CS 578 – Cryptography

Prof. Michael Backes

Active Attacks against Public-key Encryption, The Cramer-Shoup Encryption Scheme

June 6, 2006

Administrative Announcements

- Handouts:
  - New exercise sheet
- Final exam:
  - Stays on Friday, July 21
  - On 1-3pm due to room availability

Recall: Public-key Encryption

- Definition (Public-Key Encryption Scheme): A public-key encryption scheme is a triple of efficient algorithms (Gen,E,D):
  - Gen(n): Generates a secret/public key pair (pk,sk) for security parameter n
  - E(pk,m) and D(sk,m) as usual

  such that for all (pk,sk) ← Gen(k), and for all m: D(sk,E(pk,m)) = m
- n is the security parameter, tacitly considered input to all algorithms (formally again sequences of encryption schemes and (uniform) adversaries, but much more natural for public-key encryption).

Recall: ElGamal for Subgroups

- ElGamal in subgroup G_q of Z^*_p
- Technical subtlety: Now two security parameters: n for q, n* for p. Related by public function n_p: $n^* = n_p(n)$
- Generation in G_q, for security parameter n and function n_p):
  - Pick random n-bit prime q
  - Pick random $n_p(n)$-bit prime p such that q | p-1
  - Pick $g \in Z^*_p$ of order q (sometimes public q, p, g)
  - Pick random $x \in \{1,\ldots,q\}$
  - Set pk := (q, p, g, h := g^x)
  - Set sk := (q, p, g, x)
  - Output (pk,sk)

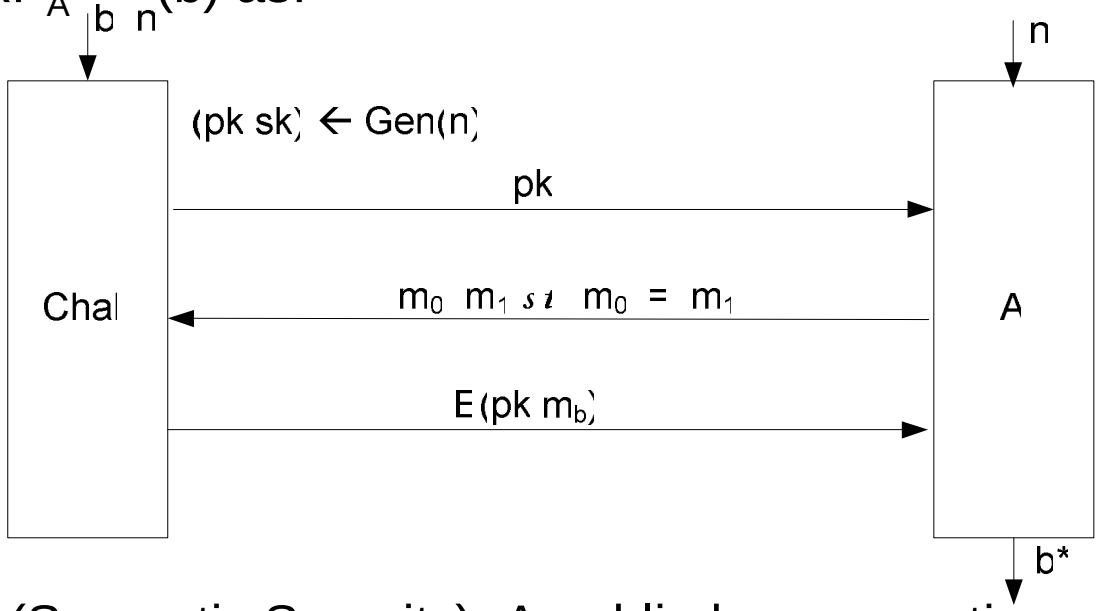
Recall: ElGamal for Subgroups

- Encryption Enc(pk,m) where
  pk = (q, p, g, h = g^x) and m $\in$ <g> =G_q
  - Pick random y $\in$ {1,…,q}
  - Set i := g^y , k := h^y
  - Output c := (i, k $\cdot$ m) $\in$ $G_q \times G_q$

- Decryption Dec(sk,c)
  where sk = (q, p, g, x) and c = (A,B)
  - Output B / A^x

Recall: Semantic Security (CPA)

- Let PE = (Gen, E, D) be a public-key encryption scheme. Define $EXP_A^{CPA}(b)$ as:



- Definition (Semantic Security). A public-key encryption scheme PE = (Gen, E, D) is semantically secure under chosen-plaintext attack (CPA) if for all efficient adversaries A, we have that $Adv^{CPA}[A,PE] = |Pr[EXP_A^{CPA}(0)=1] – Pr[EXP_A^{CPA}(1)=1]|$ is negligible.

CPA-Security of ElGamal

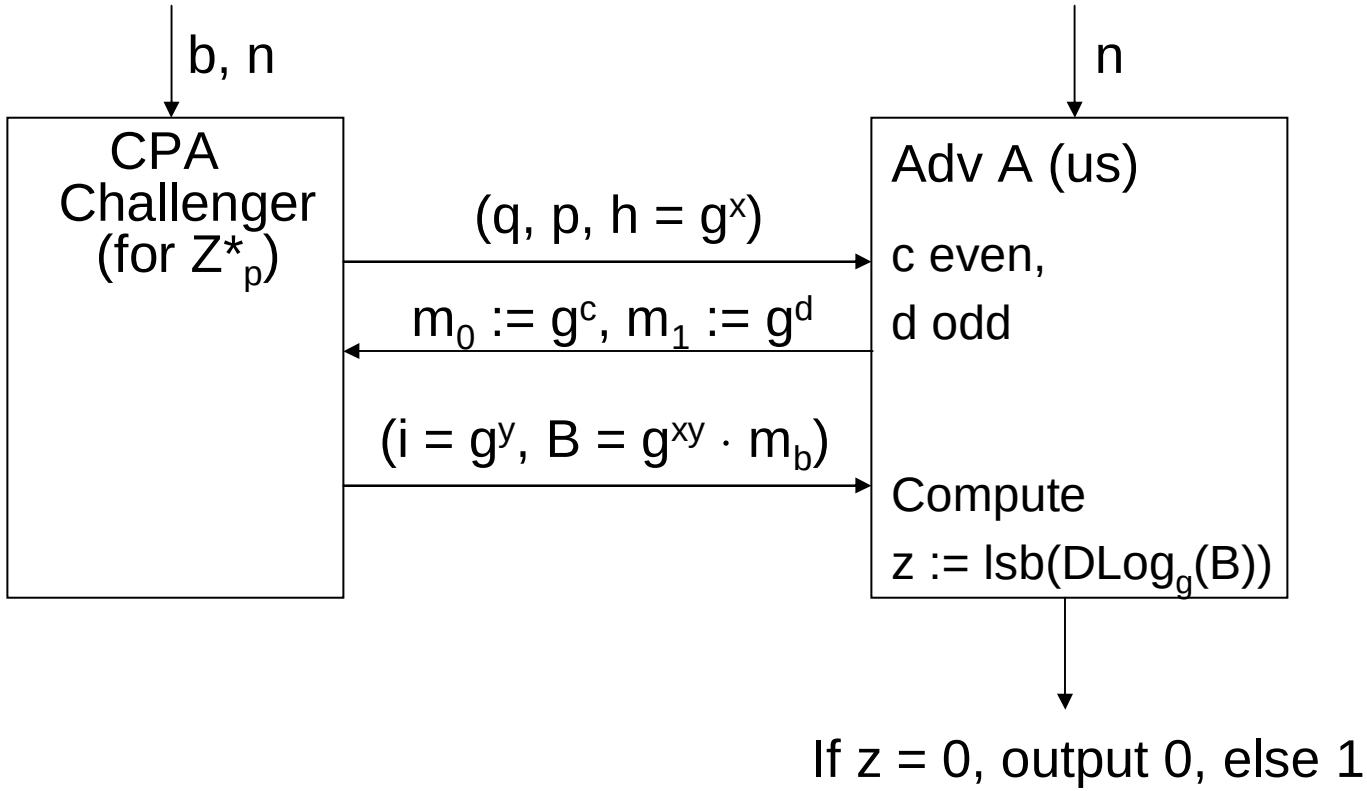
- ElGamal is semantically secure under CPA (in a subgroup G_q) if the Decisional Diffie-Hellman assumption in G_q holds:
- Decisional Diffie-Hellman Assumption (DDH): Given n-bit prime q, $n_p(n)$-bit prime p with q | p-1, and $g \in Z^*_p$ of order q, no efficient adversary (in n) can distinguish
  (g^x, g^y, g^{xy}) and (g^x, g^y, g^z)
  for x, y, z random in {1,…,q}.
- DDH not hard in Z^*_p

DDH as an Attack Game

- Definition (DDH Challenger, on input a security parameter n):
  - Challenger randomly chooses an n-bit prime q, an $n_p(n)$-bit prime p with q | p-1, and $g \in Z^*_p$ of order q and outputs (q, p, g)
  - Challenger chooses a bit b
  - Challenger chooses x, y, $z \in \{1,\ldots,q\}$ random and outputs
    - (g^x, g^y, g^{xy}) if b = 0 and
    - (g^x, g^y, g^z) if b = 1.
- The adversary outputs b* and wins if b* = b.

ElGamal in Z^*_p not CPA-secure

- ElGamal in Z^*_p not even CPA-secure:



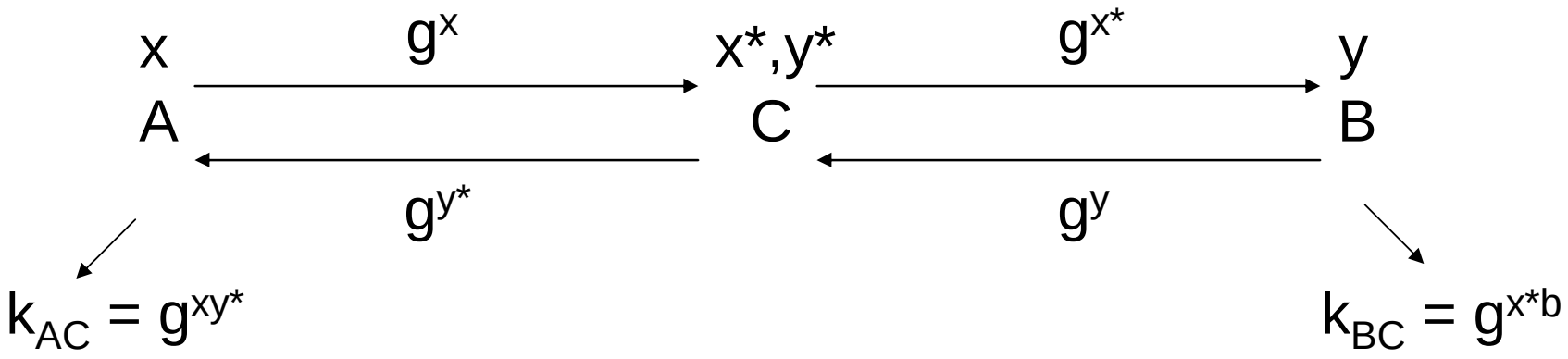
Using ElGamal Encryption for email (PGP)

- Bob wants to send mail m to Alice
- Obtain Alice's ElGamal public-key
  pk = (q,p,g,h)
- Pick a random message key $K \in G_q$
- Derive from K: AES key + IV, MAC key
- Set c ← AuthSymmEnc(K,m)
- Send ElGamal(pk,K) || c

≈ 2048 bits long

Necessity of Authentic Key Transmission

- Diffie-Hellman (and public-key encryption schemes in general) require authentic transmission of pk
- Else man-in-the-middle attack (malicious active adversary): Assume even that q, p, g public and untampered



Necessity of Authentic Key Transmission

- C now plays relay:
  Intercepts all messages, decrypts and re-encrypts them and forwards them to the intended participant.
- A and B are oblivious, but C sees all messages in clear
- Solution: Include authentication in key-exchange messages

Diffie-Hellman and ElGamal in Practice

- Group order p approx. 1024 bits or 2048 bits
- Size of the primes should be chosen to match the security of the symmetric key system
- NIST recommendation for Z^*_p (most people ignore this)

Symmetric key length	Public key length
64 bits	1024 bit prime
128 bits	4096 bit prime
256 bits	16384 bit prime

Diffie-Hellman and ElGamal in Practice

- Work in “small” subgroup G_q of Z^*_p leads to efficiency gains:
- In practice often $q \approx 2^{256} << 2^{1024}$ (i.e., twice the private key (AES) size)

A

$x \leftarrow_R \{1,\ldots,q\}$

B

$y \leftarrow_R \{1,\ldots,q\}$

$h = g^x \in Z^*_p$ →

← $i = g^y \in Z^*_p$

DH, ElGamal in Practice

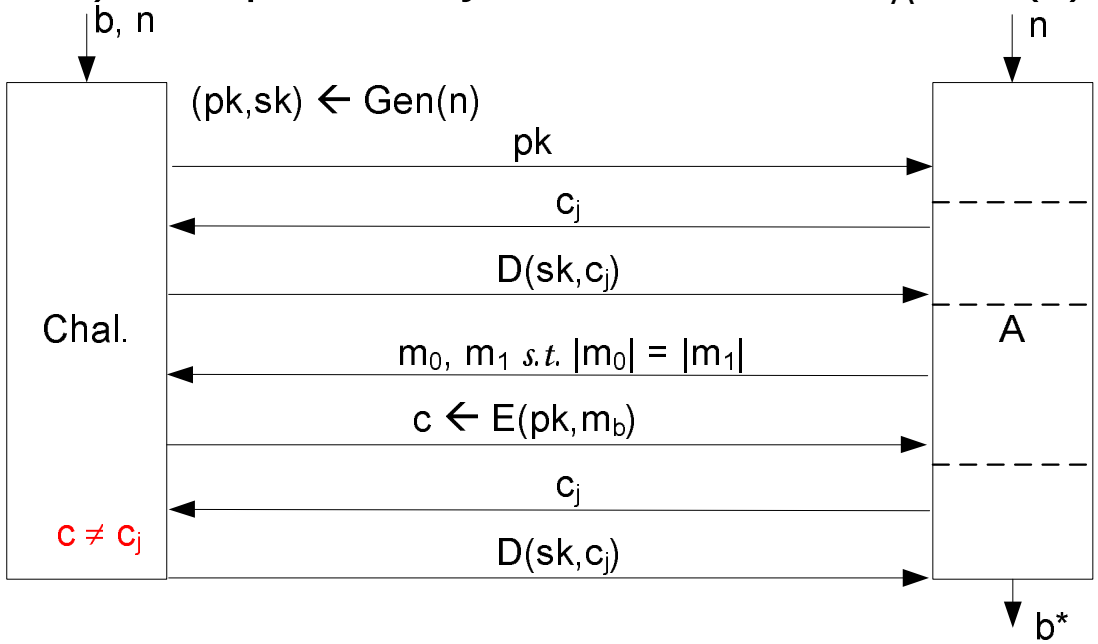
- Reason: Exponents x and y << p
- Faster Exponentiation:
  If p = 1024 bits and q = 256 bits
  → solid speedup for DH+ElGamal
- Best algorithm for solving CDH/DDH in subgroup of order q: $O(q^{1/2})$

Defining Security against Active Attacks

- Assume adversary is allowed to additionally let itself decrypt certain ciphertexts c in a CPA-game
- Called chosen-ciphertext attack (CCA2); *the* standard definition of public-key encryption security in research.
- Motivated essentially by what higher-level protocol have to be guaranteed to work

Definition of CCA2

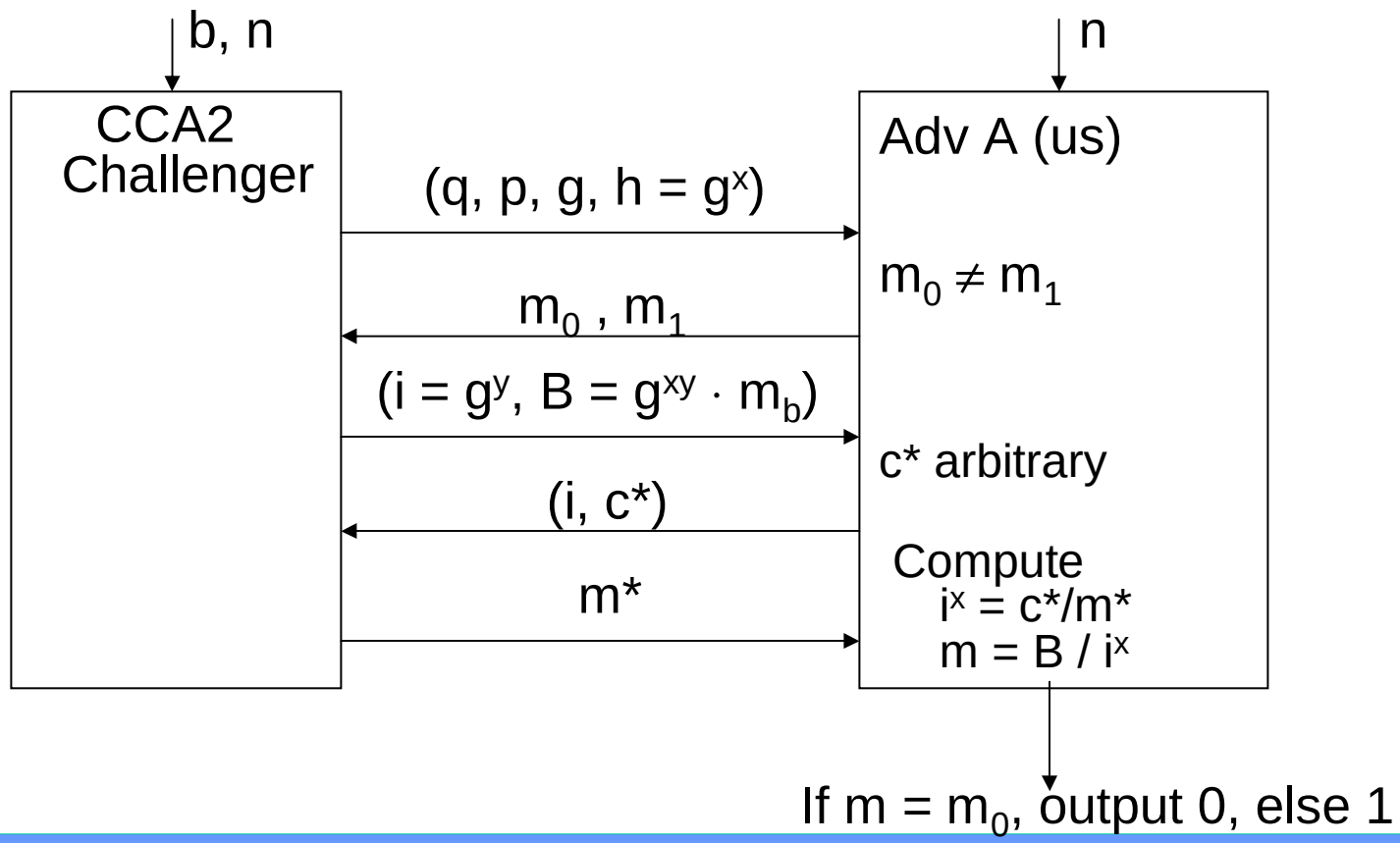
- Let (Gen, E, D) be a public-key enc. Define EXP_A^{CCA2} (b) as:



- Definition (Semantic Security against CCA2). PE = (Gen, E, D) is semantically secure under chosen-ciphertext attack (CCA2) if for all efficient adversaries A, the following is negligible:
$Adv^{CCA2}[A,PE] = |Pr[EXP_A^{CCA2}(0)=1] – Pr[EXP_A^{CCA2}(1)=1]|$.

ElGamal and CCA2-Security

- ElGamal not CCA2-secure

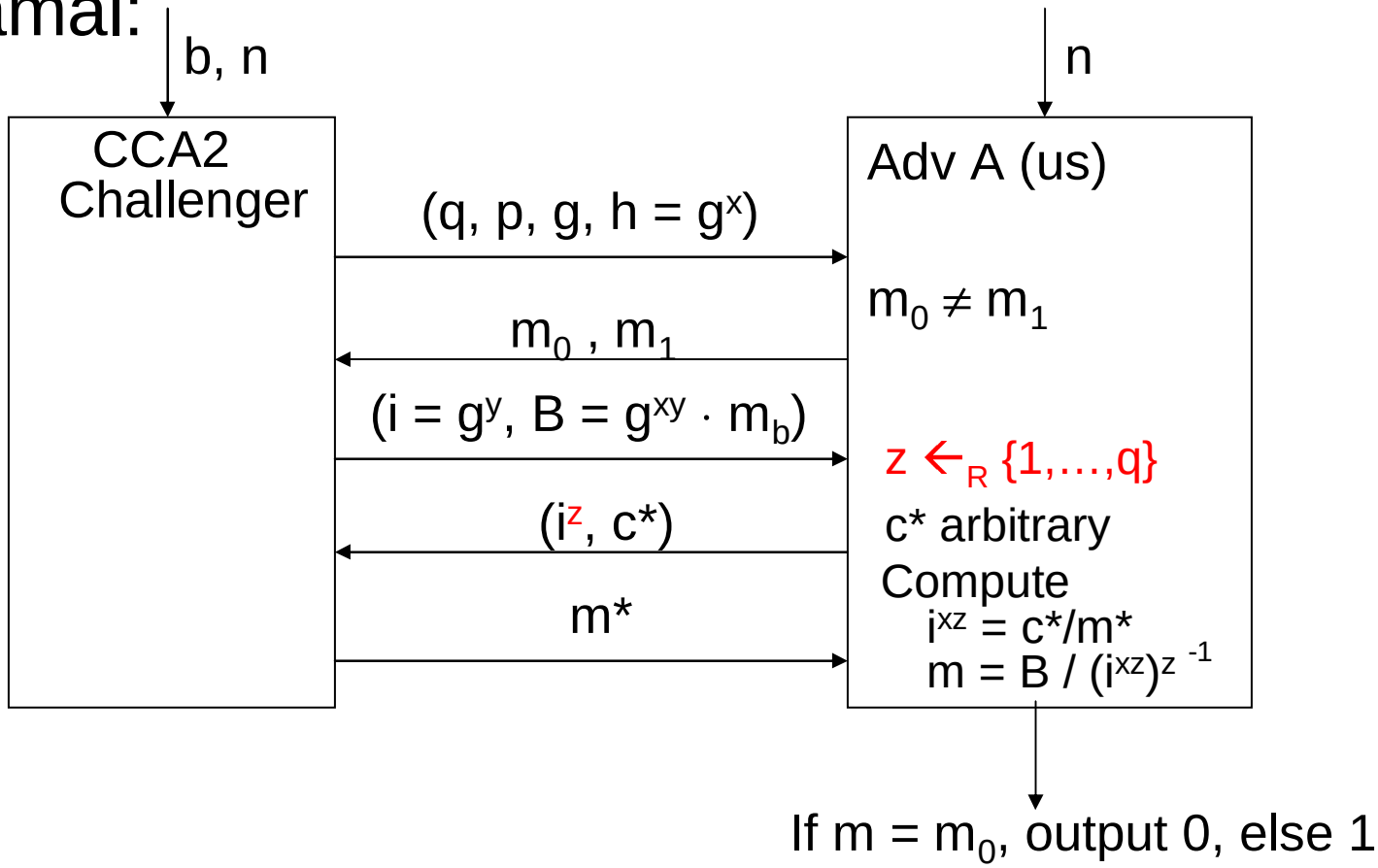


ElGamal and CCA2-Security

- ElGamal not CCA2-secure
- CCA2 attack retrieves the key g^{xy} afterwards!
- Natural fix that comes to mind:
  - Record values i in ciphertexts (i,B) and never answer decryption requests for (i,B*)
  - (in practice: store your received ciphertexts and check for replay attacks)

ElGamal and CCA2-Security

- Still not enough: blinding attack against ElGamal:



ElGamal and CCA2-Security

- Still not enough: blinding attack against ElGamal:
- Underlying reason: “Misformed” ciphertexts cannot be identified and thus cannot be sorted out.
- (Ambitious) goal now: Find a public-key encryption scheme that can be proven CCA2-secure

→ Cramer-Shoup encryption scheme

- CCA2-secure provided that DDH is hard in subgroups G_q of Z^*_p
- Constitutes sophisticated extension of ElGamal

Keyed Hash Functions

- Let *Hash* = $(H(\mathrm{pk}, \cdot))_{\mathrm{pk} \in [\mathrm{Gen}(n)]}$ be a keyed family of hash functions, i.e., $H(\mathrm{pk}, \cdot): \mathcal{M}_{\mathrm{pk}} \rightarrow \mathcal{T}_{\mathrm{pk}}$ (usually $\{0,1\}^* \rightarrow \{0,1\}^{l(n)}$)

- Definition (Collision-resistance for family of (keyed) hash functions): A family *Hash* of keyed hash functions is collision-resistant if for all efficient adversaries A (in security parameter n), we have that

  $\Pr[\, H(\mathrm{pk},m) = H(\mathrm{pk},m') \wedge m \neq m' \,;\, \mathrm{pk} \leftarrow \mathrm{Gen}(n), \; (m,m') \leftarrow A(n,\mathrm{pk}) \,]$

  is negligible (in n).

Keyed Hash Functions (cont'd)

- Definition (One-wayness for family of (keyed) hash functions): A family *H* of keyed hash functions is one-way if for all efficient adversaries A (in security parameter n), we have that

  $$\Pr[\, H(pk,m') = t;\ pk \leftarrow Gen(n),\ m \leftarrow_R \mathcal{M}_{pk},\ t := H(pk,m)\ m' \leftarrow A(n,pk,t)\,]$$

  is negligible (in n).
- Lemma: A collision-resistant family of hash functions is one-way if additionally it holds that every possible digest has ≥ 2 pre-images for all functions $H(pk, \cdot)$.

The Cramer-Shoup Encryption Scheme

- Key generation for security parameter n:
  - Pick random n-bit prime q
  - Pick random $n_p(n)$-bit prime p such that q | p-1
  - Pick $g_1 \in Z^*_p$ of order q and second generator g_2 of $\langle g_1 \rangle$ randomly
  - Pick random $x_1, x_2, y_1, y_2, z \in \{1,\ldots,q\}$
  - Set

$$s = g_1^{x_1} \cdot g_2^{x_2} \qquad t = g_1^{y_1} \cdot g_2^{y_2} \qquad h = g_1^{z}$$

  - Let $pk_{hash} \leftarrow Gen_{Hash}(n)$ (Denote $H(\cdot) := H(pk_{hash}, \cdot)$)
  - Set $pk := (q, p, g_1, g_2, s, t, h, pk_{hash})$
  - Set $sk := (pk, x_1, x_2, y_1, y_2, z)$

The Cramer-Shoup Encryption Scheme

- Key generation for security parameter n:
  - Pick random n-bit prime q
  - Pick random $n_p(n)$-bit prime p such that q | p-1
  - Pick $g_1 \in Z^*_p$ of order q and second generator g_2 of $\langle g_1 \rangle$ randomly
  - Pick random $x_1, x_2, y_1, y_2, z \in \{1,\ldots,q\}$
  - Set

$$s = g_1^{x_1} \cdot g_2^{x_2} \qquad t = g_1^{y_1} \cdot g_2^{y_2} \qquad h = g_1^{z}$$

  - Let $pk_{hash} \leftarrow Gen_{Hash}(n)$ (Denote $H(\cdot) := H(pk_{hash}, \cdot)$)
  - Set $pk := (q, p, g_1, g_2, s, t, h, pk_{hash})$
  - Set $sk := (pk, x_1, x_2, y_1, y_2, z)$

The Cramer-Shoup Encryption Scheme

- Encryption Enc(pk,m) where m $\in$ $<g_1> = G_q$ and pk = $(q, p, g_1, g_2, s, t, h, pk_{hash})$
  - Pick r randomly from {1,…,q}
  - Set

$$i_1 = g_1^r \qquad c^* = h^r \cdot m$$

  - In addition, set

$$i_2 = g_2^r \qquad \alpha = H(i_1, i_2, c^*) \qquad v = s^r \cdot t^{r\alpha}$$

- The ciphertext is

$$c = (i_1, i_2, c^*, v)$$

The Cramer-Shoup Encryption Scheme

- Encryption Enc(pk,m) where m $\in$ $<g_1> = G_q$ and $pk = (q, p, g_1, g_2, s, t, h, pk_{hash})$
  - Pick r randomly from {1,…,q}
  - Set

$$i_1 = g_1^r \qquad c^* = h^r \cdot m$$

  - In addition, set

$$i_2 = g_2^r \qquad \alpha = H(i_1, i_2, c^*) \qquad v = s^r \cdot t^{r\alpha}$$

- The ciphertext is

$$c = (i_1, i_2, c^*, v)$$

The Cramer-Shoup Encryption Scheme

- Decryption Dec(sk,c) where c = (i_1, i_2, c*, v) and sk = (pk, x_1, x_2, y_1, y_2, z)
  - Compute

$$\alpha = H(i_1, i_2, c^*)$$

  - Verify if the following holds. If not abort.

$$i_1^{x_1+y_1\alpha} \cdot i_2^{x_2+y_2\alpha} = v$$

  - If verification is true, compute

$$k = i_1^z \qquad m = \frac{c^*}{k}$$

The Cramer-Shoup Encryption Scheme

- Decryption Dec(sk,c) where c = (i_1, i_2, c*, v) and sk = (pk, x_1, x_2, y_1, y_2, z)
  - Compute

$$\alpha = H(i_1, i_2, c^*)$$

  - Verify if the following holds. If not abort.

$$i_1^{x_1+y_1\alpha} \cdot i_2^{x_2+y_2\alpha} = v$$

  - If verification is true, compute

$$k = i_1^z \qquad m = \frac{c^*}{k}$$

The Cramer-Shoup Encryption Scheme

- Correctness of decryption
  [proof on the board]

- Sketch: resistant against simple ElGamal-attack
- Intuition on why the test (using v) rejects "misformed" ciphertexts
  - If $i_2 = g_2^r$ then v necessarily of the correct form
  - If $i_2 \neq g_2^r$ then some suitable value

$$v := i_1^{x_1 + y_1 \alpha} \cdot i_2^{x_2 + y_2 \alpha}$$

exists, but we will show that an attacker not knowing the secret key can only find such a v with negligible probability.

High-level Overview of the Reduction