Univerza v Ljubljani

Fakulteta za računalništvo in informatiko
Fakulteta za matematiko in fiziko

Projekt pri predmetu
Kriptografija in teorija kodiranja II

KRIPTOGRAFIJA NA OSNOVI IDENTITETE (Identity based cryptography - IBC)

Avtor:
Peter Nose, 63040285

Mentor:
dr. Aleksandar Jurišič

7. avgust 2008

Kazalo

1 Uvod

Leta 1984, je izraelski kriptograf Adi Shamir[1] predstavil čisto novi koncept na področju kriptosistemov z javnimi ključi, tako imenovano kriptografijo na osnovi identitete (IBC - identity based cryptography). V primerjavi z PKI sistemom (PKI - public key infrastructure), kjer se za javni ključ uporabi neko 'večje' število z določenimi lastnosti, se pri IBC sistemu za javni ključ vzame kar uporabnikovo identiteto. Za njo mora veljati naslednje:

- identiteta pripada samo eni osebi (noben par oseb si ne deli iste identitete),
- oseba ne more zanikati, da identiteta ne pripada njej.

S takimi omejitvami imamo na voljo zelo veliko ključev, kot so: elektronski poštni naslov, telefonska številka, davčna številka, EMŠO, IP naslov, ... lahko pa za javni ključ uporabimo tudi kakšne biometrične podatke, kot so prstni odtisi, ter vzorci zenice ali šarenice. Številne možnosti za izbiro javnega ključa nam omogočajo, da upravljanje z javnimi ključi precej olajšamo. Tako je dobro za javni ključ uporabiti identiteto, ki je poznana vsem ali pa lahko do nje vsaj zelo hitro dostopajo. S tako izbiro javnega ključa si znatno olajšamo potrebo po vzdrževanju in upravljanju z infrastrukturo javnih ključev. Pri IBE sistemu tako ne potrebujemo črnih list niti certifikatnih agencij. Ti dve stvari pa sta dandanes najbolj nadležni stvari PKI sistema.

Poleg ideje o kriptografiji na osnovi identitete pa je Shamir objavil tudi prvo shemo za podpisovanje na osnovi identitete (IBS - identity based signature). Ni pa objavil nobene sheme za šifriranje. Izkazalo se je namreč, da je shemo za podpisovanje občutno lažje skonstruirati, kakor pa shemo za šifriranje. Tako sta se prvi rešitvi za šifriranje na osnovi identitete (IBE - identity based encryption) pojavili šele leta 2001. Takrat so neodvisno izumili taki shemi Boneh in Franklin ter Cocks. Shema prvih dveh je temeljila na osnovi bilinearnega parjenja, medtem ko je Cocksova shema temeljila na problemu kvadratnih ostankov. Obe shemi si bomo natančneje ogledali v nadaljevanju.

Kriptografijo na osnovi identitete lahko razdelimo na več delov, kot so:

- šifriranje na osnovi identitete (IBE),
- podpisovanje na osnovi identitete (IBS),
- šifriranje in podpisovanje ne osnovi identitete,
- dogovor o ključu na osnovi identitete,
- itd.

Pri tem projektu se bomo poglobili predvsem v šifriranja na osnovi identitete.

2 Osnove IBE sistema

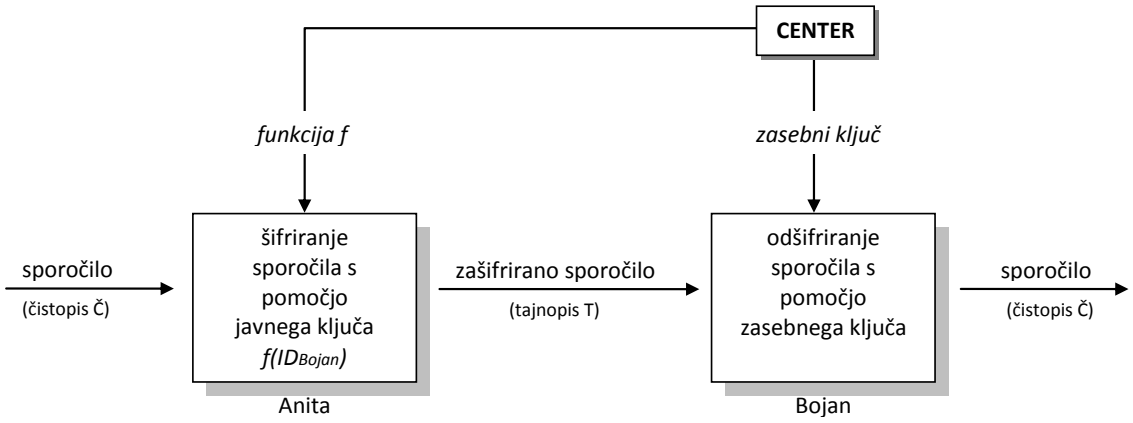
Za lažjo razlago osnov IBE sistema bomo predpostavili, da imamo dve osebi, ki želita med seboj komunicirati. Pošiljatelja bomo poimenovali Anita, prejemnika pa Bojan. Če želi Anita poslati skrivno sporočilo Bojanu, potem mora poznati nek podatek oz. neko informacijo, ki bo enolično določala Bojana. Takemu podatku pravimo *identiteta*. Identiteta se največkrat pojavi v obliki besede, lahko pa je tudi številka oz. digitalna slika. Glede na to, da si ljudje lažje zapomnimo besede, bomo v nadaljevanju kot identiteto Bojana uporabili kar njegov elektronski poštni naslov `bojan@podjetje.si`.

Anita s pomočjo identitete sporočilo zašifrira in ga pošlje Bojanu. Le ta lahko prejeto besedilo odšifrira le pod pogojem, da dobi ustrezen odšifrirni ključ. Ta odšifrirni ključ se nahaja pri neki tretji osebi, ki jo bom v našem primeru poimenovali kar center C[2]. Center bo tako Bojanu izdal ustrezen odšifrirni ključ le pod pogojem, da mu Bojan dokaže, da je lastnik te identitete.

IBE sistem lahko opišemo tudi kot zaporedje naslednjih korakov:

[1] Shamir je eden izmed treh izumiteljev RSA-ja, enega izmed najbolj znanih algoritmov javne kriptografije. Poleg tega je tudi soavtor Feige-Fiat-Shamir-jeve identifikacijske sheme ter eden izmed začetnikov diferencialne kriptoanalize. S svojimi številnimi prispevki je zelo pripomogel k razvoju kriptografije in računalništva.

[2] V literaturi zasledimo več različnih oznak za center kot so PKG (private key generator), KGC (key generation center) in SEM (security mediator).



Slika 1: IBE shema

Inicializacija

Center generira zasebni in javni[3] ključ, označili ju bomo z z_C ter j_C. Javni ključ j_C mora biti javno dostopen vsem uporabnikom sistema ter veljaven daljše časovno obdobje.

Pridobitev zasebnega ključa

Prejemnik Bojan se avtenticira (na nek način dokaže da je lastnik identitete) centru, kateri mu nato preko varnega kanala razkrije zasebni ključ $z_{\mathrm{id_{Bojan}}}$.

Šifriranje

S pomočjo Bojanove identitete $\mathrm{id_{Bojan}}$, javno znane funkcije f[4] in javno znanega ključa j_C Anita zašifrira čistopis $\check{C}$ v tajnopis T.

Odšifriranje

Bojan prejeti tajnopis T s pomočjo svojega zasebnega ključa $z_{\mathrm{id_{Bojan}}}$ odšifrira, ter tako pridobi čistopis $\check{C}$.

3 Primerjava IBE ter PKI sistema

3.1 Prednosti

Prednosti IBE sistema se v največji meri pojavijo zaradi preprostega javnega ključa. Le ta namreč ni naključno generiran, kakor je to v navadi pri PKI sistemu, ampak je v naprej določen. Ker je javni ključ kar identiteta osebe, tako identiteto poznajo vsi. Tako za komunikacijo z večjim številom oseb ne potrebujemo vseh njihovih javnih ključev, ampak le javne parametre centrov, katerim te osebe pripadajo. Če vse osebe pripadajo istemu centru (npr. nekemu podjetju), potem lahko te osebe medsebojno komunicirajo z uporabo enega samega javnega parametra centra. Tudi če osebe ne pripadajo enemu samemu centru, je v praksi centrov občutno manj kakor pa uporabnikov, tako da je šifriranje s pomočjo IBE sistema občutno lažje.

Pri IBE sistemu javni parametri centra navadno veljajo daljšo časovno obdobje in v tem obdobju ne morejo postati neveljavni. To pa nam omogoča, da si javne parametre priskrbimo le enkrat na časovno obdobje in tako ni potrebno vsakokratno preverjanje veljavnosti.

[3] Javnemu ključu centra včasih pravimo tudi javni parametri centra

[4] Za večjo varnost uporabimo injektivno funkcijo f, tako da se dve različni identiteti preslikata v različni javni ključ.

IBE javni ključ
uporabnik@podjetje.si

RSA javni ključ

Eksponent:
65537 (0x10001)

Modul (1024 bit):
00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
e8:35:1c:9e:27:52:7e:41:8f

Slika 2: Primerjava ključa IBE/RSA

Denimo da imamo svoje podjetje. V podjetju imamo poštni strežnik, katerega ves dohodni promet je šifriran s pomočjo IBE sistema. Ker so uporabniki/naslovniki internetne pošte naši zaposleni, je priporočljivo, da se center nahaja nekje na varnem mestu znotraj našega podjetja. V takšnem primeru ima center en zelo dober stranski učinek. Namreč center pozna vse zasebne ključe in zato lahko odšifrira vsa sporočila. Če torej naš center povežemo s poštnim strežnikom, lahko vsako sporočilo odšifriramo, preverimo vsebino ter zašifriramo nazaj. Če pa poznamo vsebino pošte, pa je filtriranje spama, iskanje virusov ter okuženih datotek veliko lažje.

Pri IBE sistemu je dobro še omeniti, da lahko sporočila zašifriramo neglede na to ali je naslovnik že prejel svoj zasebni ključ ali ne. Pravzaprav lahko pošljemo zašifrirano sporočilo tudi osebi, ki sploh še ne obstaja.

3.2 Slabosti

3.2.1 Veljavnost/preklic javnega ključa

Kadar želimo sporočilo šifrirati si moramo priskrbeti javni ključ naslovnika. Poleg tega pa moramo preveriti ali je pridobljeni javni ključ sploh veljaven. Lahko se namreč zgodi, da je veljavnost javnega ključa potekla ali pa je nekomu uspelo pridobiti pripadajoči zasebni ključ in je bil javni ključ zato preklican. Zavedati se namreč moramo, da šifriranje s preteklim oz. preklicanim javnim ključem ni varno. Poleg tega pa verjetno naslovnik nima več pripadajočega zasebnega ključa in tako sploh ne bo moral odšifrirati sporočila.

Pri PKI sistemu je za veljavnost javnega ključa poskrbljeno v samem certifikatu. Slednji namreč vsebuje datum, do katerega je certifikat veljaven. Če pa je potrebno v obdobju veljavnosti certifikata javni ključ preklicati, pa je pri PKI sistemu navadno to organizirano s pomočjo črnih list[5]. Le te pa sčasoma postajajo zelo velike, kar seveda otežuje delo s certifikati.

Pri IBE sistemu veljavnost javnega ključa preprosto vključimo v samo identiteto. Če pošiljamo Bojanu sporočila, zašifrirana s pomočjo njegovega poštnega naslova `bojan@podjetje.si`, potem njegovi poštni naslov spremenimo v `bojan@podjetje.si || trenutno leto`. Če nastavimo trenutno

[5]Vsa digitalna potrdila, ki so iz različnih razlogov neveljavna, objavljajo certifikatne agencije na posebnih seznamih, za katere se je uveljavila kratica CRL (certificate revocation list). Ti seznami se objavljajo na spletnih strežnikih ali pa v imenikih po standardu X.500, kjer so dostopni prek protokola LDAPv3. CRL liste morajo biti neprekinjeno in hitro na voljo.

leto na leto 2008, bo tako javni ključ pridobljen z Bojanovo identiteto `bojan@podjetje.si || 2008` veljaven samo v letu 2008. Po izteku leta 2008 bo Bojanov javni ključ postal neveljaven, novi javni ključ pa bo `bojan@podjetje.si || 2009`. Za odšifriranje sporočil bo moral Bojan vsako leto vsaj enkrat kontaktirati center za pridobitev zasebnega ključa. Naj omenimo še, da Aniti za pošiljanje sporočila Bojanu, ni potrebno vsako leto na novo pridobiti njegov javni ključ oz. certifikat. Anita preprosto Bojanovi identiteti doda trenutno leto, za vse ostalo pa poskrbi center z izdajo/neizdajo zasebnega ključa Bojanu.

Če želimo, da veljavnost javnega ključa traja samo en dan, potem za Bojanovo identiteto vzamemo `bojan@podjetje.si || trenutni dan`. Kot omenjeno že zgoraj, Aniti ni potrebno vsak dan pridobiti Bojanov javni ključ. Tako da enodnevna veljavnost javnega ključa ne oteži pošiljanja sporočil. Ima pa dodatno prednost. V primeru da Bojan zapusti podjetje, center Bojanu ne izdaja več zasebnih ključev. S tem Bojan ne more več brati sporočil, saj ni več član podjetja.

Podobno kot upravljanje z veljavnostjo ključa lahko javnemu ključu dodamo še dodatno funkcionalnost. Bojanovi identiteti `bojan@podjetje.si || trenutno leto`. dodamo dodatno polje *status*, ki pove kakšen tip sporočila je bil poslan (npr. `bojan@podjetje.si || 2008 || status: strogo zaupno`). V takem primeru bo lahko Bojan odšifriral sporočila le v premeru, če imel v letu 2008 dovoljenje za branje strogo zaupnih sporočil.

3.2.2 Obremenitev centra

Pri večjem številu uporabnikov je uporaba enega samega centra nesmiselna. Namreč center mora za vsakega uporabnika izračunati njegov zasebni ključ, preveriti njegovo identiteto ter mu sporočiti zasebni ključ preko nekega varnega kanala. Pri večjih omrežjih je tako center lahka tarča napadalcev.

Ta problem lahko rešimo s pomočjo *hierarhičnega* IBE (HIDE) sistema, ki ga bomo formalno definirali v nadaljevanju. Ta nam omogoča, da lahko korenski center razdeli svoje delo med več centrov. Tako korenski center nič več ne generira zasebnih ključev uporabnikom, ampak generira zasebne ključe samo centrom, ki se nahajajo en nivo nižje v strukturi. Vsak izmed teh centrov pa nato komunicira z določenimi uporabniki ali pa tudi on svoje delo razdeli med svoje centre, ki se nahajajo še en nivo nižje v strukturi. Ker vsak center komunicira le z določenimi uporabniki, ki tvorijo neko domeno, tak center zato poimenujemo kar *domenski* center.

Če želi Anita komunicirati z Bojanom, mora tako pridobiti samo javne parametre Bojanovega korenskega centra. Le ti pa kot pri IBE sistemu trajajo daljše časovno obdobje. HIDE sistem poleg porazdeljenega dela med centri nudi še druge prednosti. Če je eden izmed domenskih centrov napaden in je razkrit njegov zasebni ključ, to ne vpliva na ostale centre, ki se po strukturi nahajajo nad njim. Naj omenimo še da Cocks-ova ter Boneh-Franklin-ova shema, ki ju bomo spoznali v nadaljevanju, te prednosti nimata.

Definicija 1. ***n-terica identitet:*** *Uporabnikovo mesto v hierarhiji HIDE sistema je definirano kot n-terica identitet* $(\mathrm{id}_1, ..., \mathrm{id}_t)$*, kjer je* id_t *korenski center,* id_i *pa je domenski center za* id_{i+1}*,* $1 \leq i < t$*.*

HIDE sistem lahko opišemo tudi kot zaporedje naslednjih korakov:

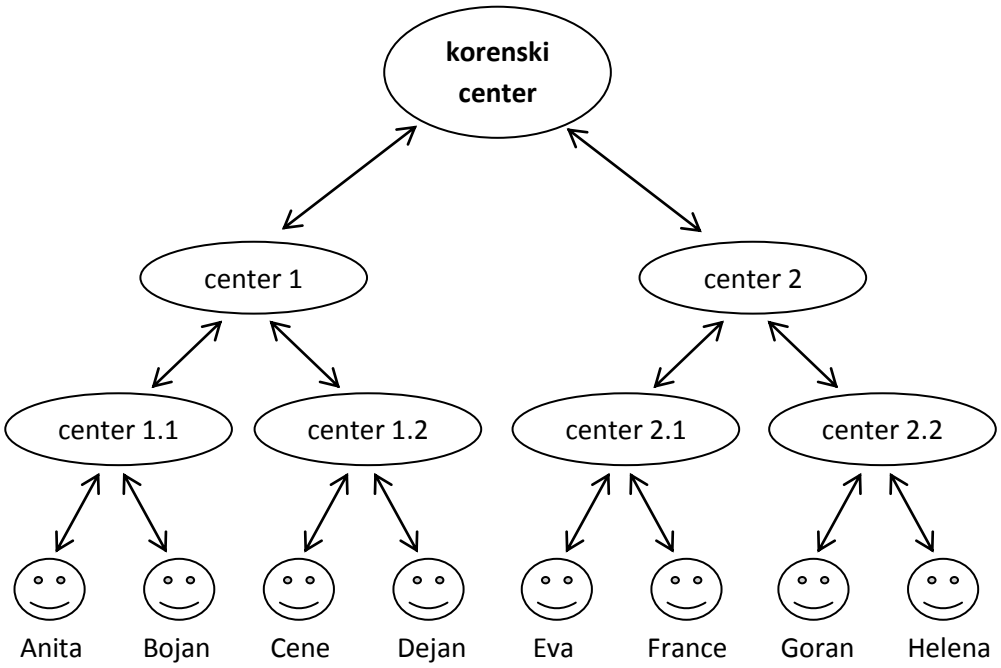
- Korenska inicializacija

  Korenski center generira javno znane parametre in zasebni ključ, katerega pozna le on.

- Nižje-nivojska inicializacija

  Vsak uporabnik na nižjem nivoju si mora priskrbeti javne parametre korenskega centra. V HIDE sistemu si uporabnik na nižjem nivoju ne sme priskrbeti nobenih nižje-nivojskih parametrov. Vseeno pa lahko nižje-nivojski centri generirajo svoje javne parametre, ter svoje zasebne ključe.

- Pridobitev zasebnega ključa



Slika 3: HIDE sistem

Center določen z n-terico identitet $(\mathrm{id}_1, ..., \mathrm{id}_t)$ lahko izračuna zasebne ključe za vse njegove uporabnike, torej za vse centre z n-terico identitet $(\mathrm{id}_1, ..., \mathrm{id}_t, \mathrm{id}_{t+1})$. Pri tem lahko uporabi javne parametre korenskega centra ter svoj zasebni ključ.

- Šifriranje

  Anita s pomočjo javnih parametrov sistema in Bojanovo n-terico identitet zašifrira čistopis $\check{C}$ in dobi tajnopis T.

- Odšifriranje

  Bojan od svojega centra pridobi zasebni ključ s katerim lahko prejeti tajnopis T odšifrira v čistopis $\check{C}$.

3.2.3 Center kot skladišče vseh šifrirnih ključev

Največja nevarnost kriptografije na osnovi identitete je seveda center. Ta ima v lasti vse zasebne ključe vseh uporabnikov. V primeru, da se center ne nahaja v našem podjetju, moramo neki tretji osebi, ki skrbi za center, brezpogojno zaupati. Da bi preprečili krhkost centra, obstaja več rešitev:

1. **uporaba večjega števila centrov**

   S pomočjo Shamir-jeve (t, n) stopenjske sheme za deljenje skrivnosti glavni zasebni ključ razdelimo med večje število centrov. Za pridobitev zasebnega ključa se mora uporabnik avtenticirati vsaj t od skupno n centrem. Tak pristop nam omogoča, da en sam center nima v lasti vseh zasebnih ključev in tako tudi ne more sam odšifrirati sporočil. Slabost takega pristopa je njegova počasnost. Namreč za pridobitev zasebnega ključa se mora uporabnik t-krat avtenticirati ter t-krat po varnem kanalu izmenjati manjši zasebni ključ. Na koncu pa s pomočjo t zasebnih ključev izračunati glavni zasebni ključ.

2. **CBE sistem (Certificate based encryption)**

Kriptografijo na osnovi certifikatov (CBE - certificate based encryption) je na EuroCrypt-u 2003 predstavil C. Gentry. Z njo je hotel združiti tako prednosti IBE kot prednosti PKI sistema. Pri CBE sistemu certifikatna agencija s pomočjo IBE sheme tvori certifikat (oz. podpis). Certifikat je generiran podobno kot pri PKI sistemu, saj vsebuje javni ključ, ime uporabnika, datum izdaje, ..., ter povezuje uporabnika z njegovim javnim ključem. Pri CBE sistemu ima certifikat še eno dodatno lastnost. Namreč certifikat skupaj z uporabnikovim zasebnim ključem deluje kot odšifrirni ključ, s katerim lahko uporabnik odšifrira vsa prejeta sporočila. Ker za odšifriranje sporočil potrebujemo tako veljaven certifikat od certifikatne agencije kot zasebni ključ uporabnika, niti certifikatna agencija brez zasebnega ključa niti uporabnik brez najnovejšega certifikata ne moreta odšifrirati sporočil.

3. **CLE sistem (Certificateless encryption)**

   Leta 2003 sta S. S. Al-Riyami in K. G. Paterson objavila nov koncept javne kriptografije, pri katerem sta se znebila potrebe po uporabi certifikatov. Poimenovala sta jo kriptografija brez uporabe certifikatov (CLE - certificateless encryption). CLE sistem je kombinacija IBE in PKI sistema, pri kateri center ne more odšifrirati sporočil.

   CLE sistem podobno kot IBE sistem uporablja center, torej center ki generira ključe. V primerjavi z IBE sistemom, center ne generira celotnega zasebnega ključa uporabnika ampak le del zasebnega ključa $d_{\text{uporabnik}}$. Tega center izračuna s pomočjo identitete uporabnika in s pomočjo svojega zasebnega ključa. Del zasebnega ključa $d_{\text{uporabnik}}$ je nato preko varnega kanala izdan samo tistemu uporabniku, ki dokaže, da je lastnik pripadajoče identitete.

   Vsak uporabnik CLE sistema od centra prejme $d_{\text{uporabnik}}$. Ker je $d_{\text{uporabnik}}$ le del zasebnega ključa, si uporabnik izbere še neko zaupno informacijo s, ki jo pozna le on. S pomočjo $d_{\text{uporabnik}}$ in s nato izračuna svoj pravi zasebni ključ $z_{\text{uporabnik}}$. Iz s in javnih parametrov centra pa izračuna svoj javni ključ $j_{\text{uporabnik}}$. Za šifriranje sporočila tako potrebujemo javni ključ uporabnika $j_{\text{uporabnik}}$ [6] , njegovo identiteto ter javne parametre centra. Naj omenimo še, da lahko $z_{\text{uporabnik}}$ in $j_{\text{uporabnik}}$ izračunamo v kakršnem koli vrstnem redu, saj za njun izračun potrebujemo le zaupno informacijo s.

   Pri CLE sistemu za šifriranje potrebujemo javni ključ uporabnika $j_{\text{uporabnik}}$ in poznati moramo njegovo identiteto. Ker pri CLE sistemu ne uporabljamo certifikatov, lahko napadalec uporabnikov javni ključ $j_{\text{uporabnik}}$ zamenja z nekim poljubnim $j_{\text{napadalec}}$. Kljub temu da napadalec lahko to stori, še zmeraj ne more odšifrirati sporočil. Namreč za odšifriranje sporočil napadalec potrebuje pravi zasebni ključ, katerega lahko izračuna s pomočjo delnega zasebnega ključa $j_{\text{uporabnik}}$. Tega pa napadalec ne more dobiti, saj bi se moral avtenticirati centru.

4 Sheme

4.1 Boneh-Franklin-ova shema na osnovi bilinearnega parjenja

Za razumevanje Boneh-Franklin-ova shema je potrebno poznati naslednje definicije:

Definicija 2. *Naj bo* $\hat{e} : \mathcal{G}_1 \times \mathcal{G}_2 \to \mathcal{G}_3$ *preslikava iz multiplikativnih grup* $(\mathcal{G}_1, *)$ *in* $(\mathcal{G}_2, *)$ *v neko tretjo multiplikativno grupo* $(\mathcal{G}_3, *)$*, kjer so vse grupe reda* q *in* q *je neko veliko praštevilo. Preslikava* $\hat{e}$ *je* ***bilinearna****, če je:*

- *linearna v prvem argumentu:* $\hat{e}(P * P, Q) = \hat{e}(P, Q) * \hat{e}(P, Q)$,
- *linearna v drugem argumentu:* $\hat{e}(P, Q * Q) = \hat{e}(P, Q) * \hat{e}(P, Q)$,

Definicija 3. *Bilinearna preslikava je* ***nedegenerirana****, če velja* $\hat{e}(P, Q) = 1 \Leftrightarrow (P = 1) \vee (Q = 1)$.

Definicija 4. *Bilinearna preslikava je* ***izračunljiva****, če obstaja učinkovit algoritem za izračun* $\hat{e}(P, Q)$ *za vsak* $P \in \mathcal{G}_1, Q \in \mathcal{G}_2$.

[6] Ker je javni ključ uporabnika izračunan s pomočjo javnega ključa $j_{\text{uporabnik}}$, le ta pa s pomočjo s-ja, taka shema ne sodi v IBE sistem.

Definicija 5. *Bilinearna preslikava je* ***popolna****, če je nedegenerirana in izračunljiva.*

Definicija 6. *Bilinearna preslikava je* ***simetrična****, če je $\mathcal{G}_1 = \mathcal{G}_2$, sicer je* ***nesimetrična****.*

Definicija 7. *Naj bo $\{a_1, ..., a_t\}$ množica elementov. Potem z*

- *$\{a_1, ..., a_t\}^*$ označimo poljubno dolgo zaporedje znakov iz množice $\{a_1, ..., a_t\}$,*
- *$\{a_1, ..., a_t\}^n$ označimo zaporedje n-tih znakov iz množice $\{a_1, ..., a_t\}$.*

Leta 2001 sta Boneh in Franklin objavila prvo IBE shemo na osnovi bilinearnega parjenja. V svoji shemi sta za simetrično bilinearno preslikavo uporabila Weil-ovo parjenje. Ker matematično ozadje tega parjenja presega meje tega projekta, se lahko bralec z njim in z ostalimi pojmi s tega področja (bilinearne preslikave in BDH, odločitveni Diffie-Hellmanov problem, BDH generator parametrov, Diffie-Hellmanova bilinearnostna predpostavka, Diffiee-Hellmanov bilinearni problem, BDH predpostavka, Weilovo parjenje, Tatovo parjenje, itd.) seznani v [9].

Inicializacija BF sheme

Center za sodo k bitno varnost naredi naslednje:

- požene BDH generator s parametrom k in generira praštevilo q, grupi $\mathcal{G}_1$ in $\mathcal{G}_2$ reda q
- izbere popolno bilinearno preslikavo $\hat{e} : \mathcal{G}_1 \times \mathcal{G}_1 \to \mathcal{G}_2$
- izbere si naključni generator $P \in \mathcal{G}_1$, naključno število $z \in \mathcal{Z}_q^*$ ter izračuna $j = sP$
- izbere si zgoščevalno funkcijo $H_1 : \{0,1\}^* \to \mathcal{G}_1^*$ in $H_2 : \mathcal{G}_2 \to \{0,1\}^n$ za nek n.
- javno objavi $params = (q, \mathcal{G}_1, \mathcal{G}_2, \hat{e}, n, P, j, H_1, H_2)$
- varno shrani glavni zasebni ključ $z \in \mathcal{Z}_q^*$

Center vsakemu uporabniku, torej vsaki osebi katera dokaže da je lastnica neke identitete, razkrije zasebni ključ $z_{\text{id}} = zQ_{\text{id}}$, kjer je $Q_{\text{id}} = H_1(\text{id}) \in \mathcal{G}_1^*$.

Šifriranje BF sheme

Za šifriranje sporočila M mora Anita storiti naslednje:

1. $Q_{\text{id}} = H_1(\text{id}) \in \mathcal{G}_1^*$
2. `izbere si naključni element` $r \in \mathcal{Z}_q^*$
3. `pošlje` $C = \langle rP, M \oplus H_2(g_{\text{id}}^r) \rangle$`, kjer je` $g_{\text{id}} = \hat{e}(Q_{\text{id}}, j) \in \mathcal{G}_2^*$

Odšifriranje BF sheme

Naj bo z_{id} zasebni ključ dobljen od centra, ter $C = \langle U, V \rangle$ prejeto šifrirano sporočilo. Potem odšifriranje poteka na naslednji način:

1. $M = V \oplus H_2(\hat{e}(z_{\text{id}}, U))$

4.2 Clifford Cocks-ova shema na osnovi problema kvadratnih ostankov

Za razumevanje Cocks-ove sheme je potrebno poznati naslednje definicije:

Definicija 8. *Število a je* ***kvadratni ostanek*** *po modulu m, če obstaja tak r, da velja:*

$$a \equiv r^2 \ (mod\ m),$$

sicer število a ***ni kvadratni ostanek*** *po modulu m.*

Primer: Če si izberemo $m = 7$, potem so kvadratni ostanki 0, 1, 2, 4.

r	0	1	2	3	4	5	6
r^2	0	1	4	9	16	25	36
$r^2 \ mod \ 7$	0	1	4	2	2	4	1

Definicija 9. ***Legendrov simbol***

$$\left(\frac{a}{p}\right) = \begin{cases} 1, \text{ če je } a \text{ kvadratni ostanek po modulu } p \\ -1, \text{ če } a \text{ ni kvadratni ostanek po modulu } p \end{cases}$$

kjer je največji skupni delitelj števil a in p enak 1.

Definicija 10. ***Jacobijev simbol*** *za* $m = p_1^{v_1} * p_2^{v_2} * ... * p_k^{v_k}$

$$\left(\frac{a}{m}\right) = \left(\frac{a}{p_1}\right)^{v_1} * \left(\frac{a}{p_2}\right)^{v_2} * ... * \left(\frac{a}{p_k}\right)^{v_k}$$

Problem kvadratnih ostankov:
Podano imamo število a ter modul m. Ali je a kvadratni ostanek po modulu m?

Izrek 1. *Naj bosta p, q praštevili. Potem za Jacobijev simbole velja:*

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$$

$$\left(\frac{a}{pq}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$$

$$\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2}$$

$$\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8}$$

$$\left(\frac{a}{p}\right) = \left(\frac{a \pm bp}{p}\right)$$

$$\left(\frac{q}{p}\right) = \left(\frac{p}{q}\right) * (-1)^{(p-1)(q-1)/4}$$

Dokaz. Ni dokaza. □

Problem kvadratnih ostankov je lahek, če je:

- m praštevilo,
- znana faktorizacija števila m.

Zaradi zgornjega izreka lahko na vsakem koraku števec reduciramo po modulu imenovalca in ju nato zamenjamo. Tak algoritem pa je zelo podoben računanju največjega skupnega deljitelja s pomočjo Evklidovega algoritma. Zato je tudi računanje Jacobijevih simbolov hitra operacija.

Izrek 2. *Če velja $p \equiv 3 \ (mod \ 4)$, kjer je p praštevilo, potem*

$$\left(\frac{p-1}{p}\right) = \left(\frac{-1}{p}\right) = -1,$$

ter natanko eno izmed števil a in $-a$ je kvadratni ostanek. Njegov kvadratni koren je $r \equiv a^{\frac{p+1}{4}} \ (mod \ p)$.

Dokaz. Za dokaz glej prosojnice predavanj predmeta KITK2 (2008). □

Izrek 3. *Naj bo $m = p*q$, kjer sta p in q praštevili za kateri velja $p \equiv 3 \ (mod\ 4)$ in $q \equiv 3 \ (mod\ 4)$. Če $\left(\frac{a}{m}\right) = 1$, potem velja da je ali a ali $-a$ kvadratni ostanek po modulu m.*

Dokaz. Iz $\left(\frac{a}{m}\right) = \left(\frac{a}{p}\right) * \left(\frac{a}{q}\right) = 1$ sklepamo:

- če je $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = 1$, potem je število a kvadratni ostanek po modulu p in q, ter posledično tudi po modulu m,
- če je $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1$, potem po (4.) velja, da je $\left(\frac{-a}{p}\right) = \left(\frac{-a}{q}\right) = 1$. Tako je število $-a$ kvadratni ostanek po modulu p in q, ter posledično tudi po modulu m.

□

Denimo, da želimo izračunati kvadratni koren števila a. Če poznamo števili p in q, potem je takšna naloga trivialna, saj je kvadratni koren kar $r = a^{(m+5-(p+r))/8}$. Za kvadratni koren torej vemo, da velja $r^2 \equiv a \ (mod\ m)$ ali $r^2 \equiv -a \ (mod\ m)$, odvisno od tega, katero izmed števil a oz. $-a$ je kvadratni ostanek po modulu m. Od tod sledi tudi naslednja predpostavka, ki ji bomo uporabili v naši shemi.

Predpostavka 1. *Če velja $m = p * q$ in $\left(\frac{a}{m}\right) = 1$, potem ne obstaja učinkovit algoritem, ki bi lahko določil ali je*

$$\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = 1 \quad ali \quad \left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1.$$

Inicializacija Cocks-ove sheme

Center določi in javno objavi:

- $m = p * q$, kjer sta p in q praštevili za kateri velja $p \equiv 3 \ (mod\ 4)$ in $q \equiv 3 \ (mod\ 4)$,
- varno zgoščevalno funkcijo h ter funkcijo $hash$, za katero velja:

```
1.  hash(identiteta):
2.      a = h(identiteta)
3.      dokler (a/m) ≠ 1
4.          a = h(identiteta)
5.      vrni a
```

- identitete uporabnikov

Center vsakemu uporabniku, torej vsaki osebi katera dokaže da je lastnica neke identitete, razkrije zasebni ključ $r = a^{\frac{m+5-(p+q)}{8}}$, kjer je $a = hash(identiteta)$.

Šifriranje Cocks-ove sheme

Anita si izbere šifrirni ključ dolžine n, s katerim bo s pomočjo simetrične kriptografije zašifrirala sporočilo. Predpostavimo lahko, da je ključ zapisan v binarni obliki kot $k_1|k_2|...|k_n$, kjer je $k_i \in \{-1, 1\}$. Ko Anita sporočilo zašifrira, mora poleg njega poslati tudi zasebni ključ. To stori na naslednji način.

```
1.  a = hash(identiteta)
2.  Za vsak k_i ∈ ključ
3.      izberi si število t_i po modulu m, tako da (t_i/m) = k_i
4.      pošlji s_i = t_i + a/t_i
```

Odšifriranje Cocks-ove sheme

Bojan kot lastnik identitete lahko od centra pridobi zasebni ključ $r = a^{\frac{m+5-(p+q)}{8}}$, kjer je $a = hash(identiteta)$. Od Anite poleg zašifriranega sporočila prejme tudi $s = s_1|s_2|...|s_n$, s katerim lahko izračuna simetrični ključ. To stori po naslednjem postopku:

```
1.  Za vsak s_i ∈ s
2.        k_i = ((s_i + 2r)/m)
```

Izrek 4. *Algoritem deluje pravilno.*

Dokaz. Pri dokazu bomo uporabili, da velja $r^2 \equiv a \ (mod\ m)$. To pomeni da Anita ve, da ima Bojan za zasebni ključ kvadratni koren od števila a. Če se Anita ne zaveda, za katerega izmed števil a, $-a$ ima Bojan v lasti njegov kvadratni koren, potem mora Anita postopek ponoviti, izbrati nove naključno izbrane t_i-je in namesto $s_i = t_i + \frac{a}{t_i}$ poslati $s_i = t_i - \frac{a}{t_i}$. Ker velja

$$s_i + 2r \equiv t_i + \frac{a}{t_i} + 2r \equiv t_i + \frac{r^2}{t_i} + 2r \equiv (t_i + r) * (1 + \frac{r}{t_i}) \equiv t_i * (1 + \frac{r}{t_i}) * (1 + \frac{r}{t_i}) \ (mod\ m)$$

$$k_i = \left(\frac{s_i + 2r}{m}\right) = \left(\frac{t_i}{m}\right)$$

in ker poznamo vrednost zasebnega ključa r, lahko izračunamo k_i ne da bi poznali naključno izbrano število t. □

Zahtevnost Cocks-ove sheme

Računska zahtevnost Cocksove sheme je seveda odvisna od dolžine simetričnega ključa. Če je simetrični ključ dolg n bitov, potem mora Anita izračunati n Jacobijevih simbolov in n delitev po modulu m. Bojan pa mora izračunati samo n Jacobijevih simbolov.

Največji problem Cocksove sheme je v velikosti prenosa med Anito in Bojanom. Namreč za vsak bit simetričnega ključa je potrebno prenesti število manjše od m. Če za simetrični ključ vzamemo ključ velikosti 128 bitov in 1024 bitni modul m, potem mora Anita poslati Bojanu 16KB podatkov samo za izmenjavo ključa. Če upoštevamo še, da Anita ne pozna za katerega od števil a, $-a$ je Bojan prejel kvadratni koren, potem skupni prenos za izmenjavo ključa znaša 32KB.

4.3 IB-mRSA shema na osnovi mRSA-ja

Za razumevanje IB-mRSA (identity based mediated RSA) sheme je potrebno poznati naslednje definicije:

1. **RSA** (privzamemo da uporabnik pozna osnove RSA-ja)
2. **mRSA** je enostaven in eleganten postopek, kako zasebni ključ RSA-ja razdeliti med dve osebi. Pri mRSA-ju zasebni ključ razdelimo med nekega uporabnika in med SEM (security mediator). Ker ima vsaka stran v posesti le del celotnega zasebnega ključa, sta za odšifriranje ter za podpisovanje nujno potrebni obe strani. Vsaka stran sama ne more odšifrirati sporočil, prav tako pa ne more učinkovito izračunati preostali del zasebnega ključa, ki ga ima v lasti druga stran. Ena izmed dobrih lastnosti mRSA-ja je odvzemanje pravic oz. preklic certifikata/javnega ključa. Ker se to lahko opravi zelo hitro in učinkovito, je to lastnost dobro vključiti tudi v IB-mRSA.

   Za delovanje mRSA je potreben SEM. SEM je strežnik, do katerega lahko vedno dostopamo in mu delno tudi zaupamo. Če želi Bojan odšifrirati sporočilo, potem mora od SEM-a prejeti še nek dodaten ključ. Ta ključ ni SEM-ov zasebni ključ, ampak je ključ dobljen s pomočjo sporočila in SEM-ovega zasebnega ključa. Ko Bojan prejme ta dodatni ključ lahko odšifrira sporočilo, ne more pa izračunati zasebnega ključa SEM-a. Tako je za odšifriranje sporočil vedno nujno prisoten tudi SEM. To nam omogoča lahko odvzemanje pravic. Če se odločimo

da Bojan nima več pravice odšifrirati sporočil, potem naročimo SEM-u naj ne izdaja več ključev Bojanu.

Obstaja več različni tipov mRSA-ja. Eden izmed njih je aditivni mRSA (+mRSA), ki se uporablja tudi pri IB-mRSA shemi. Kot zanimivost lahko omenimo še, da podobno obstaja tudi multiplikativni mRSA (*mRSA).

Definirajmo osnovne metode za uporabo +mRSA-ja:

Inicializacija

Certifikatna agencija najprej tvori zasebni in javni ključ, ter ga razbije na dva dela. En del pripada uporabniku, drugi del pa SEM-u. Algoritem za sodo k bitno varnost je sledeči:

```
1.  CA: p, q - dve naključni k/2 bitni praštevili
2.  CA: n = p * q
3.  CA: e naključni obrnljiv element iz Z*_φ(n)
4.  CA: d = 1/e (mod φ(n))
5.  CA: d_u naključni element iz Z_n − {0}
6.  CA: d_sem = d − d_u (mod φ(n))
7.  CA: sk = d
8.  CA: pk = (n, e)
```

Ko certifikatna agencija izračuna zasebne in javne ključe, po varnem kanalu pošlje d_{sem} SEM-u in d_u uporabniku. Javni ključ pk pa je javno objavljen v certifikatu.

Šifriranje

Šifriranje poteka na isti način kakor pri RSA-ju.

Odšifriranje

Uporabnik odšifrira prejeto sporočilo s pomočjo SEM-a.

```
1.  Uporabnik:  prejmi zašifrirano sporočilo C
2.  Uporabnik:  pošlji zašifrirano sporočilo SEM-u
3.  Vzporedno:
4.      SEM:
5.        če uporabnik nima več pravic, uporabniku vrni napako
6.        sicer uporabniku vrni PD_sem = C^d_sem (mod n)
7.      Uporabnik:
8.        PD_u = C^d_u (mod n)
9.  Uporabnik:  m = PD_u * PD_sem (mod n)
```

Pri IB-mRSA za razliko od mRSA uporabljamo enaki modul n za vse uporabnike v sistemu. Modul n izda certifikatna agencija in je zato javno dostopen. Poleg modula pa je javno objavljena tudi funkcija f, ki preslika uporabnikovo identiteto v javni ključ. Zaradi večje varnosti od preslikave f zahtevamo da je injektivna, torej da se dve različni identiteti ne preslikata v isti javni ključ. Funkcija f ima podobno vlogo kot funkcija $hash$, ki smo jo uporabili pri Cocksovi shemi.

Posebno je potrebno poudariti, da je tako definirana IB-mRSA shema vse prej kot varna. Namreč vsi uporabniki uporabljajo isti modul n. V primeru da neki osebi uspe razbiti oz. faktorizirati število n, potem lahko taka oseba izračuna vse zasebne ključe oseb, ki pripadajo SEM-u.

Inicializacija IB-mRSA sheme

Certifikatna agencija za sodo k bitno varnost naredi sledeče:

1.	CA: p', q' - dve naključni $\frac{k}{2}$ bitni praštevili, tako da sta števili $p = 2p' + 1$ in $q = 2q' + 1$ tudi praštevili
2.	CA: $n = p * q$
3.	CA: Za vsakega uporabnika: $k' = k - \lvert ID_{uporabnik} \rvert - 8$ $e_{uporabnik} = f(ID_{uporabnik}) = 0^{k'} \lvert\lvert ID_{uporabnik} \rvert\rvert 00000001$ $d_{uporabnik} = \frac{1}{e_{uporabnik}} \pmod{\phi(n)}$ $d_{uporabnik,u}$ naključni element iz $Z_n - \{0\}$ $d_{uporabnik,sem} = d - d_{uporabnik,u} \pmod{\phi(n)}$

Če predpostavimo da števili p' in q' nista enaki 2, potem je število n Blum-ovo število. Namreč za števili p in q velja $p \equiv 3 \pmod 4$ in $q \equiv 3 \pmod 4$. Ker je n Blum-ovo število, je zelo malo verjetno, da ima naključno izbrani element iz Z_n s številom $\phi(n)$ največji skupni delitelj večji od 1. To nam omogoča da lahko uporabimo čim bolj enostavno funkcijo f.

Denimo da za identiteto uporabnika uporabimo njegov poštni naslov. Zapis poštnega naslova v binarni obliki označimo z $ID_{uporabnik}$. Če poštnemu naslovu na konec dodamo še osem bitov 00000001, spredaj pa ga razširimo do želene velikosti k, potem smo dobili liho število. Za tega pa z veliko verjetnostjo velja, da ima z Blum-ovim številom n, največji skupni delitelj enak 1. Tako definirana funkcija f nam pri 1024 bitni velikosti modula n omogoča dolžine poštnih naslovov do 127 znakov. To število lahko še povečamo, če vsi uporabniki pripadajo isti domeni. Tako lahko poštni naslov `bojan@podjetje.si` zmanjšamo na `bojan`, domeno `podjetje.si` pa certifikatna agencija vključi skupaj z javnim modulom n v javni certifikat. Ker so poštni naslovi večino sestavljeni iz imen in priimkov, je taka dolžina poštnega naslova sprejemljiva.

Šifriranje IB-mRSA sheme

1.	Uporabnik: $e_{uporabnik} = f(ID_{uporabnik}) = 0^{k'} \lvert\lvert ID_{uporabnik} \rvert\rvert 00000001$
2.	Uporabnik: preberi modul n iz certifikata izdanega od certifikatne agencije
3.	Uporabnik: z uporabo RSA-ja in javnega ključa (e, n) zašifriraj sporočilo

Odšifriranje IB-mRSA sheme

Sporočilo odšifriramo na isti način kot pri mRSA-ju.

Literatura

[1] J. Baek, J. Newmarch, R. Safavi-Naini, W. Susilo: *A survey of identity based cryptography*, 2004,

[2] C. Gentry, A. Silverberg: *Hierarchical ID based cryptography*, 2002,

[3] D. Boneh, M. Franklin: *Identity based encryption from the Weil pairing*, 2003,

[4] D. Boneh, X. Ding, G. Tsudik: *Identity based mediated RSA*, 2002,

[5] C. Cocks: *An identity based encryption scheme based on quadratic residues*, 2001,

[6] S. S. Al-Riyami and K. G. Paterson: *Certificateless public key Cryptography*, 2003,

[7] B. Libert, J.J. Quisquater: *On constructing certificateless cryptosystems from identity based encryption*, 2006,

[8] C. Gentry: *Certificate based encryption and the certificate revocation problem*, 2003,

[9] M. Korče: *Weilovo parjenje v shemah za šifriranje*, 2006.