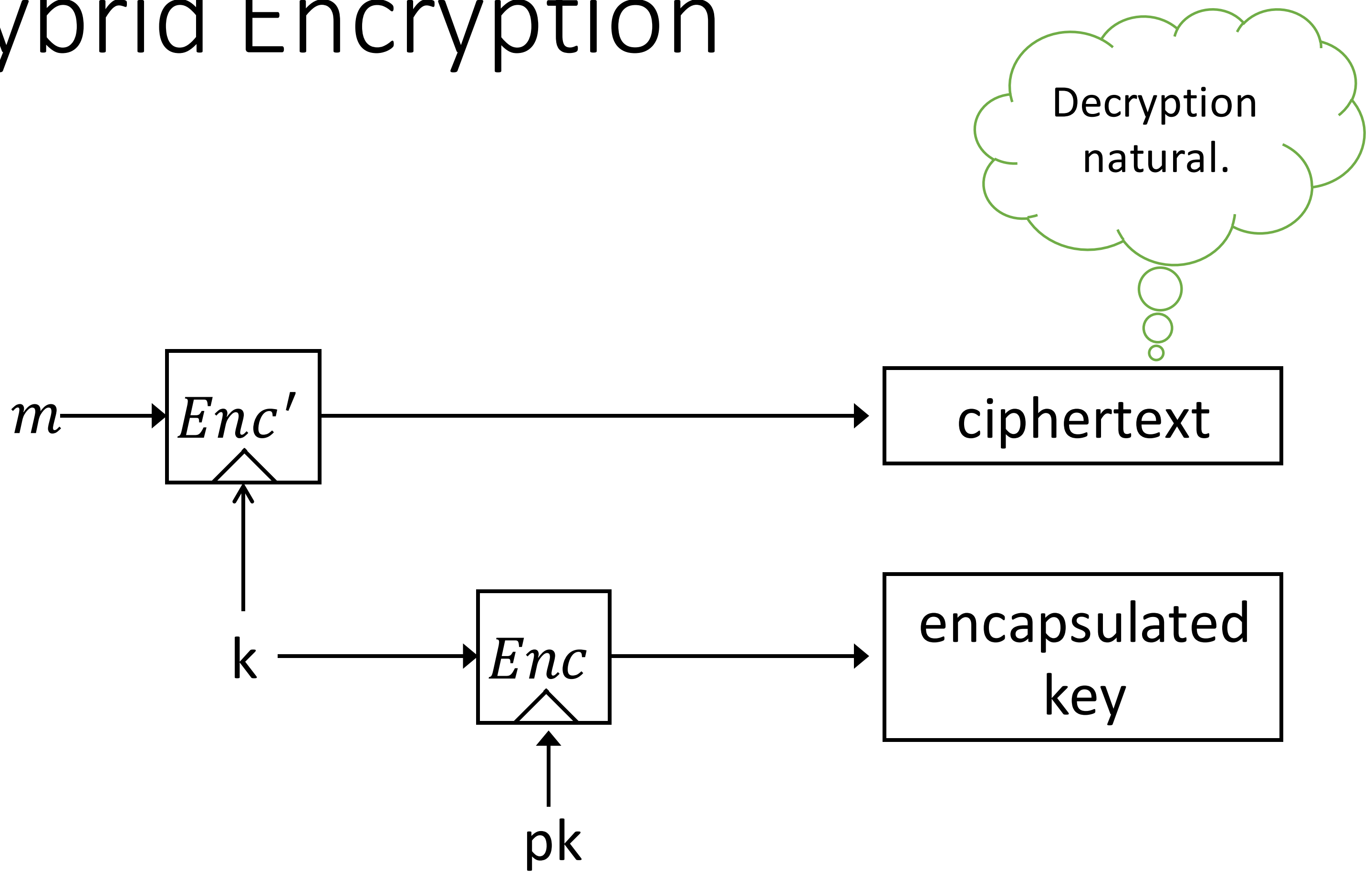
CS171: Cryptography

Lecture 15

Sanjam Garg

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')
  - $\mathrm{Dec}_{\mathrm{hy}}\left(\mathrm{sk}, (\mathrm{c}, \mathrm{c}')\right)$
    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.

CPA Security Proof

- H_0: A's input is $Enc(pk, k), Enc'_k(m_b)$ where $k \leftarrow \{0,1\}^n$
- H_1: A's input is $Enc(pk, r), Enc'_k(m_b)$ where $k, r \leftarrow \{0,1\}^n$
- H_2: A's input is $Enc(pk, r), Enc'_k(0^{|m_b|})$ where $k, r \leftarrow \{0,1\}^n$

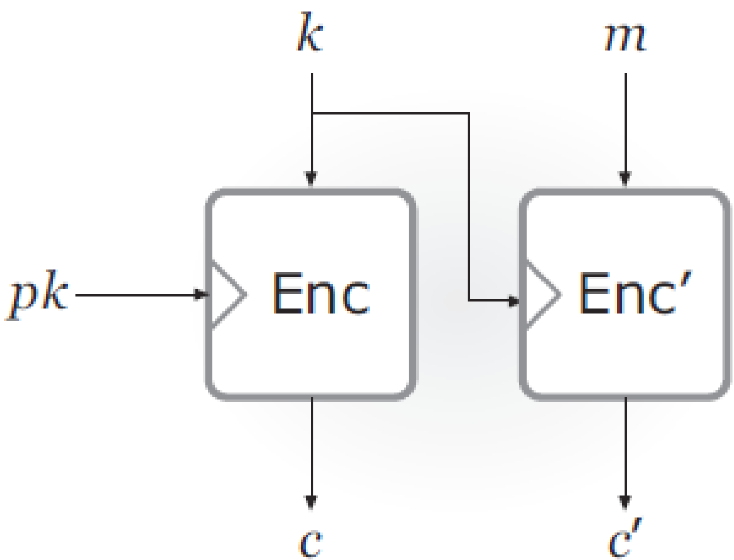
ElGamal Hybrid Encryption

- 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 $k = H\ (M)$

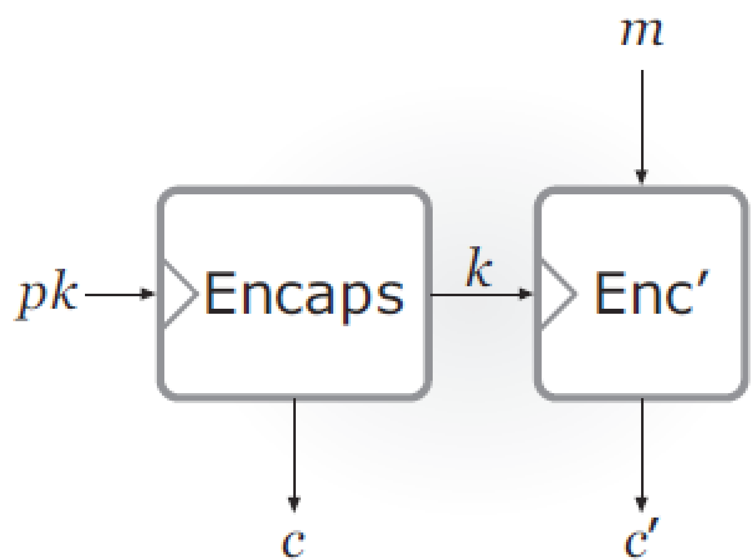
Key Encapsulation Mechanism

- Lesson: Do not need CPA secure PKE for CPA secure 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. $Gen(1^n) \rightarrow (pk, sk)$
   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. $Encap(pk) \rightarrow (c, k)$
   1. Parse $pk = (G, g, q, h)$
   2. Sample $r \leftarrow Z_q$ and set $c = g^r$ and $k = H(h^r)$
3. $Decap(sk, c) \rightarrow k$
   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 $c_2' = c_2 \cdot h^r$

Homomorphic Properties of ElGamal Encryption

- Given two ciphertexts
  - (c_1, c_2) encrypting message m
  - (c_1', c_2') encrypting message m'
- Can we obtain an encryption $m \cdot m'$?

- Answer: $(c_1 \cdot c_1', c_2 \cdot c_2')$ is an encryption of $m \cdot m'$.

- Any homomorphic encryption scheme cannot be CCA secure.

RSA Encryption

Group Z_N^*

- Consider the set $Z_N = \{0, \ldots, N-1\}$
- This is a group with respect to addition. Is it a group with respect to multiplication?

$$Z_N^* = \{b \in \{1, \ldots N-1\} \mid \gcd(b, N) = 1\}$$

- Removes 0 and elements that are not-coprime to N.
- Interested in $N = p \cdot q$, where p and q are prime

$$\phi(N) = (p-1)(q-1)$$

Factoring Problem

The RSA cryptosystem is not known to be as hard as factoring.

- Multiplication can be done in polynomial time; but factoring a number in general is believed to be hard.

- It's not hard to factor most numbers
  - Half the numbers are even.
  - 1/3 of the numbers are divisible by 3 and so on..

- Numbers obtained as products of two equal length primes are hardest to factor

The RSA Problem

- Let $N = p \cdot q$, where p and q are distinct odd primes

$$Z_N^* = \{b \in \{1, \ldots N-1\} \mid \gcd(b, N) = 1\}$$

- Fix an e such that $\gcd(e, \phi(N)) = 1$
  - $x \rightarrow x^e$ mod N is a permutation of Z_N^*
- If $ed = 1 \ mod \ \phi(N)$ then:
  - $(x^e)^d = x$ mod N

Can be computed if p and q are known!

The RSA Problem

$RSA_{A,\mathcal{G}}$(n)

1. Run $\mathcal{G}(1^n)$ to obtain (N, e, d).
2. Pick uniform $y \in Z_N^*$.
3. A is given (N, e, y) and it outputs x.
4. Output 1 if $x^e = y$ and 0 otherwise

RSA Problem is hard relative to $\mathcal{G}$ if

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

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

``Plain'' RSA Encryption

1. $Gen(1^n) \rightarrow (pk, sk)$
   1. Run $\mathcal{G}(1^n)$ to obtain (N, e, d).
   2. Set $pk = (N, e)$ and $sk = d$.
2. $Enc(pk, m \in Z_N^*) \rightarrow c$
   1. Parse $pk = (N, e)$
   2. Set $c = m^e \bmod N$
3. $Dec(sk, c) \rightarrow m/\perp$
   1. Output $c^{sk} \bmod N$

Is this secure?

- CPA Secure?

- This scheme is deterministic and thus not CPA-secure.

- c leaks partial information about m.

Plain RSA should never be used.

Secure RSA Encryption

1. $Gen(1^n) \rightarrow (pk, sk)$
   1. Run $\mathcal{G}(1^n)$ to obtain (N, e, d).
   2. Set $pk = (N, e)$ and $sk = d$.
2. $Enc(pk, m \in \{0,1\}) \rightarrow c = (c_1, c_2)$
   1. Parse $pk = (N, e)$. Sample a random $r \in Z_N^*$.
   2. Set $c_1 = r^e \bmod N$ and $c_2 = hc(r) \oplus m$
3. $Dec(sk, c) \rightarrow m/\bot$
   1. Output $hc(c_1^{sk} \bmod N) \oplus c_2$

Interestingly lsb(r) is also hardness concentrate function for the RSA function.

Thank You!