Cryptography and Security Protocols **IST-UL, 2024/2025**

More on public-key encryption

Lecturer: João Ribeiro

Introduction

In the last lecture we introduced the notion of public-key encryption (PKE) and saw how to adapt Diffie-Hellman key agreement into a PKE scheme (yielding "ElGamal encryption"). In these notes we will discuss additional PKE schemes.

1 RSA PKE

Recommended reading: Katz-Lindell, Sections 11.5.1–11.5.3 and 8.2.4.

We begin by discussing one of the oldest widely used PKE schemes, called *RSA* for their "public" inventors in 1977, Rivest, Adleman, and Shamir [RSA78]. This PKE scheme was actually developed (discovered?) earlier by Clifford Cocks at GCHQ in 1973, but this was kept classified until the late 90s.

The security of the RSA PKE scheme hinges on the "RSA assumption", which bears some connections to the problem of factoring integers (i.e., given an integer, find its prime factorization). In order to discuss this, we must first introduce the necessary concepts from number theory.

1.1 Some basic number theory

Fix an integer N. Recall that $\mathbb{Z}_N^*$ denotes the multiplicative group modulo N. More precisely,

$$\mathbb{Z}_N^* = \{x \in \{1, \ldots, N-1\} : \ x \text{ and } N \text{ are coprime}\},$$

and the group operation is multiplication modulo N. Recall that two integers a and b are called *coprime* if they do not share any prime factors (i.e., their greatest common divisor is 1). We define

$$\phi(N) = |\mathbb{Z}_N^*|.$$

In words, $\phi(N)$ is the number of positive integers smaller than N that are coprime to N. This is called *Euler's totient function*. It has many nice properties which you are welcome to explore. We record here some that are directly useful to us:

- When $N = pq$ for distinct primes p and q, we have $\phi(N) = (p-1)(q-1)$.

- For any $x \in \mathbb{Z}_N^*$, we have
$$x^e = x^{e \pmod{\phi(N)}} \pmod{N}.$$
In particular, $x^{\phi(N)} = 1 \pmod{N}$.
- For an integer e, define the function $f_e : \mathbb{Z}_N^* \to \mathbb{Z}_N^*$ as $f_e(x) = x^e \pmod{N}$. Then, f_e is a bijection when e is coprime to $\phi(N)$, and f_d with $d = e^{-1} \pmod{\phi(N)}$ is its inverse.

For a more detailed discussion of these concepts, see Section 8.1.4 of Katz-Lindell.

1.2 An insecure version of RSA PKE

We begin by presenting a "plain" version of RSA PKE with deterministic encryption (and hence not CPA-secure). Then, we discuss how this scheme can be modified to achieve CPA-security under the "RSA assumption".

- To generate a public-key/secret-key pair, $\mathsf{Gen}(1^n)$ proceeds as follows: It samples n-bit primes p and q and sets $N = pq$. Then, it samples some e coprime with $\phi(N)$ and computes $d = e^{-1} \pmod{\phi(N)}$. The public key is $pk = (N, e)$ and the secret key is $sk = (N, d)$.
- To encrypt a message $m \in \mathbb{Z}_N^*$, we compute the ciphertext $c = m^e \pmod{N}$.
- To decrypt a ciphertext c, we compute $m = c^d \pmod{N}$.

This PKE scheme is correct, since if $c = m^e \pmod{N}$ then

$$c^d = (m^e)^d = m^{ed} = m^{ed \pmod{\phi(N)}} = m \pmod{N},$$

since $ed = 1 \pmod{\phi(N)}$.

It is clear that this scheme is not CPA-secure, since encryption is deterministic.

It is instructive to reflect at an intuitive level on the hardness assumption required for this scheme to satisfy at least some basic form of security. The adversary knows the public key $pk = (N, e)$ and sees the ciphertext $x^e \pmod{N}$. Therefore, intuitively, it should be infeasible for an efficient adversary to recover x based on this information. This motivates the *RSA game*, defined next.

Definition 1 (RSA game) *The* RSA game with respect to Gen *is the following game between an adversary $\mathcal{A}$ and a challenger, parameterized by a security parameter n:*

1. *The challenger samples $(N, e, d) \leftarrow \mathsf{Gen}(1^n)$ and $x \leftarrow \mathbb{Z}_N^*$. They compute $c = x^e \pmod{N}$ and send (N, e, c) to $\mathcal{A}$.*
2. *$\mathcal{A}(1^n, N, e, c)$ outputs x' and wins if $x' = x$.*

The *RSA assumption* (with respect to Gen) is then simply the assumption that for any PPT adversary $\mathcal{A}$ there exists a negligible function $\varepsilon(n)$ such that $\mathcal{A}$ wins the RSA game with respect to Gen with probability at most $\varepsilon(n)$. This assumption is required for the RSA PKE scheme to be secure, but it is certainly not sufficient, at least in its current form. For example, the RSA assumption means that it is hard to recover a *uniformly random* message, but we would like to encrypt arbitrary messages.

1.3 CPA-secure RSA PKE

We now discuss ways to get more secure PKE schemes from the RSA assumption.

One possible modification is to first pad the message m (seen as a bitstring) with a sufficiently long string of uniformly random bits. Variants of this approach have been used in some implementations of RSA PKE, and the security depends, for example, on the length of the padding.

In this section we will focus on a particularly clean way of getting CPA-secure encryption based on the RSA assumption. It hinges on the following theorem, which we will not prove. Intuitively, it states that, under the RSA assumption, it is not only hard to recover a uniformly random x, but it is actually already hard to recover the least significant bit of x, denoted $\mathrm{lsb}(x)$.

Theorem 1 (hardcore predicate for RSA) *Suppose that the RSA assumption holds with respect to* Gen*. Then, for any PPT adversary* $\mathcal{A}$ *there exists a negligible function* $\varepsilon(n)$ *such that the probability that the output* x' *of* $\mathcal{A}$ *in the RSA game with respect to* Gen *satisfies* $\mathrm{lsb}(x') = \mathrm{lsb}(x)$ *is at most* $\frac{1}{2} + \varepsilon(n)$*.*

Using Theorem 1, we can get a CPA-secure PKE scheme based on the RSA assumption in a manner similar to ElGamal PKE. We will present it for the special case of encrypting single-bit messages.

- $\mathsf{Gen}(1^n)$ samples $pk = (N, e)$ and $sk = (N, d)$ as above.
- To encrypt a bit $b \in \{0, 1\}$, we sample $x \leftarrow \mathbb{Z}_N^*$ and output the ciphertext

$$(x^e \pmod N, \mathrm{lsb}(x) \oplus b).$$

- To decrypt a ciphertext c, we parse it as $c = (c_1, c_2)$, compute $x = c_1^d \pmod N$ and $b = \mathrm{lsb}(x) \oplus c_2$.

Using Theorem 1, the following theorem can be proved via the same strategy that we employed to establish the analogous result for the ElGamal PKE scheme.

Theorem 2 (CPA-secure PKE from the RSA assumption) *The PKE scheme defined above is CPA-secure under the RSA assumption with respect to* Gen*.*

1.4 RSA assumption vs. hardness of factoring

The RSA assumption is a bit convoluted. It is instructive to compare it with hardness assumptions based on cleaner and more basic problems. One such problem is *factoring*: Given an integer N, our goal is to find its prime factorization. This problem has been studied for thousands of years, but we still do not have efficient classical algorithms.

The RSA assumption is *stronger* than the assumption that factoring *semi-prime*[1] integers is hard (for an appropriate formalization of this hardness assumption). If there is an efficient algorithm that given $N = pq$ outputs (p, q) with probability α, then there is an efficient adversary $\mathcal{A}$ that wins the RSA game with probability at least α. Given $(N, e, c = x^e \pmod N)$, the adversary $\mathcal{A}$ first uses the algorithm above to factor N into (p, q). Then, $\mathcal{A}$ efficiently computes $\phi(N) = (p-1)(q-1)$ and $d = e^{-1} \pmod \phi(N)$, after which it efficiently recovers $x = c^d \pmod N$.

We still do not know whether the RSA assumption is strictly stronger than the hardness of factoring. However, we do have some partial results. For example, we know that these assumptions are equivalent for "generic" algorithms [AM09].

2 PKE from noisy linear algebra

Variants of the RSA and ElGamal PKE schemes we saw are widely used in practice. However, their security is based on the hardness of problems that are easy for quantum computers. As mentioned before, there has been a significant recent effort (by NIST and other organizations) to migrate our infrastructure to *post-quantum* PKE schemes whose security is based on the hardness of problems conjectured to be hard for quantum computers.

We will now discuss a more recent PKE scheme due to Regev [Reg09], whose security is based on the hardness of *Learning With Errors* (LWE) problem.

2.1 The Learning With Errors problem

We begin by discussing the Learning With Errors (LWE) problem, also introduced by Regev, which can be interpreted as the problem of solving systems of *noisy* linear equations. The following definition presents a more formal description of the problem.

Definition 2 (Search Learning With Errors problem) *The* search Learning With Errors *(search-LWE) problem parameterized by positive integers n, m, q and an error distribution χ on $\mathbb{Z}_q$, denoted by* $\mathsf{Search\text{-}LWE}_{n,m,q,\chi}$*, corresponds to the following search problem:*

1. *Sample a uniformly random matrix $A \in \mathbb{Z}_q^{n \times m}$, a uniformly random secret $s \in \mathbb{Z}_q^m$, and an error vector $e = (e_1, \ldots, e_n)$ with each e_i sampled independently according to χ.*
2. *Given A and $As + e \pmod q$, find s.*

[1] An integer N is *semi-prime* if $N = pq$ for primes p, q.

The above definition presents the search version of LWE, where we are tasked with finding the input secret s. It is also useful to consider the natural decision version of this problem, where we wish to distinguish $As + e \pmod q$ from a uniformly random vector.

Definition 3 (Decision Learning With Errors problem) *The* decision Learning With Errors *(decision-LWE) problem parameterized by positive integers n, m, q and an error distribution χ on $\mathbb{Z}_q$, denoted by* Decision-LWE$_{n,m,q,\chi}$*, corresponds to the following distinguishing game played between an adversary and a challenger:*

1. *The challenger samples a uniformly random matrix $A \in \mathbb{Z}_q^{m \times n}$, a uniformly random secret $s \in \mathbb{Z}_q^n$, and an error vector $e = (e_1, \ldots, e_m)$ with each e_i sampled independently according to χ.*
2. *The challenger also samples a challenge bit b uniformly from $\{0, 1\}$. If $b = 0$, then the challenger sends $(A, As + e \pmod q)$ to the adversary. Otherwise, if $b = 1$ then the challenger samples $u \leftarrow \mathbb{Z}_q^m$ and sends (A, u) to the adversary.*
3. *The adversary outputs b' and wins if $b' = b$.*

Search-LWE is at least as hard as Decision-LWE (in fact, both problems have essentially equivalent hardness, as you will show in the problem set), and both problems are easily solved via Gaussian elimination if no errors are introduced. A bit informally, the (decision) LWE assumption (with respect to n, m, q, χ) is that for every PPT adversary $\mathcal{A}$ there is a negligible function $\varepsilon(nm \log q)$ such that $\mathcal{A}$ solves Decision-LWE$_{n,m,q,\chi}$ with probability at most $\frac{1}{2} + \varepsilon(nm \log q)$. For usual parameter settings m and q are polynomial in n, and so the problem input size is also polynomial in n.

Usual parameters and error distribution for LWE. It is instructive to discuss the usual parameter and error distribution regime for LWE. The modulus q is usually taken to be polynomial in n. The error distribution is usually taken to be a mean-zero gaussian distribution rounded to the nearest integer and then reduced modulo q (a so-called *discrete gaussian*) with variance $q/\mathsf{poly}(n)$. One property that is often exploited in LWE-based cryptographic schemes is that the error vector has norm $\ll q$ with high probability, and so can be reliably rounded-off in crucial steps of the protocol.

LWE-based protocols have large keys because we need to set $m > n \log q$. This has led researchers to consider more structured versions of LWE, such as the ring-LWE problem [SSTX09, LPR10], which yield more efficient cryptographic protocols under less standard hardness assumptions.

We note also that LWE and ring-LWE are sometimes also studied with other error distributions, such as binary errors sampled uniformly at random from $\{0, 1\}$ [MP13].

2.2 Regev PKE from LWE

The conjectured hardness of Decision-LWE can be used to construct PKE schemes with post-quantum security.[2] We will discuss Regev's PKE scheme [Reg09], which is secure provided that Decision-LWE (with an appropriate parameterization) is hard to solve by PPT adversaries.

We describe the scheme for single-bit messages. For this scheme to be correct (i.e., for us to be able to correctly decrypt ciphertexts), we need to use an error distribution χ that generates short error vectors with high probability. We discuss this in more detail below, and remark that usual instantiations of $\mathsf{Decision\text{-}LWE}_{n,m,q,\chi}$ believed to be hard have this property. For the security proof we will additionally require that $m \geq (n+1)\log q$, which is also fine from a hardness perspective. Below, we interpret $\mathbb{Z}_q$ as the set $\{-\frac{q-1}{2}, \ldots, 0, \ldots, \frac{q-1}{2}\}$, as it leads to a cleaner discussion.

- **Key generation:** Sample a secret $s \leftarrow \mathbb{Z}_q^n$, a matrix $A \leftarrow \mathbb{Z}_q^{m \times n}$, and an error vector $e = (e_1, \ldots, e_m)$ with each e_i sampled independently according to an error distribution χ over $\mathbb{Z}_q$. Let $w = As + e \pmod q \in \mathbb{Z}_q^m$. Then, we set the secret key sk and public key pk as

$$sk = s \quad \text{and} \quad pk = A' = [A \mid w] \in \mathbb{Z}_q^{m \times (n+1)}.$$

- **Encryption:** To encrypt a bit $b \in \{0,1\}$ using $pk = A'$, sample $x \leftarrow \{0,1\}^m$ and compute the ciphertext

$$c = \mathsf{Enc}(b, pk) = x^T A' + \left(0^n, b \cdot \left\lceil \frac{q}{2} \right\rceil\right) \pmod q.$$

- **Decryption:** To decrypt c using the secret key $sk = s$, we first compute

$$\begin{aligned} c \begin{bmatrix} s \\ -1 \end{bmatrix} &= x^T(As - w) - b \cdot \left\lceil \frac{q}{2} \right\rceil \pmod q \\ &= -x^T e - b \cdot \left\lceil \frac{q}{2} \right\rceil \pmod q. \end{aligned}$$

  Since $x \in \{0,1\}^m$, by the triangle inequality we have that

$$|x^T e| \leq \sum_{i=1}^{m} |e_i| = \|e\|_1.$$

  Therefore, if $\|e\|_1 = \sum_{i=1}^m |e_i| < q/4$ then we can recover b by checking whether the computation above yields a value closer to 0 or $q/2$.

We prove the following.

Theorem 3 *The PKE scheme defined above is CPA-secure if* $\mathsf{Decision\text{-}LWE}_{n,m,q,\chi}$ *is hard to solve by PPT adversaries and* $m \geq (n+1)\log q$.

[2] Other high-profile applications of LWE include fully homomorphic encryption [BV11] and indistinguishability obfuscation [JLS21].

Proof: We follow a hybrid argument. Recall that the goal is to show that a PPT adversary $\mathcal{A}$ can only win the following CPA-security game with probability at most $1/2 + \varepsilon(n)$ for some negligible function $\varepsilon(n)$:

1. The challenger samples keys $(sk, pk) \leftarrow \mathsf{Gen}(1^n)$ and a bit $b \leftarrow \{0,1\}$, and sends the public key pk and the ciphertext $c = \mathsf{Enc}(b, pk)$ to the adversary $\mathcal{A}$;
2. $\mathcal{A}$ outputs $b' \leftarrow \mathcal{A}(1^n, c, pk)$ and wins if $b' = b$.

The first hybrid H_0 corresponds exactly to this CPA-security game played between the challenger and $\mathcal{A}$. The second hybrid H_1 is the same game as H_0 with the exception that the public key $pk = A' \in \mathbb{Z}_q^{(m+1)\times n}$ is now sampled uniformly at random from $\mathbb{Z}_q^{(m+1)\times n}$ and independently of sk. The final hybrid H_2 is like H_1 with the exception that the ciphertext c is also sampled uniformly at random and independently of everything else.

Note that in H_2 the adversary's input is independent of the message bit b. Therefore, the winning probability for any adversary in H_2 is exactly $1/2$. As a result, if we show that the "transcripts" of H_0 and H_2 are indistinguishable to the eyes of any PPT algorithm, then this implies that no PPT adversary $\mathcal{A}$ can win H_0 with probability better than $1/2 + \mathsf{negl}(n)$ (because otherwise the winning event of $\mathcal{A}$ could be used to distinguish between H_0 and H_2). We argue this in steps, first by showing that the transcripts of H_0 and H_1 are indistinguishable, and then that those of H_1 and H_2 are indistinguishable.

We begin by arguing that the transcripts of H_0 and H_1 are computationally indistinguishable, provided that $\mathsf{Decision\text{-}LWE}_{n,m,q,\chi}$ is hard to solve by PPT adversaries. To do this, we describe how we can transform a PPT algorithm $\mathcal{D}$ that correctly distinguishes between the transcripts of H_0 and H_1 with probability at least $1/2 + f(n)$ for a non-negligible function f into a PPT algorithm $\mathcal{D}'$ that correctly solves $\mathsf{Decision\text{-}LWE}_{n,m,q,\chi}$ with the same probability. Consider the following PPT algorithm $\mathcal{D}'$ for $\mathsf{Decision\text{-}LWE}_{n,m,q,\chi}$ which runs $\mathcal{D}$ as a subroutine. The algorithm $\mathcal{D}'$ receives as input a uniformly random matrix $A \leftarrow \mathbb{Z}_q^{m\times n}$ and a vector $w \in \mathbb{Z}_q^m$. Recall that in the "real" case we have that $w = As + e \pmod q$ for a uniformly random $s \leftarrow \mathbb{Z}_q^n$ and a short error vector e, while in the "ideal" case we have that $w \leftarrow \mathbb{Z}_q^m$ independently of A. The algorithm $\mathcal{D}'$ arranges (A, w) into the augmented matrix

$$A' = [A \mid w],$$

plays the PKE CPA-security game with A' as the public key, and calls $\mathcal{D}$ on the resulting transcript. Observe that if we are in the real case where $w = As + e \pmod q$, then this game corresponds exactly to H_0, while if we are in the ideal case where $w \leftarrow \mathbb{Z}_q^m$, then the game corresponds exactly to H_1. Therefore, if $\mathcal{D}$ correctly guesses whether the transcript comes from H_0 or H_1, then $\mathcal{D}'$ also correctly guesses whether it is in the real or ideal case, thus solving $\mathsf{Decision\text{-}LWE}_{n,m,q,\chi}$. This implies that H_0 and H_1 are computationally indistinguishable.

Finally, we claim that the transcripts of H_1 and H_2 are indistinguishable, even to computationally-unbounded adversaries. The only difference between the two games lies in how the ciphertext is computed. In H_1 the ciphertext is computed as $c = x^T A' + (0^n, b \cdot \lceil \frac{q}{2} \rfloor)$ with $A' \leftarrow \mathbb{Z}_q^{m\times(n+1)}$ and $x \leftarrow \{0,1\}^m$, while in H_2 it is sampled uniformly at random from $\mathbb{Z}_q^n$. The desired claim follows directly from the fact that any (even computationally-unbounded) adversary cannot distinguish

between $(A', x^T A')$ and a uniform sample from $\mathbb{Z}_q^{m\times(n+1)} \times \mathbb{Z}_q^{n+1}$ except with negligible probability when $m \geq (n+1)\log q$. This is not obvious. It is a special case of the beautiful *Leftover Hash Lemma*. You will prove a version of this lemma in the problem set. ■

References

[AM09] Divesh Aggarwal and Ueli Maurer. Breaking RSA generically is equivalent to factoring. In *Advances in Cryptology - EUROCRYPT 2009*, pages 36–53, 2009.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011)*, pages 97–106, 2011.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 60–73, New York, NY, USA, 2021. Association for Computing Machinery.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, pages 1–23, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[MP13] Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with small parameters. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, pages 21–39, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), September 2009.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, pages 617–635, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.