EECS 475: Introduction to Cryptography Notes

Noah Peters

April 4, 2023

Abstract

Lecture notes for EECS 475 at the University of Michigan. LaTeXtemplate by Pingbang Hu.

Contents

Chapter 4

Number Theory and Public Key Cryptography

Lecture 19: Number Theory

4.1 Modular Arithmetic and Euclid's Algorithm

20 Mar. 10:30

We define the set of integers, $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$, and natural numbers, $\mathbb{N} = \mathbb{Z}^+ = \{1, 2, 3, \ldots\}$.

Theorem 4.1.1 (Product of primes). Every integer $N > 1$ can be written *uniquely* as a product of (power of) primes.

Lemma 4.1.1 (Division with remainder). Let $a \in \mathbb{Z}, b \in \mathbb{Z}^+$. $\exists$ unique integers q, r such that $a = q.b + r$ where $0 \leq r < b$, and they can be efficiently computed in *polynomial time* relative to the *bit length*: i.e. $\log_2 a + \log_2 b + O(1)$

With the ability to perform division in polynomial time, we are able to find the **greatest common divisor** of two integers a, b:

Definition 4.1.1 (Greatest common divisor). Let $a, b \in \mathbb{Z}^+$. Then, there exists $x, y \in \mathbb{Z}$ such that $\gcd(a, b) = a.x + b.y$. Further, $\gcd(a, b)$ is the *smallest* positive integer that can be written this way.

We claim there is an efficient algorithm, the **Extended Euclidean Algorithm**, that computes not only $\gcd(a, b)$, but also the x, y coefficients above:

Algorithm 4.1: Extended Euclidean Algorithm

Data: $a, b \in \mathbb{Z}$ where $a \geq b > 0$
if $b|a$ **then**
  return $x = 0, y = 1$
else
  write $a = q.b + r$ where $0 < r < b$
$x', y' \leftarrow$ **ExtEuclid**(b, r)
return $x = y', y = x' - y'.q$

Verification of Extended Euclidean Algorithm. Assuming the recursive call is successful (by induction, we can), we will get back x', y' that are the gcd of b and r:

$$b.x' + r.y' = \gcd(b, r) = \gcd(a, b)$$
$$b.x' + (a - q.b).y' = a.y' + b(x' - q.y')$$

We note that $y' = x$ and $x' - q.y' = y$ and so we are done. ■

We claim that the Extended Euclidean Algorithm makes $O(n)$ recursive calls.

4.2 Group Theoretic View of Numbers

Definition 4.2.1. Let $\mathbb{Z}_N = \{0, 1, 2, \ldots, N-1\}$, indicating the set of all possible remainders of division by N, further: $\mathbb{Z}_N^* = \{x \in \mathbb{Z}_N : \gcd(x, N) = 1\}$.

Remark. For example, $\mathbb{Z}_6^* = \{1, 5\}$. Alternatively, $\mathbb{Z}_7^* = \{1, 2, 3, 4, 5, 6\}$, showing how $\mathbb{Z}_p^* = \mathbb{Z}_p \setminus \{0\}$, where p is any prime number.

Lecture 20: Group Theory

22 Mar. 10:30

We continue our review of modular arithmetic. Modular addition, subtraction, and multiplication are all closed in modular arithmetic. We have the property $a = b \mod N \Leftrightarrow N | a - b$ and further, if $a = a' \mod N$ and $b = b' \mod N$, we have:

- $a + b = a' + b' \mod N$
- $a - b = a' - b' \mod N$
- $a.b = a'.b' \mod N$

However, division is not always possible:

$$3 \cdot 2 = 15 \cdot 2 \mod 24 \not\Rightarrow 3 = 15 \mod 24$$

Definition 4.2.2 (Invertability). An integer b is **invertible** if $\exists c$ where $b \cdot c = 1 \mod N$.

Lemma 4.2.1. If $b \geq 1, N > 1$, b is invertible mod $N \Leftrightarrow \gcd(b, N) = 1$.

Proof. Let $b \cdot c = 1 \mod N$.

$$\Rightarrow b.c - 1 = N.q$$
$$\Rightarrow b.c - N.q = 1.$$

Thus, $\gcd(b, N) = 1$ since gcd is the smallest positive integer that is expressible in this way. ■

Using the Extended Euclidean Algorithm, we can compute inverse mod N efficiently.

Definition 4.2.3 (Group). $(G, \circ)$, where $\circ : G \times G \to G$ and $\circ(g, h)$ is denoted as $g \circ h$, is a **group** if it satisfies the properties of:

- identity: $\exists e \in G$ such that $\forall g \in G$: $e \circ g = g \circ e = g$
- invertability: $\forall g \in G, \exists g^{-1}$ (or $-g$) such that $g \circ g^{-1} = e$
- associativity: $\forall g_1, g_2, g_3 \in G$: $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$
- (abelian groups): $\forall g, h \in G, g \circ h = h \circ g$ (i.e commutativity)

Example. For $(\mathbb{Z}_N, + \mod N)$ we have:

- identity: $a + 0 \equiv 0 + a \equiv a \mod N$
- invertability: $a + (-a) \equiv 0 \mod N$
- associativity: $a + (b + c) = (a + b) + c \mod N$

Example. For $(\mathbb{Z}_N^*, . \mod N)$ we have:

- identity: $a.1 \equiv 1.a \equiv a \mod N$
- invertability: $a.(a^{-1}) \equiv 1 \mod N$
- associativity: $a.(b.c) = (a.b).c \mod N$

The size of a group, called the **group order**, is denoted as $|G|$. For example, $(\mathbb{Z}_N, +)$ has order N, and $\mathbb{Z}_p^*, .$ has order $p-1$

Theorem 4.2.1. For a group, G, where $m = |G|$, we have that $\forall g \in G$: $g^m = 1$ (where $g^m = (((g \circ g) \circ g) \circ g) \ldots$).

Proof. For simplicity, assume G is abelian and suppose $G = \{g_1, g_2, \ldots, g_m$ and let $g \in G$ be arbitrary. Note that

$$g \circ g_i = g \circ g_j \Rightarrow g_i = g_j$$

and so the set $\{g \circ g_i : i \in \{1, m\}\}$ covers all elements of G exactly once. Therefore:

$$g_1 \circ -g_m = g^m(g_1 \circ \ldots \circ g_m) \Rightarrow 1 = g^m.$$

■

Corollary 4.2.1 (Fermat's Little Theorem). $\forall$ prime p, $\gcd(a, p) = 1 \Rightarrow a^{p-1} \equiv 1 \mod p$

Theorem 4.2.2 (Euler's Theorem). For $\Phi(N) = \left|\{a | 1 \leq a \leq N, \gcd(a, N) = 1\}\right|$, $|\mathbb{Z}_N^*| = \Phi(N)$ we have:

$$\text{if } \gcd(g, N) = 1 \Rightarrow g^{\Phi(N)} \equiv 1 \mod N$$

Corollary 4.2.2. For $m = |G| > 1, e \in \mathbb{Z}, \gcd(e, m) = 1$, define $d = e^{-1} \mod m$ and function $f_e : G \to G$ as $f_e(g) = g^e$. Then, f_e is a bijection whose inverse is f_d.

Definition 4.2.4 (Cyclic Groups). A group, G, is considered **cyclic** if $\exists g \in G$ such that $G = \{1 = g^0, g^1, g^2, \ldots, g^{m-1}\}$. If this is the case, we say that g generates G.

Example. For $\mathbb{Z}_7^* = \{1, \ldots, 6\}$:

$$\text{powers of 3: } \{3^0 \equiv 1, 3^1 \equiv 3, 3^2 \equiv 2, 3^3 \equiv 6, 3^4 \equiv 4, 3^5 \equiv 5\}$$

$$\text{powers of 2: } \{2^0 \equiv 1, 2^1 \equiv 2, 2^2 \equiv 4, 2^3 \equiv 1, 2^4 \equiv 2, \ldots\}$$

and so 3 is a generator of $\mathbb{Z}_7^*$, but 2 is not.

If there is no $g \in G$ that generates G, then G is not cyclic. Furthermore, when g does not generate G, it generates a **subgroup**:

Definition 4.2.5. $G' \subseteq G$, is a subgroup if $(G', \circ)$ is a group.

Theorem 4.2.3 (Lagrange's Theorem). If $G' \subseteq G$ is a subgroup, then $|G'|$ divides $|G|$.

4.3 Fast Exponentiation

Suppose we have an element g and we want to compute g^M. Instead of naively multiplying g M times, we can observe that if $M = 2^m$:

$$g^M = g^{2^m} = g^{2^{m-1}} \cdot g^{2^{m-1}} = (g^{2^{m-1}})^2.$$

So for any M we can rewrite it as

$$M = \sum_{i=0}^{l} m_i.2^i,$$

allowing us to calculate g^m:

$$g^m = \prod_{i=0}^{l} g^{2^i},$$

where each g^{2^i} can be calculated using the above trick. This results in $O(l^2)$ multiplications altogether if $|M| = l$.

Corollary 4.3.1. Fast exponentiation allows us to compute inverses efficiently because $g^{-1} = g^{|G|-1}$ since $g^{|G|} = 1$.

We've shown we can compute g^m efficiently given g and m. But, can we efficiently compute m from g^m? It's unknown, but conjectured to be extremely difficult to do so.

Lecture 21: Diffie-Hellman Key Exchange, DDH Assumption, Public Key Encryption and CPA Security

4.4 Diffie-Hellman

27 Mar. 10:30

The key exchance problem occurs when two individuals seek to communicate over an insecure channel. The canonical story has *Alice* and *Bob* attempting to communicate over a channel being *passively monitored* by an eavesdropper, *Eve*.

Definition 4.4.1 (Diffie-Hellman Protocol).

1. let Alice fix a large cyclic group G of known order q. ($|q| \approx$ security parameter).
2. Alice discovers a generator g for $G = \mathbf{Z}_{q+1}^*$ for $q+1$ prime. In other words, Alice finds a number g that enumerates all elements of the group G when raised to the powers $\{0, 1, \ldots, q-1\}$.
3. Alice chooses random $a \leftarrow \mathbb{Z}_q$, let $\mathcal{A} = g^a \in G$.
4. Alice sends $\mathcal{A} = g^a, g$ to Bob over the insecure channel. Eve has access to this information.
5. Bob receives the message and chooses a random $b \leftarrow \mathbb{Z}_q$ and sends $\mathcal{B} = g^b \in G$ back to Alice.
6. Alice and Bob both calculate $\mathcal{K} = (g^a)^b = (g^b)^a = g^{ab}$ as their shared secret key.

An eavesdropper will have to take *discrete log* to break this scheme (to find a or b). Formally, the security of this scheme is based on **DDH: Decisional Diffie-Hellman Assumption**.

Definition 4.4.2 (Decisional Diffie-Hellman Assumption (DDH)). DDH holds for a group $G = \langle g \rangle$ (i.e. G is *generated* by g) if

$$(g, g^a, g^b, g^{ab}) \in G^4$$

is indistinguishable (for $a, b \leftarrow \mathbb{Z}_q, q = |G|$) from

$$(g, g^a, g^b, g^c), \text{ where } c \leftarrow \mathbf{Z}_q$$

Remark. Key derivation can simply be an algorithm for turning a group element into a random bit-string.

4.5 Modeling Public Key Encryption

We want to have a protocol (Gen, Enc, Dec) that can *directly* handle public key encryption. We can model this analogously to EAV/CPA security, just in a public key setting:

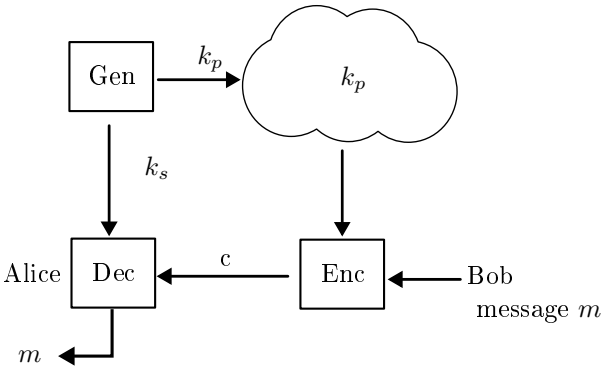

Figure 4.1: Analog of EAV/CPA security in the context of public keys.

Definition 4.5.1 (Public Key CPA (EAV) Secure Scheme)**.** For our **CPA secure public key scheme** we have:

- $\text{Gen}(1^n)$: outputs (k_p, k_s)
- $\text{Enc}(k_p, m \in M)$: outputs ciphertext c
- $\text{Dec}(k_s, c)$: outputs $m \in M$(or fail "$\perp$")

We analyze it's correctness. For $(k_p, k_s) \leftarrow \text{Gen}(1^n)$, we always have $\forall m \in M$:

$$\text{Dec}(k_s, \text{Enc}(k_p, m)) = m$$

Just like previous instances of security, we can define public key CPA security in the context of a game. Here we have adversary $\mathcal{A}$ against $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$:

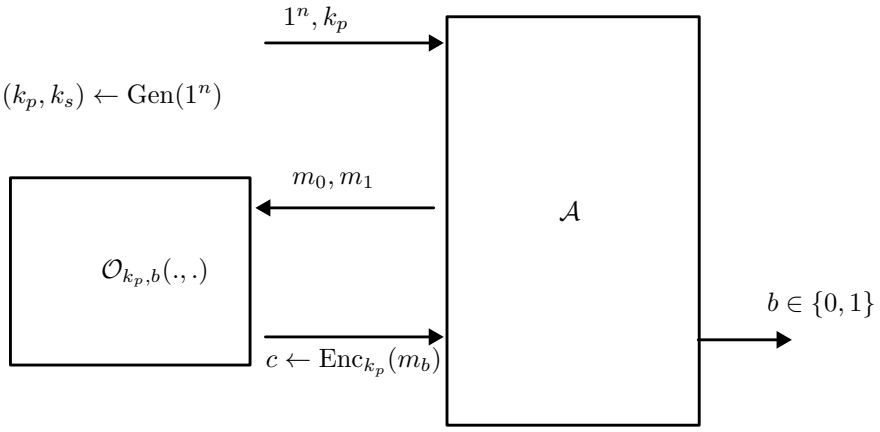

Figure 4.2: The public key CPA game.

Definition 4.5.2 (Public CPA Security). A public key encryption scheme Π as defined above is secure if $\forall$ p.p.t. $\mathcal{A}$:

$$\mathbf{Adv}_{\Pi}(\mathcal{A}) := \left| \Pr_{(k_p,k_s)\leftarrow \mathrm{Gen}(1^n)}(\mathcal{A}^{\mathcal{O}_{k_p,0}(\cdot,\cdot)}) - \Pr_{(k_p,k_s)\leftarrow \mathrm{Gen}(1^n)}(\mathcal{A}^{\mathcal{O}_{k_p,1}(\cdot,\cdot)}) \right| = \mathrm{negl}(n).$$

Remark. The number of queries to the LR oracle, $\mathcal{O}$ doesn't matter. I.e. if we have EAV-security allowing $\mathcal{A}$ to query the oracle once, then we have security even if multiple queries are allowed.

Proof. To show that

$$\textit{one-query public CPA security} \Rightarrow \textit{many query public CPA security}$$

we can use a similar proof that we used for CPA security originally and note that the adversary in this scenario can, in addition to the abilities of the original adversary, encrypt things themselves since the keys are public.

Imagine a many-query attacker, $\mathcal{A}$ that makes up to $q : \mathsf{poly}(n)$ queries. Consider the following worlds:

(i) *Hybrid 0*: All queries answered by $c \leftarrow Enc_{k_p}(m_0)$, then $Enc_{k_p}(m_0)$ thereafter.

(ii) *Hybrid 1*: First query (m_0, m_1) answered by $c \leftarrow Enc_{k_p}(m_1)$, then $Enc_{k_p}(m_0)$ thereafter.

(iii) *Hybrid 2*: First **two** queries are answered by $c \leftarrow Enc_{k_p}(m_1)$, then $Enc_{k_p}(m_0)$ thereafter.

(iv) *Hybrid q*: All queries answered by $c \leftarrow Enc_{k_p}(m_1)$

Using these hybrid worlds we can construct the *left* and *right* worlds as follows:

(i) **Left world**: all queries (m_0, m_1), to the LR oracle, $\mathcal{O}$, answered with $c \leftarrow Enc_{k_p}(m_0)$.

(ii) **Right world**: each query to the LR oracle, $\mathcal{O}$, answered with $c \leftarrow Enc_{k_p}(m_1)$.

We can use the idea that if an adversary, $\mathcal{A}$ can distinguish between the left and right worlds, they can distinguish between the ith hybrid world and the $i+1$th hybrid world where the ith hybrid world is defined as

$$HybridWorld_i = \text{First } i \text{ queries of } (m_0, m_1) \text{ answered by } c \leftarrow Enc_{k_p}(m_1), \text{ then } c \leftarrow Enc_{k_p}(m_0) \text{ thereafter.}$$

We first note that the difference between Hyb_{i-1} and Hyb_i is only in how the ith query is answered. We can build a "simulator" $\mathcal{S}_i^{\mathcal{O}}(k_p,)$ that gets *one query* and simulates either Hyb_{i-1} or Hyb_i depending on b. ■

Lecture 22: CPA Model, El Gamal Cryptosystem

29 Mar. 10:30

Appendix

Appendix A

Additional Proofs

A.1 Proof of ??

We can now prove **??**.

Proof of ??. See here. ■