Massachusetts Institute of Technology
6.857: Network and Computer Security
Professors Ronald L. Rivest and Yael Tauman Kalai

Handout R4
February 28, 2020
Adrian Sy

Recitation 4: Finite Fields and Security Definitions

1 Groups and Finite Fields

1.1 Groups

A group is defined by a tuple $(S, \cdot)$ where S is a set of numbers and $\cdot$ is an operation with the following properties, $\forall a, b, c \in S$:

Associative - $(a \cdot b) \cdot c = a \cdot (b \cdot c)$

Commutative - $a \cdot b = b \cdot a$

Identity - $a \cdot 1 = a$ (for some identity element $1 \in S$)

Invertibility - $a \cdot (a^{-1}) = 1$

The "order" (ord) of a finite group refers to the number of elements in the set. Some commonly used examples include $\mathbb{Z}_p^* = \mathbb{Z}_p \setminus 0$, $\mathbb{Z}_n^* = \{a \in \{1, 2, \ldots, n-1\} \mid \gcd(a, n) = 1, n = p \cdot q\}$.

A cyclic group is a group G for which an element g exists such that $\{g^1, \ldots, g^{ord(G)}\} = G$. g is referred to as a generator of G.

1.2 Fields

A field is defined by a tuple $(S, +, \cdot)$ where S is a set of numbers and $+$ and $\cdot$ are operations satisfying the following:

1. S contains 0 (additive identity) and 1 (multiplicative identity).

2. $(S, +)$ is an abelian (commutative) group with identity 0.

3. $(S \setminus \{0\}, \cdot)$ is an abelian (commutative) group with identity 1.

Some examples of fields include the following:

1. **$(\mathbb{Q}, +, \cdot)$ (set of rationals)**

For a rational $\frac{p}{q}$, additive inverse is $-\frac{p}{q}$ and multiplicative inverse is $\frac{q}{p}$.

2. **$(\mathbb{R}, +, \cdot)$ (set of reals)** and **$(\mathbb{C}, +, \cdot)$ (set of reals)**

For a real or complex x, additive inverse is $-x$ and multiplicative inverse is $\frac{1}{x}$.

3. **$(\mathbb{Z}_p, +, \cdot)$ (set of non-negative integers $< p$)**

For $x \in \mathbb{Z}_p$, additive inverse is $p - x$ and multiplicative inverse is p^{-1} (use Extended Euclidean Algorithm, see R1).

1.3 Finite Fields and Construction

Finite fields, also referred to as Galois fields (GF), are fields that have a finite number of elements. Any finite field has order p^n, where order is the number of elements in the set, p is prime and n is some natural number.

For $n = 1$, $\mathbb{Z}_p$ is such a finite field. For $n > 1$, we let $\pi(x)$ be some n-degree irreducible polynomial where the coefficients are all in $\mathbb{Z}_p$. Consider the following set of polynomials $F_p[x]$:

$$c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1},\ c_i \in \mathbb{Z}_p$$

Then $(F_p[x]/\pi(x), +, \cdot)$ is a finite field with order p^n.

$+$ is defined as with normal polynomial addition except the coefficients are all computed in $\mod n$. $\cdot$ is similar to normal polynomial multiplication, except when the degree exceeds n, the value is treated as the remainder when doing polynomial long division with $\pi(x)$.

A notable use of finite fields is in Shamir's secret sharing ($\mathbb{Z}_p$) and in AES ($GF(2^8)$).

2 Security Definitions

2.1 CPA Security

In the context of a symmetric (same key to encrypt and decrypt) encryption scheme, a Chosen Plaintext Attack (CPA) refers to an attack where the adversary can choose a number of plaintexts and see their corresponding ciphertexts and use this information to break the encryption scheme, e.g. discover some information about the plaintext of some new ciphertext, discover the key.

Indistinguishable under Chosen Plaintext Attack (IND-CPA) security means that any adversary $\mathcal{A}$ cannot efficiently win in the following game.

1. The challenger generates a key k and gives $\mathcal{A}$ access to encryption oracle Enc_k.
2. As many times as needed, $\mathcal{A}$ can generate a message m and see the ciphertext $Enc_k(m)$.
3. $\mathcal{A}$ chooses two messages m_0 and m_1 and sends them to the challenger.
4. The challenger returns ciphertext c which is either $Enc_k(m_b)$ for $b \in \{0, 1\}$.
5. (Adaptive case) $\mathcal{A}$ can continue to query the encryption oracle.
6. $\mathcal{A}$ wins if they can guess b correctly.

By necessity, for any encryption scheme to be CPA secure, it must be randomized. Otherwise $\mathcal{A}$ can just choose m_0 and m_1 from the plaintexts for which they know the ciphertext.

(For any public key cryptosystem to be usable, it must also be CPA secure since everyone has the public key and thus access to the "encryption oracle".)

2.2 MAC Authentication

Message Authentication Codes provide integrity to a message, i.e. makes it detectable when a message has been altered. CPA, in the context of MACs, is when an adversary can choose a number of plaintexts and see their corresponding MACs, and use this to break the MAC scheme, e.g. be able to forge a MAC for some/all possible messages.

Defined by three algorithms: $Gen, MAC(k, m), Ver(k, m, M)$.

A MAC scheme is secure against adaptive chosen message/ plaintext attack if any adversary $\mathcal{A}$ cannot efficiently win in the following game.

1. The challenger generates a key k and gives $\mathcal{A}$ access to MAC oracle MAC_k.
2. As many times as needed, $\mathcal{A}$ can generate a message m and see the MAC $M = MAC_k(m)$.
3. $\mathcal{A}$ wins if they can generate a **new** message m' and MAC M' such that $Ver_k(m', M') = 1$.

2.3 CCA Security

For symmetric encryption schemes, a Chosen Ciphertext Attack (CCA) is when the adversary has access to both an encryption and decryption oracle and can use this to break the encryption scheme. The game corresponding to this type of security is as follows:

1. The challenger generates a key k and gives $\mathcal{A}$ access to encryption oracle Enc_k.
2. As many times as needed, $\mathcal{A}$ can generate a message m and see the ciphertext $Enc_k(m)$.
3. Similarly, $\mathcal{A}$ can generate a ciphertext c and see the decrypted plaintext $m = Dec_k(c)$ if valid.
4. $\mathcal{A}$ chooses two messages m_0 and m_1 and sends them to the challenger.
5. The challenger returns ciphertext c which is either $Enc_k(m_b)$ for $b \in \{0, 1\}$.
6. $\mathcal{A}$ can continue to query the encryption and decryption oracles (but cannot query for decryption of c).
7. $\mathcal{A}$ wins if they can guess b correctly.

2.4 Making a CCA Secure Encryption Scheme

Any CPA secure encryption scheme (Gen, Enc, Dec) can be converted into a CCA secure encryption scheme by using the MAC $(Gen_{MAC}, MAC, Verify)$. To be specific, key generation, encryption and decryption algorithms are as follows:

- $Gen' = (Gen, Gen_{MAC})$ (generates separate keys for the encryption scheme and MAC scheme)
- $Enc'((k_1, k_2), m) = Enc(k_1, m), MAC(k_2, Enc(k_1, m))$ (ciphertext and MAC of ciphertext)
- $Dec'((k_1, k_2), (c_1, c_2)) = Dec(k_1, c_1)$ if $Ver(k_2, c_1, c_2)$

Since the original encryption scheme is CPA secure and the MAC scheme is independently generated from the encryption scheme, the entire scheme is still CPA secure.

For CCA security, the only difference is the adversary's access to the decryption algorithm. However, because the MAC is unforgeable, the adversary cannot generate a valid ciphertext so the decryption algorithm provides no additional information.

As such, this scheme is CCA secure.