CPSC/PMAT 669

Randomized Encryption, ElGamal, and Semantic Security

Mike Jacobson

Department of Computer Science
University of Calgary

Topic 6

Outline

1. Probabilistic Encryption
2. The El Gamal PKC
3. Provable Security Under Passive Attacks

Other Public-Key Cryptosystems

Need other cryptosystems whose security relies on problems other than factoring. Eg.

- Rabin, Rabin-Williams: computing square roots modulo pq
- El Gamal: discrete logarithms
- Merkle-Hellman: subset-sum problem (1st realization of PKC, but insecure)
- Chor-Rivest: secure subset-sum based PKC
- McEliece: decoding linear error-correcting codes
- XTR: subgroup of a finite field
- NTRU: shortest vector in a lattice

Provable Security

Provable security is also important:

- provide proof that breaking the cryptosystem (for particular adversarial goals and capabilities) reduces to a computational problem believed to be hard (eg. Diffie-Hellman key exchange and DLP)
- would like equivalence when possible (eg. Diffie-Hellman key exchange and DHP, Rabin and square roots mod pq)

Next topic: more formal notions of security for PKC

- try to achieve computational analogue of perfect security under strong attack model (CCA2)

Probabilistic Encryption

One disadvantage of deterministic PKCs is that identical messages always encrypt to the same ciphertext (like block ciphers in ECB mode).

- particularly problematic if the message space is small (*e.g.* electronic yes/no vote)

Probabilistic or *randomized encryption* utilizes randomness to attain a provable, stronger level of security.

As a result, every message can have many possible encryptions, so a small message space is no longer a problem.

- leads to the notion of *semantic* security.

The ElGamal PKC

Randomized, security based on DLP (alternative to RSA which was based on IFP)

Set-up: the designer produces her public and private keys as follows:

1. Selects a large prime p and a primitive root g of p
2. Computes $y = g^x \pmod{p}$ where $0 < x < p - 1$.

Public key: $\{p, g, y\}$
Private key: $\{x\}$

ElGamal Encryption

Messages for the designer are integers M, $0 < M < p$ (so $M \in \mathbb{Z}_p^*$).

To send M encrypted, proceed as follows:

1. Select a random $k \in \mathbb{Z}$, $0 < k < p$.
2. Compute and send (C_1, C_2) where

$$\begin{aligned} C_1 &\equiv g^k \pmod{p}, \quad 0 < C_1 < p, \\ C_2 &\equiv My^k \pmod{p}, \quad 0 < C_2 < p \ . \end{aligned}$$

ElGamal Decryption

To decrypt (C_1, C_2), the designer computes

$$\begin{aligned} C_2 C_1^{p-1-x} &\equiv (My^k)(C_1^{p-1-x}) \\ &\equiv (Mg^{xk})(g^{k(p-1-x)}) \\ &\equiv Mg^{xk+k(p-1)-kx} \\ &\equiv M(g^{p-1})^k \\ &\equiv M \pmod{p} \ . \end{aligned}$$

Think of C_1 as a “clue” that can be used to remove the “mask” y^k in C_2, thus “unmasking” the encrypted message M.

Summary of ElGamal

As with DH key establishment, the security of this system relies on the presumed difficulty of the DLP, but it is unknown whether there are other ways of breaking ElGamal.

Disadvantages:

- Message expansion by a factor of 2 (ciphertext is twice as long as the plaintext).
- Twice as much computational work for encrypting as RSA:
  - two exponentiations (and one multiplication), as opposed to one exponentiation only for RSA.
- A new random number k must be generated for each message.

Advantages: different security assumption, works in other settings (eg. elliptic curves)

Polynomial Security

Goal: public-key cryptosystems that

- computational security: best-known attack involves solving a hard problem (eg. RSA, El Gamal)
- provable security: breaking a particular security property reduces to a problem believed to be difficult (eg. factoring, DLP)

Definition 1 (Polynomial security, IND-CPA security)

A PKC is said to be *polynomially secure* or *IND-CPA secure* if no passive adversary can in expected polynomial time select two plaintexts M_1 and M_2 and then correctly distinguish between encryptions of M_1 and M_2 with probability significantly greater than $1/2$.

IND-CPA: indistinguishability under chosen plaintext attacks.

Semantic Security

Definition 2 (Semantic security)

A PKC is said to be *semantically secure* if for all probability distributions over the message space, anything that can be computed by a passive adversary in expected polynomial time about the plaintext given the ciphertext can also be computed in expected polynomial time without the ciphertext.

Intuitively, semantic security is a weaker version of perfect security

- an adversary with polynomially-bounded computational resources (as opposed to infinite resources in perfect security) can learn nothing about the plaintext from the ciphertext.

Equivalance

Theorem 1

A PKC is semantically secure if and only if it is polynomially secure.

Idea of Proof.

Use contrapositive: prove that a PKC that is not polynomially secure is not semantically secure (and conversely). □

Although El Gamal is randomized, it is *not* semantically secure as presented here (Assignment 3).

We will soon look at a PKC that is semantically secure assuming that a certain number theoretic problem (not DLP or IFP) is hard. But first, we need a bit more number theory.