Public-Key Cryptography

Lecture 8

CCA Secure PKE

Hybrid Encryption

CCA Secure PKE

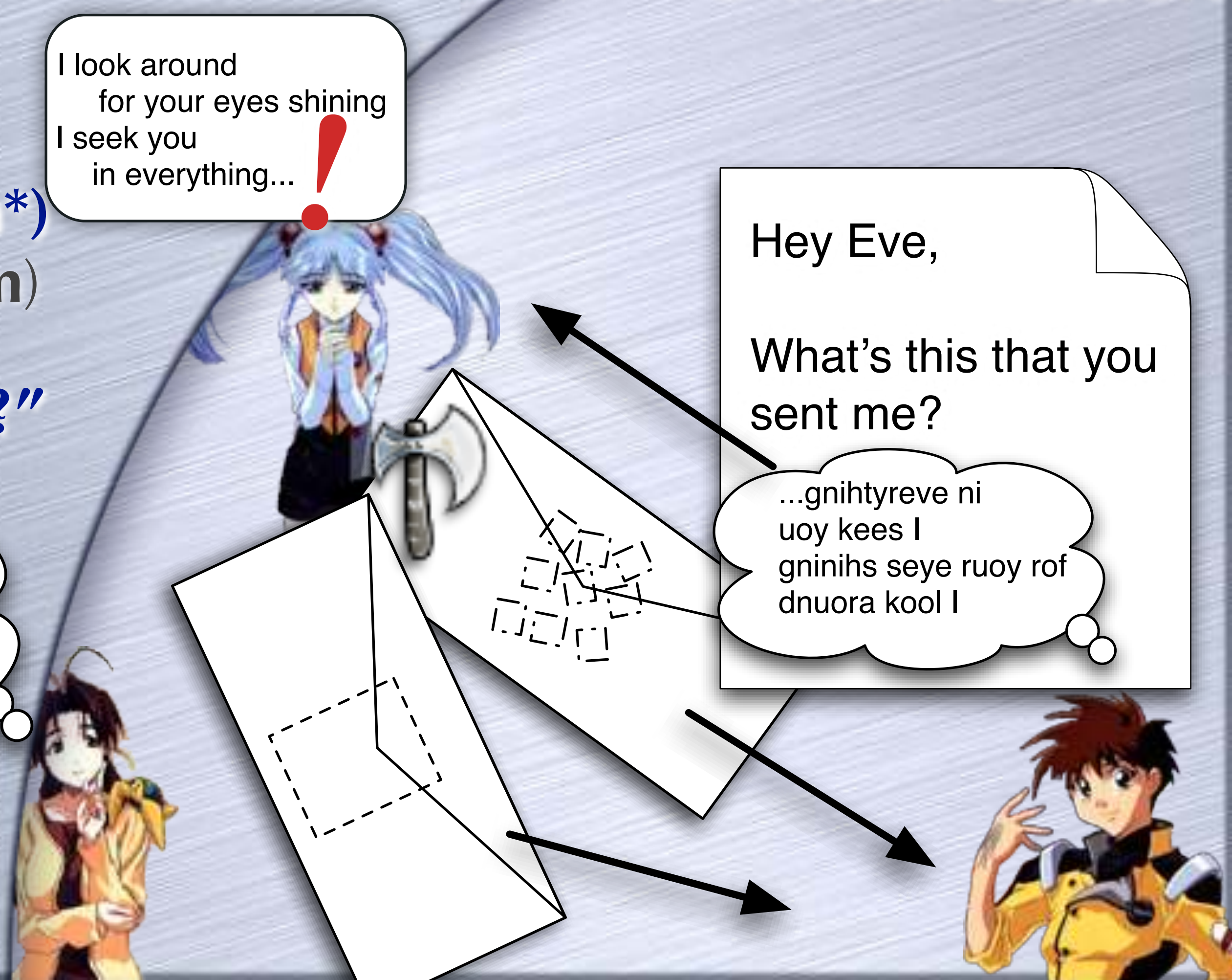
- In SKE, to get CCA security, we used a MAC
  - Bob would accept only messages from Alice
- But in PKE, Bob <u>wants to</u> receive messages from Eve as well!
  - But only if it is indeed Eve's "own message": she should "know" her own message!

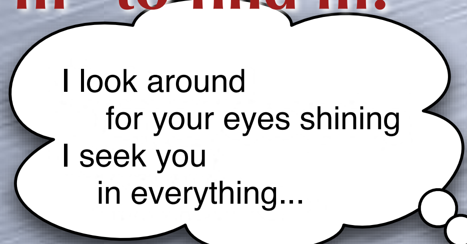
Chosen Ciphertext Attack

- Suppose Enc SIM-CPA secure
  - Suppose encrypts a character at a time (still secure)

A subtle e-mail attack

Alice → Bob: Enc(m)
Eve: Hack(Enc(m)) = Enc(m*)
(where **m*** = Reverse of **m**)
Eve → Bob: Enc(m*)
Bob → Eve: "what's this: m*?"
Eve: Reverse m* to find m!





Malleability

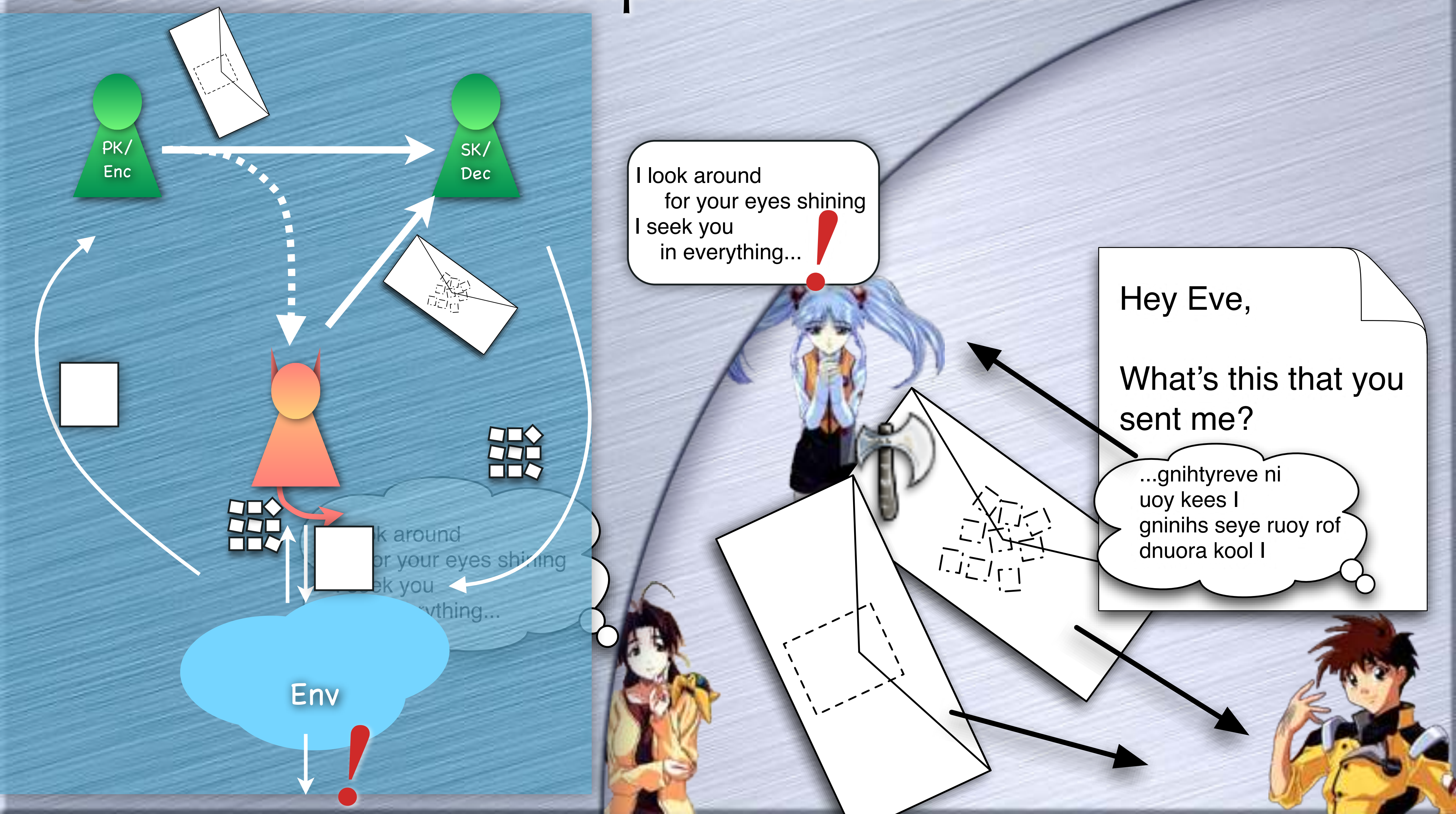
- Malleability: Eve can "malleate" a ciphertext (without having to decrypt it) to produce a new ciphertext that would decrypt to a "related" message

More subtly, the 1 bit - valid or invalid - may leak information on message or SK

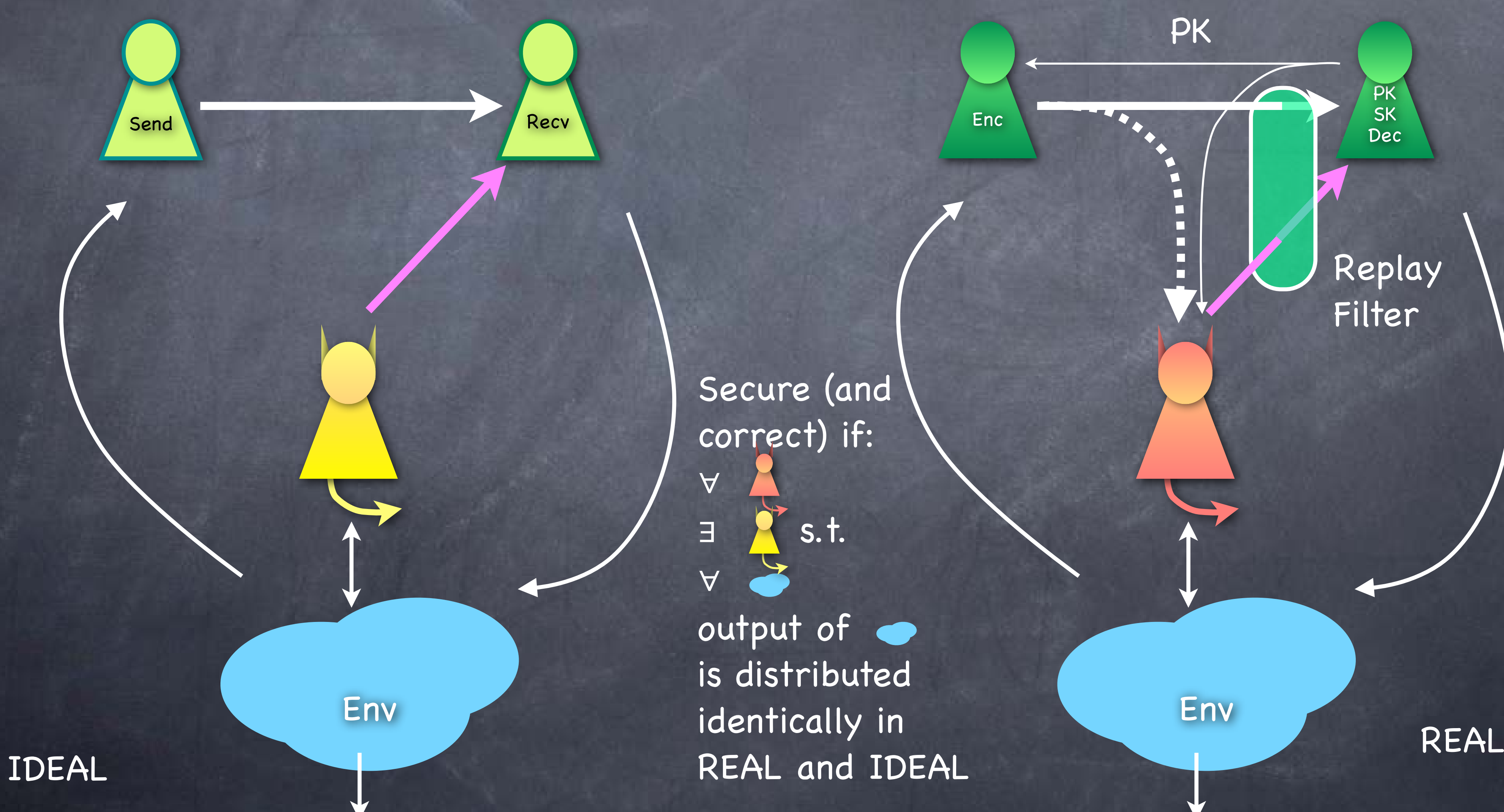
- E.g.: Malleability of El Gamal
  - Recall: $\mathrm{Enc}_{(G,g,Y)}(m) = (g^x, M.Y^x)$
  - Given (X,C) change it to (X,TC): will decrypt to TM
  - Or change (X,C) to (X^a, C^a): will decrypt to M^a
- If chosen-ciphertext attack possible
  - i.e., Eve can get a ciphertext of her choice decrypted
  - Then Eve can exploit malleability to learn something "related to" Alice's messages

Chosen Ciphertext Attack

- SIM-CCA: does capture this attack



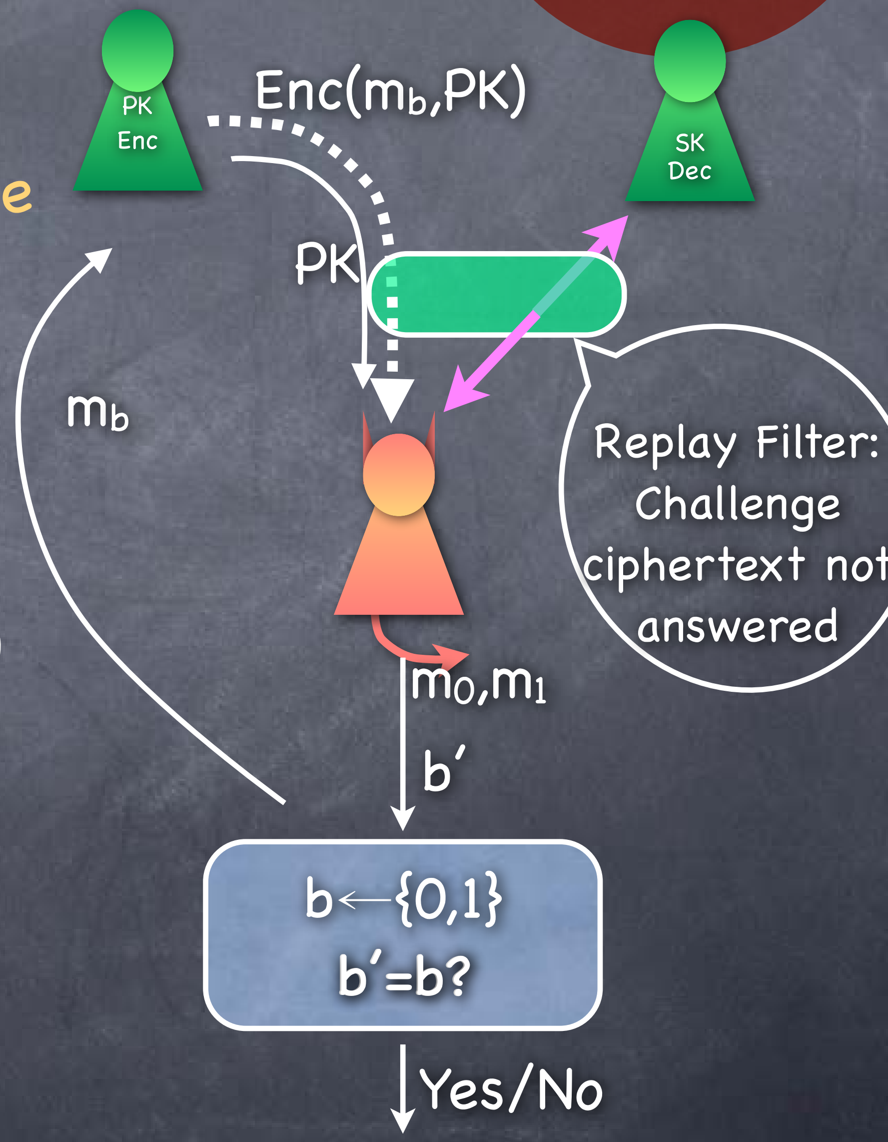
SIM-CCA Security (PKE)



IND-CCA (PKE version)

IND-CCA + ~correctness equivalent to SIM-CCA

- Expt picks a random bit b. It also runs KeyGen to get a key (PK,SK). Adv gets PK and (guarded) access to Dec_{SK} oracle
  - Adv sends two messages m_0, m_1 to Expt
  - Expt returns $Enc(m_b,K)$ to the adversary (and installs replay filter)
  - Adversary returns a guess b′
  - Experiment outputs 1 iff b′=b
- IND-CCA secure if for all PPT adversaries $\Pr[b'=b] - 1/2 \leq \nu(k)$



CCA Secure PKE Schemes

- Several schemes in the heuristic "Random Oracle Model"
  - RSA-OAEP
  - Fujisaki-Okamoto
  - DHIES (doesn't need the full power of ROM)
- Cramer-Shoup Encryption: Provably secure CCA scheme, under DDH assumption

RSA function

- $f_{RSA[N,e]}: \mathbb{Z}_N \rightarrow \mathbb{Z}_N$ defined as $f_{RSA[N,e]}(x) = x^e \pmod{N}$ where:
  - N is the product of two large primes, say N=PQ
  - $\gcd(e,\phi(N)) = 1$ where $\phi(N) = (P-1)(Q-1)$
    - Ensures that $\exists d$ s.t. $ed \equiv 1 \pmod{\phi(N)}$ and so $x^{ed} \equiv x \pmod{N}$
      - Can easily compute given $\phi(N)$ using Euclid's algorithm
    - $f_{RSA[N,d]}$ is the inverse of $f_{RSA[N,e]}$
- Smallest (and a common) choice for e is 3 (taking P-1 and Q-1 to be not multiples of 3)
  - However d would be a large number that is (believed to be) hard to find without knowing P, Q
- RSA Assumption: $f_{RSA[N,e]}$ is a OWF
  - Makes it a Trapdoor One-Way Permutation (trapdoor being d)

Random Oracle Model

- Random Oracle: a mythical oracle that, when initialized, picks a random function $R:\{0,1\}^* \rightarrow \{0,1\}^{n(k)}$ and when queried with x, returns R(x)
  - All parties have access to the same RO
- In ROM, evaluating some "hash function" H would be modeled as accessing an RO
  - Hope: the code for H has "no simple structure" and only way to get anything useful from it is to evaluate it on an input
- Sometimes security definitions need to be adapted for ROM
- Rigorous proofs of security, after moving to the ROM

Random Oracle Model

- **There is no Pseudo-RO**
  - Unlike PRF, RO must be locally evaluable for all parties. (think: giving out the seed of a PRF)
- There are schemes secure in ROM, such that for any instantiation of the RO, the scheme is insecure!
  - Also natural <u>constructs/primitives</u> which are realizable in ROM, but not in the standard model!
- What does a proof in ROM tell us?
  - Secure against attacks that treat H as a blackbox (and for which H is pseudorandom)

RSA-OAEP

- RSA-OAEP
  - "Text-book RSA encryption" (i.e., the Trapdoor OWP candidate f_{RSA}) applied to an "encoding" of the message
    - Encoding is randomized
    - Encoding uses a hash function modeled as a Random Oracle
    - Security in the RO Model, assuming f_{RSA} a OWP
  - Part of RSA Cryptography Standard (PKCS#1, since Ver 2.0, in 1998). Commonly used in SSL/TLS implementations

A Bit of RSA History

- In 1977 Rivest, Shamir, Adleman proposed using the RSA function directly as encryption ("text-book RSA encryption")
  - Being deterministic, it is not IND-CPA secure
- PKCS#1 V1.5 (1993) defined Enc(m;N,e) $\leftarrow$ $f_{RSA[N,e]}$(<header>||r||m), where r is a 0-terminated random byte sequence. Decryption returns error if $f_{RSA[N,d]}$(ciphertext) doesn't have the right format
  - Considered to be CPA secure
  - But is malleable: For $c = f_{RSA[N,e]}(pad(m))$ and $c' = s^e \cdot c$; decryption of c' (if not error) gives $s\cdot(pad(m))$
    - Considered only a theoretical concern in protocols like SSL, as it was not clear how a decryption oracle will be effected
  - Bleichenbacher (1998) showed that d can be recovered from access (a few million times) to the <u>decryption error</u> oracle, which was exposed by SSL

As we'll see, long-term encryption keys prevent "forward secrecy" and are not recommended by protocols like TLS 1.3. But they are unavoidable in applications like encrypted e-mail (S/MIME, OpenPGP, etc.)

CCA Secure PKE Schemes

- Several schemes in the heuristic "Random Oracle Model"
  - RSA-OAEP
  - Fujisaki-Okamoto
  - DHIES (doesn't need the full power of ROM)
- Cramer-Shoup Encryption: Provably secure CCA scheme, under DDH assumption (next time)

Hybrid Encryption schemes

Hybrid Encryption

- PKE is far less efficient compared to SKE (even in ROM)
  - RSA-OAEP uses modular exponentiations, DDH based schemes uses exponentiations in a group, etc.
  - SKE and MAC (e.g., using Block Ciphers like AES) are very fast
- Hybrid encryption: Use (CCA secure) PKE to transfer a key for the (CCA secure) SKE. Use SKE with this key for sending data
  - Hopefully the combination remains (CCA) secure
  - Note: PKE used to encrypt only a (short) key for the SKE
    - Relatively low overhead on top of the (fast) SKE encryption

Hybrid Encryption

Or to generate a key

- Hybrid Encryption: KEM/DEM paradigm
  - Key Encapsulation Method: a public-key scheme to transfer a key
  - Data Encapsulation Method: a symmetric-key scheme (using the key transferred using KEM)
- For what KEM/DEM is a hybrid encryption scheme CCA secure?
  - Works if KEM is a SIM-CCA secure PKE scheme and DEM is a SIM-CCA secure SKE scheme
    - Easy to prove using "composition" properties of the SIM definition
  - Less security sufficient: KEM used to transfer a random key; DEM uses a new key every time.

Another CCA Secure PKE: DHIES

- Diffie-Hellman Integrated Encryption Scheme
  - Part of some standards
- Essentially a hybrid scheme
  - Data Encapsulation: CPA secure SKE, and MAC
  - Key Encapsulation: $X=g^x$. Let $K=Y^x$, where Y is the PK (as in El Gamal), and (K_{SKE},K_{MAC}) = Hash(K) (where $K=Y^x=X^y$)
- CCA secure if Hash is modelled as a Random Oracle
  - Alternately, in the standard model, can be based on a complex (non-standard) assumption involving Hash and the group: "Oracle Diffie-Hellman Assumption"

Today

- CCA secure PKE
  - RSA-OAEP, Cramer-Shoup, DHIES, ...
- The Random Oracle model
- Hybrid Encryption: KEM/DEM
- Next up: Hash functions, Digital Signatures