CS171: Cryptography

Lecture 14

Sanjam Garg

Cryptographic Group

- If p and q are primes such that $2q \ = \ p - 1$ and let g $\in Z_p^*$ be an elements of order q. Let $H \ = \langle g \rangle$ be the group of order q.
- Example, p = 23 and q = 11
- $Z_p^* = \{1, 2, \ldots 22\}$ and $a \cdot b = ab \; mod \; 23$

$\langle g \rangle$

- $Z_p^* = \{1, 2, \ldots 22\}$
- $\langle 1 \rangle = \{1\}$
- $\langle 2 \rangle = \{2, 4, 8, 16, 9, 18, 13, 3, 6, 12, 2^{11} = \ 1\}$
- $\langle 5 \rangle = \{5, 2, 10, 4, 20, 8, 17, 16, 11, 12, \ldots 5^{22} = 1\}$
- $\langle 22 \rangle = \{22, 22^2 = 1\}$

- Pick any g such that $g^{11} = 1$.
- For example, $\mathrm{H} = \langle 2 \rangle$ is of prime order
- For hardness use large primes.

The Discrete-Log Problem

- Let $\mathcal{G}(1^n)$ be a PPT algorithm that generates description of a cyclic group, i.e., order q (where $|q| = n$) and a generator g.
- Unique bit representation for each element and group operation can be performed in time polynomial in n.
- Sampling a uniform group element: Sample $x \leftarrow Z_q$ and compute g^x.

DLOG Problem

$\mathrm{DLog}_{\mathrm{A},\mathcal{G}}$(n)

1. Run $\mathcal{G}(1^n)$ to obtain (G, g, q).
2. Pick uniform $h \in G$.
3. A is given (G, g, q, h) and it outputs x.
4. Output 1 if $g^x = h$ and 0 otherwise

Discrete-Log Problem is hard relative to $\mathcal{G}$ if

$\forall$ $PPT\ A$ $\exists$ $negl$ such that:

$\left|\Pr\left[\mathrm{DLog}_{\mathrm{A},\mathcal{G}}(\mathrm{n}) = 1\right]\right| \le$ negl(n).

The Diffie-Hellman Problems

- The computational variant: given g^x and g^y compute g^{xy}
- The decisional variant: given g^x and g^y distinguish between g^{xy} and a random group element.

Computational Diffie-Hellman Problem

$\mathrm{CDH}_{\mathrm{A},\mathcal{G}}$(n)

1. Run $\mathcal{G}(1^n)$ to obtain (G, g, q).
2. $a, b \leftarrow Z_q^*$.
3. A is given (G, g, q, g^a, g^b) and it outputs h.
4. Output 1 if $g^{ab} = h$ and 0 otherwise

CDH is hard relative to $\mathcal{G}$ if

$\forall$ $PPT\ A$ $\exists$ $negl$ such that:

$\left|\Pr\left[\mathrm{CDH}_{\mathrm{A},\mathcal{G}}(\mathrm{n}) = 1\right]\right| \leq$ negl(n).

Decisional Diffie-Hellman Problem

$\mathrm{DDH}_{A,\mathcal{G}}$ (n)

1. Run $\mathcal{G}(1^n)$ to obtain (G, g, q).
2. $a, b, r \leftarrow Z_q^*$. Sample a uniform bit c.
3. A is given $(G, g, q, g^a, g^b, g^{ab+cr})$ and it outputs c'.
4. Output 1 if $c = c'$and 0 otherwise

DDH is hard relative to $\mathcal{G}$ if

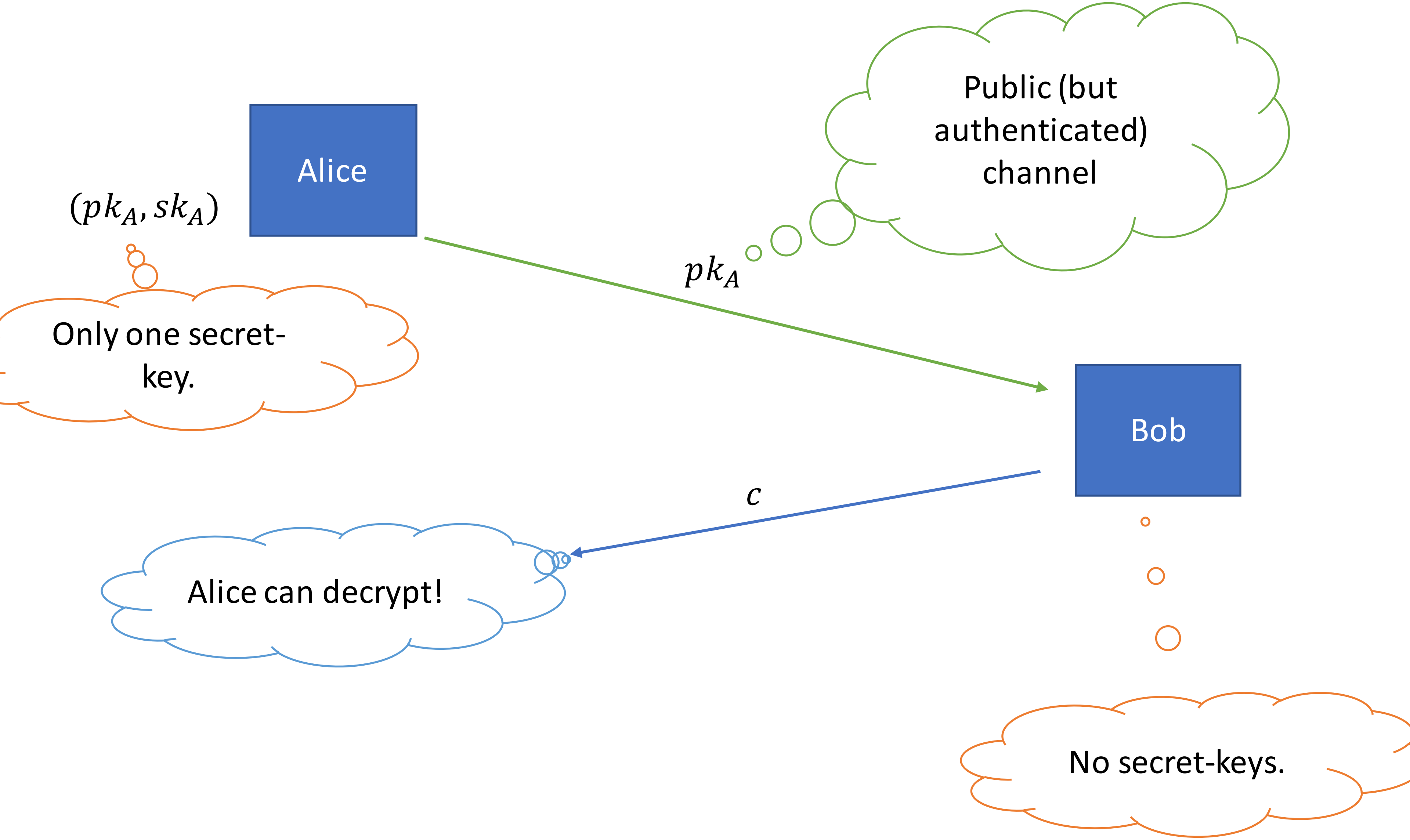
$\forall$ $PPT\ A$ $\exists$ $negl$ such that:

$\left|\Pr\left[\mathrm{DDH}_{A,\mathcal{G}}(\mathrm{n}) = 1\right]\right| \leq$ ½ + negl(n).

Public-Key Cryptography

- Public-Key Encryption
- Digital Signatures

Public-Key Encryption



Public-Key Encryption vs Private-Key Encryption

- Public-key encryption is strictly stronger than private-key encryption

- Then why even use private-key encryption?
  - Public-key encryption is roughly 2-3 orders of magnitude slower than private-key encryption

Public-Key Encryption

- A public-key encryption scheme is a triple of PPT algorithms (Gen, Enc, Dec) such that:

1. $Gen(1^n) \rightarrow (pk, sk)$
2. $Enc(pk, m) \rightarrow c$
3. $Dec(sk, c) \rightarrow m/\perp$

- Correctness: For all (pk, sk) output by $Gen(1^n)$, we have that $\forall$ (legal) $m, Dec\left(sk, Enc(pk, m)\right) = m$
- Security: EAV-security, CPA-security?

EAV Security

$\mathrm{PubK}^{\mathrm{eav}}_{\mathrm{A},\Pi}(n)$

1. $(pk, sk) \leftarrow G(1^n)$ and give pk to A.
2. A outputs $m_0, m_1 \in \{0,1\}^*, |m_0| = |m_1|$.
3. $\mathrm{b} \leftarrow \{0,1\}, c \leftarrow Enc(pk, m_b)$
4. c is given to A and it outputs b'
5. Output 1 if $b = b$' and 0 otherwise

Encryption scheme $\Pi = (Gen, Enc, Dec)$ is indistinguishable in the presence of an eavesdropper, or is *EAV-secure* if

$\forall$ PPT A it holds that:

$$\Pr\left[\mathrm{PubK}^{\mathrm{eav}}_{\mathrm{A},\Pi} = 1\right] \leq \frac{1}{2}$$

+ negl(n)

EAV-security vs CPA Security

- In the public-key setting the two notions are identical.

- Since, given the public-key, encryption can be performed (without any secret values)

- Hence, encryption must be randomized

What about security of multiple messages?

- CPA-security implies security for encrypting multiple messages (same as the private-key setting)

- $Enc(pk, m_1 \ldots m_n)$: $Enc(pk, m_1) \ldots Enc(pk, m_n)$

- Proof via a direct hybrid argument

CCA Security (A bigger concern in the PKE setting)

- Attacker can obtain decryptions of ciphertexts of its choice itself

- Attacker can more easily come up with illegitimate ciphertexts (cannot have a MAC on a ciphertext)

- Malleability: An attacker can given a ciphertext c encrypting a message m could obtain a ciphertext c' of a related message m' (without knowing m' itself)

CCA Security

Much harder in the PKE setting.

$\text{PubK}^{\text{CCA}}_{\text{A},\Pi}(n)$

1. $(pk, sk) \leftarrow G(1^n)$ and give pk to A.
2. $A^{Dec(sk,\cdot)}$ outputs $m_0, m_1 \in \{0,1\}^*, |m_0| = |m_1|$.
3. b ← {0,1}, $c^* \leftarrow Enc(pk, m_b)$
4. c is given to $A^{Dec(sk,\cdot)}$ and it outputs b' (query c^* not allowed)
5. Output 1 if $b = b'$ and 0 otherwise

Encryption scheme $\Pi = (Gen, Enc, Dec)$ is indistinguishable in the presence of a CCA attacker, or is *CCA-secure* if

$\forall$ PPT A it holds that:

$$\Pr\left[\text{PubK}^{\text{cca}}_{\text{A},\Pi} = 1\right] \leq \frac{1}{2}$$

+ negl(n)

Construction of PKE

ElGamal Encryption

Correctness?

1. $Gen(1^n) \rightarrow (pk, sk)$
   1. Run $\mathcal{G}(1^n)$ to obtain (G, g, q).
   2. Sample $x \leftarrow Z_q$ and set $h = g^x$
   3. Set $pk = (G, g, q, h)$ and $sk = x$.
2. $Enc(pk, m \in G) \rightarrow c = (c_1, c_2)$
   1. Parse $pk = (G, g, q, h)$
   2. Sample $r \leftarrow Z_q$ and set $c_1 = g^r$ and $c_2 = m \cdot h^r$
3. $Dec(sk, c) \rightarrow m/\perp$
   1. Parse $c = (c_1, c_2)$
   2. Output $\frac{c_2}{c_1^r}$

Security based on DDH!

Encrypting long messages

- Encrypting block-by-block is inefficient
  - Ciphertext expands for each block
  - Public-key encryption is “expensive”

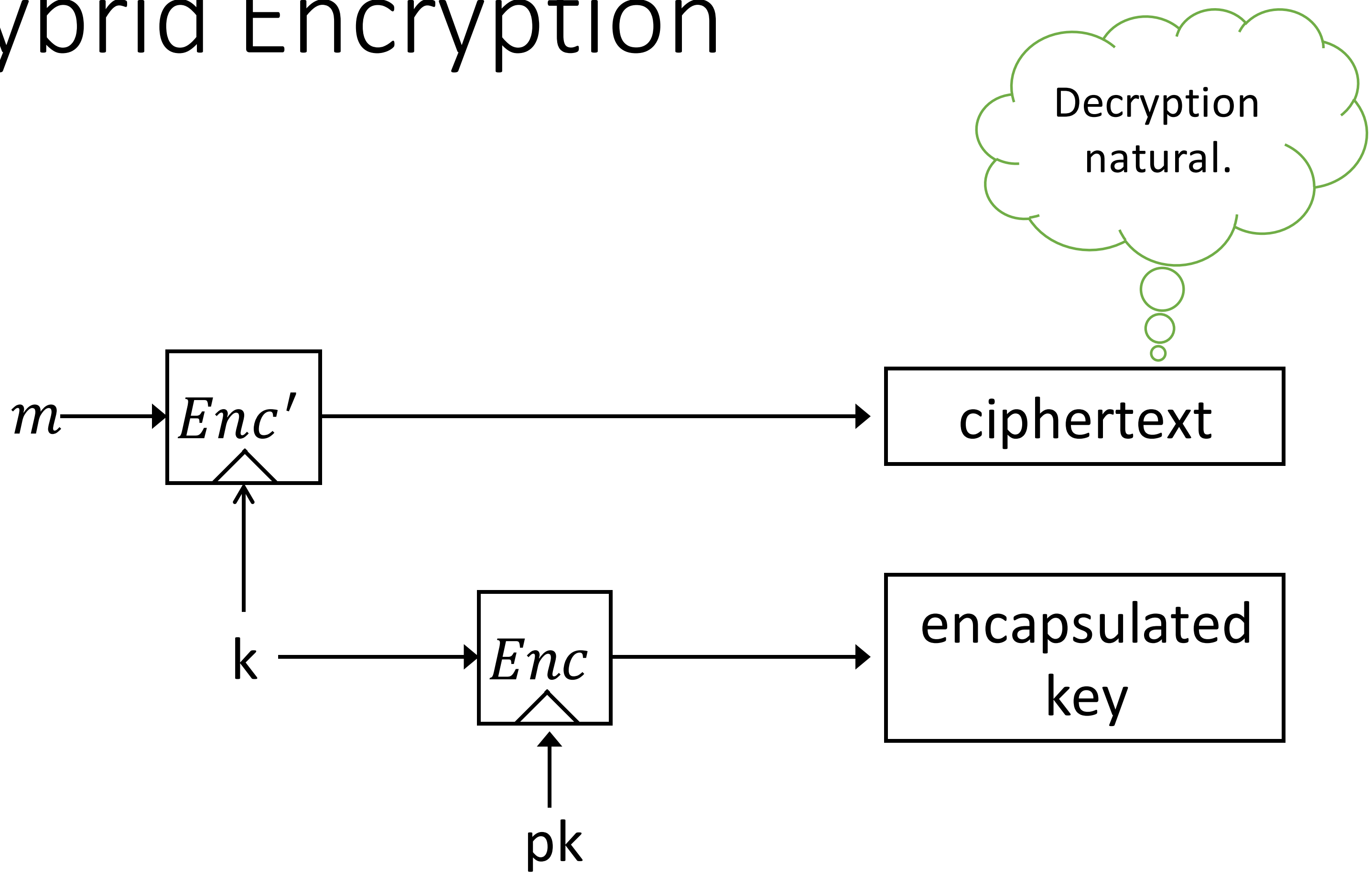
- Anything better?

Hybrid Encryption

- Use public-key encryption to set up a shared secret-key k which is then used to encrypt the message itself

- Benefits:
  - The inefficiency of the public-key encryption is not the bottleneck; i.e. we get amortized efficiency as the message is large
  - The ciphertext expansion over the message is small

Hybrid Encryption



The *functionality* of public-key encryption
at the (asymptotic) *efficiency* of private-key encryption!

Hybrid Encryption: More Formally

- Let Π be a public-key scheme, and let Π' be a private-key scheme
- Define Π_{hy} as follows:
  - $\mathrm{Gen}_{\mathrm{hy}}$ = Gen_{Π}
  - $Enc_{hy}(pk, m)$
    1. Sample $k \leftarrow \{0,1\}^n$
    2. $c \leftarrow Enc\,(pk, k)$
    3. $c' \leftarrow Enc'_k(m)$
    4. Output (c, c')
  - $\mathrm{Dec}_{\mathrm{hy}}\big(\mathrm{sk}, (\mathrm{c}, \mathrm{c}')\big)$
    1. Decrypt c to get k
    2. Use k to decrypt c and recover m.

Security of hybrid encryption

- If Π and Π' are CPA secure, then
  Π_{hy} is also CPA secure.
  - In fact, even if Π' is EAV secure

- If Π and Π' are CCA secure, then
  Π_{hy} is also CCA secure.

ElGamal Hybrid Encryption

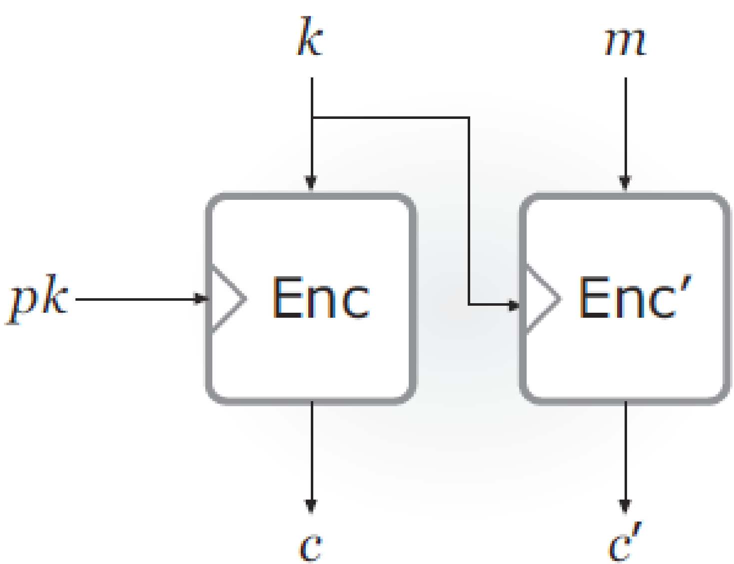
- The private key k needs to be encoded as a group element
  - Not clear how to do it!

- Alternative: Rather than encryption a specific key k, encrypt a random group element M
  - And derive the key as $k = H\ (M)$

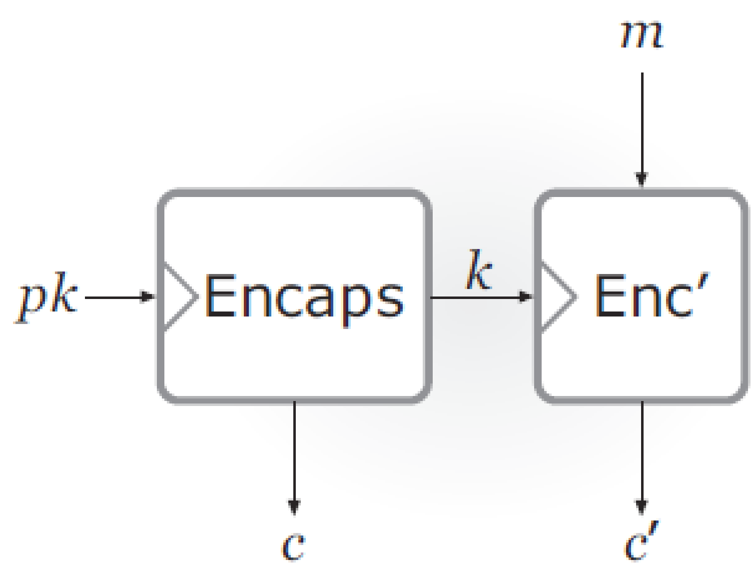
Key Encapsulation Mechanism

- Lesson: Do not need CPA security for hybrid encryption
- Sufficient to have a key encapsulation mechanism, or KEM for short
  - Takes as input a public-key and outputs a ciphertext c and a key k encapsulated in c
  - Correctness: k can be recovered from c using sk
  - Security: k is indistinguishable from uniform given pk and c (analogues of CPA/CCA security)
- Can be used to construct PKE by combining with private-key encryption

Hybrid Encryption (PKE vs KEM)



Hybrid encryption



KEM/DEM

Security

- If Π (KEM) and Π' are CPA secure, then

  Π_{hy} is also CPA secure.
  - In fact, even if Π' is EAV secure

- If Π (KEM) and Π' are CCA secure, then

  Π_{hy} is also CCA secure.

KEM based on ElGamal

1. $Gen(1^n) \to (pk, sk)$
   1. Run $\mathcal{G}(1^n)$ to obtain (G, g, q).
   2. Sample $x \leftarrow Z_q$ and set $h = g^x$
   3. Set $pk = (G, g, q, h)$ and $sk = x$.
2. $Encap(pk) \to (c, k)$
   1. Parse $pk = (G, g, q, h)$
   2. Sample $r \leftarrow Z_q$ and set $c = g^r$ and $k = H(h^r)$
3. $Decap(sk, c) \to k$
   1. Output $k = H(c^{sk})$

KEM security based on DDH and H can extract the randomness in the input

Efficiency

- For short messages: Directly use PKE
- For long messages: Use hybrid encryption
  - This is how things are done in practice

Is ElGamal Encryption CCA Secure?

- ElGamal Ciphertext $c_1 = g^r$ and $c_2 = m \cdot h^r$
- Given this ciphertext construct another ciphertext that encrypts the same message.
- Sample uniform s.
- $c_1' = c_1 \cdot g^s$ and $c_2' = c_2 \cdot h^r$

Thank You!