

# CS 578 – Cryptography

Prof. Michael Backes

---

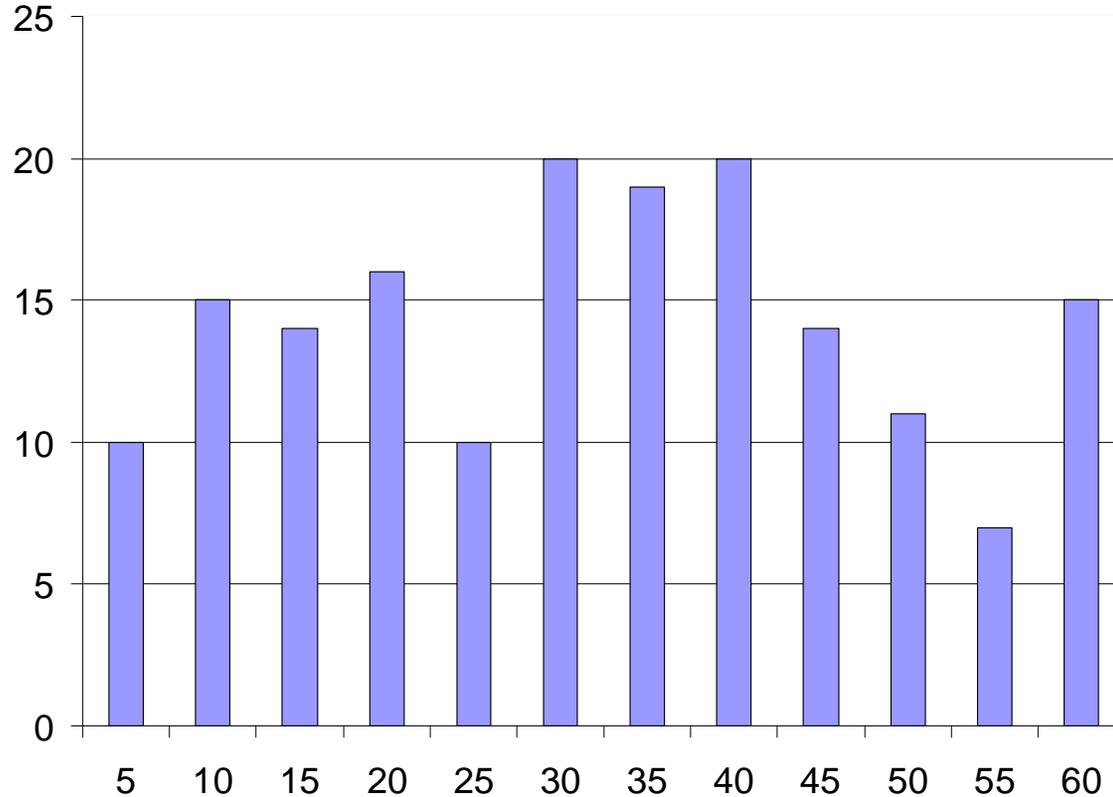
## The Cramer-Shoup Encryption Scheme - Security Proof -

---

June 9, 2006

# Administrative Announcements

- Time for investigating your exam:
  - Friday, June 16, 1.30 – 4.30



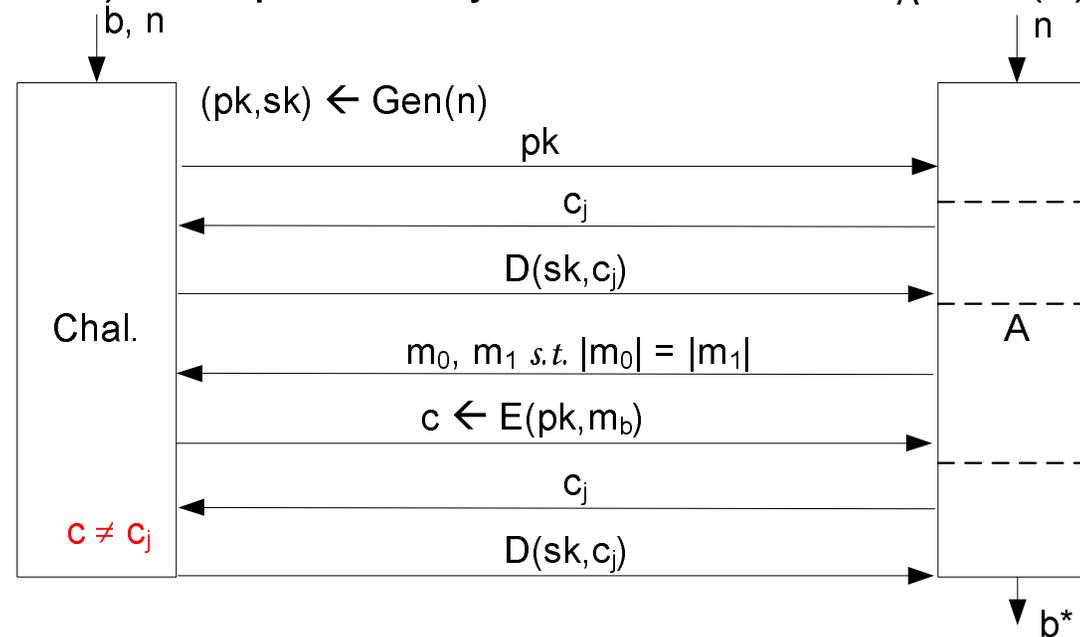
# Recall: Public-key Encryption

---

- Definition (**Public-Key Encryption Scheme**): A public-key encryption scheme is a triple of efficient algorithms  $(\text{Gen}, E, D)$ :
  - $\text{Gen}(n)$ : Generates a secret/public key pair  $(pk, sk)$  for security parameter  $n$
  - $E(pk, m)$  and  $D(sk, m)$  as usualsuch that for all  $(pk, sk) \leftarrow \text{Gen}(k)$ , 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).

# Definition of CCA2

- Let  $(\text{Gen}, E, D)$  be a public-key enc. Define  $\text{EXP}_A^{\text{CCA2}}(b)$  as:



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

# Keyed Hash Functions

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

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

is negligible (in  $n$ ).

# 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  $q \mid p-1$
  - Pick  $g_1 \in \mathbb{Z}_p^*$  of order  $q$  and second generator  $g_2$  of  $\langle g_1 \rangle$  randomly
  - Pick random  $x_1, x_2, y_1, y_2, z \in \{1, \dots, q\}$
  - Set
$$s = g_1^{x_1} \cdot g_2^{x_2} \quad t = g_1^{y_1} \cdot g_2^{y_2} \quad h = g_1^z$$
  - Let  $pk_{\text{hash}} \leftarrow \text{Gen}_{\text{Hash}}(n)$  ( Denote  $H(\cdot) := H(pk_{\text{hash}}, \cdot)$  )
  - Set  $pk := (q, p, g_1, g_2, s, t, h, pk_{\text{hash}})$
  - 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  $q \mid p-1$
  - Pick  $g_1 \in \mathbb{Z}_p^*$  of order  $q$  and second generator  $g_2$  of  $\langle g_1 \rangle$  randomly
  - Pick random  $x_1, x_2, y_1, y_2, z \in \{1, \dots, q\}$
  - Set
 
$$s = g_1^{x_1} \cdot g_2^{x_2} \quad t = g_1^{y_1} \cdot g_2^{y_2} \quad h = g_1^z$$
  - Let  $pk_{\text{hash}} \leftarrow \text{Gen}_{\text{Hash}}(n)$  ( Denote  $H(\cdot) := H(pk_{\text{hash}}, \cdot)$  )
  - Set  $pk := (q, p, g_1, g_2, s, t, h, pk_{\text{hash}})$
  - Set  $sk := (pk, x_1, x_2, y_1, y_2, z)$

# The Cramer-Shoup Encryption Scheme

---

- Encryption  $\text{Enc}(\text{pk}, m)$  where  $m \in \langle g_1 \rangle = G_q$  and  $\text{pk} = (q, p, g_1, g_2, s, t, h, \text{pk}_{\text{hash}})$ 
  - Pick  $r$  randomly from  $\{1, \dots, q\}$
  - Set

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

- In addition, set

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

- The ciphertext is

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

# The Cramer-Shoup Encryption Scheme

- Encryption  $\text{Enc}(\text{pk}, m)$  where  $m \in \langle g_1 \rangle = G_q$  and  $\text{pk} = (q, p, g_1, g_2, s, t, h, \text{pk}_{\text{hash}})$

- Pick  $r$  randomly from  $\{1, \dots, q\}$
- Set

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

- In addition, set

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

- The ciphertext is

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

# The Cramer-Shoup Encryption Scheme

---

- Decryption  $\text{Dec}(\text{sk}, c)$  where  $c = (i_1, i_2, c^*, v)$  and  $\text{sk} = (\text{pk}, x_1, x_2, y_1, y_2, z)$

- 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

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

# The Cramer-Shoup Encryption Scheme

---

- Decryption  $\text{Dec}(\text{sk}, c)$  where  $c = (i_1, i_2, c^*, v)$  and  $\text{sk} = (\text{pk}, x_1, x_2, y_1, y_2, z)$

- 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**

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

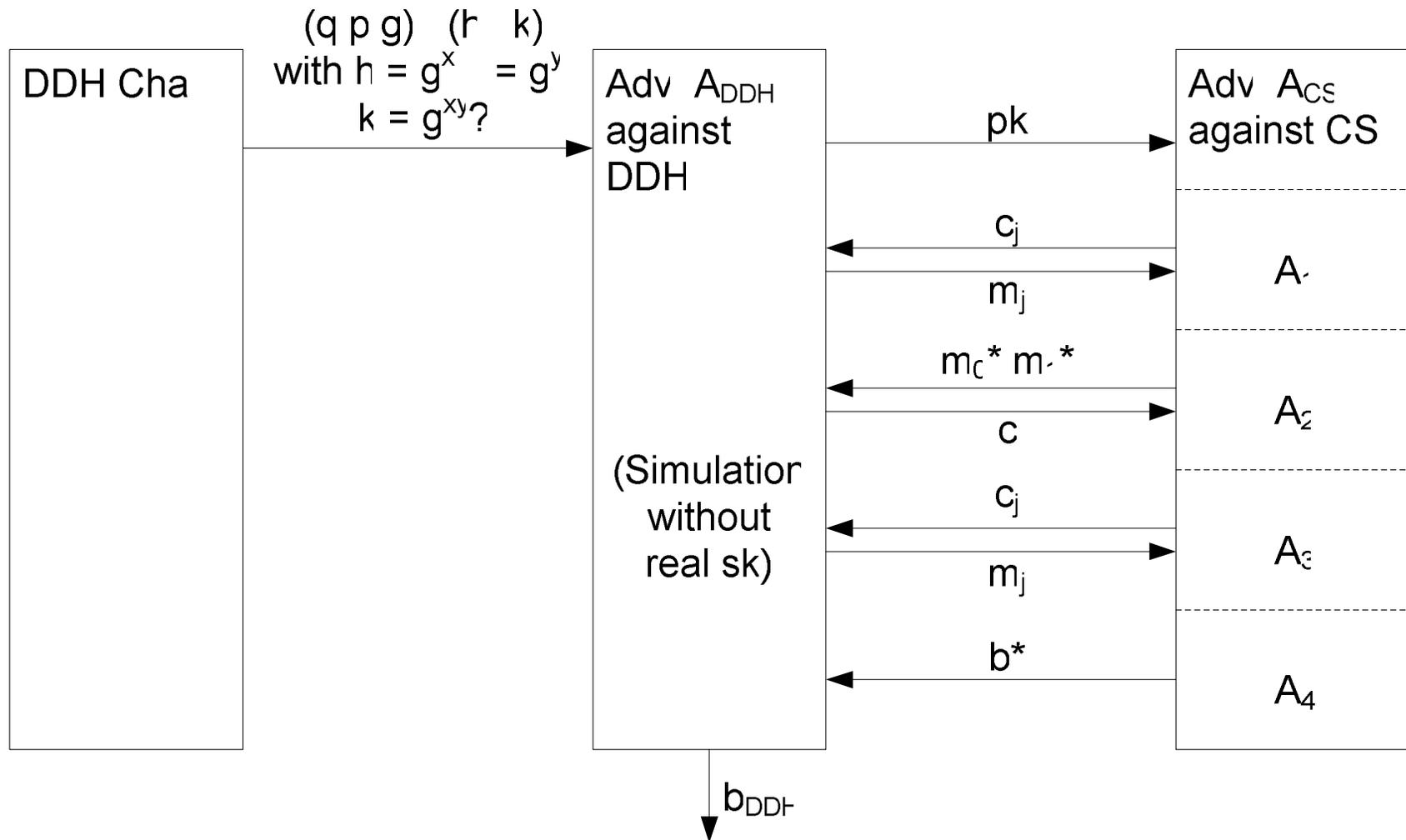
# The Cramer-Shoup Encryption Scheme

---

- Correctness of decryption, resistance against naïve ElGamal attacks shown last time
- 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



# Intuition on the Reduction

---

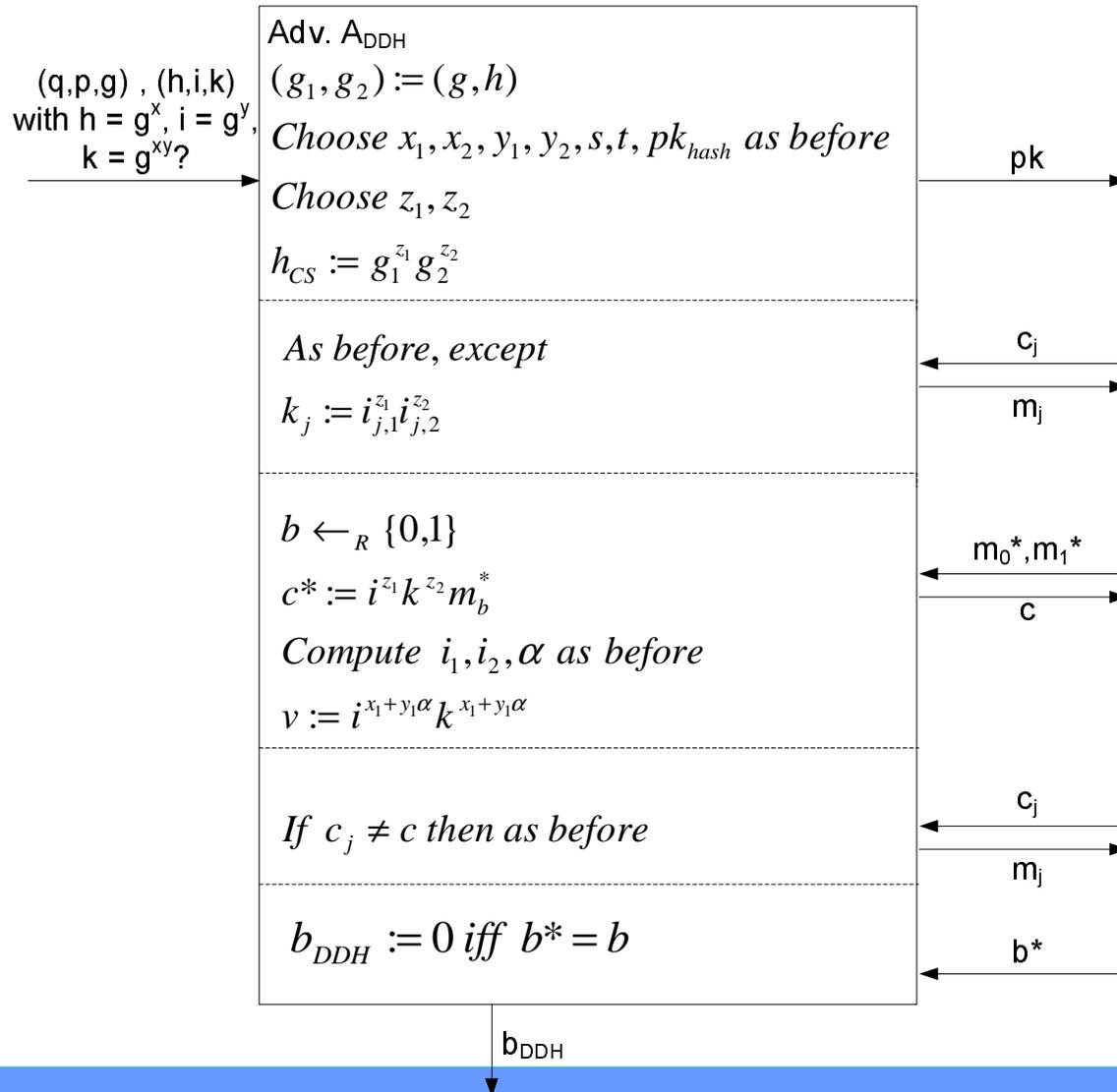
- Basic idea: Use given DH triple  $(h, i, k)$  (which is either  $(g^x, g^y, g^{xy})$  or  $(g^x, g^y, g^z)$ ), in public key and challenge-ciphertext such that:
    - We get a correct simulation if  $k=g^{xy} \rightarrow$  success probability of  $A_{CS}$  and hence of  $A_{DDH}$  significantly better than  $\frac{1}{2}$ .
    - The adversary  $A_{CS}$  learns no information about  $b$  if  $k=g^z$ , i.e., if  $k$  random  $\rightarrow$  success probability of  $A_{CS}$  and hence of  $A_{DDH} = \frac{1}{2}$ .
- $\rightarrow$  Difference not negligible  $\rightarrow$  DDH broken

# Intuition on the Reduction (cont'd)

---

- Where to use the DH triple?
- Set
$$g_1 := g \quad (g_2, i_1, i_2) := (h, i, k)$$
- If  $k=g^{xy}$  then  $i_1, i_2$  chosen according to correct probability distribution
- Problem:  $A_{\text{DDH}}$  does not know  $y$  and hence has to compute remaining components of the ciphertext differently.

# Detailed Overview of the Reduction



# Simulation of Key Generation

- Given  $(p, q, g)$ ,  $(h, i, k)$
- Use  $g_1 := g$  and  $g_2 := h$
- Pick random  $x_1, x_2, y_1, y_2, z_1, z_2 \in \{1, \dots, q\}$
- Set

$$s = g_1^{x_1} \cdot g_2^{x_2} \quad t = g_1^{y_1} \cdot g_2^{y_2} \quad h_{CS} = g_1^{z_1} g_2^{z_2}$$

- Let  $pk_{\text{hash}} \leftarrow \text{Gen}_{\text{Hash}}(n)$  ( Denote  $H(\cdot) := H(pk_{\text{hash}}, \cdot)$  )
- Output  $pk := (q, p, g_1, g_2, s, t, h, pk_{\text{hash}})$
- Remark: Only cheated with  $h_{CS}$  so far. But ok since suitable  $z$  can be defined as

$$z = z_1 + xz_2$$

# Simulation of Ciphertext Decryption

- Given  $c = (i_{j,1}, i_{j,2}, c_j^*, v_j)$
- $A_{\text{DDH}}$  knows entire secret key, but “wrong”  $z$  and  $h_{\text{CS}}$
- Verification of  $v$  as usual (uses neither  $z$  nor  $h_{\text{CS}}$ )
- Usually, one would compute  $i_{j,1}^z$ , but  $z$  unknown to  $A_{\text{DDH}}$ .
- Let  $i_{j,1} = g_1^{r_j}$  (but  $A_{\text{DDH}}$  does not know this  $r_j$ !). Then
 
$$k_j = i_{j,1}^z = g_1^{r_j z} = (g_1^{z_1} \cdot g_2^{z_2})^{r_j} = g_1^{r_j z_1} \cdot g_2^{r_j z_2}$$
- If  $i_{j,2} = g_2^{r_j}$  (correct ciphertext) then
 
$$k_j = i_{j,1}^{z_1} \cdot i_{j,2}^{z_2}$$
- Return  $m_j = c_j^* / k_j$
- Does not work for incorrect ciphertexts!
- Later show lemma: Adversary only finds correct  $v$ 's for incorrect (misformed) ciphertexts with negligible probability.

# Simulation of Encryption Challenge

- Given  $(h = g^x, i = g^y, k = g^{xy}?)$  and  $(m_0, m_1)$
- Choose bit  $b$  randomly and set  $i_1 := i$  and  $i_2 := k$ .
- For computing  $c^* = h_{CS}^y \cdot m_b$ ,  $A_{DDH}$  had to know  $y$ , but doesn't.
- Instead, it computes

$$k^* := h_{CS}^y = (g_1^{z_1} \cdot g_2^{z_2})^y = g_1^{yz_1} \cdot g_2^{yz_2} = i^{z_1} \cdot k^{z_2} \quad c^* = i^{z_1} \cdot k^{z_2} \cdot m_b$$

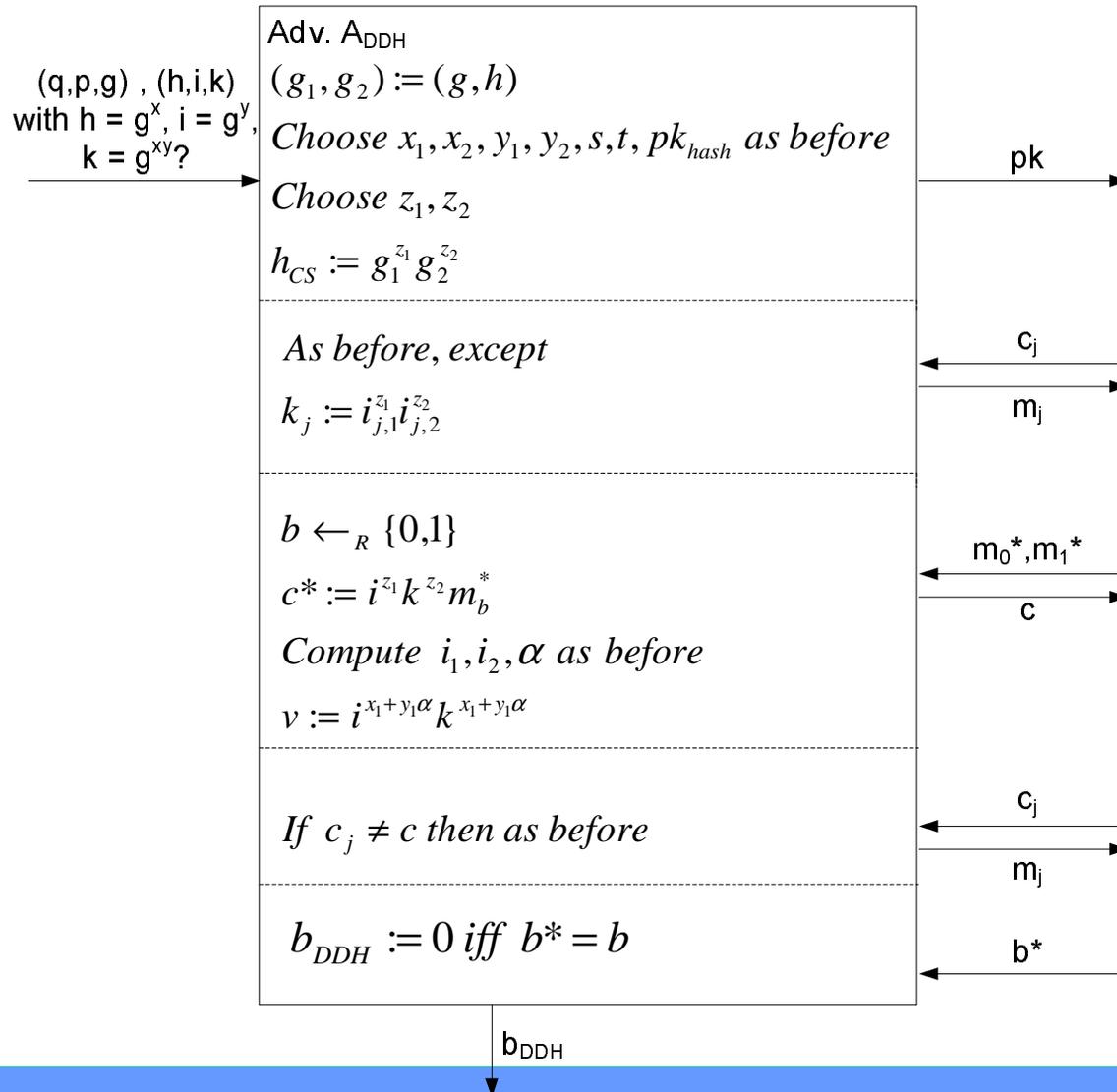
- $v$  again hard to compute because  $y$  unknown. Instead compute

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

i.e., as one would in the decryption routine.

- Perfectly correct simulation if  $k = g^{xy}$  although  $y$  unknown to  $A_{DDH}$  (always found alternative ways to compute the needed values)

# Detailed Overview of the Reduction



# Correct Simulation if $k = g^{xy}$

---

- Shown correct simulation in case  $k = g^{xy}$  except for decryption of malformed ciphertexts
- Lemma: If  $k = g^{xy}$ , the probability that  $A_{CS}$  succeeds in sending any ciphertext  $c_j$  whose first two components are not of the form

$$i_{j,1} = g_1^{r_j} \quad i_{j,2} = g_2^{r_j}$$

for some  $r_j$ , but which nevertheless passes the verification, is exponentially small.

- Lemma holds even for information-theoretic adversaries (easier to prove)

# Correct Simulation if $k = g^{xy}$

- Fix a ciphertext  $(i_{j,1}, i_{j,2}, c_j^*, v_j)$  and assume

$$i_{j,1} = g_1^{r_j} \quad i_{j,2} = g_2^{r_j^*}$$

for some  $r_j \neq r_j^* \pmod q$ .

- This ciphertext passes verification if

$$v_j = i_{j,1}^{x_1 + y_1 \alpha_j} \cdot i_{j,2}^{x_2 + y_2 \alpha_j} = g_1^{r_j(x_1 + y_1 \alpha_j)} \cdot g_2^{r_j^*(x_2 + y_2 \alpha_j)}$$

- Let  $\beta_j$  such that  $v_j = g_1^{\beta_j}$ . Then verification is true if and only if  $\pmod q$

$$\beta_j = r_j(x_1 + y_1 \alpha_j) + x r_j^*(x_2 + y_2 \alpha_j)$$

- Secrets here are  $x_1, x_2, y_1, y_2$ .

# Correct Simulation if $k = g^{xy}$

- If  $x_1, x_2, y_1, y_2$  were perfectly secret
  - clear that  $A_{CS}$  gets no information about  $\beta_j$
  - cannot construct  $v_j$  except for purely guessing it
- But  $A_{CS}$  gets information on  $x_1, x_2, y_1, y_2$ :
  - 1. Key generation:  $s$  and  $t$  contain information

$$s = g_1^{x_1} \cdot g_2^{x_2} \quad t = g_1^{y_1} \cdot g_2^{y_2}$$

yielding equations

$$\sigma = x_1 + xx_2 \quad \tau = y_1 + xy_2$$

- 2. Decryption of correct ciphertexts: Attacker only learns

$$k_j = i_{j,1}^{z_1} \cdot i_{j,2}^{z_2}$$

which is independent of  $x_1, x_2, y_1, y_2$

# Correct Simulation if $k = g^{xy}$

- But  $A_{CS}$  gets information on  $x_1, x_2, y_1, y_2$  (cont'd):
  - 3. Challenge encryption: Here  $c^*$  only depends on  $z_1, z_2$ , but

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

depends on  $x_1, x_2, y_1, y_2$ .

However this gives no new info since

$$v = s^r \cdot t^{r\alpha}$$

**in this correct case**, and  $s$  and  $t$  are known already.

- 4. Answers on incorrect ciphertexts: The case we are currently treating: With overwhelming probability, the answer will be the fixed error message.
  - Investigate linear equations to see what the adversary has learned (prob.  $\approx 1/q$  of guessing  $\beta_j$  right)

# The case k random

- Show that  $m_b$  perfectly hidden by  $k^* = i^{z_1} \cdot k^{z_2}$  (in a OTP manner)
- Problem again: If  $k^*$  perfectly secret, this would be clear, but  $A_{CS}$  gets information on  $k^*$  via  $z_1, z_2$ :
  - 1. Key generation:  $h_{CS}$ :

$$h_{CS} = g_1^{z_1} g_2^{z_2}$$

yielding an equation

$$z = z_1 + xz_2$$

- 2. Decryption of correct ciphertexts: Attacker only learns

$$k_j = i_{j,1}^{z_1} \cdot i_{j,2}^{z_2} = (g_1^{z_1} g_2^{z_2})^{r_j} = h_{CS}^{r_j}$$

and thus no new info on  $z_1, z_2$ .

# The case k random

- But  $A_{CS}$  gets information on  $z_1, z_2$  (cont'd):
  - 3. Challenge encryption: This is the case we are just considering
  - 4. Answers on incorrect ciphertexts: Show below: With overwhelming probability, the answer is the fixed error message.
- In summary, the attacker learns at most one linear equation about the two variables  $z_1, z_2$ .
  - $q$  pairs  $(z_1, z_2)$  still possible for the adversary
- The equation  $k^* = i^{z_1} \cdot k^{z_2}$  can be written as
 
$$\gamma = yz_1 + \delta z_2$$

for  $k^* = g_1^\gamma$  and  $k = g_1^\delta$
- Linearly independent of above eq. (except if  $k = g^{xy}$ )

# The case $k$ random

---

- Final lemma: If  $k = g^z$ , i.e.,  $k$  random, the probability that  $A_{CS}$  succeeds in sending any ciphertext  $c_j$  whose first two components are not of the form

$$i_{j,1} = g_1^{r_j} \quad i_{j,2} = g_2^{r_j}$$

for some  $r_j$ , but which nevertheless passes the verification, is exponentially small.

- Lemma holds again for information-theoretic adversaries
- Proof very similar to lemma for  $k = g^{xy}$

# The case k random

---

- Fix a ciphertext  $(i_{j,1}, i_{j,2}, c_j^*, v_j)$  and assume

$$i_{j,1} = g_1^{r_j} \quad i_{j,2} = g_2^{r_j^*}$$

for some  $r_j \neq r_j^* \pmod q$ .

- This ciphertext passes verification if and only if

$$\beta_j = r_j(x_1 + y_1\alpha_j) + xr_j^*(x_2 + y_2\alpha_j)$$

where  $\beta_j$  such that  $v_j = g_1^{\beta_j}$

- Secrets here again  $x_1, x_2, y_1, y_2$ .

# The case k random

- If  $x_1, x_2, y_1, y_2$  were perfectly secret  
 → clear that  $A_{CS}$  gets no information about  $\beta_j$   
 → cannot construct  $v_j$
- But  $A_{CS}$  gets information on  $x_1, x_2, y_1, y_2$ :

- 1. Key generation:  $s$  and  $t$

$$s = g_1^{x_1} \cdot g_2^{x_2} \quad t = g_1^{y_1} \cdot g_2^{y_2}$$

yielding equations

$$\sigma = x_1 + xx_2 \quad \tau = y_1 + xy_2$$

- 2. Decryption of correct ciphertexts: Attacker only learns

$$k_j = i_{j,1}^{z_1} \cdot i_{j,2}^{z_2}$$

which is independent of  $x_1, x_2, y_1, y_2$

# The case k random

- But  $A_{CS}$  gets information on  $x_1, x_2, y_1, y_2$  (cont'd):
  - 3. Challenge encryption (**Difference to last lemma!!**): Here  $c^*$  depends only on  $z_1, z_2$ , but

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

depends on  $x_1, x_2, y_1, y_2$ . **In the last lemma, this was**

$$v = s^r \cdot t^{r\alpha}$$

**for known s and t.** In this case, we get an additional equation:

$$\varepsilon = y(x_1 + y_1\alpha) + \delta(x_2 + y_2\alpha)$$

where  $k = g_1^\delta$  and  $v = g_1^\varepsilon$ .

# The case $k$ random

---

- But  $A_{CS}$  gets information on them (cont'd):
  - 4. Answers on incorrect ciphertexts: The case we are currently treating: With overwhelming probability, the answer will be the fixed error message.
- Finally (and then we are done):

Investigate linear dependency of equations to see what the adversary has learned

  - 1. Case: If  $\alpha \neq \alpha_j$  and  $k \neq g^{xy}$   
→  $q$  tuples possible from the adversaries point of view, but only one is correct (prob.  $1/q$  of guessing  $\beta_j$  right)
  - 2. Case:  $k = g^{xy}$  → Exponentially small probability.
  - 3. Case:  $\alpha = \alpha_j$  → Collision found (or guessed the challenge ciphertext before the challenge phase)!