Admin:

Outline:

- El Gamal sigs (review)
- Digital Signature Standard (DSS)
- Gap groups & bilinear maps
- BLS (Boneh, Lynn, Shacham) signatures
- Secret-sharing & threshold cryptography

El Gamal signatures

Public system parameters p prime
g generator

Keygen: $x \in_R \{0,1,\ldots,p-2\}$ $SK = x$
$y = g^x$ $PK = y$

Sign(M): $m = h(M)$
$k \in_R Z_{p-1}^*$ [$\gcd(k, p-1) = 1$]
randomized signing
$r = g^k$ [hard work is indep of M]

$ks + rx = m$

$$s = \frac{(m - rx)}{k} \pmod{p-1}$$

$\sigma(M) = (r, s)$

Verify: check $0 < r < p$
" $y^r r^s = g^m \pmod p$ where $m = h(M)$
Return True if both checks pass else return False

Correctness:

$$g^{rx} g^{sk} = \underbrace{g^{rx+sk} \stackrel{?}{=} g^m} \pmod p$$

$$\equiv$$

$$rx + ks \stackrel{?}{=} m \pmod{p-1}$$

$$\equiv$$

$$s = \frac{(m-rx)}{k} \pmod{p-1}$$

(if $\gcd(k, p-1) = 1$)

[El Gamal signatures, cont.d]

That was original version.

Theorem: El Gamal is existentially forgeable (without h fn or h = identity)
↑ note: CR!

Proof: Let $e \in_R Z_{p-1}$

$$r \leftarrow g^e y \pmod p$$
$$s \leftarrow -r \pmod{p-1}$$

(r,s) is sig for message $m = es \pmod{p-1}$

$$y^r r^s \stackrel{?}{=} g^m$$
$$g^{xr}(g^e y)^{-r} = g^{-er} = g^{es} = g^m \quad \text{for } m = es \pmod{p-1}$$

But: It is easy to fix.

Modified El Gamal (Pointcheval/Stern 1996)

sign(M): $k \in_R Z_p^*$

$$r = g^k \pmod p$$
$$m = h(M \| r) \quad \Leftarrow ***$$
$$s = \frac{(m - rx)}{k} \pmod{p-1}$$

$$\sigma(M) = (r,s)$$

Verify: check $0 < r < p$
check $y^r r^s = g^m$ where $m = h(M\|r)$.

10/23/06 L12.12

Thm: (Modified) El Gamal is existentially unforgeable against adaptive chosen message attack, in ROM, assuming DLP is hard.

Digital Signature Standard (DSS - NIST 1991)

Public parameters: q prime $\quad |q| = 160$ bits

$p = nq+1$ prime $\quad |p| = 1024$ bits

g_0 generates Z_p^*

$g = g_0^n$ generates G_q - subgroup of Z_p^* of order q

Keygen: $x \in_R Z_q$ $\quad$ SK $\quad |x| = 160$ bits

$y = g^x \pmod p$ $\quad$ PK $\quad |y| = 1024$ bits

Sign(M): $k \in_R Z_q^*$ $\quad$ (i.e. $1 \le k < q$)

$r = (g^k \bmod p) \pmod q$ $\quad |r| = 160$ bits

$m = h(M)$

$s = (m + rx)/k \pmod q$ $\quad |s| = 160$ bits

redo if $r = 0$ or $s = 0$

$\sigma(M) = (r, s)$ $\quad |\sigma| = 320$ bits

Verify $(PK, M, (r,s))$

Check $y^{r/s} g^{m/s} \pmod p \pmod q \stackrel{?}{=} r$

where $m = h(M)$

Correctness: $g^{(rx+m)/s} \stackrel{?}{=} r \pmod p \pmod q$

$g^k \stackrel{?}{=} r \pmod p \pmod q$ ✓

Security proof works if we had done $m = h(M||r)$, as before.

As it stands, existentially forgeable for h = identity.

6.857 Rivest
3/11/09 L11.5

Gap groups (DDH easy, CDH hard)

Bilinear groups:

Let G_1 be group of prime order q (multiplicative) gen g

Let G_2 be group of prime order q (") gen h

Suppose we also have (bilinear) map

$$e: G_1 \times G_1 \to G_2$$

s.t.

$$(\forall a,b)\ e(g^a, g^b) = h^{ab} \qquad (!!)$$

$$= e(g, g^{ab}) = e(g,g)^{ab}$$

$$= e(g, g^b)^a = e(g^a, g)^b$$

Then:

DDH is easy in G_1:

given g^a, g^b, g^c

$$c = ab \pmod q \iff e(g^a, g^b) = e(g, g^c)$$

$$h^{ab} \stackrel{?}{=} h^c$$

$$ab = c \pmod q$$

Doesn't make CDH easy, though; we have gap between DDH & CDH

How:

G_1 is elliptic curve (supersingular: $y^2 = x^3 + ax + b \pmod p$)

$$|E(\mathbb{F}_p)| = p + 1 \qquad (p \geq 5)$$

$p \equiv 2 \pmod 3$, $b \in \mathbb{Z}_p^*$, $a = 0$ — supersingular $y^2 = x^3 + b$

or $p \equiv 3 \pmod 4$, $a = 1$, $b = 0$ — $y^2 = x^3 + x$

G_2 is $\mathbb{F}_{p^k}$ some small k.

e: a "Weil pairing" or "Tate pairing" ...

Boneh, Lynn, Shacham (2001)

BLS signature scheme

Public: groups G_1, G_2 of prime order q (multiplicative notation)
pairing function $e: G_1 \times G_1 \to G_2$
g generator of G_1
H: hash fn mapping messages $\to G_1$ (C.R.)

Secret key: $x \quad 0 < x < q$

Public key: $y = g^x \quad (\text{in } G_1)$

To sign message m:

$$\sigma = \sigma_x(m) = (H(m))^x \quad (\text{in } G_1)$$

← short! (e.g. 160 bits)

To verify:

check $e(g, \sigma) \stackrel{?}{=} e(y, H(m))$

$\left(= e(g, H(m))^x \text{ in both cases} \right)$

Thm: BLS secure against existential forgery under chosen-message attack (in ROM) assuming CDH hard in G_1.

Secret Sharing (Threshold cryptography)

Alice has a secret document (or key) s.

She wants to protect it as follows:

she has n friends $A_1, A_2, \ldots, A_n$

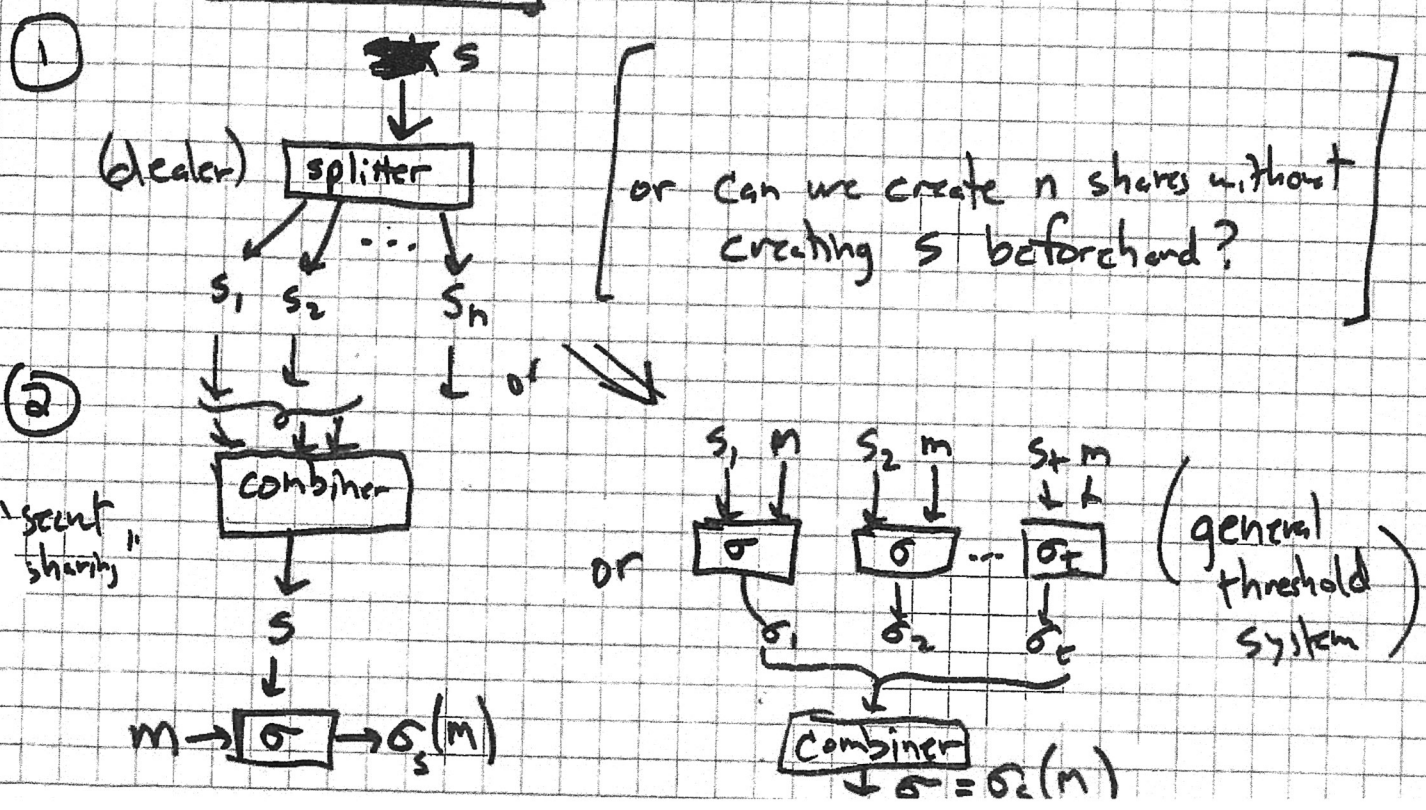
she wants to give each of them a "share" of s.

she picks a "threshold" t, $1 \le t \le n$.

she wants it to be possible that any t or more of her friends can reconstruct (or use) s, while any set of $< t$ friends can not.

[If s is secret key, note distinction between
- reconstructing s (for use)
- getting same effect via shares]

Example: signing:



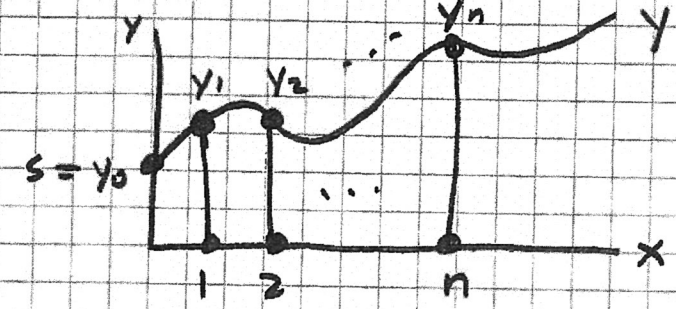
Secret Sharing:

t = 1: $s_1 = s \quad 1 \le i \le n$

t = n: $s_1, \ldots, s_{n-1}$ random

s_n chosen so $s_1 \oplus s_2 \oplus \cdots \oplus s_n = s$

1 < t < n ??:



$y = f(x)$ degree $t-1$

$= a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \cdots + a_1 x + a_0 \pmod p$

t points (x_i, y_i) $1 \le i \le t$ determine a unique polynomial of degree $t-1$.

Given s: let $y_0 = s = a_0$

pick $a_1, a_2, \ldots, a_{t-1}$ at random from Z_p

s_i = pair (i, y_i) : $y_i = f(i)$ $1 \le i \le n$

How to reconstruct??

Evaluation

dual is Interpolation

t pt-value pairs $\xleftrightarrow[\text{Interpolation}]{\text{Evaluation}}$ t coeffs

Interpolation

Given $(x_i, y_i) \quad 1 \le i \le t$

Then $f(x) = \sum_{i=1}^{t} f_i(x) \cdot y_i$

where $f_i(x) = \begin{cases} 1 & \text{at } x = x_i \\ 0 & \text{for } x = x_j,\ j \neq i,\ 1 \le j \le t \end{cases}$

is clearly ok

But $f_i(x) = \dfrac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}$

polynomial of degree $t-1$
$\therefore$ f is poly of degree $t-1$

$(= 0$ or 1 at x_j as appropriate, $x_j \neq x_i$ $x_j = x_i)$

Evaluating $f(0)$ to get $y_0 = s$: simplifies to

$$s = f(0) = \sum_{i=1}^{t} y_i \cdot \frac{\prod_{j \neq i} (-x_j)}{\prod_{j \neq i} (x_i - x_j)}$$

Thm: Secret-sharing is **information-theoretically secure**
Adversary with $< t$ shares has **no** information re s.

Pf: Can create consistent polynomial of degree $t-1$ going through given points at **any** point $(0, s)$ $0 \le s < p$.

Refs: Reed-Solomon codes, erasure codes, error-correction, information dispersal (Rabin)