UNIVERZITET CRNE GORE

Prirodno-matematički fakultet Podgorica

Ana Boljević

Upravljanje kriptografskim ključevima

SPECIJALISTIČKI RAD

Podgorica, 2017.

UNIVERZITET CRNE GORE

Prirodno-matematički fakultet Podgorica

Upravljanje kriptografskim ključevima

SPECIJALISTIČKI RAD

Kriptografija

Mentor: prof. dr Vladimir Božović

Ana Boljević

Matematika i računarske nauke

Podgorica, Septembar 2017.

Apstrakt

Upravljanje kriptografskim ključevima je vjerovatno najbitnija ali i najviše potcijenjena oblast kriptografije. Ono daje podršku bilo kom kriptografskom sistemu i aspekt je kriptografije u kom korisnici dobijaju priliku da utiču na pitanja koja se tiču kriptografije. U ovom radu proučićemo proces upravljanja ključevima prateći ključeve od njihovog nastanka do uništenja. Na ovaj način, moguće je vidjeti koliko daleko uticaj upravljanja ključevima doseže.

Abstract

Key management is arguably the most important, and often overlooked, area of cryptography from a practical perspective. This underpins the security of any cryptographic system and is the aspect of cryptography where users are given a chance to become involved in decisions concerning cryptography. In this paper we will discuss process of key management following keys from their creation to their destruction. This way, we will see how far the influence of key management can reach.

Sadržaj

1 Uvod 1

2 Osnove upravljanja kriptografskim ključevima 4

2.1 Životni ciklus ključa 5

2.2 Osnovni zahtjevi procesa upravljanja ključevima 6

3 Generisanje ključeva 9

3.1 Direktno generisanje ključeva 9

3.2 Izvođenje ključeva 12

3.2.1 Fistelov algoritam 13

3.3 Generisanje ključeva iz komponenti 15

4 Uspostavljanje ključeva 17

4.1 Hijerarhija ključeva 17

4.2 Šeme jedinstvenog po transakciji ključa 19

4.2.1 Motivacija za korištenje UKPT šema 20

4.2.2 Primjer UKPT šema 21

5 Skladištenje ključeva 24

5.1 Izbjegavanje skladištenja ključeva 24

5.2 Skladištenje ključeva u softveru 25

5.2.1 Čuvanje neenkriptovanih ključeva (eng. Storing keys in the clear) 25

5.2.2 Skladištenje ključeva korištenjem kriptografije 26
5.3 Skladištenje ključeva na hardveru 27
5.3.1 Hardverski bezbjednosti modul (HSM) 27
5.3.2 Skladištenje ključeva na HSM-u 29
5.3.3 Drugi tipovi hardvera 29
5.4 Faktori rizika kod skladištenja ključeva 30
5.5 Bekap i Arhiviranje ključeva 32
5.5.1 Bekap ključeva 32
5.5.2 Arhiviranje ključeva 33

6 Upotreba ključeva 34
6.1 Separacija ključeva 34
6.1.1 Potreba za separacijom ključeva 34
6.1.2 Primjena separacije ključeva 36
6.2 Promjena ključeva 38
6.3 Uništenje ključeva 39

7 Zaključak 40

Bibliografija 42

Glava 1

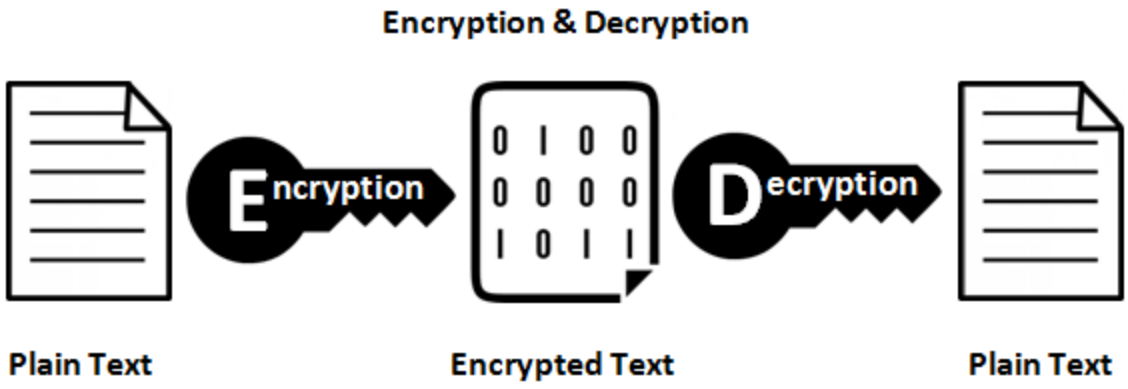
Uvod

Zahvaljujući brzom razvoju informacionih tehnologija danas nam je omogućena brza i laka razmjena velike količine podataka. Sa njihovim razvojem dolazi i do razvoja raznih tehnika napada na tajne podatke. Stoga, jasno je da je neophodno razviti tehnike koje će obezbijediti njihovu zaštitu. Nauka koja nam ovo omogućava je *kriptografija*.

Osnovni zadatak kriptografije je omogućavanje dvijema osobama (pošiljaocu i primaocu i u radu ćemo za njih rezervisati imena Alisa i Bob) da komuniciraju preko nesigurnog komunikacionog kanala (na primjer interneta) tako da treća osoba ili napadač ne može razumjeti njihove poruke.

Enkripcija je osnovni proces u kriptografiji kojim se vrši izmjena podataka na način da se učine nečitljivim za osobe koje ne posjeduju određenu informaciju - ključ. Ovo je proces kojim se osnovna poruka pretvara u šifrat. Vraćanje šifrata u prvobitnu formu, prije enkripcije naziva se *dekripcija*.

Kriptografski algoritam ili šifra je matematička funkcija koja se koristi za enkripciju i dekripciju poruka. Njeni argumenti mogu biti ključ, otvoreni tekst i/ili šifrat. Stoga je kriptografski algoritam ono što se najčešće smatra glavnim faktorom zaštite naših podataka.



Slika 1.1: Enkripcija i dekripcija - Šifrat i osnovni tekst

Simetrična kriptografija koristi isti ključ za enkripciju i dekripciju. Za razliku od nje *asimetrična kriptografija* koristi javni ključ (poznat svima) za enkripciju poruke i tajni ključ za dekripciju šifrata. Cjelokupan sistem koji se sastoji od svih kriptografskih algoritama, poruka i šifrata naziva se *kriptosistem*.

Ono što će ovaj rad pokazati jeste da je za bezbjednost kriptosistema potrebno više od jakog kriptografskog algoritma.

Oblast kriptografije koja se bavi administracijom kriptografskih ključeva - informacija koje iza ″sigurnih vrata″stavljaju naše podatke, je *upravljanje kriptografskim ključevima*. To je oblast koju ćemo upoznati kroz ovaj rad. Upravljanje kriptografskim ključevima je od suštinskog značaja za sigurnost kriptosistema. Postoji interesantna analogija između kriptografskih ključeva i kombinacija za sef. Ukoliko je kombinacija za sef poznata nije bitno koliko je sef siguran jer napadač ima ″ključ″od njegovih vrata. Na isti način, loše upravljanje ključevima može lako ugroziti i najjače algoritme.[1]

Ova oblast kriptografije u analizu stavlja same uređaje koji omogućavaju bezbjednost naših podataka, okruženje u kom se svi procesi koji se transporta podataka tiču

odvijaju, znanje i djelovanje korisnika i interakciju i koordinaciju između svih ovih elemenata. Dakle, ne radi se o čistoj matematičkoj praksi. Stvari nisu automatizovane već zavise od navedenih i brojnih drugih činilaca.

Upravljanje ključevima je kompleksan i prilično važan aspekt svakog kriptosistema. Ne prožima se samo kroz jedan dio procesa zaštite podataka već se miješa u situacije koje na prvi pogled ne djeluju kao dio ovog procesa. Kako je upravljanje ključevima zapravo interfejs između kriptografskog mehanizma i sigurnosti stvarnog sistema, to ono mora biti prilagođeno konkretnoj aplikaciji odnosno organizaciji. Na primjer, različita rješenja su potrebna za upravljanje ključevima u banci, vojnoj organizaciji, mobilnoj telefonskoj mreži ili na personalnom računaru. Ne postoji jedinstven način za upravljanje ključevima.

Vidjećemo da svaki ključ ima svoj životni ciklus koji se sastoji od faza koje će dalje otkriti koliko duboko upravljanje ključevima ulazi u problematiku bezbjednosti podataka odnosno kojih se sve oblasti tiče.

Glava 2

Osnove upravljanja kriptografskim ključevima

Upravljanje kriptografskim ključevima je skup procedura koje obuhvataju period od nastanka do uništenja ključa i koje za cilj imaju da zaštite ključ od raznih manipulacija. Pomenuti period obuhvata različite faze koje moraju biti međusobno usklađene kako bi u svakom trenutku sigurnost ključeva bila obezbijeđena.

Kriptografski ključevi su suštinski samo specijalna vrsta podataka. Upravljanje ključevima zato podrazumijeva dosta različitih procesa koji na kraju svi imaju isti cilj - zaštitu podataka. Tehnička kontrola je proces koji se primjenjuje u različitim fazama procesa upravljanja ključevima, kao na primjer kada je potreban specijalan hardverski uređaj za skladištenje ključeva.

Već je rečeno da je proces upravljanja ključevima različit za različite organizacije, pa tako mora biti prilagođen i sredini u kojoj organizacija ili aplikacija pripada. Takođe, fizička lokacija ključeva je jako bitna jer je na osnovu nje moguće odrediti tehnike upravljanja ključevima. Kriptografski sistem se nerijetko oslanja na čovjeka i na ručne procese, pa je ljudski faktor je često prisutan, što zna znatno da poveća i rizik od javljanja bezbjedonosnih problema.

Ključevi obuhvataju samo mali procenat od svih podataka kojim jedna organizacija mora da upravlja. Međutim većina bezbjedonosnih problema koja se javlja u organizacijama ima veze sa upravljanjem ključevima. Paradoksalno, vidjećemo da iako proces upravljanja ključevima postoji kako bi podržao upotrebu kriptografije, kriptografija se koristi da obezbijedi upravljanje ključevima.

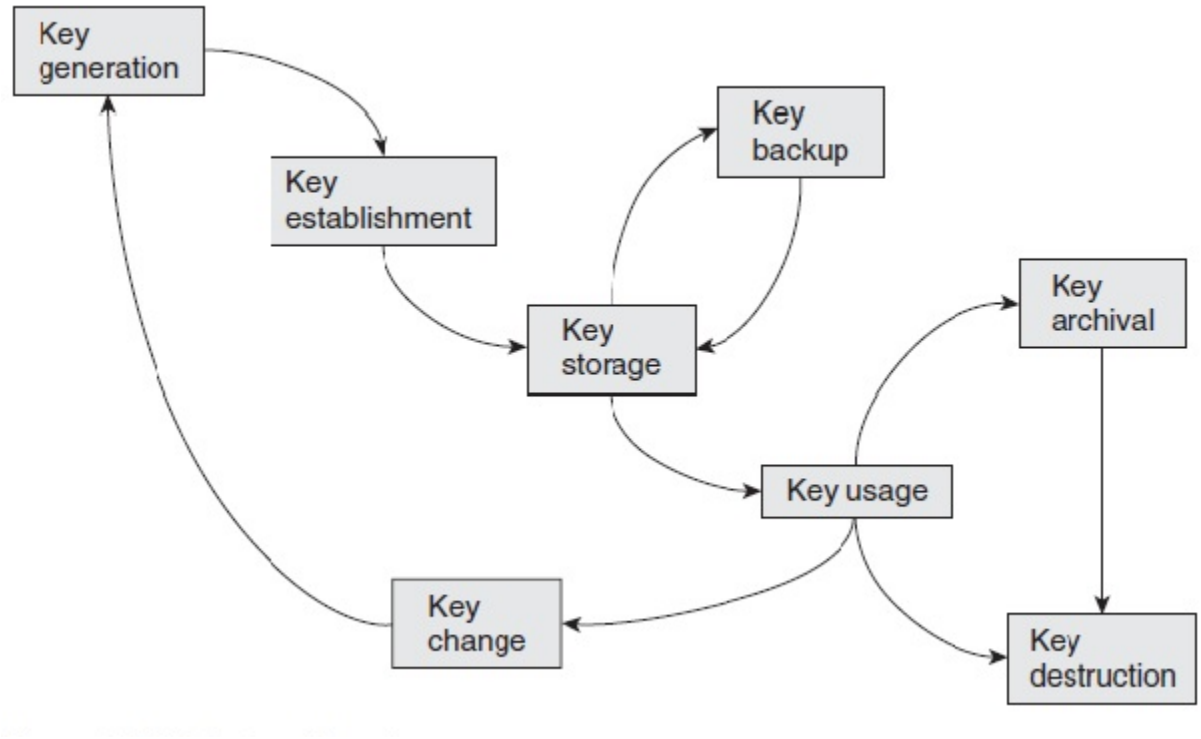
2.1 Životni ciklus ključa

Svaki kriptografski ključ ima svoj *životni ciklus* koji predstavlja skup različitih procesa vezanih za taj ključ u jednom vremenskom intervalu. Zadatak procesa upravljanja ključevima se može bolje shvatiti detaljnijom analizom životnog ciklusa ključa.

Glavne faze u životom ciklusu kriptografskog ključa (Slika 2.1) su:

- Generisanje ključeva - Stvaranje ključeva.
- Uspostavljanje ključeva - Dovođenje ključeva na pozicije sa kojih će se koristiti i njihova enkripcija.
- Skladištenje ključeva - Sigurno čuvanje ključeva. U skopu ove faze pominjaće se i arhiviranje i bekap ključeva.
- Upotreba ključeva - Pregled načina korišćenja ključeva i u sklopu ove faze pominjaće se i promjena ključa i njegovo uništenje.

Sve faze će se kroz poglavlja koja slijede detaljno razraditi i na taj način ćemo otkriti kako svaka pojedinačno utiče na cijeli proces.



Slika 2.1: Životni ciklus ključa

2.2 Osnovni zahtjevi procesa upravljanja ključevima

Postoje dva osnovna zahtjeva procesa upravljanja ključevima koji se protežu kroz sve faze i to su tajnost ključeva i osiguranje postojanja svrhe ključeva.

Tokom svog životog ciklusa tajni ključevi moraju **ostati tajni** odnosno nedostupni i nepoznati za sve entitete osim onih koji su ovlašćeni da ih znaju. Ova osobina ključeva se mora sačuvati kroz sve faze životog ciklusa ključa.

Prvo, ako se koristi slab mehanizam generisanja onda je šansa da napadač otkije informacije o ključevima mnogo veća. Takođe, ključevi su ranjiviji kada su smješteni negdje duže vrijeme pa mehanizam za skladištenje ključeva mora biti dovoljno jak da ih sačuva od bilo koga ko ima pristup uređaju na kom se nalaze, jer na svakog moramo gledati kao na potencijalnog napadača. Na kraju, u slučaju da ključevi nisu uništeni na pravi način postoje razne mogućnost njihove obnove i time može biti narušena

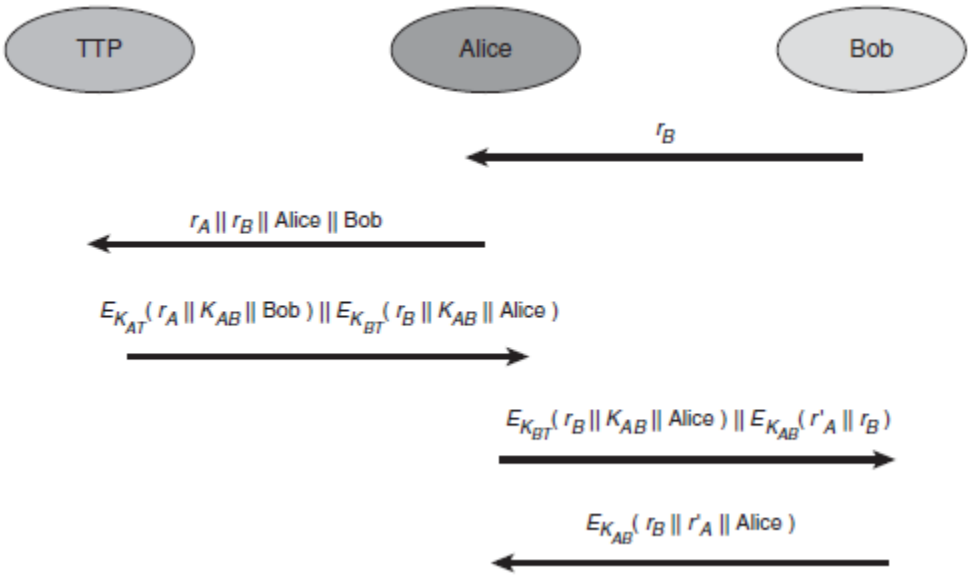
zaštita čitavog sistema.

One strane koje koriste ključ moraju **znati svrhu** u koju će se koristiti taj ključ. Pod svrhom ključa podrazumijeva se informacija o tome koje sve strane su povezane sa tim ključem, kriptografski algoritam za koji se ključ koristi ili ograničenja vezana za korišćenje ključa (npr. da se simetrični ključ može koristiti samo za provjeru i verifikovanje MAC-a).

Osiguranje postojanja svrhe ključa mora postojati u svim fazama životnog ciklusa ključa. U suprotnom, korišćenje jednog ključa u više svrha može izazvati brojne probleme o čemu će biti riječi kasnije.

Najčešće je postojanje svrhe ključa određeno implicitno kao što je slučaj kod AKE (authentication and key establishment) protokola.

AKE protokol uključuje interakciju tri entiteta Alise, Boba i TTP-a (eng. trusted third party). Alisa i Bob vjeruju TTP-u i sa njim imaju uspostavljene long-term ključeve K_{AT} i K_{BT}. Protokol je prikazan na Slici 2.2 i odvija se na sledeći način:



Slika 2.2: AKE protokol

1. Bob generiše slučajan broj r_B (eng. nonce).

2. Alisa generiše broj r_A i šalje zahtjev za simetričnim ključem TTP-u. Ovaj zahtjev uključuje i Alisino i Bobovo ime, kao i slučajne brojeve r_A i r_B

3. TTP generiše simetrični ključ K_{AB} i enkriptuje ga dva puta. Prvi šifrat je namijenjen Alisi i enkriptovan je sa K_{AT}. Osnovni tekst se sastoji od r_A, K_{AB} i Bobovog imena. Drugi Bobu, enkriptovan sa K_{BT}, dok se osnovni tekst sastoji od r_B, K_{AB} i Alisinog imena. Oba šifrata se šalju Alisi.

4. Alisa dekriptuje prvi šifrat koristeći K_{AT} i provjerava da sadrži r_A i Bobovo ime. Izvlači K_{AB} i potom generiše novi slučajan broj $r`_A$. Nakon toga generiše i novi šifrat enkriptujući $r`_A$ i r_B koristeći K_{AB}. Konačno, drugi šifrat koji je primila od TTP-a zajedno sa novim šalje Bobu.

5. Bob dekriptuje prvi šifrat koji prima (onaj koji prvenstveno dolazi od TTP-a) koristeći K_{BT} i provjerava da li sadrži r_B i Alisino ime. Izvlači K_{AB}. Potom dekriptuje drugi šifrat koristeći K_{AB} i provjerava da li sadrži r_B. Izvlači $r`_A$. Konačno, enkriptuje $r_B$, $r`_A$ i Alisino ime koristeći K_{AB} i šalje Alisi.

6. Alisa dekriptuje primljeni šifrat koristeći K_{AB} i provjerava da li sadrži r_B, $r`_A$ i Alisino ime. Ako sadrži protokol je uspješno završen.

Ono što vidimo iz ovog primjera jeste da je osiguranje postojanja svrhe ključa postignuto zahvaljujući tome što Alisa, Bob i TTP čine siguran zatvoren sistem u kom dijele simetričan tajni ključ. Kod asimetrične kriptografije gdje su javni ključevi bukvalno javni podaci dostupni svima najčešće se mora eksplicitno ostvariti postojanje svrhe ključeva.

Glava 3

Generisanje ključeva

Ovom glavom započinjemo objašnjavanje svake faze životnog ciklusa ključa detaljnije. Faza generisanja ključeva se logično nameće kao prva i ovom fazom započinju svi uticaji ključeva na bezbjednost kriptosistema. Razlog neuspjeha mnogih kriptosistema bio je generisanje ključeva na nedovoljno bezbjedan način, što govori o važnosti ove faze.

3.1 Direktno generisanje ključeva

Simetrični ključevi su samo slučajno generisani brojevi. Stoga je najočigledniji način za dobijanje kriptografskog ključa generisanje slučajnog ili najčešće pseudoslučajnog broja. Pri izboru tehnike koja se koristi za generisanje ključeva treba uzeti u obzir prirodu aplikacije i važnost ključa koji se generiše. Npr. upotreba hardverskog nedeterminističkog generatora može biti adekvatna za master ključeve, dok softverski nedeterministički generator zasnovan na pokretima miša može biti dovoljan za generisanje lokalnog ključa koji se koristi za čuvanje fajlova na nekom privatnom računaru. Nešto detaljnije o pomenutim tehnikama:

Nedeterministički generator generiše slučajan broj u nizu na način da je on nezavisan u odnosu na svog prethodnika oslanjajući se na nepredvidljive uticaje u fizičkom svijetu. Ovo je dobar ali obično skup pristup proizvodnji slučajnog broja. Nedeterministički generatori mogu biti hardverski ili programski orjentisani.

Hardverski nedeterministički generator može biti zasnovan na prirodnim pojavama poput zvučnih šumova, radioaktivnog raspadanja, termalnih šumova otpornika, fotoelektričnih efekata i raznih kvantnih fenomena.

Kod ovih izvora najčešće je potrebno dodatno programski obraditi dobijene podatke. Naime, može se primijetiti da dobijeni niz često prati određeni šablon (pojava određenih brojeva je vjerojatnija), a moguća je i međusobna povezanost između brojeva u nizu (pojava jednog broja u nizu povećava vjerojatnost da sledeći broj bude neki tačno određeni broj). Navedeni izvori se moraju vanjski povezati sa računarom, tj. generator je zapravo periferni uređaj. U nekim slučajevima ovakvo generisanje brojeva je toliko sporo da nije od praktične koristi.

Programski nedetermistički generator može biti zasnovan na rezultatu sistemskog sata, vremenu između dva pritiska na dugmad tastature ili miša, pokretima miša i slično.

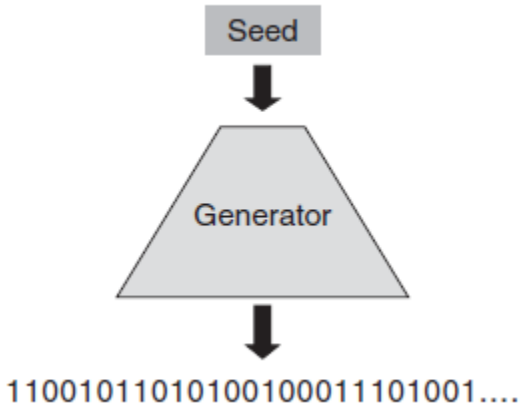
Iako se na prvi pogled čini da ova skupina izvora može dati nepredvidivi niz brojeva, to ne mora nužno biti slučaj. Naime ponašanje sistemskog sata se lako može ″predvidjeti″. Takođe, interakcija korisnika (drugi i treći primjer) u stvarnosti slijedi određenu pravilnost koja se može preslikati na generisani niz brojeva.

Pametnim kombinovanjem više različitih izvora slučajnih brojeva moguće je dobiti dobar generator slučajnih brojeva.

Ideja *determinističkog generatora* slučajnog broja može zvučati kao oksimoron jer sve što je determinističko ne može istinski biti slučajno.

Deterministički generator je algoritam koji za rezultat ima pseudoslučajan niz bita ili

niz bita koji nije u potpunosti slučajan. U sustini, svako ko zna ulaznu vrijednost za deterministički generator može znati i rezultat. Odnosno, svaki put kada se pokrene algoritam sa istom ulaznom vrijednošću dobiće se ista izlazna vrijednost. Međutim, ako koristimo tajnu ulaznu vrijednost onda je moguće da će se dobiti rezultat koji nema očiglednu strukturu. Odnosno, za nekog ko ne zna ulaznu vrijednost rezultat će djelovati u potpunosti slučajan.



Slika 3.1: Deterministički generator

Razlikuju se dvije komponenete determinističkog generatora i to: ulazna vrijednost (eng. seed) i sam generator, kao što se i vidi na Slici 3.1. Ulazna vrijednost je tajna informacija koja suštinski predstavlja kriptografski ključ. Ovo je jedina informacija koja definitivno nije poznata napadaču. Kako bi se obezbijedila nepredvidivost pseudoslučajnog izlaza bitno je da se ova informacija zaštiti ali i da se dovoljno često mijenja. Generator je kriptografski algoritam koji proizvodi pseudoslučajan rezultat na osnovu ulazne vrijednosti. Takođe, obično se pretpostavlja da su detalji o generatoru javno dostupni.

3.2 Izvođenje ključeva

Termin izvođenje ključeva označava generisanje ključeva iz nekog ″roditeljskog″ključa. Izvođenje ključeva ima mnoge prednosti pa se na mnogim mjestima nameće kao jedno od najboljih rješenja. Generisanje ključeva i njhovo uspostavljanje mogu biti relativno skupi procesi. Jedna efikasna tehnika za zaobilaženje ovog problema je generisanje i uspostavljanje jednog ključa koji nazivamo bazični ključ a potom korištenje njega za izvođenje drugih.

Na primjer, mnoge aplikacije zahtjevaju i povjerljivost i autentifikaciju podataka. Ako se odvojeni kriptografski mehanizmi koriste da obezbjede ova dva bezbjednosna servisa onda su potrebni i MAC i enkriptovani ključ. Kako bi se povećala efikasnost, praktično rješenje je da se generiše i uspostavi jedan ključ K a onda iz njega da se izvedu ključevi K_1 i K_2 koji će predstavljati MAC i enkripcioni ključ. Više informacija o MAC-ovima možete naći u knjizi Kejt Martin, Everyday cryptography ili u Specijalističkom radu kolege Vladimira Pekovića.

Jedan jednostavan proces generisanja uključuje računanje:

$$K_1 = h(K||0) \text{ i } K_2 = h(K||1)$$

gdje je h heš funkcija o kojoj takođe možete više saznati iz navedenih izvora.

U nekim aplikacijama, dugoročni simetrični ključevi su unaprijed učitani na uređajima prije nego što su stavljeni u upotrebu. Direktna upotreba ovih kriptografskih ključeva za enkripciju podataka izlaže ih kriptoanalizi. Stoga, ako bi se iz takvog ključa izvodili ključevi koji bi se dalje koristili za enkripciju podataka onda bazični ključ ne bi bio izložen kao ranije direktnom upotrebom.

Kako bi se bazični ključ zaštitio u slučaju kada su ključevi izvedeni iz njega kompromitovani potrebno je koristiti *jednosmjernu funkciju* za izvođenje. Razlog za to je jer jednosmjerna funkcija f ima osobinu lakog izračunavanja $f(x)$ za zadato x i teško

je naći bilo kakvu informaciju o x kada je poznato $f(x)$.

Postoje brojni standardi za izvođenje ključeva. Na primjer, PKCS#5 definiše kako ključ može biti izveden iz PIN-a ili šifre (što se inače smatra prilično nebezbjednim kriptografskim ključem). U ovom slučaju ključevi se izvode računanjem $f(P, S, C, L)$ gdje je: f funkcija izvođenja koja objašnjava kako se kombinuju različite ulazne vrijednosti kako bi se izveo ključ, P šifra ili PIN, S niz pseudoslučajnih bita koji obezbjeđuje da ista šifra P ne izvede pri svakom pokretanju isti ključ, C brojač koji određuje broj iteracija za izračunavanje i L dužina izvedenog ključa.

3.2.1 Fistelov algoritam

U kriptografiji, Fistelov algoritam je simetrična struktura koja se koristi pri konstrukciji blok šifri, nazvana po njemačkom fizičaru i kriptografičaru Horstu Fistelu. Dio Fistelovog algoritma zahvata upravo izvođenje određenog broja ključeva iz jednog ključa K i njega ćemo u nastavku i opisati.

Postupak generisanja n 48-bitnih djelova ključeva $(K_1, K_2, \ldots, K_n)$ od zadatog tajnog 64-bitnog ključa K sprovodi se u nekoliko koraka.

Prvo se iz ključa K odstrane bitovi parnosti tj. 8-, 16-, 24-, 32-, 40-, 48-, 56- i 64- bit. Nakon toga se formira novi ključ $K+$ tako što svaki bit sa pozicije i ključa K ide na poziciju novog ključa $K+$ koja predstavlja poziciju broja i u tabeli PC-1. Preciznije, kako je prva vrijednost u tabeli PC-1 57 to znači da 57-mi bit originalnog ključa K postaje prvi bit novog permutovanog ključa $K+$. [2] [4]

Tabela PC-1:

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Sledeći korak podrazumijeva generisanje dva bloka od po 28 bita C_0 i D_0 od ključa $K+$.

Na primjer, ako je ključ $K+$:

$K+ = 1111000011001100101010101111010101010110011001111100011111$

onda su C_0 i D_0:

$C_0 = 1111000011001100101010101111$

$D_0 = 0101010101100110011110001111$

Sada kada su formirani C_0 i D_0, kreira se n blokova C_i i D_i gdje je $1 \leq i \leq n$. Svaki par blokova C_i i D_i se formira od para koji mu prethodi C_{i-1} i D_{i-1}, $1 \leq i \leq n$ tako što se koristi pomjeranje bitova bloka ulijevo. Preciznije, blok C_1 se dobija od bloka C_0 tako što se svaki bit pomjeri ulijevo, osim prvog koji ide na poslednje mjesto, pa C_1 dobija vrijednost:

$C_1 = 1110000110011001010101011111$

Tabela PC-2:

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Sada se formira ključ K_i, za $1 \leq i \leq n$, primjenjujući tabelu permutacija PK-2 za svaki blok C_iD_i. Svaki blok C_iD_i ima 56 bita, ali se tabela PC-2 primjenjuje samo na 48 bita bloka C_iD_i.

Na primjer, iz bloka C_1D_1:

$C_1D_1 = 1110000110011001010101011111101010101100110011110001111 0$

dobija se ključ K_1:

$K_1 = 000110110000001011101111111111000111000001110010$

3.3 Generisanje ključeva iz komponenti

Prethodno objašnjeni procesi generisanja ključeva - direktno generisanje ključeva i izvođenje ključeva su dva procesa koja mogu biti izvedena samo ako postoji strana kojoj se potpuno može vjerovati i prepustiti proces generisanja ključa. Ovo je često i realna pretpostavka, međutim za ekstremno bitne tajne ključeve to nije slučaj. U ovim situacijama proces generisanja ključeva prepušta se grupi entiteta na takav način da ni jedan član grupe pojedinačno nema potpunu kontrolu nad ovim procesom ali kolektivno imaju. Jedan od načina da se ovo ostvari jeste generisanje ključeva iz

komponenti. Ovaj pristup postaje najjasniji razmatranjem scenarija koji uključuje tri entiteta: Alisu, Boba i Čarlija. Pretpostavimo da je cilj da se generiše 128-bitni ključ:

1. Alisa, Bob i Čarli zasebno generišu tri slučajne komponente od 128 bita. Ove komponente su vrsta kriptografskih ključeva pa se za njihovo generisanje koristi bilo koji mehanizam generisanja ključeva. Označimo dobijene komponente sa K_A, K_B i K_C respektivno.

2. Alisa, Bob i Čarli bezbjedno tranportuju svoje komponente do još jednog sigurnog entiteta koji će u mnogim aplikacijama biti hardverski bezbjedonosni modul (eng. harware security module) ili HSM. U mnogim situacijama bezbjedan transfer će podrazumijevati ručnu dostavu i za velike internacionalne organizacije prenošenje nekih komponenti fizički sa jedne na drugu stranu svijeta.

3. Pomenuti četvrti sigurni entitet izvodi ključ K iz dobijenih komponenti. U ovom slučaju najbolja funkcija izvođenja je XOR:

$$K = K_A \oplus K_B \oplus K_C$$

   XOR je dobar izbor za funkciju izvođenja jer čak iako neko zna dvije komponente to mu ne daje nikakve informacije o izvedenom ključu K. Razmotrimo slučaj u kom Alisa i Bob žele da saznaju nešto o ključu K. U tu svrhu oni računaju:

$$R = K_A \oplus K_B.$$

   Primijetimo da je $K = R \oplus K_C$, što znači da je $R = K \oplus K_C$. Dakle, R se može smatrati enkripcijom ključa K upotrebom OTP-a (one-time pad) sa ključem K_C. OTP nudi savršenu bezbjednost, što znači da to što oni znaju R (šifrat) ne odaje nikakvu informaciju o ključu K (originalnoj poruci).

Glava 4

Uspostavljanje ključeva

Nakon što je ključ generisan slijedi proces njegovog uspostavljanja ili distribucije. Ova faza životnog vijeka ključa podrazumijeva ručno ili elektronsko dovođenje ključa na mjesto na koje će se koristiti i smatra se najtežom fazom za realizaciju. Ovo važi u slučajevima kada ključ nije lokalno generisan, jer ako jeste, proces uspostavljanja ključa je pravolinijski. Kada je potrebno učitati ključ na uređaj na koji će se koristiti, ideja je da se on ekriptuje sa novim ključem i time se javlja ideja hijerarhije ključeva koja će u ovoj glavi biti detaljno objašnjena.

4.1 Hijerarhija ključeva

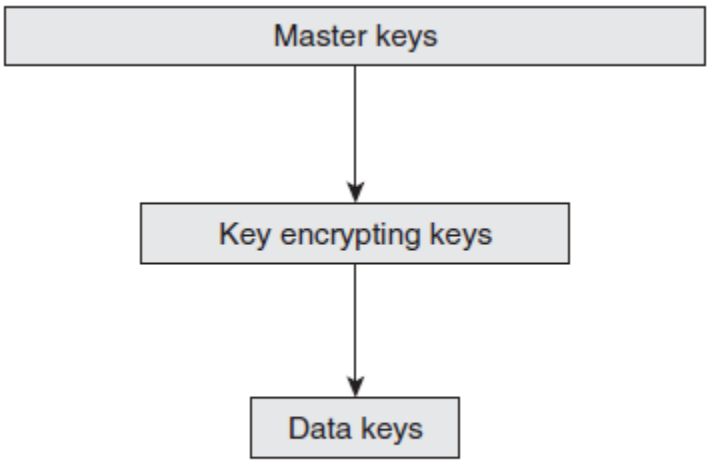
U cilju lakšeg upravljanja ključevima često se koristi rangiranje ključeva po nivoima ili upotreba hijerarhije ključeva. U hijerarhiji ključeva ključevi na većem nivou smatraju se važnijim od ključeva na nižim nivoima.

Veća je šansa da budu kompromitovani ključevi koji se nalaze na nižim nivoima, odnosno ključevi koji se direktno koriste za enkripciju podataka (data ključevi). Samim tim, često se može javiti potreba za njihovom promjenom. Korištenje hijerarhije ključeva olakšava promjenu ovih ključeva bez potrebe za promjenom ključeva na višim

nivoima čije je generisanje i uspostavljanje mnogo skuplje.

Problem upravljanja ključevima na koji upotreba hijerarhije ključeva stavlja fokus je kako koristiti, distribuirati i skladištiti ključeve na višim nivoima. Moguće je skoncentrisati se samo na ovaj problem jer zahvaljujući hijerarhiji ključeva dobro upravljanje ključevima na višim nivoima gotovo potpuno osigurava ključeve na nižim nivoima.

Primjer jednostavne hijerarhije ključeva sa tri nivoa predstavljen je na slici 4.1.



Slika 4.1: Hijerarhija ključeva

Tri nivoa uključuju:

- Master ključeve - Ovo su top-level ključevi koji zahtijevaju pažljivo upravljanje. Oni se samo koriste za enkripciju key encrypting ključeva. Kako je upravljanje master ključevima zahtjevno i skupo oni obično imaju relativno dug vijek trajanja.

- Key encriptyng ključeve - Ovo su ključevi koji su distribuirani i skladišteni u

enkriptovanoj formi upotrebom master ključeva. Oni se samo koriste za enkripciju data ključeva i imaju kraći životni vijek od master ključeva jer su više izloženi i jednostavnije ih je promijeniti.

- Data ključeve - Oni su distribuirani i skladišteni u enkriptovanoj formi upotrebom key encrypting ključeva. Ovo su ″radni″ključevi koji se koriste za enkripciju podataka. Najviše su izloženi od svih u hijerarhiji i imaju najkraći vijek trajanja.

Dužina vijeka trajanja ključeva u hijerarhiji raste što se krećemo ka višim nivoima, a isto se dešava i sa dužinom ključeva. Ključevi na jednom nivou bi trebali da budu bar dugi kao oni ispod njih. U većini aplikacija dovoljno je postojanje samo master i data ključeva, a srednji nivo nije neophodan.

Potrebno je sa većim stepenom bezbjednosti pristupiti upravljanju top-level (master) ključevima inače će cijela hijerarhija biti kompromitovana. Zbog toga će se većina sistema za upravljanje ključevima koji koriste hijerarhiju odlučiti za hardverski bezbjedonosni modul (HSM) za čuvanje master ključeva, koji ih nikada neće ostaviti u nezaštićenoj formi.

Generisanje master ključeva je ključna operacija. Najčeće se oni generišu, uspostavljaju i bekapuju u formi komponenti. Ako je potrebno da se master ključ dijeli između dva HSM-a onda je jedna opcija da se uspostavi key agreement protocol (na primjer Difi-Helman) između ovih HSM-ova kako bi se uspostavio šerovani master ključ.

4.2 Šeme jedinstvenog po transakciji ključa

Drugačiji način za uspostavljanje kriptografskih ključeva je pomoću šema jedinstvenog po transakciji ključa ili UKPT (eng. Unique key per transaction) koje se tako

zovu jer se ključ uspostavlja svaki put kada se koriste.

4.2.1 Motivacija za korištenje UKPT šema

Većina mehanizama za uspostavljanje ključeva uključuju ili upotrebu top-level tajnih ključeva (npr. upotreba master ključeva u hijerarhiji ključeva) ili specijalan transfer podataka za potrebe uspostavljanja ključeva. Iako su ovo većinom prihvatljivi zahtjevi, ipak postoje okruženja u kojima nisu. Za prvi zahtjev neophodan je uređaj koji može skladištiti i koristiti top-level ključeve, a drugi uključuje višak komunikacije. Do ovih zahtjeva dolazi jer se novi ključ koji se uspostavlja generiše nezavisno, u smislu da nema veze sa postojećim podacima ili postojećim ključevima. Alternativni metod je da se generišu novi ključevi tako što će se izvesti iz informacija koje Alisa i Bob već dijele. Ono što je najbitnije jeste da dijeljena informacija ne mora da bude top-level tajni ključ, može biti kratkotrajni ključ (short-term), podatak ili njihova kombinacija.

Ako se koristi izvođenje ključeva za generisanje novih ključeva onda se procesi generisanja i uspostavljanja stapaju u jedan proces. Prednosti ovog metoda su što Alisa i Bob ne moraju da skladište dugoročne (long-term) ključeve, ne zahtjeva se da učestvuju u bilo kakvoj dodatnoj komunikaciji i generisanje i uspostavljanje se mogu automatizovati.

Dobar primjer aplikacije koja koristi UKPT šeme je maloprodajni terminal koji se koristi u trgovini da verifikuje PIN-ove i potvrdi uplate karticom (transakcije). Potreba za korišćenjem UKPT šema kod maloprodajnih terminala dolazi od ograničene bezbjedonosne kontrole (nalaze se na javim mjestima i moraju biti dovoljno jeftini da bi se masovno koristili). Najčešće su locirani na nebezbjednim javnim mjestima kao što su radnje i restorani. Takođe su prenosivi, pa ih je stoga često lako i ukrasti. (Sve navedeno se odnosi na Zonu 1 koju ćemo kasnije pominjati u sekciji 5.4. Faktori rizika kod skladištenja ključeva). Stoga, jasno je da je nepoželjno da sadrže važne

top-level ključeve.

Terminalima može upravljati, i najčešće upravlja, nestručno osoblje, pa je potpuna automatizacija procesa uspostavljanja ključeva neophodna. Što je još jedan od razloga zašto se UKPT šeme koriste kod ovih terminala.

4.2.2 Primjer UKPT šema

Razmotrimo UKPT šemu koja operiše između maloprodajnog terminala i hosta (banke ili servera (card payment server)). Terminal čuva registar (eng. key register) koji je u suštini pokretački ključ koji će se update-ovati nakon svake transakcije.

Opisaćemo generičku UKPT šemu u terminima protokola koji se odvija između hosta i terminala tokom transakcije. Pretpostavićemo da na početku protokola host i terminal dijele neku početnu vrijednost koja se čuva u registru terminala. Ovo može i ne mora biti tajna vrijednost. Opisaćemo jednostavan protokol koji koristi transakciju ključa kako bi izračunao MAC-ove na razmijenjenim porukama. U realnosti, ovakvi protokoli su ipak komplikovaniji jer npr. neki enkripcioni ključ može se koristiti i da enkriptuje PIN kartice.

Generička UKPT šema:

1. Terminal izvodi transakcioni ključ koristeći sadržaj registra i dijeljenu informaciju koja će biti dostupna hostu.

2. Terminal šalje zahtjev u vidu poruke hostu. Transakcioni ključ se koristi da izračuna MAC na poslatoj poruci (zahtjevu).

3. Host izvodi transakcioni ključ (tehnike variraju od šeme do šeme).

4. Host provjerava validnost MAC-a na zahtjevu.

5. Host šalje odgovor u vidu poruke terminalu. Transakcioni ključ se koristi da izračuna MAC na odgovoru.

6. Terminal provjerava validnost MAC-a na odgovoru.

7. Terminal update-uje sadržaj registra.

Kako bi se proizvela realna UKPT šema iz generičke UKPT šeme potrebni su odgovori na sledeća pitanja:

-Šta je početna vrijednost registra?

-Kako se izvodi transakcioni ključ na način da i host i terminal izvedu isti?

-Kako treba da se update-uje registar terminala na način da ga i terminal i host update-uju na istu vrijednost?

Cilj je da terminal i host budu sinhronizovani, a to se može postići na različite načine. Primjeri realnih UKPT šema su Racal UKPT šema i Izvedena UKPT šema.

Odgovori **Racal UKPT šeme** na navedena pitanja su:

-Početna vrijednost je tajni seed koji se ugovoren između terminala i host.

-Host uspostavlja identičan registar terminalovom. Transakcioni ključ se izvodi iz registra i podatka sa kartice, preciznije primarnog broja računa na kartici, a oba podatka posjeduju i terminal i host pa će izvedeni ključ biti isti za oboje.

-Na kraju protokola nova vrijednost registra se računa kao funkcija od stare vrijednosti registra, podatka sa kartice i transakcionog podatka (dio podatka dobijen iz izračunatih MAC-ova na porukama zahtjeva i odgovora). I host i terminal vrše isto izračunavanje pa na isti način i update-uju registre.

Izvedena UKPT šema je podržana od strane Vize između ostalog i njeni odgovori na postavljena pitanja su:

-Početna vrijednost je jedinstveni početni ključ koji je instaliran na terminalu.

-Transakcioni ključ je izveden od strane terminala iz sadržaja registra terminala,

transakcionog broja i terminalovog jedinstvenog identifikatora. Host ima specijalan bazični (master) ključ. Iz njega, transakcionog broja i terminalovog identifikatora može izvući isti transakcioni ključ pa mu nije potrebno da uspostavlja posebno transakcioni ključ.

-Na kraju protokola, nova vrijednost registra terminala je izvedena iz stare vrijednosti registra i transakcionog broja. Host ne mora da čuva ovu vrijednost jer može da izračuna transakcione ključeve direktno, kao što je upravo objašnjeno.

Glavna prednost Izvedene UKPT šeme je što host ne mora da uspostavlja registar i što može da izvede transakcioni ključ direktno. Osnovni problem kod ove šeme je što će napadač koji pristupi vrijednosti registra moći da izračuna buduće transakcione ključeve terminala. Dok bi u Racalovoj UKPT šemi ovaj napadač morao još i da pristupi i svim budućim podacima kartice. Izvedena UKPT šema zahtijeva i pažljiv proces inicijalizacije, otkad kompromitovanost početnog terminalnog ključa vodi ka kompromitovanosti svih budućih transakcionih ključeva.

Svi pomenuti problemi sa ovim UKPT šemama mogu se riješiti kroz pažljiviji proces upravljanja. UKPT šeme su veoma efikasni sistemi za upravljanje ključevima i omogućavaju lociranje poteškoća povezanih sa uspostavljanjem ključeva u raznim tipovima okruženja za koje su dizajnirani.

Glava 5

Skladištenje ključeva

Tajni ključevi ne smiju biti dostupni licima odnosno entitetima koji nisu planirani da budu njihovi ″vlasnici″. Stoga je veoma bitno da se bezbjedno skladište. U ovoj sekciji razmatraće se načini skladištenja ključeva i kako se nositi sa potencijlnim gubitkom ili nedostupnošću ključeva.

5.1 Izbjegavanje skladištenja ključeva

Najbolje od svih rješenja bi bilo da se kriptografski ključevi ne skladište nigdje i da se samo generišu u letu, onda kada je potrebno.

Ovo je moguće u nekim aplikacijama. Kako jedan ključ mora biti generisan u letu svaki put kada je potrebno da se iskoristi, zahtijeva se deterministički generator ključeva da generiše taj ključ. Pominjano je da deterministički generator zahtijeva seed ili ulaznu vrijednost, pa je potrebno da se ovaj seed koristi dosljedno svaki put kada se generiše ključ. Međutim, samim tim i seed se mora čuvati na sigurnom, pa se postavlja pitanje gdje?

Većina aplikacija koje koriste ovu tehniku prilikom skladištenja seed-a oslanja se na ljudski faktor odnosno čovjekovu memoriju. U ovim situacijama seed se čuva u

obliku šifre (password), niza tajnih riječi ili drugog teksta - fraze. Fraza se potom koristi za deterministički generator koji generiše ključ u letu. Očigledna mana ovog procesa je što sigurnost skladištenog ključa zavisi od sigurnosti seed-a (fraze) koji se koristi da generiše ključ.

Svakako, ovo je opšte koristan i pragmatičan pristup koji predstavlja balans između sigurnosti i korisnosti i koji je odgovarajući za većinu aplikacija.

Međutim, nije uvjek moguće izbjeći skladištenje ključeva. Na primjer, javni parovi ključeva su skupi za generisanje, zbog svoje dužine. Smatra se neefikasnim da se generišu svaki put kada se ukaže potreba za njima.

5.2 Skladištenje ključeva u softveru

Jedna opcija za skladištenje ključeva jeste da se ključ ugradi u softver. Međutim, sprovođenje bilo kog kriptografskog procesa kroz softver dolazi sa određenim rizicima. Sa druge strane, skladištenje ključeva u softveru mnogo je jeftinije od skladištenja u hardveru, pa je najčešće neophodno izabrati između bezbjedonosnih rizika i troškova.

5.2.1 Čuvanje neenkriptovanih ključeva (eng. Storing keys in the clear)

Do sada najjeftiniji ali i najrizičniji pristup je da se ključevi čuvaju neenkriptovani u softveru. Drugim riječima, ključevi se posmatraju kao djelovi podataka koji su skladišteni na hard disku u nezaštićenoj formi. Ovo je nerijedak i svakako opasan pristup jer se oslanja na to da napadač neće pronaći ključeve. U suštini, postoje dva osnovna problema ovog pristupa. Diveloper (Inženjer) koji dizajnira softver će znati gdje se ključevi nalaze, pa samim tim već postoji potencijalni napadač koji zna gdje da traži ključeve. Sa druge strane, pretpostavljajući da su skriveni ključevi

spacifični za različite verzije softvera, napadač koji posjeduje dvije verzije može naći ključ upoređivanjem ove dvije verzije. Tada, sve lokacije na kojima se uočavaju razlike su potencijalne lokacije ključeva ili bar materijala vezanog za ključeve.

Normalno, ovaj pristup je najbolje izbjegavati. Zbog navedenih problema nije neuobičajeno što je standardima upravljanja ključevima ovaj pristup čak i zabranjen u mnogim sistemima.

5.2.2 Skladištenje ključeva korištenjem kriptografije

Očigledno rješenje za zaštitu ključeva koji se oslanjaju na softver jeste da se oni enkriptuju. Međutim, ovo ne rešava već samo preusmjerava problem skladištenja. Kako bi se enkriptovao ključ potreban je key encrypting ključ, što ponovo nameće pitanje gdje skladištiti taj ključ?

Jedna od opcija jeste da se pomenuti ključ generiše u letu. Ovo i jeste čest pristup koji se koristi u aplikacijama u kojima rješenje zasnovano na hardveru nije dostupno. Ukoliko je moguće ispuniti zahtijev za odgovarajućim hardver uređajem, onda je ovo jedno od boljih rješenja i o njemu će biti detaljnije govoreno u poglavlju 5.3. Skladištenje ključeva na hardveru.

Još jedna od opcija je da se skladište u formi komponenti. Na ovaj način se otežava pristup ključu, jer da bi se pristupilo ključu neophodno je pristupiti svakoj komponenti. Međutim, opet problem skladištenja nije riješen jer se moraju negdje skladištiti i komponente. Najadekvatnije rješenje za to je da se komponente, koje su u stvari ključevi, storniraju na hardveru pa ovo rješenje nije nova alternativa već proizilazi iz prethodnog, zasnovanog na hardveru.

5.3 Skladištenje ključeva na hardveru

Najbezbjedniji način skladištenja podataka je na hardveru. Naravno, postoje različiti hardverski uređaji sa različitim karakteristikama i nivoima bezbjednosti.

5.3.1 Hardverski bezbjednosti modul (HSM)

Najbezbjedniji hardverski medij za čuvanje kriptografskih ključeva je hardverski bezbjednosni modul ili HSM (eng. Hardware security modules). Mnogi HSM-ovi mogu obezbijediti većinu kriptografskih operacija i to najčešće velikom brzinom. HSM može biti periferan uređaj, a može biti ugrađen u neki uređaj opštije namjene kao što je maloprodajni terminal.

Iako se sada HSM pominje u kontekstu mehanizma za bezbjedno čuvanje kriptografskih ključeva, bitno je naglasiti da se HSM često koristi i u drugim fazama životnog ciklusa ključa.

Ključevi koji su skladišteni na HSM-u su fizički zaštićeni hardverom. Ukoliko bi neko pokušao da probije neki HSM, na primjer da izvuče ključ iz uređaja, aktivira se kolo otporno na zloupotrebu i ključ se najčešće briše iz memorije HSM-a. Postoje različite tehnike koje obezbjeđuju otpornost na zloupotrebu i one uključuju:

- Mikro-svičeve - jednostavan mehanizam koji pokreće svič (okidač) ako je HSM otvoren. Ovo nije naročito efikasan mehanizam jer profesionalni napadač može uvjek izbušiti rupu i iskoristiti lijepak da isključi svič.

- Elektronsku mrežu - mreža koja okružuje osjetljive komponente, koja ako se slomi prilikom pokušaja pristupa ovim komponentama aktivira kolo za otpornost na zloupotrebu.

- Smolu - čvrsta supstanca koja se koristi da obloži osjetljive komponente. Nekada

je elektronska mreža ugrađena u zaštitnu smolu. Bilo kakav pokušaj bušenja kroz smolu ili korištenja hemikalija će oštetiti komponente i aktivirati kolo za detekciju zloupotrebe.

- Detektore temperature - senzori koji su dizajnirani da detektuju varijacije temperature izvan uobičajenih granica. Drastična promjena temperature često zna biti nagovještaj napada, jer na primjer jedan od napada podrazumijeva bukvalno smrzavanje memorije uređaja.

- Diode osjetljive na svjetlost - senzori koji detektuju prodor ili otvaranje kućišta HSM-a.

- Detektore pokreta ili nagiba - senzori koji detektuju pokušaj pomjeranja HSM-a. Jedan od pristupa je da se ukine protok struje ukoliko se naruši fizička stabilnost HSM-a.

- Detektore napona i struje - senzori koji detektuju varijacije napona ili struje izvan normalnog opsega, jer takve anomalije takođe mogu biti nagovještaj napada.

- Sigurnosne čipove - specijalni mikroprocesori unutar kojih kriptografski ključevi mogu ostati zaštićeni i pored uspješnih napada na sve ostale sigurnosne sisteme HSM-a.

Različiti HSM-ovi koriste različite kombinacije navedenih tehnika kako bi izgradili višeslojnu zaštitu od napadača. Takođe, HSM je najčešće podržan rezervnom baterijom, pa ne može, zajedno sa podacima, biti ugrožen prostim isključivanjem struje.

5.3.2 Skladištenje ključeva na HSM-u

Uvjek postoji bar jedan ključ i to najčešće lokalni master ključ koji se nalazi u HSM-u cijelo vrijeme. Neki HSM-ovi čuvaju veliki broj lokalnih master ključeva, gdje svaki ima specijalnu namjenu. Ostali ključevi koji moraju biti skladišteni mogu se ili čuvati u HSM-u ili izvan njega ali su enkriptovani korišćenjem lokalnog master ključa. U slučaju kada je potrebno iskoristiti ključ koji se čuva izvan HSM-a, on prvo mora biti dostavljen HSM-u gdje se osvježava korišćenjem lokalnog master ključa a tek onda upotrebljava. Ovaj pristup podrazumijeva značajno oslanjanje na lokalni master ključ, pa je stoga veoma važno bekapovati ovaj ključ u slučaju njegovog gubitka. Ovaj gubitak je moguć ako HSM padne ili ukoliko je napadnut, jer u tom slulčaju kontrole za otpornost na zloupotrebu mogu obrisati njegovu memoriju. Dakle, odluka da li da se ključ čuva izvan ili unutar HSM-a uključuje izbor između efiksanosti (skladištenje ključeva na HSM-u je efikasnije u smislu brzine jer ne moraju da se učitavaju u HSM i osvježavaju sa lokalnim master ključem) i potrebe za bekapovanjem, što opet ima svoju cijenu.

5.3.3 Drugi tipovi hardvera

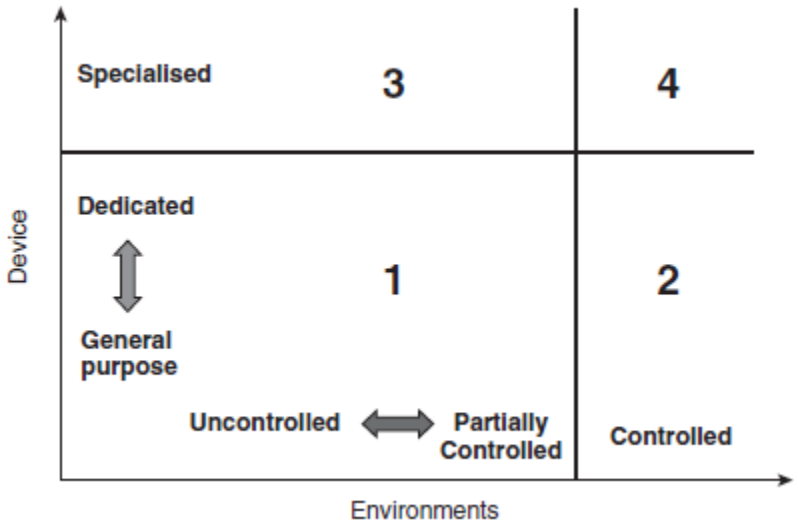
Iako su HSM-ovi najbezbjedniji hardverski uređaji za skladištenje ključeva postoji veliki broj drugih hardverskih uređaja. Ovi hardverski uređaji mogu uključivati neke od mjera otpornosti na zloupotrebu koji su pominjani kao tehnike koje koriste HSM-ovi, dok se neki samo oslanjaju na sam hardver kako bi obezbijedili neki vid otpornosti na napad.

Jedna klasa ovih hardverskih uređaja jesu smart tokeni koji uključuju i smart kartice. Smart tokeni su dizajnirani tako da budu prenosivi i jeftini, pa su samim tim mjere bezbjednosti ograničene. Iako su obično smart tokeni pogodni za skladištenje

ključeva specifičnih za svakog korisnika, neki smart tokeni koji se koriste na primjer da generišu dinamičke šifre (password) obično nisu dovoljno bezbjedni da skladište kriptografske ključeve koji su bitni za zaštitu čitavog sistema, kao što su master ključevi.

5.4 Faktori rizika kod skladištenja ključeva

Rizici koji se javljaju prilikom skladištenja ključeva ne zavise samo od uređaja na kom se ključevi čuvaju, već i od okruženja u kom se uređaji nalaze.



Slika 5.1: Zone rizika pri skladištenju ključeva

Veza ova dva faktora prikazana je kroz dvije dimenzije na grafiku sa Slike 4.1.

Gruba kategorizacija okruženja po bezbjednosti daje nam na razmatranje nekontrolisana, djelimično kontrolisana i kontrolisana okruženja.

Nekontrolisanim (eng. Uncontrolled) okruženjima smatraju se javne sredine kao što su prodavnice i restorani, gdje je nemoguće implementirati mehanizam stroge kontrole pristupa. Na ovim mjestima nekontrolisano se smjenjuju ljudi i prisustvo

bilo kog uređaja čini ga potpuno izloženim potencijalnim napadima. *Djelimično kontrolisane* (eng. Partially controlled) sredine su kancelarije i domovi gdje je moguće implementirati mehanizam osnovne kontrole pristupa (ključ od vrata od kancelarije). Dok se pod *kontrolisanim* (eng. Controlled) okruženjem misli na vojne prostorije ili visoko zaštićene kancelarije gdje je moguće implementirati mehanizam stroge kontrole pristupa (biometrijske kartice za kontrolu pristupa). Ovo su mjesta na kojima važnog samih padataka koji se koriste nameće visok stepen bezbjednosti.

Na sličan način pravimo razliku između uređaja opšte namjene (eng. General purpose) sa centralizovanim operativnim sistemima (laptop), posvećenih uređaja (eng. Dedicated) koji nude limitiranu otpornost na zloupotrebu (maloprodajni terminali) i specijalnih uređaja (eng. Specialised) čija je glavna funkcionalnost da obezbjede sigurnost podataka (HSM).

Ove različite vrijednosti dimenzija obezbjeđuju više konceptualnu podjelu u četri zone koje jasno ilustruju važnost obije dimenzije:

Zona 1 je najmanje sigurna zona i samim tim nudi najveći rizik. Međutim, za većinu aplikacija ovdje je dovoljno obezbijeđena sigurnost. Na primjer, ključ koji je skladišten u enkriptovanoj formi na hard disku nekog računara, može biti dovoljna zaštita za korisnikove lične fajlove.

Zona 2 nastaje pomjeranjem uređaja iz Zone 1 u bezbjedniju sredinu čime se nivo bezbjednosti znatno povećava. Na primjer, za ključ koji se u neenkriptovanoj formi čuva na nekom računaru opšte namjene je obezbijeđena znatno bolja zaštita ukoliko taj računar nije u mreži i nalazi se u fizički bezbjednoj sobi kojoj se pristupa karticom nego ako se nalazi na računaru u javnoj biblioteci.

Zona 3 je zona koja nastaje kada specijalizovani uređaji moraju biti smješteni u nebezbjednim okolinama zbog prirode njihovih aplikacija. Najbolji primjer su automati koji moraju biti dostuni kornicima. Ovakvi uređaji su izloženi različitim vrstama

napada upravo zbog sredine u kojoj se nalaze.

Zona 4 je najbezbjednija zona u kojoj su specijalni uređaji čuvani u kontrolisanoj sredini. Pored toga što je najbezbjednije ovo je i očekivano, najskuplje rješenje.

Primijetimo da se broj zona značajno može proširiti razmatranjem drugih faktora kao što je način aktivacije skladištenih ključeva.

5.5 Bekap i Arhiviranje ključeva

Jedna od gotovo opštih pretpostavki je da upotreba kriptografije donosi sigurnosne benefite, međutim postoje situacije u kojima upotreba kriptografije ima štetne posljedice. Na primjer, pri gubitku dekripcionog ključa za neki podatak skaldišten u enkriptovanoj formi gubi se i sam taj podatak jer nije moguće povratiti šifrat u upotrebljivu formu, ili slično pri gubitku verifikacionog ključa za digitalni potpis.

Ovi primjeri nameću potrebu za realizacijom ideje o bekapu osjetljivih (bitnih) ključeva kao i za njihovim arhiviranjem što predstavlja dugoročno skladištenje do nihovog isteka.

5.5.1 Bekap ključeva

Može biti iznenađujuće lako ″izgubiti″osjetljive kriptografske ključeve. Kako se najčešće ovi ključevi čuvaju na HSM-u jedan od najočiglednijih napada je pokušaj da se fizički pristupi HSM-u i njegovom sadržaju na način da se aktivira neki od okidača za otpornost na zloupotrebu čime dolazi do brisanja memorije uređaja. U ovom slučaju napadač ne dolazi do traženih skrivenih informacija, ali bez bekapa uticaj na organizaciju koja se oslanja na ovaj uređaj može biti velik. Čak i u Zoni 4 postoje razni scenariji koji mogu dovesti do slučajnog ili namjernog brisanja memorije odnosno ključeva.

Kao što je rečeno, ključevi su samo specijalna vrsta podataka, pa se bekap ključeva ne razlikuje mnogo od bekapa običnih (opštih) podataka. Očigledna ali i veoma bitna činjenica je da bezbjednost sistema za bekap ključeva mora biti bar na istom nivou kao bezbjednost samog ključa. Zato, kada su u pitanju top-level ključevi možda je i jedini način adekvatnog bekapa upotreba komponeneti.

5.5.2 Arhiviranje ključeva

Arhiviranje ključeva je u suštini specijalna vrsta bekapa, koja je neophodna u situacijama kada dođe do potrebe za ključevima u vremenu između njihovog isteka i uništenja.

Može postojati zakonski uslov da se neki podaci čuvaju tačno određeno vrijeme. Ukoliko se ti podaci čuvaju u enkriptovanoj formi onda mora biti zakonski određeno da se arhiviraju i njihovi ključevi kako bi se podaci mogli povratiti u odgovarajućem obliku. Na primjer, Londonska berza zahtijeva da se ključevi čuvaju 7 godina.

Često se u organizacijama javlja potreba za verifikacijom digitalnog potpisa nekog dokumenta i nakon perioda isteka ključa koji je korišćen za njegovo potpisivanje. Samim tim, potrebno je arhivirati odgovarajući verifikacioni ključ kako bi se zadovoljile potrebe u budućnosti. Na primjer, Belgijsko zakonodavstvo zahtijeva da se verifikacioni ključevi koji se koriste za elektronske potpise u online bankarskim aplikacijama arhiviraju 5 godina.

Upravljanje procesom skladištenja arhiviranih ključeva jednako je zahtjevno kao kod bekapa ključeva. Jednom kada više ne postoji potreba za arhiviranim ključem on mora biti uništen.

Glava 6

Upotreba ključeva

Nakon razmatranja i obrade procesa generisanja, uspostavljanja i skladištenja ključeva, nastavak studija o životnom ciklusu ključeva vodi ka istraživanju pitanja koja se vezuju za upotrebu kriptografskih ključeva. Najvažnije od njih je separacija ključeva.

6.1 Separacija ključeva

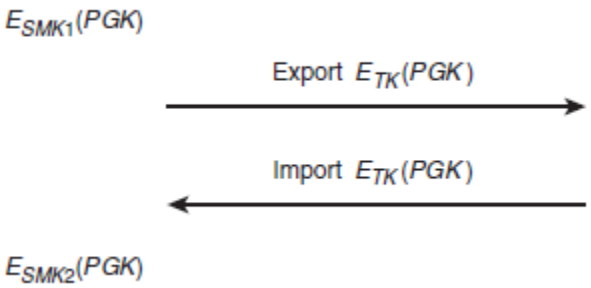
Princip separacije kriptografskih ključeva zasniva se na tome da se ključevi moraju koristi samo u namijenjene svrhe. U ovoj sekciji biće razmotreno zašto je separacija ključeva dobra ideja i kako se realizuje.

6.1.1 Potreba za separacijom ključeva

Problemi koji se mogu javiti zbog nekorišćenja separacije ključeva mogu biti ozbiljni. Potreba za separacijom ključeva u nekim aplikacijama može biti prilično očigledna. Na primjer, može biti slučaj da se za enkripciju ključa i autentifikaciju entiteta sprovode različiti procesi, svaki sa svojim posebnim zahtjevima u pogledu dužine ključa, što čini upotrebu jednog ključa u pogrešne svrhe nemogućim. Sa druge

strane, u drugim aplikacijama može biti prilično primamljivo da se ključ koji je već uspostavljen u jednu svrhu iskoristi u potpuno druge svrhe.

Kao i šifre (passwords), PIN-ovi se ne bi trebali čuvati bilo gdje neenkriptovani. Stoga, PIN-ovi su najčešće skladišteni u enkriptovanoj formi korišćenjem PIN enkripcionog ključa. Ovaj ključ bi se trebao uvjek koristiti samo za enkripciju PIN-a, a nikada za dekripciju enkriptovanog PIN-a. Sa druge strane, običan simetričan ključ koristi se i za enktipciju i za dekripciju. Ako se ova dva ključa nekim slučajem zamijene od strane HSM-a, može nastati ozbiljan problem. Prije svega, može biti moguće da se dekriptuje i otkrije PIN, a sa druge strane i da postane nemoguće da se povrate podaci enkriptovani PIN enkripcionim ključem u normalnom obliku.



Slika 6.1:

Jedna od metoda da se nametne separacija ključeva jeste da se u HSM-u skladište ključevi enkriptovani master ključem koji ima specificiranu jednu namjenu. Na ovaj način, pristup ključu je direktno povezan sa upotrebom master ključa koji identifikuje namjenu ključa. Međutim, mnogi HSM-ovi imaju ulazne i izlazne (import i eksport) funkcije koje omogućavaju transport ključa od jednog do drugog HSM-a. Ključevi se enkriptuju transportnim ključem tokom importa i eksporta.

Na Slici 6.1 je prikazano kako se ova metoda može iskoristiti da se promijeni početna namjena ključa.

1. PIN generacioni ključ - PGK je skladišten na HSM-u, enkriptovan master ključem SMK1 koji je lokalni ključ na HSM-u koji se koristi za skladištenje PIN generacionih ključeva.

2. Kada je HSM-u poslat zahtjev za eksport PGK-a on dekriptuje PGK koristeći SMK1, potom ga ponovo enkriptuje sa transportnim ključem TK i takvog eksportuje.

3. Od strane napadača, HSM-u je data instrukcija da importuje novi MAC ključ. Napadač dolazi do PGK-a koji je enkriptovan sa TK-om.

4. HSM dekriptuje enkritovani PGK koristeći TK, potom ga enkriptuje koristeći master ključ SMK2, koji je HSM-ov lokalni ključ koji se koristi da skladišti MAC ključeve. Na taj način HSM PGK vidi kao MAC ključ.

Ovaj napad neće biti moguć ako se koriste različite varijante transportnog ključa za različite ulazne i izlazne funkcije. Takođe, navedene i slične slabosti najčešće proizilaze iz bezbjedonosne slabosti interfejsa između uređaja na kojima se skladište ključevi iz spoljašnjeg svijeta.

6.1.2 Primjena separacije ključeva

Kako bi se izbjegli navedeni i slični problemi, koriste se različiti mehanizmi primjene separacije ključeva.

Ključevi su najčešće nestruktuirani nizovi bita, pa ne postoji očigledan način na koji se može odrediti namjena ključeva iz njegove osnovne forme. Jedna od glavnih tehnika koja se može koristiti za razlikovanje namjene ključeva je *ugrađivanje ključeva u veći blok podataka*. Najpoznatije tehnike su tehnika upotrebe viška i smještanje ključeva u specijalno formatirane blokove.

Na primjer, ključevi kao što je DES ključ (eng. Data Encryption Standard; najrasprostranjeniji blok algoritam korišćen širom svijeta) su 64-bitne vrijednosti koje u stvari imaju efektivnu dužinu od 56 bita. Dakle, kod njih postoji 8 rezervnih bita koji se mogu iskoristiti u razne svrhe. Originalni DES standard predlaže da se rezervni bitovi iskoriste za detektovanje grešaka u slučaju da DES ključ bude oštećen. Kako ovo nije standardom prihvaćeno, ideja označavanja ključeva je stupila na snagu. Na ovaj način, 8 rezervnih bita iskorišteno je za definisanje namjene ključeva. Nakon što je HSM-u predstavljen ključ i komanda, HSM provjerava oznaku ključa i potvrđuje da li je taj ključ validan za komandu za koju se koristi.

Jedan od primjera bloka ključeva je ANSI TR-31 blok koji omogućava da se ključ predstavi zajedno sa drugim podacima koji su povezani sa njim i ima sledeća polja:

- heder koji uključuje informaciju koja razjašnjava namjenu ključa;
- opcioni heder koji uključuje opcione podatke kao što je datum isteka ključa;
- sam ključ enkriptovan odgovarajućim key encrypting ključem;
- MAC - autentifikator koji obezbjeđuje provjeru porijekla podataka bloka ključa.

Slika 6.2: ANSI TR-31 blok

Bitno je napomenuti da dok je razlikovanje svrhe kriptografskih ključeva korisno, time ipak nije primijenjena separacija ključeva, jer njena primjena zahtijeva kontrolu procedura.

Na intuitivnom nivou, princip separacije ključeva ima smisla, jer postojanje odvojenih ključeva za različite svrhe čini stvari jednostavnim. Međutim, princip separacije

ključeva je upravo to - princip. Njegova primjena ne dolazi bez određenih troškova. Na primjer, njegova primjena može značiti da sistem upravljanja ključevima ima znatno više ključeva da njima upravlja nego kada se princip ne bi koristio. Opet, sa druge strane, u nekim situacijama je neophodan, pa se kao najbolji kompromis nameće korišćenje izvođenja ključeva.

6.2 Promjena ključeva

Većina sistema za upravljanje ključevima zahtijeva sposobnost za promjenom ključeva.

Potreba za promjenom ključa može da bude planirana i neplanirana. Planirana promjena ključa se najčešće odvija u fiksiranim intervalima. Jedan od razloga za planiranom promjenom ključa može biti zbog isteka životnog vijeka ključa ili prosto radi uvježbavanja za neplanirane zahtjeve za promjenom, što je u većini organizacija i najčešći razlog jer je vijek trajanja ključeva prilično velik.

Neplanirana promjena ključa može biti izazvana na različite načine. Na primjer, ukoliko je ključ kompromitovan, ukoliko operativni sistem postane ranjiv ili prosto neki zaposleni neočekivano napusti organizaciju. U nekim od ovih situacija moguće je samo ukloniti odgovarajući ključ iz upotrebe, bez potrebe za njegovim mijenjanjem. Na primjer, ako neki radnik napusti kompaniju najbolje rješenje je da se njegovi lični ključevi, kao na primjer ključevi koje je dijelio samo sa centralnim sistemom, uklone, dok bi grupne ključeve koje je dijelio sa nekim zaposlenim iz kompanije bilo potrebno promijeniti.

Promjena ključeva može biti veoma skupa, u zavisnosti od važnosti samog ključa koji se mijenja. Neplanirana promjena ključa može biti posebno problematična, naročito kada je u pitanju kompromitovanost ključa, jer se onda dovodi u pitanje bilo

koja kriptografska operacija koja se izvršila korišćenjem tog ključa. Jedna od problematičnih posljedica je to da će u tom slučaju biti potrebno promijeniti sve ključeve enkriptovane tim ključem. Minimalna posljedica promjene ključa je potreba za generisanjem i uspostavljanjem novog. Jedna od situacija kada posljedice promjene ključa nisu velike je kada je u pitanju ključ za digitalni potpis. Tada je samo potrebno poništiti sve potpise generisane tim ključem nakon trenutka kada je ključ kompromitovan.

6.3 Uništenje ključeva

Kada više ne postoji potreba za ključem onda on mora biti uništen u sigurnosne svrhe. Do potrebe za uništenjem ključa može doći usljed kraja životnog vijeka tog ključa ili perioda na koji je odlučeno da se ključ arhivira.

Kada je ključ povučen prije svog isteka, zbog neke neplanirane situacije javlja se potreba ili samo za njegovom zamjenom ili potpunim uništenjem, u zavisnosti od važnosti ključa kao i svrhe u koju se koristio.

Kako su ključevi samo specijalne vrste podataka, mehanizmi dostupni za njihovo uništenje su svi mehanizmi dostupni za uništavanje bilo kog podatka. Ipak, kako su ključevi osjetljivi podaci, određeni bezbjedonosni mehanizmi moraju biti korišteni. Ukoliko je potrebno stvarno uništiti ključ, onda nije dovoljno samo obrisati ga. Ne samo da ovo ne uništava ključ, već postoji mogućnost da operativni sistem ima druge privremene kopije ključa na različitim lokacijama.

Bitna napomena je da bilo koji medij koji skladišti podatke o ključevima koje treba uništiti, kao na primjer papir, treba takođe biti uništen. Odgovarajući standardi sadrže vodič o tome kako treba sprovesti sve usputne operacije koje dolaze uz zahtjev za uništenjem ključa.[3]

Glava 7

Zaključak

Već je par puta naznačeno da je upravljanje ključevima glavni interfejs između tehnologije koja koristi kriptografiju i korisnika i sistema koji se oslanja na nju. Upravljanje ključevima je mali ali veoma važan dio mnogo šireg upravljanja bezbjednošću čitavog informacionog sistema.

Za korisnika koji privatno upravlja ključevima sam na svojoj mašini, upravljanje ključevima može uključivati samo selekciju odgovarajućih tehnika za sprovođenje svake od faza životnog ciklusa ključa. Međutim, upravljanje ključevima je monogo kompleksiniji proces za neku organizaciju, shodno raznovrsnosti procesa koji utiču na upravljanje ključevima. Dakle, upravljanje ključevima u sklopu neke organizacije treba da se vodi pravilima i procesima.

Unutar neke organizacije, najčešći način za sprovođenje upravljanja ključevima je preko specifikacija o: polisama upravljanja ključevima, praksi upravljanja ključevima i procedurama upravljanja ključevima.

Naravno, različite organizacije će imati različite formulacije polisa, praksi i procedura upravljanja ključevima, ali rezultat ovih procesa bi trebao da bude da implementacija upravljanja ključevima bude:

- Po dizajnu - Da je cijeli životni cikljus ključa isplaniran unaprijed, a ne u hodu

kao odgovor na događaje koji se usput dešavaju;

- Koherentna - Sve faze životnog ciklusa ključa iako različite, treba da se smatraju međusobno povezanim komponentama uniformnog procesa - potrebno je na njih gledati kao na djelove jedne slike;
- Integrisana - Faze životnog ciklusa ključa su integrisane sa širim zahtjevima i prioritetima organizacije.

U ovom radu razmatrali smo upravljanje ključevima, aspekt kriptografije koji je od velikog značaja za njene korisnike, kao dio koji se u najvećem dijelu oslanja na donošenje odluka i na procese dizajna u individualnom okruženju aplikacije.

Cilj je bio objasniti kroz izlaganje važnosti i detalje o fazama životnog ciklusa upravljanja ključevima koji moraju ostati tajni, dok se upravljanje javnim parovima ključeva suštinski razlikuje i ono nije izučavano u ovom radu.

Bibliografija

[1] Elaine Barker. *Recommendation for Key Management.* National Institute of Standards and Technology, 2016.

[2] J. Orlin Grabbe. *The DES Algorithm Illustrated.* 2006.

[3] Keith Martini. *Everyday Cryptography.* Oxford, 2012.

[4] Kovačević Vladimir. *SEMINARSKI RAD Zaštita podataka primenom kriptografskih metoda.* 2010.