Notes for Lecture 19

Summary

Today we continue to discuss number-theoretic constructions of CPA-secure encryption schemes.

First, we return to the Decision Diffie Hellman assumption (the one under which we proved the security of the El Gamal encryption scheme) and we show that it fails for $\mathbb{Z}_p^*$, although it is conjectured to hold in the subgroup of *quadratic residues* of $\mathbb{Z}_p^*$.

Then we introduce the notion of *trapdoor permutation* and show how to construct CPA-secure public-key encryption from any family of trapdoor permutations. Since RSA is conjectured to provide a family of trapdoor permutations, this gives a way to achieve CPA-secure encryption from RSA.

1 Decision Diffie Hellman and Quadratic Residues

Recall that if $\mathbb{D}$ is a distribution over triples $(\mathbb{G}, g, q)$, in which $\mathbb{G}$ is a cyclic group of q elements and g is a generator, then $\mathbb{D}$ satisfies the (t, ϵ) Decision Diffie Hellman Assumption if for every algorithm A of complexity $\leq t$ we have

$$|\mathbb{P}[A(\mathbb{G}, g, q, g^x, g^y, g^z) = 1] - \mathbb{P}[A(\mathbb{G}, g, q, g^x, g^y, g^{x \cdot y}) = 1]| \leq \epsilon \qquad (1)$$

where $\mathbb{G}, g, q$ are distributed as in $\mathbb{D}$, and x, y, z are integers uniformly distributed in $\{0, \ldots, q-1\}$.

In Lecture 17, we gave the group $\mathbb{Z}_p^*$, p prime, as an example of cyclic group for which the discrete logarithm problem is conjectured to be hard. The Decision Diffie Hellman assumption, however, is always false for groups of the type $\mathbb{Z}_p^*$.

To see why, we need to consider the notion of *quadratic residue* in $\mathbb{Z}_p^*$. An integer $a \in \{1, \ldots, p-1\}$ is a quadratic residue if there exists $r \in \{1, \ldots, p-1\}$ such that $a = r^2 \bmod p$. (In such a case, we say that r is a *square root* of a.) For every odd primes p, exactly $(p-1)/2$ of the elements of $\mathbb{Z}_p^*$ are quadratic residues, and each has two square roots. Furthermore, there is an efficient algorithm (polynomial in the number of digits of a) to check whether a is a quadratic residue. This fact immediately gives an algorithm that contradicts (1) if we take $\mathbb{G}$ to be $\mathbb{Z}_p^*$, when $\epsilon < 1/4$.

Note, however, that the set $\mathbb{Q}_p$ of quadratic residues of $\mathbb{Z}_p^*$ is itself a cyclic group, and that if g is a generator of $\mathbb{Z}_p^*$ then $g^2 \bmod p$ is a generator of $\mathbb{Q}_p$.

It is believed that if p is a prime of the form $2q+1$, where q is again prime then taking $\mathbb{G} = \mathbb{Q}_p$ and letting g be any generator of $\mathbb{G}$ satisfies (1), with t and $1/\epsilon$ exponentially large in the number of digits of q.

2 Trapdoor Permutations and Encryption

A *family of trapdoor permutations* is a triple of algorithms (G, E, D) such that:

1. $G()$ is a randomized algorithm that takes no input and generates a pair (pk, sk), where pk is a *public key* and sk is a *secret key*;
2. E is a deterministic algorithm such that, for every fixed public key pk, the mapping $x \to E(pk, x)$ is a bijection;
3. D is a deterministic algorithm such that for every possible pair of keys (pk, sk) generated by $G()$ and for every x we have

$$D(sk, E(pk, x)) = x$$

That is, syntactically, a family of trapdoor permutations is like an encryption scheme except that the "encryption" algorithm is deterministic. A family of trapdoor permutations is secure if inverting $E()$ for a random x is hard for an adversary that knows the public key but not the secret key. Formally,

Definition 1 *A family of trapdoor permutations (G, E, D) is (t, ϵ)-secure if for every algorithm A of complexity $\leq t$*

$$\mathop{\mathbb{P}}_{(pk,sk)\leftarrow G(),x}[A(pk, (E(pk, x))) = x] \leq \epsilon$$

It is believed that RSA defines a family of trapdoor permutations with security parameters t and $1/\epsilon$ that grow exponentially with the number of digits of the key. Right now the fastest factoring algorithm is believed to run in time roughly $2^{O(k^{1/3})}$, where k is the number of digits, and so RSA with k-bit key can also be broken in that much time. In 2005, an RSA key of 663 bits was factored, with a computation that used about 2^{62} elementary operations. RSA with keys of 2048 bits may plausibly be $(2^{60}, 2^{-30})$-secure as a family of trapdoor permutations.

In order to turn a family of trapdoor permutations into a public-key encryption scheme, we use the notion of a *trapdoor predicate*.

Definition 2 *Let (G, E, D) be a family of trapdoor permutations, where E takes plaintexts of length m. A boolean function $P : \{0,1\}^m \to \{0,1\}$ is a (t, ϵ)-secure trapdoor predicate for (G, E, D) if for every algorithm A of complexity $\leq t$ we have*

$$\mathop{\mathbb{P}}_{(pk,sk)\leftarrow G(),\ x}[A(pk, E(pk, x)) = P(x)] \leq \frac{1}{2} + \epsilon$$

Remark 3 *The standard definition is a bit different. This simplified definition will suffice for the purpose of this section, which is to show how to turn RSA into a public-key encryption scheme.*

Essentially, P is a trapdoor predicate if it is a hard-core predicate for the bijection $x \to E(pk, x)$. If (G, E, D) is a secure family of trapdoor permutations, then $x \to E(pk, x)$ is one-way, and so we can use Goldreich-Levin to show that $\langle x, r\rangle$ is a trapdoor predicate for the permutation $E'(pk, (x, r)) = E(pk, x), r$.

Suppose now that we have a family of trapdoor permutations (G, E, D) and a trapdoor predicate P. We define the following encryption scheme (G', E', D') which works with *one-bit* messages:

- $G'()$: same as $G()$
- $E'(pk, b)$: pick random x, output $E(pk, x), P(x) \oplus b$
- $D'(pk, (C, c)) := P(D(pk, C)) \oplus c$

Theorem 4 *Suppose that P is (t, ϵ)-secure trapdoor predicate for (G, E, D), then (G', E', D') as defined above is $(t - O(1), 2\epsilon)$-secure public-key encryption scheme.*

The encryption scheme described above is only able to encrypt a one-bit message. Longer messages can be encrypted by encrypting each bit separately. Doing so, however, has the undesirable property that an ℓ bit message becomes an $\ell \cdot m$ bit ciphertext, if m is the input length of P and $E(pk, \cdot)$. A "cascading construction" similar to the one we saw for pseudorandom generators yields a secure encryption scheme in which an ℓ-bit message is encrypted as a cyphertext of length only $\ell + m$.