koja sadrži žigosan dokument i vremenski žig.
2. Korisnička aplikacija izračunava hash vrednost žigosanog dokumenta i poredi je sa hash vrednošću koja se nalazi u vremenskom žigu. Ako su te dve hash vrednosti identične, to znači da je očuvan integritet žigosanog dokumenta. U protivnom, aplikacija prikazuje poruku da je sadržaj žigosanog dokumenta izmenjen.
3. Korisnička aplikacija kreira zahtev za verifikovanje vremenskog žiga i zatim ga šalje serveru.
4. Server poredi hash vrednost i vreme iz vremenskog žiga sa vrednostima koje se nalaze u Univerzalnom registru.
5. Server šalje korisničkoj aplikaciji odgovor da li je vremenski žig ispravan ili neispravan. Ako je vremenski žig neispravan ili korisnička aplikacija ne može da uspostavi konekciju sa serverom, tada aplikacija prikazuje poruku da je nemoguće sprovesti verifikovanje vremenski žigosanog dokumenta. Ako je vremenski žig ispravan, a integritet žigosanog dokumenta očuvan na osnovu provere iz tačke 2, tada aplikacija prikazuje poruku da su vreme žigosanja i sadržaj žigosanog dokumenta ispravni.



Slika 4. Verifikovanje vremenski žigosanog dokumenta

Verifikovanje vremenski žigosanog dokumenta kod čijeg vremenskog žiga je dodata zbirna hash vrednost, može da se uradi i bez konsultovanja servera.

IV. OPTEREĆENJE SERVERA

Iako je upotreba ulančane šeme u potpunosti opravdana, mogu se javiti problemi koji stvaraju smetnje u njenoj praktičnoj primeni. Server za izdavanje novih vremenskih žigova može da bude dobro projektovan po pitanju svog kapaciteta, ali ono o čemu se u praksi retko vodi računa je povećanje broja zahteva zbog obnavljanja već postojećih vremenskih žigova [9].

Pretpostavimo da nam je poznat broj zahteva koje server može da obradi u jednoj rundi, bez stavljanja zahteva u red čekanja. Ovo ćemo nazvati *kapacitet runde*, pri čemu je kapacitet *k*-te runde označen sa N_k. Odnos između broja novih zahteva (koji ne uključuju obnavljanje) i kapaciteta runde označićemo sa konstantom α, pri čemu polazimo od toga da je $0 < \alpha < 1$, odnosno da je izdavalac vremenskih žigova uložio dovoljno u infrastrukturu da bi obezbedio ovu relaciju. U tom slučaju, $1 - \alpha$ predstavlja tzv. *parametar margine kapaciteta*. Odnos između broja zahteva za obnavljanje u okviru jedne runde i broja postojećih vremenskih žigova označićemo sa konstantom β i nazvati *parametar zahteva za obnavljanjem*. Usvajamo da je $0 < \beta < 1$.

Pretpostavimo da je u inicijalnoj rundi izdato αN_0 vremenskih žigova. Da bi se u sledećoj rundi prihvatili svi zahtevi za obnavljanjem, bez stavljanja u red čekanja, moralo bi da važi:

$$\alpha\beta N_0 \le (1 - \alpha) N_1. \quad (1)$$

Ukoliko pređemo na sledeću rundu, onda važi :

$$\beta(\alpha\beta N_0 + \alpha N_1) \le (1 - \alpha) N_2. \quad (2)$$

Na osnovu toga, u *k*-toj rundi će važiti relacija:

$$\sum_{j=0}^{k-1} \alpha\beta^{k-j} N_j \le (1-\alpha) N_k. \quad (3)$$

A. Broj zahteva je konstantan

Ukoliko je broj zahteva za izdavanje vremenskih žigova αN_k konstantan, N_k je konstantan, pa će važiti:

$$\frac{\beta(1-\beta^k)}{1-\beta} \le \frac{1-\alpha}{\alpha}. \quad (4)$$

Ako je *k* dovoljno veliko da je $\beta^k = 0$, tada se (4) svodi na

$$\alpha + \beta \le 1. \quad (5)$$

Ako važi jednačina (5), tada je (4) zadovoljena za svako *k* jer uslov $0 < 1 - \beta^k < 1$ i jednačina (5) daju:

$$\frac{1-\alpha}{\alpha} \ge \frac{\beta}{1-\beta} > \frac{\beta(1-\beta^k)}{1-\beta}. \quad (6)$$

B. Broj zahteva se linearno povećava

U situaciji da se broj zahteva za izdavanje vremenskih žigova αN_k linearno povećava, tj. da je $N_k = N_0 + \gamma k$ (γ je pozitivna konstanta), uslov (3) postaje:

$$\alpha\beta^{k+1} N_0 + \alpha\beta\gamma + \frac{\beta^2 - \beta^{k+1}}{1-\beta} \ge (\alpha + \beta - 1)(N_0 + \gamma k). \quad (7)$$

I u ovom slučaju, ako je $\alpha + \beta \le 1$, uslov (7) je zadovoljen za svako *k*, imajući u vidu da je

$$\alpha\beta^{k+1} N_0 + \alpha\beta\gamma + \frac{\beta^2 - \beta^{k+1}}{1-\beta} > 0, \quad (8)$$

i

$$N_0 + \gamma k > 0. \quad (9)$$

Ukoliko broj zahteva u praksi preraste onaj koji je predviđen projektom, prednost će imati oni sistemi koji su po svojoj prirodi skalabilni. Pri tome, TSA može da koristi slojevitu arhitekturu, gde su na slojevima primenjene različite ulančane šeme. Npr., na prvom sloju implementira se delimično uređena ulančana šema i ona radi drektno sa zahtevima klijenata u kratkim rundama. Na drugom sloju implementira se totalno uređena ulančanu šema, koja radi u dugačkim rundama. Ovaj sloj obrađuje samo zahteve koje je dobio od prvog sloja.

V. ZAKLJUČAK

Očekuje se da upotreba digitalnih podataka tokom narednih godina bude generalizovana i da će postepeno zameniti tradicionalni papirni metod obrade informacija. Ova tendencija već je očigledna u naprednim administrativnim okruženjima, kao što je bankarski sektor ili osiguranje, a pre ili kasnije osvojiće i dravnu upravu, parlament i sudove. Zbog toga se nameće neophodnost dokazivanja da je određeni dokument bio kreiran pre nekog vremenskog trenutka, što se postiže vremenskim žigosanjem

Izdavaoci vremenskih žigova moraju da rade u skladu sa pravnom regulativom koja po pravilu propisuje upotrebu protokola RFC 3161. Glavni problem razmene inforamcija upotrebom ovog protokola je to što je on zasnovan na prostoj šemi. Kao što je već napomenuto, prosta šema omogućuje malicioznom insajderu u TSA da izmeni vremenski parametar u okviru vremenskog žiga, bez bojazni da može da bude otkriven. Ovaj nedostatak otklanja ulančana šema, koja je u mnogim pitanjima bolja od proste šeme, uz nedostatak da je nešto složenija za implementaciju. U radu je prikazana praktična primena jednog softverskog rešenja, koje funkcioniše po principu delimično uređene ulančane šeme. Na kraju su date relacije koje ukazuju na važnost dobrog projektovanja kapaciteta servera za vremensko žigosanje. Pokazano je i da ulančana šema ne zahteva upotrebu složene PKI infrastrukture niti je neophodna upotreba kriptografskih ključeva. To omogućuje da više entiteta na jednostavniji način stekne dozvolu za izdavanje vremenskih žigova, pod pretpostavkom da se u ovoj oblasti izvrši deregulacija, tj. da se ne insistira na implementaciji (npr. po kojoj šemi, ili po kom standardu treba raditi), već samo na funkcionalnosti.

ZAHVALNICA

Rad je deo istraživanja koje finansira Ministarstvo prosvete i nauke Republike Srbije, projekti: III-45003 i TR-35026.

LITERATURA

[1] C. Adams, P. Cain, D. Pinkas and R. Zuccherato, “RFC 3161: Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)”, Internet Engineering Task Force, 2001.

[2] S. Milinković, B. Milojković, D. Spasić and Lj. Lazić, “Evaluation of some time-stamping authority software”, Proc. the 6th International Conference on Methodologies, Technologies and Tools Enabling e-Government, Belgrade, Serbia, July 3-5, 2012, pp. 89-99.

[3] S. Haber and W. S. Stornetta, “How to time-stamp a digital document”, Journal of Cryptology, Vol. 3, No. 2, pp. 99-111, 1991.

[4] R. C. Merkle, “Protocols for public key cryptosystems”, Proc. the IEEE Symposium on Security and Privacy, 1980, pp. 122–134.

[5] ISO/IEC18014-3, "Information technology - Security techniques - Time-stamping services - Part 3: Mechanisms producing linked tokens", 2009.

[6] ANSI X9.95, "Trusted Time Stamp Management and Security", 2012.

[7] Surety, "Ensuring Record Integrity with AbsoluteProof", Technical Whitepaper, 2012.

[8] M. Thimblin, N. C. Kamisetty, P. Raman, A. Paila, "Implementation of an Evidentiary Record Validation Utility and Security Analysis for Surety’s AbsoluteProof", Tech. report, George Mason University, 2005.

[9] K. Matsuura and H. Imai, “Digital timestamps for dispute settlement in electronic commerce: generation, verification, and renewal”, Proc.4th Int. Conference On Enterprise Information Systems, Universidad de Castilla-La Mancha - Ciudad Real - Spain - 3-6 April, 2002. pp.962-967.

ABSTRACT

This paper provides a brief description of the practical schemes for issuing the time-stamps, along with an explanation of their advantages and disadvantages. The two types of linking schemes are described and it is shown that both are applicable in practice. Further, the paper describes the operation of a software solution based on a partially ordered linking scheme. It points to the problem of proper system design, particularly in relation to the time-stamp renewal. It stresses out that for the practical application of these schemes the main obstacle is legislation, which is still based on the use of the simple scheme. As the conclusion, it is suggested that we need deregulation in this area, where only functionalities, but not the implementation details would be prescribed.

ON PRACTICAL IMPLEMENTATION OF TIME-STAMPING USING LINKING SCHEME

Dragan Spasić, Stevan Milinković, Branislav Milojković