El Gamal. (Convert DH to public key encryption)

- Cyclic group G of order n (e.g. $G = (\mathbb{Z}_p)^*$)
- Fix a generator in G (e.g. $G = \{1, g, g^2, g^3, \ldots, g^{n-1}\}$)

Alice

choose random a in $\{1, \ldots, n\}$

Bob

choose random b in $\{1, \ldots, n\}$

Assumption: given g^a, hard to find a

$A = g^a$ — broadcast public key →

← $B = g^b$

Bob: compute $g^{ab} = A^b$; derive sym key k; encrypt m with k

$ct = [B = g^b, \text{encrypt } m \text{ with } k]$

- To decrypt compute $g^{ab} = B^a$, derive k, and decrypt.

El Gamal System

- G finite cyclic group of order n
- $(\mathcal{E}_s, \mathcal{D}_s)$: sym. AE. scheme over (k, M, c)
- $H: G^2 \to K$ — a hash function.

$\mathcal{E}(pk = (g, h), m)$:

$b \xleftarrow{R} \mathbb{Z}_n$, $u \leftarrow g^b$,

$v \leftarrow h^b$, $k \leftarrow H(u, v)$

$c \leftarrow \mathcal{E}_s(k, m)$

output (u, c)

$\mathcal{D}(sk = a, (u, c))$:

$v \leftarrow u^a$

$k \leftarrow H(u, v)$

$m \leftarrow \mathcal{D}_s(k, c)$

output m

$\mathcal{E}$

$h^b = g^{ab}$

$\mathcal{D}$:

$u^a = (g^b)^a = g^{ab}$

$k = H(g^b, g^{ab})$

- Can precompute $[g^{2^i}, h^{2^i}$ for $i = 1 \ldots \log_2 n]$

Lecture 2

Exponentiation.

- G - finite cyclic group (e.g., $G = \mathbb{Z}_p^*$)
- Goal: given g in G and x, compute g^x

Example. Suppose $x = 53_{10} = 110101_2 = 32 + 16 + 4 + 1$

Then, $g^{53} = g^{32+16+4+1} = g^{32} \cdot g^{16} \cdot g^4 \cdot g^1$

$$g \to g^2 \to g^4 \to g^8 \to g^{16} \to g^{32} \quad = g^{53}$$

- Repeated squaring: to compute g^{53}, compute only g, g^2, g^4, g^8, and g^{32}; ignore g^3 and g^5 => a lot faster than multiplying g 53 times

Repeated Squaring Algorithm.

- Input: g in G; $x > 0$
- Output: g^x
- Algorithm: write $x_{10} = (x_n x_{n-1} \dots x_1 x_0)_2$ (decimal, binary)

```
y ← g, z ← 1
for i = 0 to n:
    if (x[i] == 1), then z ← z·y
    y ← y^2
output z
```

- Example: g^{53}

y	z
g^2	g
g^4	g
g^8	g^5
g^{16}	g^5
g^{32}	g^{21}
g^{64}	g^{53}

- Every time we compute g^x, we can reuse it later, i.e., precompute

Kilroy

Lecture 3

Computational Diffie-Hellman (CDH)

- G: finite cyclic group of order n
- CDH assumption holds in G if: $g, g^a, g^b \Rightarrow g^{ab}$

$\Rightarrow$ i.e., if the Adv. knows g, g^a, g^b he cannot compute g^{ab}.

- For all eff. algorithms A:

$$Pr[A(g, g^a, g^b) = g^{ab}] < negl$$

where $g \leftarrow$ {generators of G}

$a, b \leftarrow Z_n$

Hash Diffie-Hellman.

- G; $H: G^2 \to k$

Def. Hash-DH (HDH) assumption holds for (G, H) if

$$(g, g^a, g^b, H(g^b, g^{ab})) \approx_p (g, g^a, g^b, R)$$

$g \leftarrow$ {generators of G}, $a, b \leftarrow Z_n$; $R \leftarrow k$

- H acts as extractor: distribution of $G^2 \Rightarrow$ uniform dist. on k
- HDH $\to$ CDH; if CDH is easy, so is HDH because g^{ab} can be solved.

Example. Suppose $k = \{0,1\}^{128}$ ($\forall x, y$ $msb(H(x,y)) = 0$)

$H: G^2 \to k$ only outputs strings in k which begin with 0.

Q: Can HDH hold for (G, H)?

↳ No, HDH is easy to break. If it starts with 1, it is in R.

Lecture 3

El Gamal CCA-Security.

- Security theorem: If Interactive-DH (IDH) holds in G, (E_s, D_s) provides auth. enc. and $H: G^2 \to K$ is a "random oracle", then El Gamal is CCA^{ro} secure.

- To prove CCA security based on Computational-DH (CDH), i.e. $(g, g^a, g^b \to g^{ab})$.
  1) use group G where CDH = IDH (bilinear group)
  2) change the El Gamal system.

Twin El Gamal

- $g \leftarrow$ Egens of G ; $a_1, a_2 \leftarrow Z_n$ • Now pair of keys instead of 1
- Output $pk = (g, h_1 = g^{a_1}, h_2 = g^{a_2})$
  $sk = (a_1, a_2)$

$E(pk = (g, h_1, h_2), m)$: $b \leftarrow Z_n$
- $k \leftarrow H(g^b, h_1^b, h_2^b)$
- $c \leftarrow E_s(k, m)$
- output (g^b, c)

$D(sk = (a_1, a_2), (u, c))$:
- $k \leftarrow H(u, u^{a_1}, u^{a_2})$
- $m \leftarrow D_s(k, c)$
- output m

Security theorem: If CDH holds in G, (E_s, D_s) provides auth enc, and $H: G^3 \to K$ is a "random oracle", then twin El Gamal is CCA^{ro} secure.

- Without random oracles:
  1) IDH with bilinear groups
  2) CDH with any group

Kilroy