**6.875J/18.425J Cryptography and Cryptanalysis** Mar 9, 2016

# Lecture 11: Semantically Secure Public-Key Encryption I

*Instructor: Shafi Goldwasser* *Scribes: Vahid Fazel-Rezai, Daniel Richman, Connor Sell*

## 1 RSA preprocessing

## 2 Random Oracle Model Methodology

## 3 Types of Attacks

- Lunchtime attack
- Timing attack
- Power attack
- Fault attack
- Cache attack

## 4 Public key crypto based on squaring

## 5 Nondeterministic public key crypto with trapdoor functions

## 6 Trapdoor predicates

## 7 Using single-bit encryption for arbitrary length encryption

## 8 Homomorphic encryption

**Definition 1.** *An encryption scheme is* ***homomorphic*** *if computations on the ciphertexts are reflected as computations on the messages when decrypted.*

Symbollically, given an encryption function $E$ and messages $m_1$ and $m_2$, this means

$$E(m_1) \circ E(m_2) = E(m_1 \diamond m_2).$$

Note that the operation on the cyphertexts and the messages can be different operations. Example applications of homomorphic encryption schemes:

- E-voting: all votes could be encrypted and include a 0 or 1 indicating the vote. The ciphertexts of the votes could be added and then decrypted, yielding the vote count without revealing individual votes.
- Secure cloud computing: data could be encrypted and have ciphertexts operated on (both $+$ and $\times$) without revealing the data itself. The result would then be decrypted and used.

Specifically with the quadratic residuosity encryption scheme, we have the following homomorphic properties:

- $E(m_1 \oplus m_2) = E(m_1) \cdot E(m_2)$ (can be checked with truth table)
- $E(1 \oplus m) = E(1) - E(m)$
- $E(m) = E(0) \cdot E(b)$ (effectively re-randomizing)

# 9 Probabilistic encryption scheme and examples

## 9.1 Main Idea

Given a trapdoor permutation collection $F$, we define an encryption scheme as follows:

- The key generation function $\mathsf{Gen}(1^k)$ simply chooses a $f \in F$ and its corresponding trapdoor $t$, and outputs $(f, t)$.
- The encryption function $\mathsf{Enc}(f, m)$ chooses a seed $r$ in the domain of $f$ and a PSRG $g$ based on $f$. It returns $c = (c_1, c_2) = (g(r) \oplus m, f^{|m|+1}(r))$.
- The decryption function $\mathsf{Dec}(t, (c_1, c_2))$ has access to the trapdoor. It first finds $r = f^{-(|m|+1)}(c_2)$ (by inverting over and over) then returns $m = c_1 \oplus g(r)$.

The security of this scheme follows from the assumption of a PSRG.

## 9.2 Example: RSA

The probabilistic approach can be applied to RSA as follows:

- $\mathsf{Gen}(1^k)$ is defined as choosing $(n, e)$ just as in RSA.
- $\mathsf{Enc}(n, m)$ is defined by choosing $r \in Z_n^*$ and concatenating $|m|$ bits computed by

  $$\text{pad} = \text{lsb}(r \bmod n) \quad \text{lsb}(r^e \bmod n) \quad \text{lsb}(r^{e^2} \bmod n) \quad \cdots \quad \text{lsb}(r^{e^{|m|-1}} \bmod n).$$

  We then set $c = (\text{pad} \oplus m, r^{|m|})$.
- $\mathsf{Dec}((p, q), (c_1, c_2))$ decrypts by finding $r$ as the $|c_1|$th root of $c_2$ modulo $n$ (using the factorization $n = pq$). Then, it can recompute pad as above and find $m = c_1 \oplus \text{pad}$.

## 9.3 Example: El Gamal

The El Gamal Cryptosystem is based on the discrete log problem and takes advantage of probabilistic encryption, defined as follows:

- $\mathsf{Gen}(1^k)$ chooses a random $k$-bit prime $p$ such that $p = 2q + 1$, where $q$ is also prime. Let $g$ be a generator of $QR_p$, $x$ be a number with $1 < x < q$, and $y = g^x \mod p$. Publish $(p, g, y)$ as the public key and keep the $x$ that was used secret.
- $\mathsf{Enc}((p, g, y), m)$ (where $m \in QR_p$) is defined by choosing randomly $1 \leq r \leq q$, computing $\text{pad} = y^r = g^{xr} \mod p$, and yielding $c = (\text{pad} \cdot m \mod p, g^r)$.
- $\mathsf{Dec}(x, (c_1, c_2))$ is able to decrypt the cipher by recomputing the pad as $\text{pad} = c_2^x = g^{rx} \mod p$ and finding $m = c_1 \cdot \text{pad}^{-1} \mod P$. by finding $r$ as the $|c_1|$th root of $c_2$ modulo $n$ (using the factorization $n = pq$). Then, it can recompute pad as above and find $m = c_1 \oplus \text{pad}$.

Note that $g$ and $p$ can be shared across all the users as long as $x$ and therefore $y$ are chosen differently for each key generation.

This scheme has, for message size $|m| = k$, public key of size $O(k)$, bandwidth of $O(k)$, and both encryption and decryption running time of $O(k^3)$. We also have security:

**Theorem 2.** *Under DDH, El Gamal is computationally indistinguishable.*

El Gamal also has multiplicative homomorphism. That is, if $\mathsf{Enc}(m) = (c_1, c_2)$ and $\mathsf{Enc}(m') = (c_1', c_2')$, we have $\mathsf{Enc}(m \cdot m') = (c_1 c_1' \mod p, c_2 c_2' \mod p)$.

Furthermore, we can modify the scheme to also have additive homomorphism as follows. In encrypting, instead of returning $c_1 = \text{pad} \cdot m \mod p$, we set $c_1 = \text{pad} \cdot g^m \mod p$. With this modification, multiplying $g^m \cdot g^{m'} = g^{m+m'}$ effectively adds $m + m'$. To decrypt, as long as $m$ is a member of a polynomial size known set, can try all possibilities for $g^m$ and choose the one that matches.

### 9.4 Example: Paillier

Another example of an encryption scheme that uses randomness is as follows:

- $\mathsf{Gen}(1^k)$ chooses a $n = pq$, where $p$ and $q$ are primes. It publishes $n$ and keeps $\phi(n)$ secret.
- $\mathsf{Enc}(n, m)$ (assuimg $m \in Z_n^*$) chooses a random $r \in Z_n^*$ and computes

$$c = (1+n)^m r^n \mod n^2.$$

- $\mathsf{Dec}((p, q), c)$ first computes

$$\begin{aligned} c' &= c^{\phi(n)} \mod n^2 \\ &= (1+n)^{m\phi(n)} r^{n\phi(n)} \mod n^2 \\ &= (1+n)^{m\phi(n)} \mod n^2 \\ &= 1 + nm\phi(n) \mod n^2, \end{aligned}$$

  from which we can find $m = \frac{c'-1}{n\phi n}$.

Note that the last step of decryption follows from the fact $(1+n)^t = 1 + tn + n^2(\cdots) = 1 + tn \mod n^2$ for any $t$.

The Paillier encryption scheme is used in applications such as auctions and voting due to its homomorphic properties: if $\mathsf{Enc}(n, m) = c$ and $\mathsf{Enc}(n, m') = c'$, then $\mathsf{Enc}(n, m + m' \mod n) = c \cdot c'$ and $\mathsf{Enc}(n, m - m' \mod n) = c/c'$.

The security of the scheme is guaranteed under the Decisional Composite Residuosity (DCR) assumption.

**Definition 3.** *The Decisional Composite Residuosity (DCR) assumption states that it is hard to distinguish between $(n, R^n)$ and $(n, S)$ for random $R \in Z_n$ and $S \in Z_{n^2}$.*

With DCR, Paillier is computationally indistinguishable against a passive adversary.

## References