Theory of Cryptography | **Lecture 21** | **Instructor: Chris Peikert**
Identity-Based Encryption | **Scribe: Pushkar Tripathi**

1 Identity-Based Encryption

Shamir, 1984: "What if your name could be your public key?"

Boneh and Franklin, 2001: "It can!"

Cocks, 2001: "I did it years ago . . . (again)."

Encryption in which "your name is your public key" is called Identity-Based Encryption (IBE). Shamir's original motivation for identity-based encryption was to simplify the management of public keys. Frequently, this process can be unwieldy because of the following problem: if Bob obtains Alice's public key over the network, how does Bob know that the key really belongs to Alice, and not to a malicious adversary-in-the-middle who replaced Alice's key with his own as it traveled over the network? The usual solution is to rely upon "certificate authorities" (CAs), who are trusted to implement the following process: when Alice creates a key pair, she presents her public key pk_A and some valid form of identification to the CA. The CA then generates a signature, under its own verification key vk_{CA}, attesting to the statement "this public key pk_A belongs to Alice," which Alice appends to her own public key. Later on, Bob can verify the CA's signature and gain confidence that the key really is Alice's. Of course, in order to do this Bob needs to know the CA's true public key (and to trust that the CA implemented its policy correctly), so at first sight it seems that we have gained very little. However, there need only be a few CAs in the world, and they can make their keys widely known through alternative means, such as bundling with common cryptographic software or widespread publication. Still, this whole process is rather heavyweight: every time a person generates a new public key, she must register that key with a CA, and Bob must store and manage a large number of keys for all the people with whom he corresponds. (Things become even more complicated when we introduce *revocation*, where Alice may desire for her key to be declared invalid if, for example, it becomes compromised.)

Identity-based encryption can simplify the above scenario in the following way: for a particular administrative domain (e.g., `domain.com`), there is an "authority" who generates a "master" public/secret key pair for the entire domain. When Bob wants to send a message to Alice at `alice@domain.com`, he simply encrypts using the public key string `alice` and the master public key for the domain (which should be certified as usual). There is no need for Bob to obtain Alice's individual public key or certificate, or even for Alice to generate a key at all! Meanwhile, Alice identifies herself to the domain authority, who uses its master secret key to "extract" a secret key that works just for the identity `alice`. Alice uses this to decrypt her message as usual. Notice here that the authority implicitly knows secret keys for every user in the system, so it can read everyone's encrypted messages. Depending on the scenario, this can be a blessing or a curse, but in any case authority's master secret key is very powerful, so it must be guarded carefully.

IBE also provides additional useful properties like implicit revocation (via identities that have an "expiration date" appended to them) and delegation (only a subset of messages can be decrypted by a particular agent). More generally, IBE-related techniques and implications have proved to be very versatile and powerful in the design of other important and seemingly unrelated protocols.

The problem of constructing an IBE remained open for many years following its initial conception by Shamir. Finally in 2001, Boneh and Franklin described a scheme that used a relatively new mathematical object called a bilinear map (which has since been shown to have numerous other related applications). Soon after Boneh and Franklin's announcement, it was revealed that Clifford Cocks, a mathematician in the United Kingdom's cryptography agency GCHQ, had years earlier devised a simple IBE based on a standard and elementary assumption (but his scheme was classified by the UK government). This was an interesting case of "history repeating itself," as Cocks himself had also discovered the famed RSA algorithm in 1974, three years before Rivest, Shamir and Adelman, but this discovery was also classified!

In this lecture we will present the Cocks IBE scheme.

1.1 Model

The model for IBE consists of an authority that generates a single master public key mpk for all its users, along with a master secret key msk. To encrypt a message, one needs to know the master public key mpk and the receiver's identity id, which can be any string. A user with identity id obtains a secret key sk_{id} from the authority (typically after proving his identity in some way). He can use this secret key to decrypt messages sent to him.

Formally, an IBE scheme consists of four ppt algorithms:

- $\mathsf{Setup}(1^n)$ outputs a master public key mpk and master secret key msk. This algorithm is executed once by the authority when setting up the system.
- $\mathsf{Ext}(msk, id)$ outputs a secret key for every user based on his id. Since only the authority should have access to msk, this algorithm is executed only by the authority after checking a user's identity.
- $\mathsf{Enc}(mpk, id, m)$ outputs a ciphertext corresponding to a message m.
- $\mathsf{Dec}(sk_{id}, c)$ decrypts a ciphertext and outputs a message.

1.2 Security Definition

Here we present a formal definion of the notion of IND-CPA security for an IBE scheme. The intuition behind this definition is that a message encrypted to a particular "target" identity id^* (chosen by the adversary) should remain secret, *even if* the adversary gets the secret keys for a polynomial number of other identities that are different from id^*.

Definition 1.1. An IBE scheme is IND-CPA-secure if:

$$\left\langle mpk, \mathsf{E}(\cdot), C^0(id^*, m_0, m_1)\right\rangle \overset{c}{\approx} \left\langle mpk, \mathsf{E}(\cdot), C^1(id^*, m_0, m_1)\right\rangle,$$

where $(mpk, msk) \leftarrow \mathsf{Setup}$, and $\mathsf{E}(id)$ returns $\mathsf{Ext}(msk, id)$, except if queried on id^* in which case it returns $\perp$. Furthermore, for an id^* different from all prior queries to E, the oracle $C^b(id^*, m_0, m_1)$, returns $\mathsf{Enc}(mpk, id^*, m_b)$, defines id^* as the target identity, and terminates.

The definition can be strengthened in various ways, such as adding chosen-ciphertext security. In such a definition the adversary would additionally get a decryption oracle for the identity id^* that it is allowed to query on any string except for the challenge ciphertext that was output by C^b.

2 Number Theory Background

Throughout this lecture we will let N be a *Blum integer*, i.e., $N = pq$ for primes p and q that are both 3 modulo 4. We use $\mathbb{QR}_N^*$ to denote the set of quadratic residues modulo N. We use $\left(\frac{a}{p}\right) = a^{p-1/2} = \pm 1 \bmod p$ to denote the Legendre symbol. (The symbol itself is defined as an integer, not a quantity modulo p).

A natural generalization of the Legendre symbol is the *Jacobi symbol*, defined as $\left(\frac{a}{pq}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{a}{q}\right)$. Note that $\left(\frac{-1}{p}\right) = \left(\frac{-1}{q}\right) = -1$, hence $\left(\frac{-1}{N}\right) = 1$ for a Blum integer N. That is, $-1 \in \mathbb{J}_N^*$ but is a non-square.

Also note that $\left(\frac{ab}{N}\right) = \left(\frac{a}{N}\right) \cdot \left(\frac{b}{N}\right)$, by the multiplicative property of the Legendre symbol. Finally, we define $\mathbb{J}_N^* = \left\{a \in \mathbb{Z}_N^* | \left(\frac{a}{N}\right) = 1\right\}$, which is a multiplicative subgroup of $\mathbb{Z}_N^*$. The following claim can be established easily.

Claim 2.1. *Given a and N (in binary), we can efficiently compute $\left(\frac{a}{N}\right)$ without knowing the factoring of N.*

Proof. The following *reciprocity laws* can be proved with some effort. We omit their proofs.

$$\text{for odd } a \text{ and } N, \quad \left(\frac{a}{N}\right) = \left(\frac{N}{a}\right) \cdot (-1)^{\frac{a-1}{2}\cdot\frac{N-1}{2}} \tag{2.1}$$

$$\text{for any } N, \quad \left(\frac{-1}{N}\right) = (-1)^{\frac{N-1}{2}} \tag{2.2}$$

$$\left(\frac{2}{N}\right) = (-1)^{\frac{N^2-1}{8}} \tag{2.3}$$

For any even number a, we can reduce the problem to the case when a is odd by using **??** 1. Using **??** 1, we can iteratively reduce a or N until one of them becomes eauls to 1, when $\left(\frac{a}{N}\right)$ can be calculated trivially. The overall algorithm and its complexity is analogous to Euclid's algorithm for finding the gcd of two numbers. □

We also present another claim that would be useful in the following section.

Claim 2.2. $|\mathbb{QR}_N^*| = \frac{1}{2}|\mathbb{J}_N^*|$.

Proof. Let $a \in \mathbb{Z}_N^*$. Note that $a \in \mathbb{QR}_N^*$ iff $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = 1$, so $|\mathbb{QR}_N^*| = (p-1)(q-1)/4$. Also, $a \in \mathbb{J}_N^*$ iff $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = \pm 1$, so $|\mathbb{J}_N^*| = \frac{(p-1)(q-1)}{2}$. □

The security of Cocks' IBE is based on the following assumption which was proposed in the seminal work of Goldwasser and Micali on public-key encryption.

Conjecture 2.3 (Quadratic Residuosity Assumption). For a modulus generator S,

$$\langle N \leftarrow S(1^*), a \leftarrow \mathbb{QR}_N^* \rangle \overset{c}{\approx} \langle N \leftarrow S(1^*), a \leftarrow \mathbb{J}_N^* \setminus \mathbb{QR}_N^* \rangle$$

In words, the assumption says that given N, it is hard to distinguish between a random quadratic residue and a random quadratic nonresidue *with Jacobi symbol* 1. Note that it is crucial that the element have Jacobi symbol 1, otherwise the tuples could be distinguished trivially by computing the Jacobi symbol $\left(\frac{a}{N}\right)$ using Claim 2.1.

2.1 Interlude: Goldwasser-Micali Encryption Scheme

As a warm-up to the IBE, we briefly recall the *original* application of the quadratic residuosity assumption to standard public-key encryption. The Goldwasser-Micali scheme for message space $\{0, 1\}$ is defined as follows.

- $\mathsf{Gen}(1^n)$: Generate a Blum integer $pk = N$ as a public key with known factorization $sk = (p, q)$.
- $\mathsf{Enc}(N, b \in \{0, 1\})$: Choose $r \leftarrow \mathbb{Z}_N^*$ and output the encryption of b as $c = r^2 \cdot (-1)^b \bmod N$.

- $\mathsf{Dec}(sk = (p,q), c)$: Return $m = 0$ if $c \in \mathbb{QR}_N^*$, and return 1 otherwise. This test can be done efficiently by testing whether both $\left(\frac{c}{p}\right) = 1$ and $\left(\frac{c}{q}\right) = 1$.

The proof of security follows directly from Conjecture 2.3, which yields the following theorem.

Theorem 2.4. *Under QRA, the Goldwasser-Micali scheme is IND-CPA-secure.*

3 The Cocks IBE Scheme

In this section we define Cocks' IBE scheme and prove its security. To begin, we give a basic public key encryption scheme that works relative to a modulus N, but where decryption can be performed *without needing the factorization of* N. Instead, decryption will only require knowledge of a square root of a user's public key. Looking ahead, the factorization of N will be the *master* secret key of the system, which will allow the authority to extract a secret key for any user. Later in Section 3.2, we will convert this PKE into an IBE scheme.

3.1 Warm-Up: Public-Key Encryption Scheme

The PKE is for the message space $\{\pm 1\}$, and is defined by the following algorithms, which are all implicitly provided with the modulus N.

- $\mathsf{Gen}(1^n)$: Generate secret key $sk = r \leftarrow \mathbb{Z}_N^*$ and public key $pk = r^2 = a \in \mathbb{QR}_N^* \subseteq \mathbb{Z}_N^*$.
- $\mathsf{Enc}(a, m \in \{\pm 1\})$: Choose uniformly random $t \in \mathbb{Z}_N^*$ such that $\left(\frac{t}{N}\right) = m$. (This can be done by repetition, since the probability of success for a random choice of t is $1/2$.) Output the ciphertext $c = t + a/t \bmod N$.
- $\mathsf{Dec}(r, c)$: Output $\left(\frac{c+2r}{N}\right)$.

From inspection it is not immediately apparent that decryption is correct, but the following claim establishes that this is so.

Claim 3.1. *For any* $r \in \sqrt{a} \bmod N$ *and any message* $m \in \{\pm 1\}$, *we have* $\mathsf{Dec}(r, \mathsf{Enc}(a, m)) = m$.

Proof. Let t be the random value chosen by Enc where $\left(\frac{t}{N}\right) = m$. The claim follows from the equations given below.

$$\begin{aligned}\left(\frac{c+2r}{N}\right) &= \left(\frac{t + r^2/t + 2r}{N}\right) \cdot \left(\frac{t}{N}\right)^2 \\ &= \left(\frac{t^2 + r^2 + 2rt}{N}\right) \cdot \left(\frac{t}{N}\right) \\ &= \left(\frac{(t+r)^2}{N}\right) \cdot \left(\frac{t}{N}\right) \\ &= \left(\frac{t}{N}\right).\end{aligned}$$

□

In the remainder of this section we will argue that the above scheme is IND-CPA secure under the QRA by showing that if the public key $pk = a$ is instead an element of $\mathbb{J}_N^* \setminus \mathbb{QR}_N^*$, then the encryption algorithm Enc is "lossy," i.e., the distributions of $\mathsf{Enc}(a, +1)$ and $\mathsf{Enc}(a, -1)$ are *identical*. The following claim establishes this lossy property. Following the claim, we easily prove that this lossiness property implies IND-CPA security.

Claim 3.2. *If $pk = a \in \mathbb{J}_N^* \setminus \mathbb{QR}_N^*$, then $\mathsf{Enc}(a, +1) \equiv \mathsf{Enc}(a, -1)$.*

Proof. Note that $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1$. Let $c = t + a/t$ (for some $t \in \mathbb{Z}_N^*$) be some fixed ciphertext that could be output by $\mathsf{Enc}(a, \cdot)$. For this fixed c and public key a, consider all solutions t to this equation $c = t + a/t$. Let $t = t_0 \in \mathbb{Z}_N^*$ be an arbitrary such solution. Written using the Chinese remainder representation of $\mathbb{Z}_N^*$, there are a total of four solutions:

- $\langle t_0 \bmod p; t_0 \bmod q\rangle$ has Jacobi symbol $\left(\frac{t_0}{N}\right)$.
- $\langle a/t_0 \bmod p; t_0 \bmod q\rangle$ has Jacobi symbol $-1 \cdot \left(\frac{t_0}{N}\right)$, because $\left(\frac{a/t_0}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{t_0}{p}\right)^{-1} = -1\left(\frac{t_0}{p}\right)$.
- $\langle t_0 \bmod p; a/t_0 \bmod q\rangle$ has Jacobi symbol $-1 \cdot \left(\frac{t_0}{N}\right)$, for similar reasons.
- $\langle a/t_0 \bmod p; a/t_0 \bmod q\rangle$ has Jacobi symbol $\left(\frac{t_0}{N}\right)$.

Hence, $\Pr\left[\mathsf{Enc}(a, 1) = c\right] = \Pr\left[\mathsf{Enc}(a, -1) = c\right]$, and the proof is complete. □

Corollary 3.3. *Under the QRA, the basic Cocks public-key encryption scheme is IND-CPA secure.*

Proof. By the QRA, composition lemma, and Claim 3.2, we have the following:

$$\begin{aligned}\langle N, a \leftarrow \mathbb{QR}_N^*, \mathsf{Enc}(a, +1)\rangle &\overset{c}{\approx} \langle N, a \leftarrow \mathbb{J}_N^* \setminus \mathbb{QR}_N^*, \mathsf{Enc}(a, +1)\rangle \\ &\equiv \langle N, a \leftarrow \mathbb{J}_N^* \setminus \mathbb{QR}_N^*, \mathsf{Enc}(a, -1)\rangle \\ &\overset{c}{\approx} \langle N, a \leftarrow \mathbb{QR}_N^*, \mathsf{Enc}(a, -1)\rangle.\end{aligned}$$

□

3.2 Making It Identity-Based

We make the scheme identity-based in the random oracle model. Let $H\colon \{0,1\}^* \to \mathbb{J}_N^*$ be modelled as a truly random function, which will maps identities to public keys in the basic scheme. (Note that such a function can be obtained from a random function $H'\{0,1\}^* \to \{0,1\}$ by using the random output bits of H' to sample from $\mathbb{J}_N^*$.) There is only one subtlety: we cannot detect whether the output of the hash function H is a "proper" or "lossy" public key. The solution is to encrypt (using independent randomness) to both $a_0 = H(id)$ and $a_1 = -H(id)$, both of which have Jacobi symbol 1, and exactly one of which is a quadratic residue (because $-1 \in \mathbb{J}_N^* \setminus \mathbb{QR}_N^*$).

- $\mathsf{Setup}(1^n)$ chooses a Blum integer $mpk = N = pq$ as the master public key, and lets $msk = (p, q)$ be the master secret key.
- $\mathsf{Ext}^H(msk, id)$ lets $a_0 = H(id)$ and $a_1 = -1 \cdot H(id)$. It returns $r_{id} \in \mathbb{Z}_N^*$ as a *random* square root of a_0 or a_1, whichever is a quadratic residue.

- $\mathsf{Enc}^H(mpk = N, id, m \in \{\pm 1\})$ lets $a_0 = H(id)$ and $a_1 = -1 \cdot H(id)$, and outputs $c_0 = \mathsf{Enc}(a_0, m)$ and $c_1 = \mathsf{Enc}(a_1, m)$ where $\mathsf{Enc}(\cdot, \cdot)$ is the encryption algorithm of the basic scheme as defined in Section 3.1
- $\mathsf{Dec}(sk_{id} = \sqrt{a_b}, c = (c_0, c_1))$ outputs $\mathsf{Dec}(sk_{id}, c_b)$, where $\mathsf{Dec}(\cdot, \cdot)$ is the decryption algorithm of the basic scheme as defined in Section 3.1.

We now provide an outline of the proof of security for the above scheme; a good exercise is to fill in the details carefully. In order to prove security, we need to give a simulator $\mathcal{S}(N, a)$ that attempts to solve the QR problem using an adversary for the IBE scheme. The simulator must provide an msk, and answer oracle queries $H(\cdot)$ and extraction queries $\mathsf{Ext}_{msk}(\cdot)$ for arbitrary identities (but without knowing the msk!). The basic idea is that $\mathcal{S}$ sets $msk = N$, and "programs" the random oracle H so that it knows a random square root for $\pm H(id)$ for all but a single $H(id^*) := \pm a$, where id^* is a randomly chosen query to H that $\mathcal{S}$ "guesses" will be the adversary's declared target identity. If $\mathcal{S}$ does happen to guess this identity correctly, then it prepares a challenge ciphertext by using the basic scheme to encrypt two *opposite* bits in random order (i.e., either $+1, -1$ or $-1, +1$) under a and $-a$, respectively. Note that by the lossiness property, such a ciphertext is distributed identically as either $\mathsf{Enc}(id^*, +1)$ or $\mathsf{Enc}(id^*, -1)$, depending on whether a or $-a$ is a quadratic residue. Based on whether the adversary identifies the ciphertext as an encryption of $+1$ or -1, then, the simulator answers whether $a \in \mathbb{QR}^*_N$ or not.