Nevarnosti pri napačni uporabi ElGamalovega sistema

1. Če naključno število k ne ostane skrito, lahko izračunamo
$$a = (x - k\delta)\gamma^{-1} \bmod (p-1).$$
2. Število k lahko uporabimo le enkrat, sicer ga je mogoče zlahka izračunati.

Digital Signature Standard

DSS je modifikacija ElGamalovega sistema za podpisovanje. Kot ameriški standard je bil predlagan leta 1991, sprejet pa leta 1994.

Algoritem: Naj bo p praštevilo velikosti L bitov, kjer je $512 \le L \le 1024$ in $64 \,|\, L$, q 160−bitno praštevilo, da $q \,|\, p-1$, ter $\alpha \in \mathbb{Z}_p^*$ q−ti koren enote po modulu p. Definirajmo $\mathcal{P} = \mathbb{Z}_p^*, \mathcal{A} = \mathbb{Z}_q \times \mathbb{Z}_q$ in
$$\mathcal{K} = \{(p, q, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}\}.$$
Vrednosti p, q, α in β so javne, število a pa skrito.

Podpisovanje: podpisnik izbere naključno skrito število k, $1 \le k \le q-1$ in določi
$$\mathrm{sig}_K(x, k) = (\gamma, \delta),$$
kjer je
$$\gamma \equiv (\alpha^k \bmod p) \bmod q$$
in
$$\delta \equiv (x + a\gamma)\, k^{-1} \pmod{q}.$$

Za število δ mora veljati $\delta \not\equiv 0 \pmod{q}$.

Preverjanje podpisa: najprej izračunamo
$$e_1 \equiv x\delta^{-1} \quad \text{in} \quad e_2 \equiv \gamma\delta^{-1}.$$
Potem je
$$ver_K(x, \gamma, \delta) = \text{true}$$
$$\Updownarrow$$
$$(\alpha^{e_1}\ \beta^{e_2} \bmod p) \bmod q = \gamma.$$

Podobno kot pri ElGamalovi shemi je podpisovanje hitrejše od preverjanja (za razliko od RSA).

Prikrit kanal v algoritmu DSA

V algoritmu DSA obstaja prikrit kanal, ki omogoča:

(a) vključitev šifriranega sporočila v podpis, ki ga lahko prebere le tisti, ki pozna dodaten ključ;

(b) razkritje skritega ključa, brez vednosti njegovega lastnika.

Eno možnost za (a) si oglejmo na naslednji foliji, točko (b) pa prihranimo za domačo nalogo.

Primer: Izberimo n tajnih praštevil $p_1, \ldots, p_n$ in poskusimo v podpis skriti binarno zaporedje $b_1, \ldots, b_n$. Naključno število k izbiramo toliko časa, da za vsak $1 \le i \le n$ velja

$b_i = 1 \Longrightarrow \gamma$ je kvadratni ostanek po modulu p_i,

$b_i = 0 \Longrightarrow \gamma$ ni kvadratni ostanek po modulu p_i,

kjer je $\mathrm{sig}_K(x, k) = (\gamma, \delta)$.

Napadi

Uganjevanje fraz, ki jih uporabljamo za gesla

primer	število znakov	zahtevnost	dolžina gesla	čas za razbijanje
mucka	5	25 (majhne črke)	12 bitov	40 minut
br1a9Az	7	62 (črke in številke)	24 bitov	22 let
TH,X1lb<V+	10	95 (znaki na tipkov.)	40 bitov	nedosegljivo

Če uporabimo angleško ali slovensko besedo, dobimo zaporedje s približno 1.3 biti entropije na en znak (t.j. prostor za besedo proti popolnoma naključnim znakom).

Napadi z grobo silo (angl. Brute Force Attack)

posameznik ima 1 PC in programsko opremo ($2^{17} - 2^{24}$ ključev/sek.).

majhna skupina, 16 PC ($2^{21} - 2^{28}$ ključev/sek.).

akademska omrežja, 256 PC ($2^{25} - 2^{32}$ ključev/sek.).

veliko podjetje z \$1.000.000 za strojno opremo (2^{43} ključev/sek.).

vojaška obveščevalna organizacija z \$1.000.000.000 za strojno opremo in napredno tehnologijo (2^{55} ključev/sek.).

Napadi z grobo silo

dolžina ključa (v bitih)	posamični napadalec	majhne skupine	razisko-valna omrežja	velika podjetja	vojaške obveščevalne službe
40	tedni	dnevi	ure	milisekunde	mikrosekunde
56	stoletja	desetletja	leta	ure	sekunde
64	tisočletja	stoletja	destletja	dnevi	minute
80	∞	∞	∞	stoletja	stoletja
128	∞	∞	∞	∞	tisočletja

Povprečen čas za napad z grobo silo

dolžina ključev (v bitih)	število možnih ključev	potreben čas za eno šifriranje/μsek.	potreben čas za 10^6 šifriranj/μsek.
32	$2^{32}=4.3\times10^{9}$	$2^{31}\mu\text{sec}\approx36$ min	≈2 milisek.
56	$2^{56}=7.2\times10^{16}$	$2^{55}\mu\text{sec}\approx1142$ let	≈10 ur
128	$2^{128}=3.4\times10^{38}$	$2^{127}\mu\text{sec}\approx5\times10^{24}$	$\approx5\times10^{18}$ let

Napadi na PKS

Napadi na DSA

- Metoda Index Calculus ($p \approx 2^{1024}$)

- Pollardova ρ-metoda ($\sqrt{\pi q/2}$, $q \approx 2^{160}$)

Napadi na ECDSA

- Pollardova ρ-metoda ($\sqrt{\pi n/2}$, $n \approx 2^{160}$)

Programski napadi

MIPS računalnik lahko opravi 4×10^4 seštevanj točk na eliptični krivulji na sekundo.

(Ta ocena je precej konzervativna. Posebaj prirejeno integrirano vezje s frekvenco ure 40 MHz, ki opravlja operacije na eliptični krivulji nad obsegom $GF(2^{155})$ in lahko izvede 40.000 seštevanj na sekundo.)

Na osnovi tega zaključimo, da je število seštevanj na eliptični krivulji na $GF(2^{155})$ izvedeno na MIPS računalniku v času enega leta naslednje

$$(4 \times 10^4) \cdot (60 \times 60 \times 24 \times 365) \approx 2^{40}.$$

Spodnja tabela nam kaže kolikšno računsko moč potrebujemo za računanje problema diskretnega logaritma z uporabo Pollard ρ-methodo za različne vrednosti števila n. MIPS leto je ekvivalentno računski moči 1 MIPS računalnika, ki je na voljo eno leto.

velikost obsega (v bitih)	velikost števila n	$\sqrt{\pi n/2}$	MIPS let
155	150	2^{75}	3.8×10^{10}
210	205	2^{103}	7.1×10^{18}
239	234	2^{117}	1.6×10^{23}

Npr. če imamo na voljo 10.000 računalnikov z močjo 1.000 MIPS in je $n \approx 2^{150}$, potem je lahko problem diskretnega logaritma na eliptični krivulji rešen v 3.800 letih.

Prejšnjo tabelo je zanimivo primerjati s Odlyzko-vo tabelo, ki kaže kolikšno računsko moč potrebujemo za faktorizacijo celih števil s sedanjo verzijo splošnega NFS algoritma.

velikost števila n (v bitih)	MIPS let
512	3×10^4
768	2×10^8
1024	3×10^{11}
1280	1×10^{14}
1536	3×10^{16}
2048	3×10^{20}

Hardwarski napadi

Za bolj perspektiven napad (s strani dobro financiranega napadalca) na ECC, bi bilo potrebno narediti specializirano programsko opremo za paralelno iskanje na osnovi Pollard ρ-metode.

Van Oorschot and Wiener ocenjujeta: za $n \approx 10^{36} \approx 2^{120}$ bi računalnik z $m = 325.000$ procesorji (cena okoli 10 milijonov USD) lahko izračunal diskretni logaritem v približno 35 dneh.

Poudariti moramo, da računanje diskretnega logaritma na $E(\mathbb{Z}_p)$ v zgoraj omenjenih napadih odkrije **en sam** *zasebni ključ.*

M. Blaze, W. Diffie, R. Rivest, B. Schneier, T. Shimomura, E. Thompson, and M. Wiener, January 1996, (`http://theory.lcs.mit.edu/rivest/publications.html`) govorijo o minimalnih dolžinah ključev potrebnih za varen simetrični sistem (npr. DES ali IDEA):

> *Da bi zagotovili ustrezno zaščito proti najbolj resnim grožnjam (npr. velike komercialne ustanove in vladne agencije) mora ključ biti dolg vsaj 75 bitov. Za zaščito za naslednjih 20-let morajo ključi biti dolgi vsaj 90 bitov (pri tem upoštevamo pričakovano rast računske moči).*

Če posplošimo te zaključke na eliptične kripto-sisteme, mora biti praštevilo n, ki zagotovlja kratkoročno varnost, dolgo vsaj 150 bitov, za srednjeročno varnost pa vsaj 180 bitov.

Dolžina ključev

simetrične šifre (AES)	**asimetrične** (RSA, DSA, DH)	**eliptične krivulje**
40 bitov	274 bitov	80 bitov
56 bitov	384 bitov	106 bitov
64 bitov	512 bitov	132 bitov
80 bitov	1024 bitov	160 bitov
96 bitov	1536 bitov	185 bitov
112 bitov	2048 bitov	237 bitov
120 bitov	2560 bitov	256 bitov
128 bitov	3072 bitov	270 bitov

Digitalni podpisi v $\mathbb{Z}_p$ in na EC

grupa	$\mathbb{Z}_p^*$	$E(\mathbb{Z}_p)$
elementi	množica celih števil $\{1, 2, \ldots, p-1\}$	točke (x, y), ki zadoščajo enačbi eliptične krivulje E in še točka v neskončnosti
operacija	množenje po modulu p	seštevanje točk na eliptični krivulji
oznake	elementi: g, h množenje: $g \times h$ multiplikativni inverz: h^{-1} deljenje: g/h potenciranje: g^a	elementi: P, Q seštevanje: $P + Q$ nasprotna točka: $-Q$ odštevanje: $P - Q$ skalarno množenje točke: aP
problem diskretnega logaritma	Za dana $g, h \in \mathbb{Z}_p^*$ poišči tako celo število a da je $h = g^a \bmod p$.	Za dani točki $P, Q \in E(\mathbb{Z}_p)$ poišči tako celo število a da je $Q = aP$.

Grupe

Digital Signature Algorithm (DSA) eliptični analog ECDSA

DSA	ECDSA
1. Izberi praštevili p in q velikosti $2^{1023} < p < 2^{1024}$, $2^{159} < q < 2^{160}$, tako da $q \mid p-1$.	1. Izberi tako eliptično krivuljo E: $y^2 = x^3 + ax + b$ nad $\mathbb{Z}_q$, da je število $\lvert E(\mathbb{Z}_p)\rvert$ deljivo s praštevilom $n \approx 160$-bitov.
2. $t \in \mathbb{Z}_p^*$, izračunaj $g = t^{(p-1)/q} \bmod p$, potem je $g \neq 1$ in ima red q v $\mathbb{Z}_p^*$.	2. Izberi točko P na $E(\mathbb{Z}_q)$ katere red je praštevilo n.
3. Uporabi multiplikativno grupo $\{g^0, g^1, \ldots, g^{q-1}\}$	3. Uporabi aditivno grupo $\{\mathcal{O}, P, 2P, \ldots, (n-1)P\}$

Generiranje ključa pri DSA in ECDSA

DSA	ECDSA
1. Izberi naključno celo število $x \in [2, q-2]$, tj. **zasebni ključ**	1. Izberi naključno celo število $d \in [2, n-2]$, tj. **zasebni ključ**
2. Izračunaj $y = g^x \bmod p$, **javni ključ** je (p, q, g, y).	2. Izračunaj $Q = dP$, **javni ključ** je (E, n, q, Q).

DSA	ECDSA
q	n
g	P
x	d
y	Q

Podpisovanje sporočila m

DSA	ECDSA
1. Izberi naključno celo število $k \in [2, q-2]$.	1. Izberi naključno celo število $k \in [2, n-2]$.
2. Izračunaj $g^k \bmod p$, $r = (g^k \bmod p) \bmod q$, $0 \neq s = k^{-1}(h(m) + xr) \bmod q$.	2. Izračunaj $kP = (x_1, y_1)$, $r = x_1 \bmod n$, $0 \neq s = k^{-1}(h(m) + dr) \bmod n$.
Podpis je par (r, s).	

Preverjanje podpisa (r, s) sporočila m osebe A

DSA	ECDSA
1. Preskrbi si avtentično kopijo javnega ključa osebe A: (p, q, g, y)	(E, n, q, Q)
2. Izračunaj $s^{-1} \bmod p$ in $h(m)$, $u_1 = h(m)s^{-1} \bmod q$, $u_2 = rs^{-1} \bmod q$, $v = (g^{u_1} y^{u_2} \bmod p) \bmod q$.	2. Izračunaj $s^{-1} \bmod n$ in $h(m)$ $u_1 = h(m)s^{-1} \bmod n$, $u_2 = rs^{-1} \bmod n$, $u_1 P + u_2 Q = (x_0, y_0)$ in $v = x_0 \bmod n$.
Sprejmi podpis samo in samo če je $v = r$.	

SigGen z EC

Razvita je bila v **Certicom Corp., Kanada**, v sodelovanju s Schlumberger Smart Cards and Systems.

Uporablja Motorolin čip 68SC28:
- ROM 12.790 zlogov,
- EEPROM 8.112 zlogov,
- RAM 240 zlogov.

Vsebuje tehnologijo MULTIFLEX^{TM} ter tehnologijo eliptičnih krivulj $(CE)^2$, ki jo razvija podjetje Certicom Corp.

SigGen kartica je zelo prikladna za končnega uporabnika ter za proces prepoznavanja:

- je poceni,
- podpis je opravljen v pol sekunde,
- rabi samo 90 zlogov RAM-a,
- program ne zasede niti 4 KB.

Je edina pametna kartica, ki opravi digitalni podpis kar z obstoječim procesorjem.

Eliptični kripto-sistemi nudijo največjo moč glede na število bitov ključa med današnjimi javnimi kripto-sistemi.

Manjši ključi omogočajo

- manjše sistemske parametre,
- manjša potrdila z javnimi ključi,
- hitrejšo implementacijo,
- manjše zahteve po energiji,
- manjše procesorje,

itd.

Enkratni podpis

Z istim ključem lahko podpišemo le en dokument. Ponavadi algoritem temelji na enosmernih funkcijah.

Lamportova shema: $\mathcal{P} = \{0,1\}^{k\in\mathbb{N}}$, $|Y| < \infty$, enosmerna funkcija $f : Y \to Z$.

Naključno izberemo matriko $(y_{ij}) \in Y^{k\times 2}$ in določimo matriko enake velikosti z elementi $z_{ij} = f(y_{ij})$.

Ključ K sestavljata obe matriki, prva je skrita, druga pa javna.

Podpisovanje:

$$\mathrm{sig}_K(x_1,\ldots,x_k) = (y_{1,x_1},\ldots,y_{k,x_k}).$$

Preverjanje podpisa:

$$\mathrm{ver}_K(x_1,\ldots,x_k,a_1,\ldots,a_k) = \text{true}$$
$$\Updownarrow$$
$$f(a_i) = z_{i,x_i}, \quad 1 \le i \le k.$$

Napadalec ne more ponarediti podpisa, saj ne more obrniti enosmerne funkcije f, da bi izračunal y-e.

Če pa bi podpisali dve različni sporočili z isto shemo, potem bi napadalec lahko poneveril podpis novih sporočil.

Primer: Naj bo $f(x) = 3^x \pmod{7879}$, ključ pa sestavljen iz matrik

$$\begin{pmatrix} 5831 & 735 \\ 803 & 2467 \\ 4285 & 6449 \end{pmatrix} \text{ in } \begin{pmatrix} 2009 & 3810 \\ 4672 & 4721 \\ 268 & 5731 \end{pmatrix}.$$

Potem je podpis za $x = (1,1,0)$ enak $(y_{1,1}, y_{2,1}, y_{3,0}) = (735, 2467, 4285)$.

Pomanjkljivost te sheme je velikost podpisa (za vsak bit čistopisa število med 1 in Y).

Spernerjeva lema

Naj bo $\mathcal{F}$ taka družina podmnožic n-elementne množice, da noben njen element ni vsebovan v kakem drugem elementu iz $\mathcal{F}$. Potem ima družina $\mathcal{F}$ največ

$$\binom{n}{\lfloor n/2 \rfloor} \text{ elementov.}$$

Bos-Chaumova shema za enkratni podpis

$\mathcal{P} = \{0,1\}^{k\in\mathbb{N}}$, $n \in \mathbb{N}$ tak, da je $2^k \le \binom{2n}{n}$.
B je množica z $2n$ elementi in

$$\phi : \{0,1\}^k \to \mathcal{B}$$

injekcija, kjer je $\mathcal{B}$ množica n-teric iz B.

Naj bo $f : Y \to Z$ enosmerna funkcija.

Naključno izberemo vektor $\mathbf{y}= (y_i) \in Y^{2n}$.

Naj bo ključ K tajni vektor $\mathbf{y}$ in javni vektor $(f(y_i))$.

$$\mathrm{sig}_K(x_1,\ldots,x_k) = \{y_j \,|\, j \in \phi(x_1,\ldots,x_k)\}.$$

in

$$\mathrm{ver}_K(x_1,\ldots,x_k,a_1,\ldots,a_n) = \text{true}$$
$$\Updownarrow$$
$$\{f(a_i) \,|\, 1 \le i \le n\} = \{z_j \,|\, j \in \phi(x_1,\ldots,x_k)\}.$$

Uporabili smo $2^k \le \binom{2n}{n}$. Ocenimo binomski koeficient in dobimo

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2}$$

oziroma z uporabo Stiringove formule $2^{2n}/\sqrt{(\pi n)}$.
Od tod dobimo

$$k \le 2n - \frac{\log_2(n\pi)}{2}.$$

Asimptotično je torej n blizu $k/2$, zato smo dobili 50% redukcijo dolžine podpisa.

Slepi podpis

Želimo, da nam kdo podpiše dokument, hkrati pa nočemo, da bi podpisnik videl njegovo vsebino (npr. notarji, banke pri elektronskem denarju).

Algoritem (Chaum): Anita želi od Bojana podpis dokumenta x, $1 \leq x \leq n-1$, pri čemer je (n, e) Bojanov javni ključ za algoritem RSA, d pa zasebni ključ.

1. *Anita izbere takšno skrito naključno število k, da velja $0 \leq k \leq n-1$ in $D(n,k) = 1$.*
   Nato zastre dokument, tj. izračuna
   $$m = xk^e \bmod n,$$
   in ga pošlje Bojanu.
2. *Bojan podpiše zastrti dokument*
   $$s = m^d \bmod n.$$
3. *Anita odstre podpisani dokument*
   $$y = k^{-1}s \bmod n.$$

Podpisi brez možnosti zanikanja

Podpisa ni mogoče preveriti brez sodelovanja podpisnika, podpisnik pa tudi ne more zanikati, da bi že podpisani dokument res podpisal

(razen če odkloni sodelovanje pri podpisu, kar pa lahko pojmujemo kot priznanje, da je podpis v resnici ponarejen).

Primer algoritma (Chaum-van Antwerpen):

Naj bosta q in $p = 2q+1$ praštevili, $\alpha \in \mathbb{Z}_p^*$ element reda q, $1 \leq a \leq q-1$ in $\beta = \alpha^a \bmod p$.

Grupa G je multiplikativna podgrupa reda q grupe $\mathbb{Z}_p^*$ (G sestavljajo kvadratični ostanki po modulo p).

Naj bo $\mathcal{P} = \mathcal{A} = G$ in

$$\mathcal{K} = \{(p, \alpha, a, \beta) : \beta = \alpha^a \bmod p\}.$$

Števila p, α in β so javna, vrednost a pa je skrita.

Podpisovanje (Bojan podpiše dokument $x \in G$):

$$y = \mathrm{sig}_K(x) = x^a \bmod p.$$

Preverjanje podpisa:

1. Anita izbere naključni števili $e_1, e_2 \in \mathbb{Z}_q^*$. Nato izračuna $c = y^{e_1}\beta^{e_2} \bmod p$ in ga pošlje Bojanu.
2. Bojan izračuna $d = c^{a^{-1} \bmod q} \bmod p$ in ga vrne Aniti.
3. Anita sprejme podpis kot veljaven, če je
   $$d = x^{e_1}\alpha^{e_2} \bmod p.$$

Izrek. *Če je $y \not\equiv x^a (\bmod p)$, potem bo Anita sprejela y za veljaven podpis čistopisa x z verjetnostjo $1/q$.*

Poleg algoritmov za podpisovanje in preverjanje obstaja še algoritem (*disavowal protocol*), s katerim lahko podpisnik dokaže, da je ponarejen podpis res ponarejeni, hkrati pa ne more zanikati, da pravega podpisa ni napravil sam.

Primeri podpisov brez možnosti zanikanja

- *Entrusted undeniable signature*: *disavowal* protokol lahko izvede le za to določena ustanova, npr. sodišče.
- *Designated confirmer signature*: ob podpisu sami določimo, kdo bo namesto nas sodeloval pri preverjanjih podpisov. Podpišemo lahko še vedno le mi.
- *Convertible undeniable signature*: shema vsebuje skrito število. Do razkritja tega števila mora pri preverjanju podpisa sodelovati podpisnik.
  Po razkritju lahko kdorkoli preveri podpis sam
  (kot pri običajnem digitalnem podpisu).

Skupinski podpisi

Lastnosti:

- Dokumente lahko podpisujejo le člani določene skupine.
- Kdorkoli lahko preveri, da je dokument podpisal nekdo iz omenjene skupine, vendar ne more ugotoviti, kdo je to bil.
- V primeru spora je možno podpis "odpreti" in identificirati podpisnika.

Fail-stop podpisi

Če bi ponarejevalec z metodo grobe sile našel skriti ključ, bi lahko v večini sistemov za digitalne podpise podpis ponaredil. Fail-stop sistemi takšno možnost onemogočijo tako, da vsakemu javnemu ključu priredijo več skritih ključev.

Algoritem (van Heyst - Pedersen)

Generiranje ključa se razdeli med Anito in TTP (*Trusted Third Party*).

TTP izbere praštevili q in $p = 2q + 1$ (diskretni algoritem je težko izračunljiv), element $\alpha \in \mathbb{Z}_p^*$ reda q ter skrito naključno število a_0, $1 \leq a_0 \leq q - 1$ in izračuna $\beta \equiv \alpha^{a_0} \pmod{p}$. Nato Anita pošlje četverko (p, q, α, β) in izbere skrita naključna števila $a_1, a_2, b_1, b_2 \in \mathbb{Z}_q$, ki predstavljajo njen skriti ključ, ter določi svoj javni ključ $(\gamma_1, \gamma_2, p, q, \alpha, \beta)$, kjer je

$$\gamma_1 = \alpha^{a_1}\,\beta^{a_2} \pmod{p} \text{ in } \gamma_2 = \alpha^{b_1}\,\beta^{b_2} \pmod{p}.$$

Podpisovanje: $y = \mathrm{sig}_K(x) = (y_1, y_2)$, kjer je

$$y_1 \equiv a_1 + x\,b_1 \pmod{q}$$

in

$$y_2 \equiv a_2 + x\,b_2 \pmod{q}.$$

Preverjanje podpisa:

$$\mathrm{ver}_K(x, y_1, y_2) = \text{true} \iff \gamma_1\gamma_2^x \equiv \alpha^{y_1}\beta^{y_2} \pmod{p}.$$

Opombe:

1. Natanko q^2 četverk (a_1', a_2', b_1', b_2'), kjer so elementi iz $\mathbb{Z}_q$, da enaki vrednosti (γ_1, γ_2) v javnem ključu.
2. Teh q^2 četverk da pri istem dokumentu x q različnih podpisov.
3. Naj bo Q_1 množica q četverk, ki da pri x enak podpis. Potem da ta množica pri drugem dokumentu q različnih podpisov.

Varnost sistema

Recimo, da želi nekdo ponarediti Anitin podpis za sporočilo x'.

1. Če ponarejevalec pozna le skriti ključ, ki pripada javnemu, je verjenost $1/q$, da je njegov podpis enak Anitinemu.
2. Ponarejevalec ima dostop do drugega sporočila x in Anitinega podpisa (y_1, y_2). Po tretji opombi je verjetnost spet $1/q$.