Introduction to Cryptography CS 355

Lecture 25

Mental Poker And Semantic Security

Lecture Outline

- Review of number theory
- The Mental Poker Protocol
- Semantic security
- Semantic insecurity of RSA

Summary of Number Theory Results Covered

- Z_p^* is a cyclic group, has p-1 elements
  - has generators
- QR and QNR in Z_p^* can be easily determined by computing the Legendre symbol

$$\left(\frac{x}{p}\right) = x^{\frac{p-1}{2}} \bmod p$$

Summary of Number Theory Results Covered

- Jacobi symbol (generalizes Legendre symbol to composites)
  - can be computed without factoring n
  - Jacobi symbol does not determine QR in Z_n^*
- QR in Z_n^* is hard
- Computing square roots modulo n is as hard as factoring n
- Computing e'th root modulo n for $e \geq 3$ is believed to be as hard as factoring n

The Mental Poker Problem

- Alice and Bob want to play poker, we need a way to deal 5 cards to each of Alice and Bob so that
  - Alice's hand of 5 cards does not overlap with Bob's hand
  - Neither Alice nor Bob can control which cards they each get
  - Neither Alice nor Bob knows the other party's hand
  - Both hands should be random provided one party follows the protocol
- First solution due to Shamir, Rivest, and Adelman in 1980
  - uses commutative encryption schemes

Commutative Encryption

Definition: an encryption scheme is commutative if
$E_{K1}[E_{K2}[M]] = E_{K2}[E_{K1}[M]]$

- Given an encryption scheme that is commutative, then $D_{K1}[D_{K2}[E_{K1}[E_{K2}[M]] = M$
- Most symmetric encryption scheme (such as DES and AES) are not commutative

Examples of Commutative Encryption Schemes

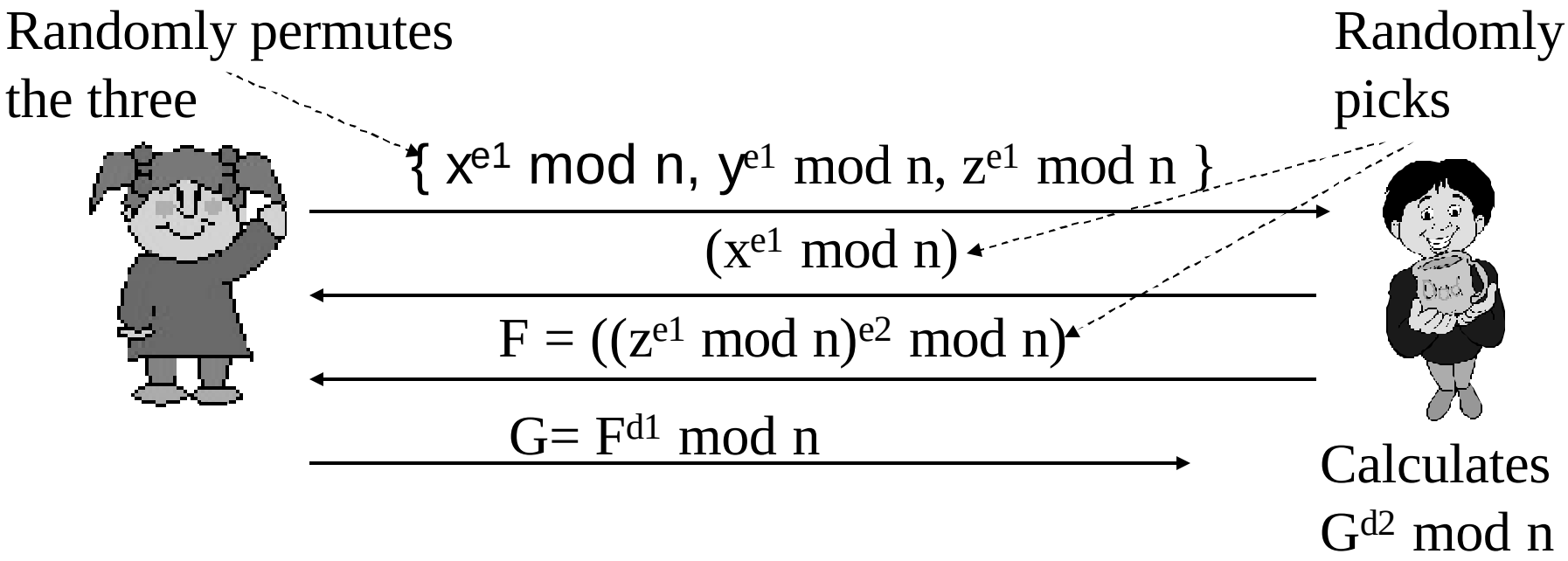
- Pohlig-Hellman Exponentiation Cipher with the same modulus p
  - encryption key is e, decryption key is d, where $ed \equiv 1 \pmod{(p-1)}$
  - $E_{e1}[M] = M^{e1} \bmod p$ and $D_{d1}[C] = C^{d1} \bmod p$
  - $E_{e1}[E_{e2}[M]] = M^{e1e2} = E_{e1}[E_{e2}[M]] \pmod p$

Examples of Commutative Encryption Schemes

- The SRA encryption scheme
  - Alice and Bob share n=pq and they both know p and q
  - Alice has encryption key e1 and decryption key d1 s.t. $e1 \bullet d1 = 1 \pmod{(p-1)(q-1)}$
    - $E_{e1}[M] = M^{e1} \pmod{n}$
  - Bob has e2, d2 s.t. $e2 \bullet d2 = 1 \pmod{(p-1)(q-1)}$
  - Also a commutative encryption scheme
  - Essentially RSA, except that e is kept private

A Simple Example with Two Cards

- Let x, y, and z denote three cards, Alice and Bob wants to each randomly picks a card without the other one knowing which one



The SRA Mental Poker Protocol

Setup: Alice and Bob share M_1, M_2, …, M_{52} denote the 52 cards, n=pq, p, and q. Alice has e1,d1 and Bob has e2,d2

Protocol:

- Alice encrypts M_1, M_2, …, M_{52} using her key, i.e., computes $C_j=M_j^{e1}$ mod n for $1 \leq j \leq 52$, randomly permute them and send the ciphertexts to Bob
- Bob picks 5 cards as Alice's hand and sends them to Alice
- Alice decrypts them to get his hand
- Bob picks 5 other cards as his hand, encrypts them using his key, and sends them to Alice
- Alice decrypts the 5 ciphertexts and sends to Bob
- Bob decrypts what Alice sends and gets his hand
- Both Alice and Bob reveals their key pairs to the other party and verify that the other party was not cheating. (Why need this step?)

“Security Analysis” of the Protocol

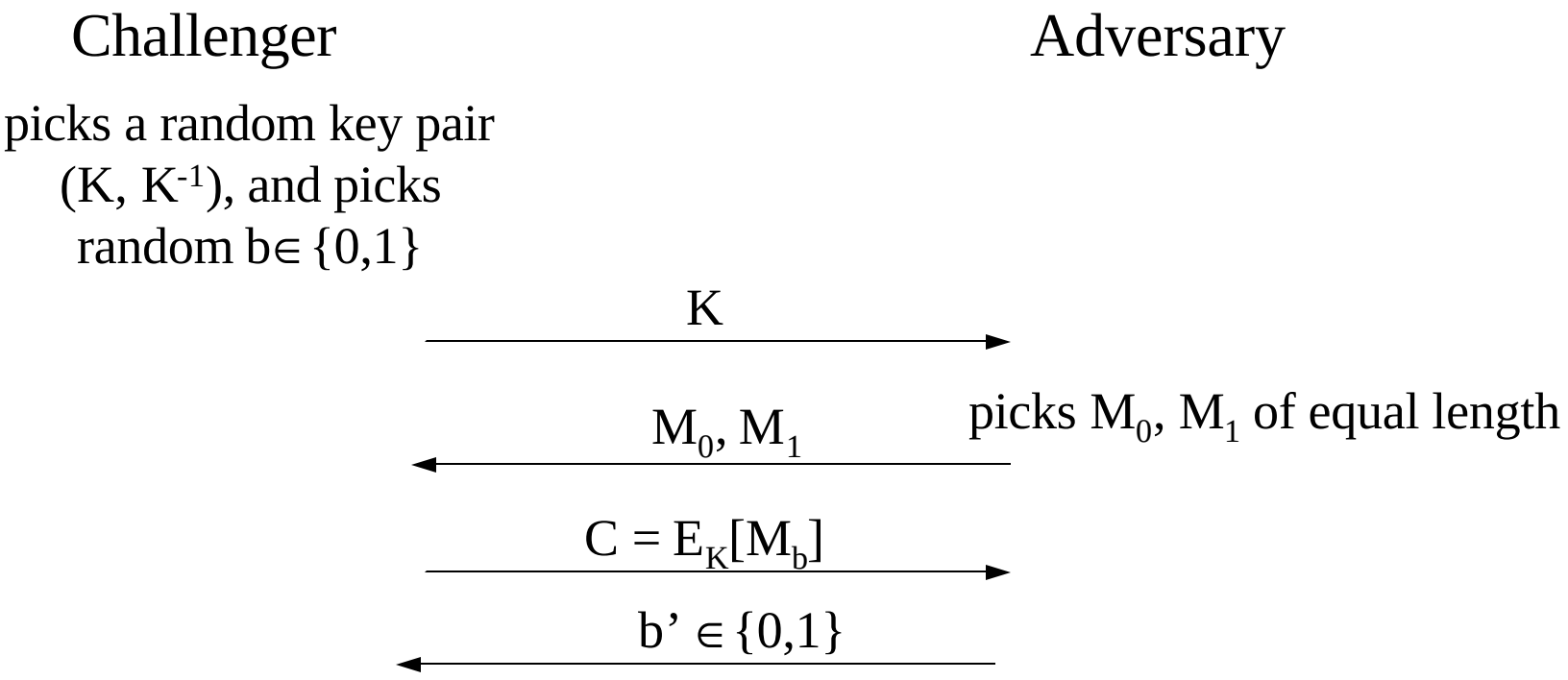
- Bob sees 52 random ciphertexts, he doesn’t know which ciphertext corresponds to which cards.
- Bob can only randomly pick Alice’s hand, and Bob does not know what Alice’s hand is.
- Bob can only randomly pick his hand, and Alice doesn’t know Bob’s hand, as it is encrypted under Bob’s key.

An Attack on the SRA Mental Poker Protocol

- The encryption function $f(x)=x^e \bmod n$ leaks information about x !
  - f(x) is QR modulo n iff. x is QR modulo n
  - $x^e \in QR_n \Leftrightarrow x^e \in QR_p$ and $x^e \in QR_q \Leftrightarrow x \in QR_p$ and $x \in QR_q$
    $\Leftrightarrow x \in QR_n$
  - Why this matters in the SRA mental poker protocol?
    - suppose that the cards that are QR are mostly large cards, and the cards that are not QR are mostly small cards, then Bob can choose large cards for him and small cards for Alice
  - Even when f(x) is a trapdoor one-way function, some bits about x can be leaked.

Semantic Security (IND-CPA for Public Key Encryption)

- The IND-CPA game



Attacker wins game if b=b'

Semantic Insecurity of the RSA

- RSA encryption is not semantically secure because it is deterministic
- In particular, the encryption function $f(x)=x^e \bmod n$ leaks information about x !
  - it leaks the Jacobi symbol of x

$$\left(\frac{x^e}{N}\right)=\left(\frac{x^e}{p}\right)\left(\frac{x^e}{q}\right)=\left(\frac{x}{p}\right)\left(\frac{x}{q}\right)=\left(\frac{x}{N}\right)$$

  - it also leaks the whether x is a QR or not, but this is not a concern, why?

Coming Attractions …

- El Gamal Encryption
- The Blum-Blum-Shun pseudo-random sequence generator