ElGamal Cryptosystem
and variants

BİL 448/548
Internet Security Protocols
Ali Aydın Selçuk

Structure of $\mathbb{Z}_p^*$

For a prime p, $\mathbb{Z}_p^*$ is all non-zero elements of $\mathbb{Z}_p$.
Fermat's (Little) Theorem: For all $x \in \mathbb{Z}_p^*$, we have

$$x^{p-1} \equiv 1 \pmod p).$$

Let <g> denote the numbers generated by powers of g in $\mathbb{Z}_p^*$; $\langle g\rangle = \{g, g^2, \ldots, g^{p-1}\}$.
E.g. for $\mathbb{Z}_5^*$:

$\langle 1\rangle = \{1\}$ $\langle 2\rangle = \{2,4,3,1\}$
$\langle 3\rangle = \{3,4,2,1\}$ $\langle 4\rangle = \{4,1\}$

- "order" of 1 is one; of 4 is two; of 2 & 3 is four.
- 2 & 3 are "generators" of $\mathbb{Z}_5^*$ (they have order p-1).
- Fact: For every prime p, $\mathbb{Z}_p^*$ has a generator.

ElGamal – Encryption

Parameters:
- p, a large prime
- g, a generator of $\mathbb{Z}_p^*$
- $\alpha \in \mathbb{Z}_{p-1}$, $\beta = g^\alpha \bmod p$
- p, g, β public; α private

Encryption:
- generate random, secret $k \in \mathbb{Z}_{p-1}$.
- E(x, k) = (r, s), where
  $r = g^k \bmod p$
  $s = x\beta^k \bmod p$
- $D(r, s) = s(r^\alpha)^{-1} \bmod p = xg^{\alpha k}g^{-\alpha k} \bmod p = x$.

ElGamal – Encryption

- Plaintext x is masked by a random factor, $g^{\alpha k} \bmod p$.
- DH problem: Given g^α, $g^k \bmod p$, what is $g^{\alpha k} \bmod p$?
- p, g can be common. Then $g^k \bmod p$ can be computed in advance.
- Same k should not be used repeatedly.
- Performance:
  - encryption: two exponentiations
  - decryption: one exponentiation, one inversion
- Size: Ciphertext twice as large as plaintext.

ElGamal – Signature

Parameters: The same as encryption.

Signature:

– generate random, secret $k \in \mathbb{Z}_{p-1}^*$.

– $S(m, k) = (r, s)$, where

$r = g^k \bmod p$

$s = (m - r\alpha)k^{-1} \bmod (p - 1)$

(i.e., $m = r\alpha + sk$)

Verification:

– Is $\beta^r r^s \equiv g^m \pmod p$?

– $\beta^r r^s = g^{\alpha r} g^{k(m - r\alpha)k^{(-1)}} = g^{\alpha r + (m - r\alpha)} = g^m \bmod p$.

ElGamal – Signature

Security:

– Only one who knows α can sign; can be verified by β.

– Solving α from β, or s from r, m, β, is discrete log.

– Other ways of forgery? Unknown.

– Same k should not be used repeatedly.

Variations:

– Many variants, by changing the “signing equation”,

$m = r\alpha + sk$.

– E.g., the DSA way:

$m = -r\alpha + sk$

with verification: $\beta^r g^m \equiv r^s \pmod p$? ($\equiv g^{m + r\alpha}$)

Digital Signature Algorithm (DSA)

- US government standard, designed by NSA.
- Based on ElGamal & Schnorr:
  - patent-free (ElGamal)
  - can’t be used for encryption
- Objections:
  - ElGamal was not analyzed as much as RSA
  - slower verification
  - industry had already invested in RSA
  - closed-door design

DSA (cont’d)

- Let $q \mid (p-1)$ be prime, and $g \in \mathbb{Z}_p^*$ be of order q.
- Schnorr group: The subgroup in $\mathbb{Z}_p^*$ generated by g, of prime order q.

$$\langle g \rangle = \{1, g, g^2, \ldots, g^{q-1}\}$$

- **Fact:** q can be much shorter than p (e.g. 160 vs. 1024 bits), and the hardness of DLP in <g> remains the same.

DSA (cont'd)

Parameters: prime p, prime q | (p-1), and $g \in \mathbb{Z}_p^*$ of order q. Hash fnc. H: $\{0,1\}^* \to \mathbb{Z}_q$.

Keys: $\alpha \in \mathbb{Z}_q$ is private; $\beta = (g^\alpha \bmod p)$ is public.

Signature: (r,s) where

- $v = g^k \bmod p$
- $r = v \bmod q$
- $s = (H(M) + r\,\alpha)\,k^{-1} \bmod q$

Verification:

- $v' = g^{H(M)\, s\hat{}(-1)}\, \beta^{r\, s\hat{}(-1)} \bmod p$
- r = v' mod q ?

Advantage: Reduced size (r, s are 160-bit)

Elliptic Curve Cryptosystems

Generalized Discrete Log Problem:

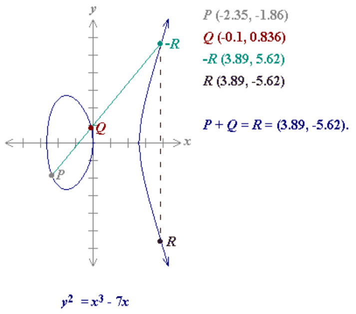
- For any group (G, •), for $x \in G$, define
  $x^n = x \bullet x \bullet \ldots \bullet x$ (n times)
- DLP: For $y = x^n$, given x, y, what is n?

Elliptic curves over $\mathbb{Z}_p$:

- Set of points $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ that satisfy
  $y^2 \equiv x^3 + ax + b \pmod p$
  and an additional point of infinity, 0.
- Group operation: P•Q is the inverse of where the line thru P & Q intersects the curve. (inverse of P = (x, y) is defined as $P^{-1} = (x, -y)$.)
- Well-defined, provided that $4a^3 \neq -27b^2 \pmod p$.

Elliptic Curve Cryptosystems (cont'd)

EC example over R^2:



Elliptic Curve Cryptosystems (cont'd)

- Facts for an EC over a finite field:
  - Exponentiation is efficient.
  - DLP is hard. In fact, harder than in $\mathbb{Z}_p$. (no sub-exponential algorithm is known)
- Hence, DH, ElGamal, etc. can be used with smaller key sizes over ECs. (160-bit EC ~ 1024-bit RSA)
- Popular for constrained devices (e.g., smart cards)
- Advantages over RSA:
  - smaller key size
  - compact in hardware
  - faster (for private key operations)
- Licensed by NSA.

ECC vs. RSA

NIST guidelines for key sizes (bits) with eqv. security levels:
http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf

Symmetric Key	RSA/DH/ElGamal	ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

(according to our current knowledge of algorithms for factorization, DLP, and EC DLP)