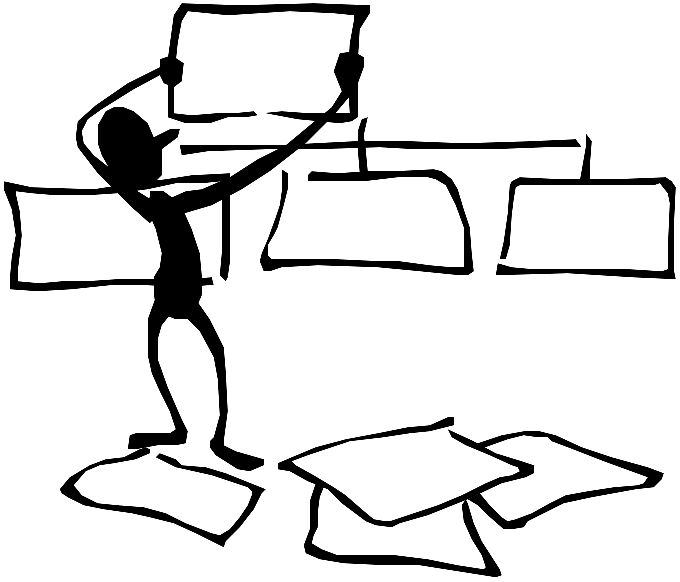
Introduction to Cryptography CS 355

Lecture 26

El Gamal

Lecture Outline

- ElGamal Encryption



ElGamal

- Published in 1985 by ElGamal
- Its security based on the intractability of the discrete logarithm problem and the CDH and DDH problems
- Message expansion: the ciphertext is twice as big as the original message
- Uses randomization, each message has many different possible ciphertexts

El Gamal

- Public key is (p; g; $\beta=g^a$ mod p)
  - p is a large random prime number such that DLP is infeasible in Z_p
  - g is a generator g of the multiplicative group Z_p^*
  - a is a random integer in [1..p-2]
- Private key is a.
- The ciphertext of M is (g^k mod p, $M\beta^k$ mod p)
  - k is randomly chosen such that $0 \le k \le p$-2
- How to decrypt?

Parameters Size

- All parties could use the same modulus *p* and generator g
  - they choose different private key a, and will have different β's
- Different encryptions should use different *k*
- Prime p should be chosen as 1024 bits to ensure that DLP is infeasible, while k should be 160 bits
- ElGamal encryption can also be defined in cyclic groups other than Z_p*
  - e.g., in elliptic curves

Security of ElGamal

- ElGamal is not semantically secure.
- WHY? An attacker can learn information about the plaintext without decrypting: given two encryptions, can say which plaintext was a quadratic residue and which one was not.

Making ElGamal Semantically Secure

- Main idea: Use only quadratic residues in the operation
  - Choose p such that $p = 2q + 1$, where q is also prime
  - Then define ElGamal in Q_q, the subgroup of quadratic residues modulo p, this subgroup is a cyclic subgroup of Z_p having order q

ElGamal and DH Problems

- Semantic security of the ElGamal algorithm (where we use only QRs) is equivalent to the infeasibility of Decision Diffie-Hellman
- ElGamal decryption (without knowing the private key) is equivalent to solving Computational Diffie-Hellman

Coming Attractions …

- Integrity/Authenticity
- Hash functions