Today in Cryptography (5830)

Hybrid encryption recap
OpenPGP standard
TextSecure

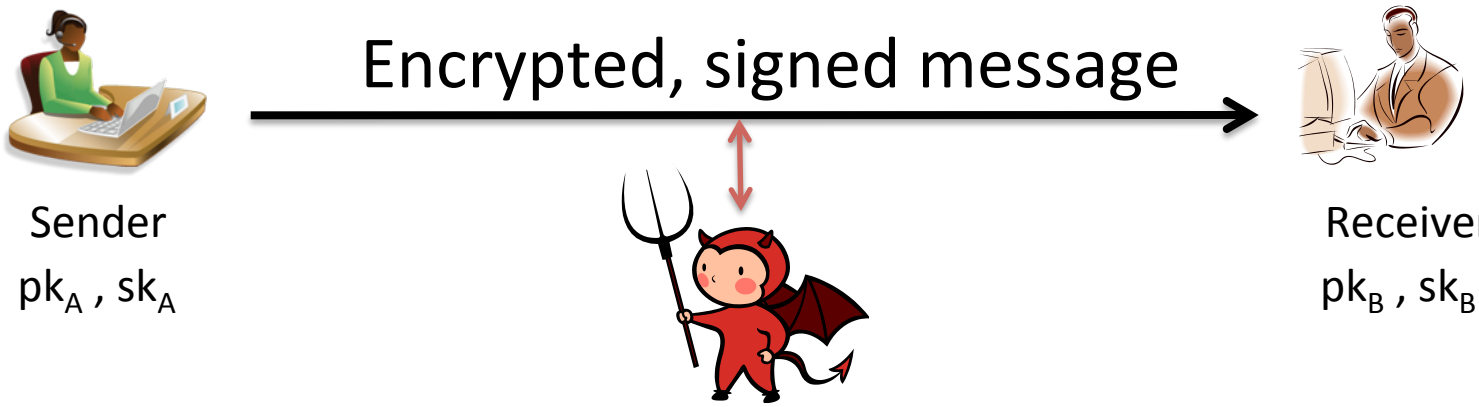
Katz-Lindell Chapter 10.3 (Hybrid Encryption)
RFC 4880 (OpenPGP standard)

Application-layer crypto

- So far focused on TLS as running example
  - Transport Layer Security
  - Provides network socket style stream interface

- What about if an application wants to encrypt discrete messages (as opposed to stream)?
  - Email
  - Text messages
  - Etc.

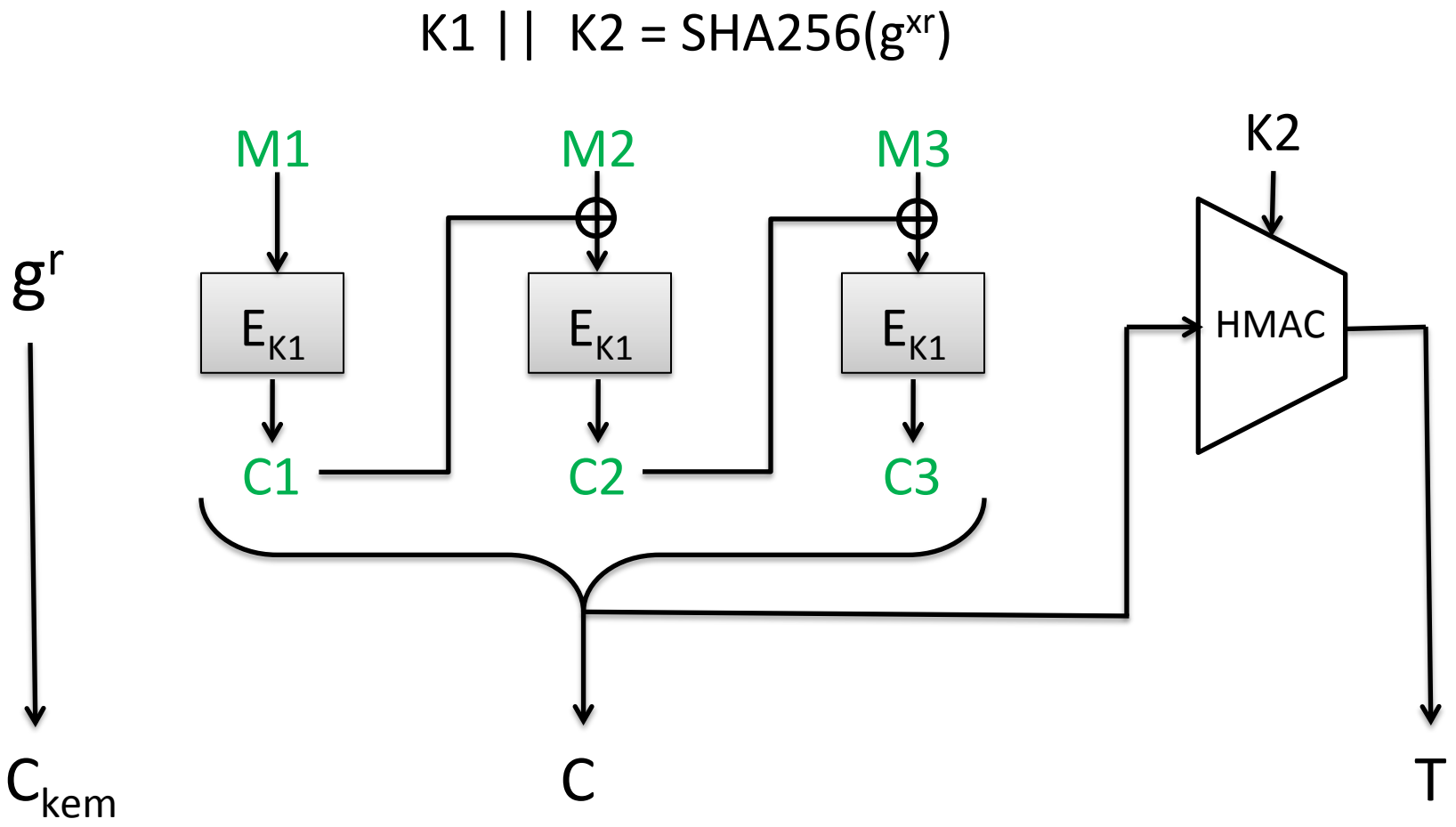
Email encryption



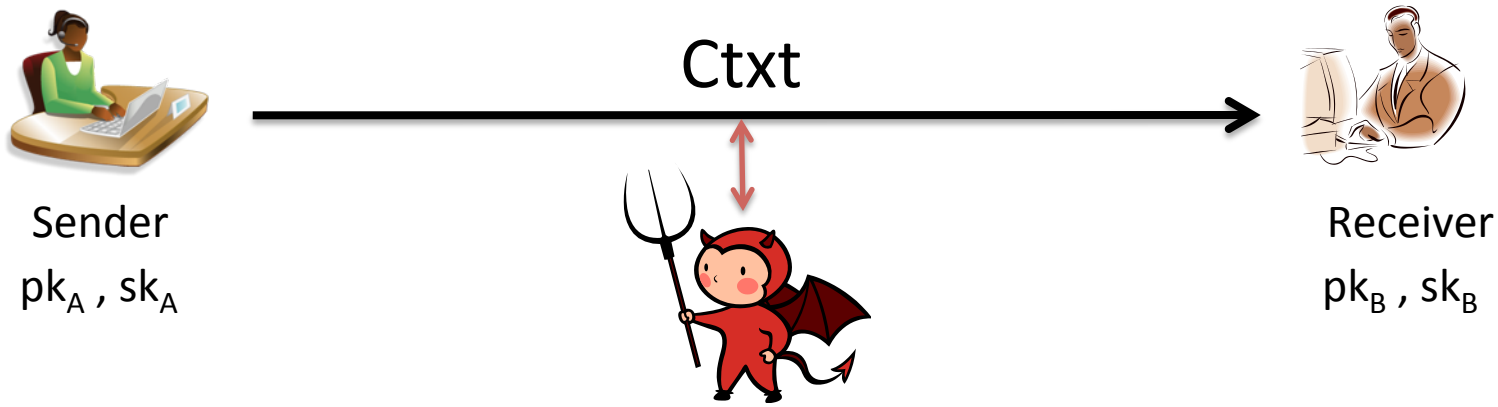
- Message may be large (body of email, PDF of attachments)
- Desire authenticity and confidentiality
- Public-keys delivered out-of-band
  - Websites, key parties, key directory servers

Example hybrid encryption

Enc(X,M):



Email encryption



- To digitally sign, let M = Msg || Sign(sk_A , Msg)
- Ctxt = Encrypt(pk_B , M)

PGP history

- Phil Zimmerman released “Pretty Good Privacy” in 1991 on a USENET post marked as “US only”

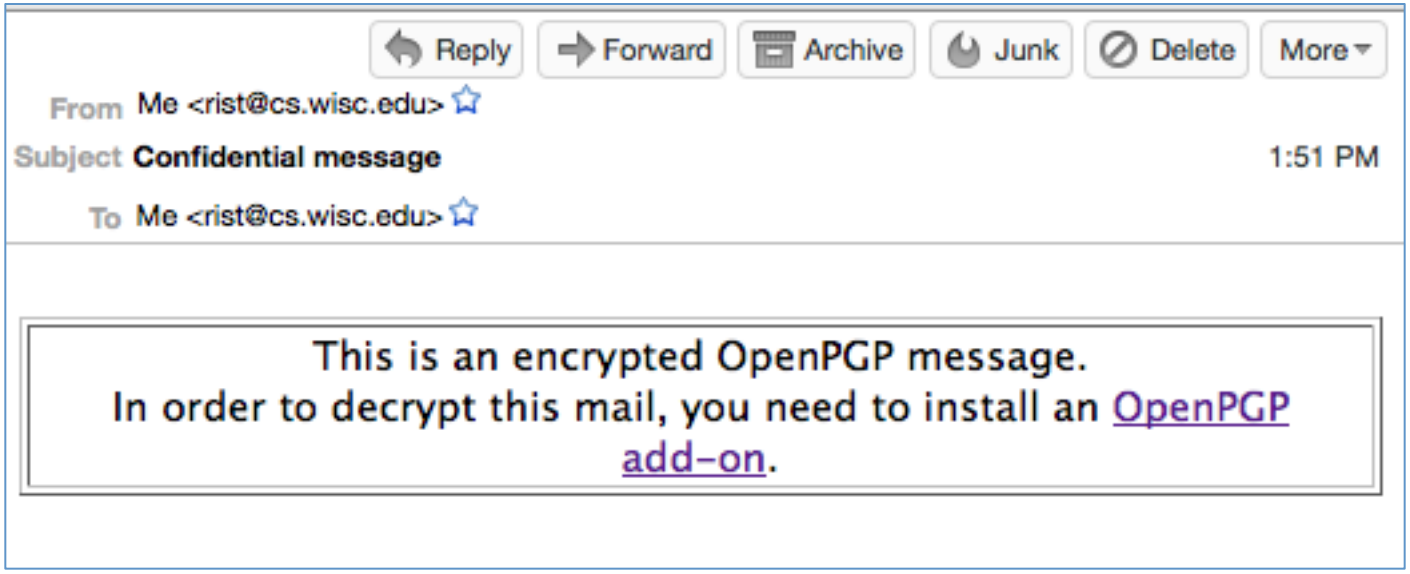
- 1993: Criminal investigation by US government for munitions export without a license.
  - Printed PGP source code into a book. First amendment gambit

OpenPGP overview

- Standard for PGP is RFC 4880
- Key encapsulation mechanism:
  - RSA PKCS#1 v1.5 encryption
  - ElGamal over finite field or elliptic curve
- Digital signatures:
  - RSA PKCS#1 v1.5 signatures
  - DSA
- Symmetric encryption:
  - Password-based key derivations using iterated hashing
  - CFB mode using block cipher

OpenPGP overview

- Security problems:
  - Padding oracle attacks against CFB & PKCS#1 v1.5
  - http://www.ssi.gouv.fr/uploads/2015/05/format-Oracles-on-OpenPGP.pdf
  - Attacks against home-brewed integrity checks (modification detection check, MDC)
  - Subject lines always in the clear
- Usability problems:
  - Users must manage their own keys
  - Copying private keys to each device
  - "Why Johnny Can't Encrypt" http://www.gaudior.net/alma/johnny.pdf
  - 90 minutes to send encrypted email, 66% failure rate



How Glenn Greenwald Began Communicating With NSA Whistleblower Edward Snowden

06/10/2013 04:23 pm ET | **Updated** Jun 11, 2013

The Switch

Yahoo's plan to get Mail users to encrypt their e-mail: Make it simple

A | 14 | Save for Later | Reading List

Messaging encryption

Encrypted/Signed SMS or IM

Sender

pk_A , sk_A

Receiver

pk_B , sk_B

- End-to-end encrypted messaging is a big topic
- TextSecure is protocol adopted by WhatsApp (~1 billion users)

TextSecure

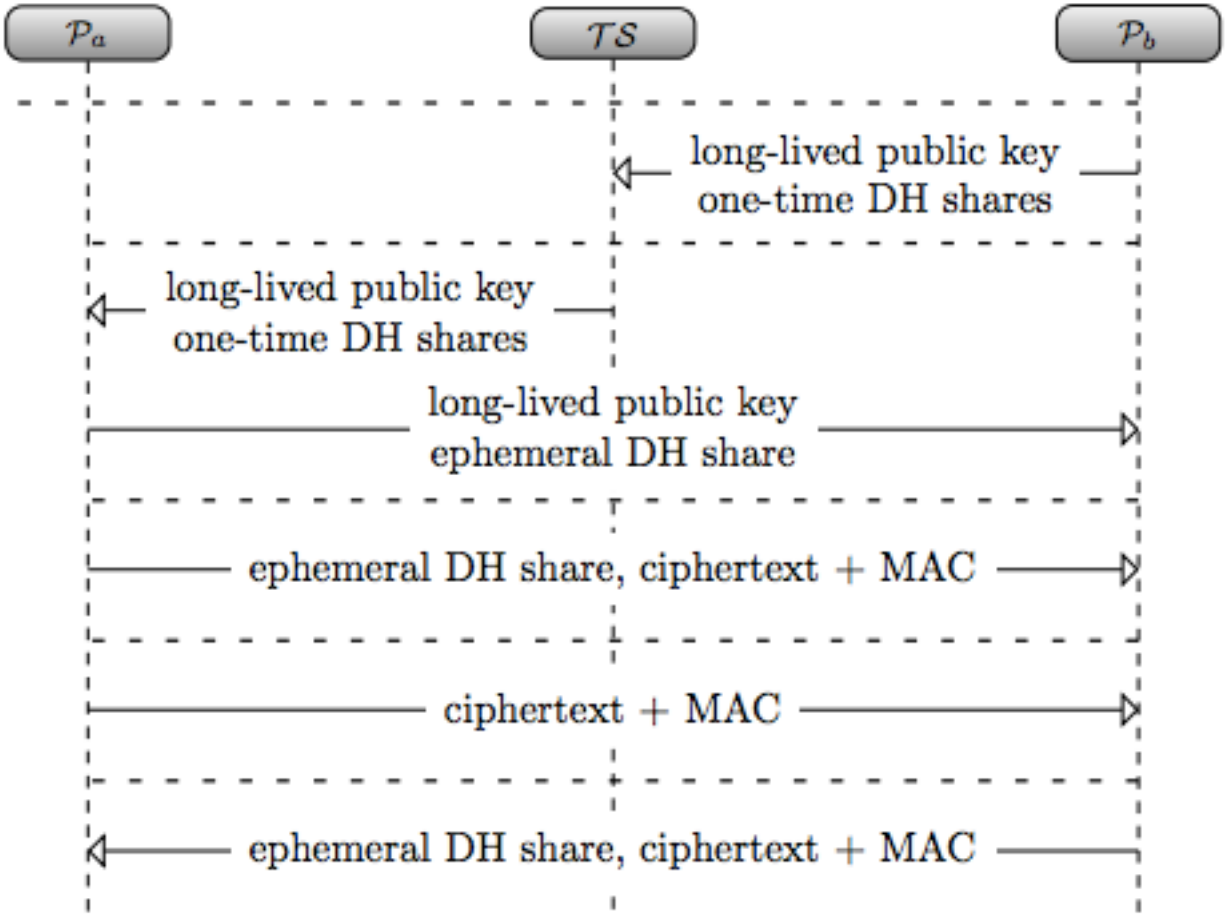
Encrypted/Signed SMS or IM

Sender

pk_A , sk_A

Receiver

pk_B , sk_B



https://eprint.iacr.org/2014/904.pdf

TextSecure

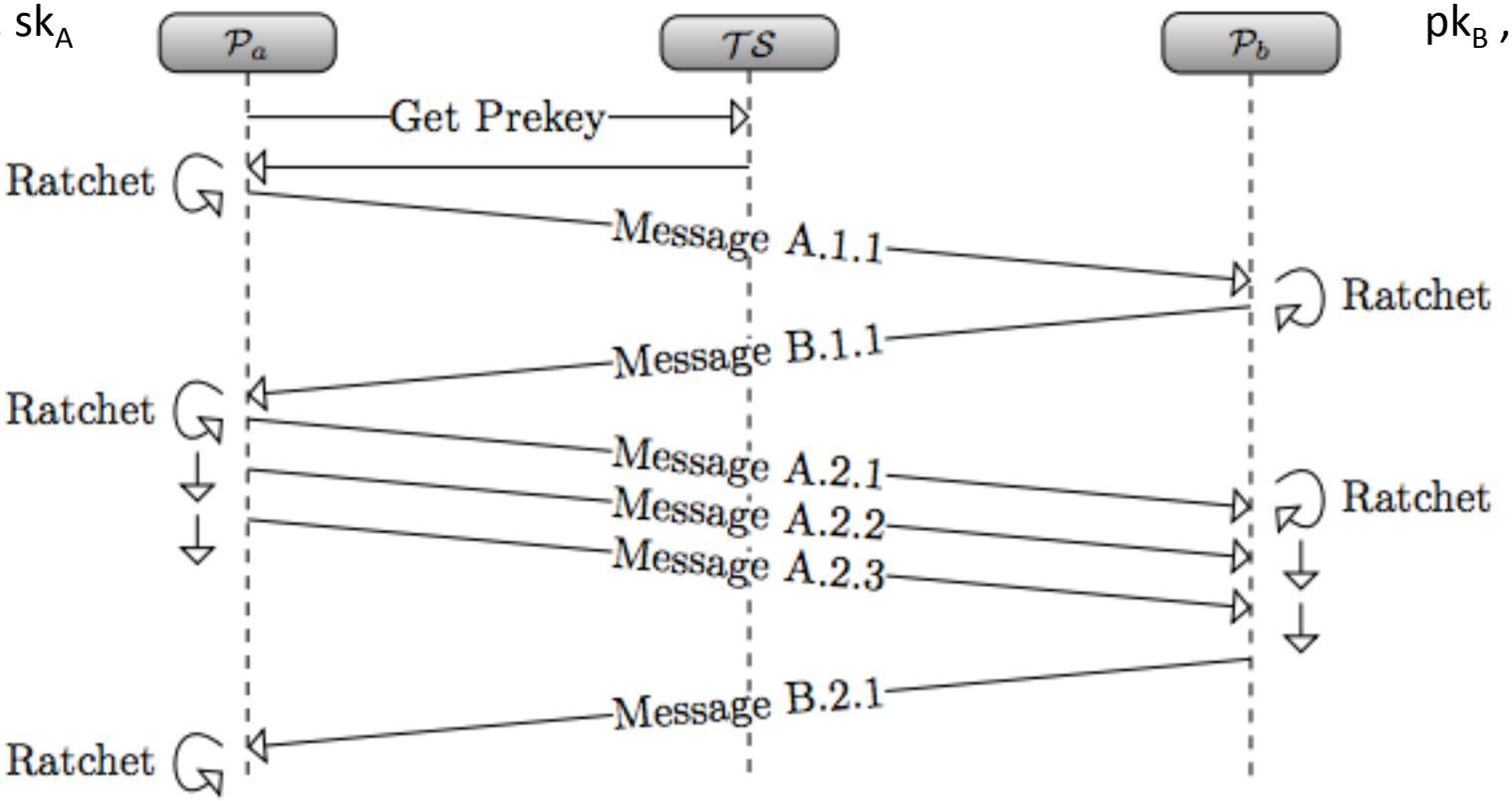
Encrypted/Signed SMS or IM

Sender

pk_A , sk_A

Receiver

pk_B , sk_B

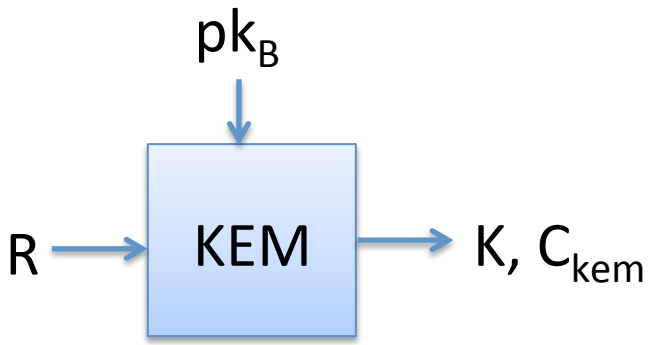


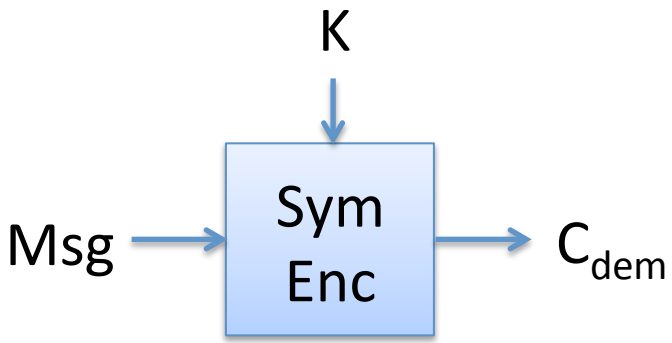
https://eprint.iacr.org/2014/904.pdf

Summary

- Hybrid encryption uses combination of asymmetric and symmetric cryptography
  - Key encapsulation mechanisms (KEM) based on secure PKE, (elliptic curve) Diffie-Hellman
  - Use an authenticated encryption scheme for data encapsulation mechanism (DEM)
- PGP is historical example (and still somewhat widely used)
- End-to-end messaging for IM/chat hotter topic, now widely deployed

Hybrid encryption (KEM/DEM)





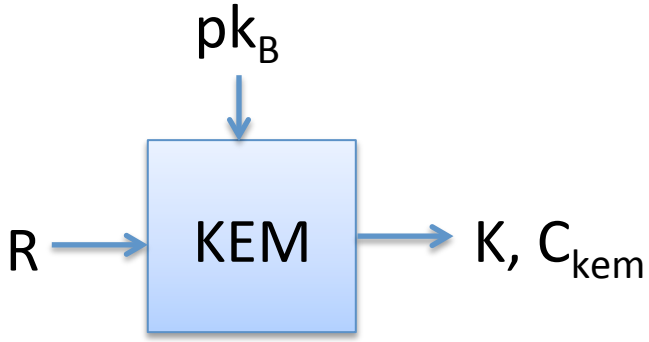
KEM = key encapsulation mechanism
Public-key primitive

DEM = data encapsulation mechanism
One-time secure authenticated encryption

HybEnc(pk, M)
Choose randomness R
K, C_{kem} <- KEM(pk,R)
C_{dem} <- Enc(K,M)
Return C_{kem} , C_{dem}

HybDec(sk, C_{kem} , C_{dem})
K <- KEM^{-1}(sk, C_{kem})
M <- Dec(K, C_{dem})
Return M

KEM from PKE



KEM(pk, R)

C_{kem} <- PKE-Enc(pk,R)

Return H(R), C_{kem}

KEM = key encapsulation mechanism

Public-key primitive

ElGamal encryption

Kg outputs pk = (g,X =g^x) and sk = (g,x)
g is generator for group of order prime p

Enc((g,X), M, R)
r = R mod p
C1 = g^r
C2 = X^r * M
Return C1 , C2

Dec((g,x), C1, C2):
Return C2 * $C1^{-x}$

This is only at most chosen-plaintext attack secure. CCA attacks?

ElGamal KEM

Kg outputs pk = (g,X =g^x) and sk = (g,x)
g is generator for group of order prime p

EG-KEM((g,X), R)
r = R mod p
$C_{kem} = g^r$
$K = X^r$
Return H(K) , C_{kem}

Dec((g,x), C_{kem}):

Return $H(C_{kem}^x)$

Secure if computational Diffie-Hellman assumption holds in group