[CS309] Foundations of Cryptography and Security Networks

Instructor: Dr. Dibyendu Roy — Autumn Semester 2024
Notes by: Tanuj Saini (202251141) — Week 5 Lecture

1 Introduction to Polynomial and Field Theory in Cryptography

Polynomial and field theory form the cornerstone of numerous algebraic structures, particularly in the domain of finite fields. These mathematical constructs play a pivotal role in contemporary cryptographic systems, providing the foundation for secure communication protocols and encryption algorithms.

1.1 Irreducible Polynomials

Definition 1 (Irreducible Polynomial) *A polynomial $P(x) \in F[x]$ of degree $n \neq 1$ is termed irreducible if it cannot be factored into the product of two non-constant polynomials in $F[x]$. Formally, if $P(x) = P_1(x) \cdot P_2(x)$, then either $P_1(x)$ or $P_2(x)$ must be a constant.*

The polynomial $P(x) = x^2 + 1$ is irreducible over $\mathbb{R}[x]$, but reducible over $\mathbb{C}[x]$, as $x^2 + 1 = (x+i)(x-i)$ in $\mathbb{C}[x]$.

Irreducible polynomials are fundamental in the construction of finite fields, also known as Galois fields. For instance, the field $\mathbb{F}_{2^n}$ can be constructed using an irreducible polynomial of degree n over $\mathbb{F}_2$.

Theorem 1 *Let $P(x)$ be an irreducible polynomial over a field F. Then:*

1. *$I = \langle P(x) \rangle = \{q(x) \cdot P(x) \mid q(x) \in F[x]\}$ defines an ideal in the ring $F[x]$.*
2. *The quotient $F[x]/\langle P(x) \rangle$ forms a field.*

1.2 Polynomial Arithmetic

1.2.1 Multiplication

Polynomial multiplication is a fundamental operation in cryptographic algorithms. Consider the following example:

$$\begin{aligned}
p(x) \cdot (x^2 + x + 1) &= (x+1) \cdot (x+1) \\
&= (x+1)(x+1)x + (x+1)x^2 + x^3 \\
&= x^4 + x^3 + x^2 + x = 1 + x^3
\end{aligned}$$

This exemplifies polynomial arithmetic within a ring structure, which is crucial in algebra and its applications to coding theory and cryptography.

1.2.2 Division

Theorem 2 (Polynomial Division) *For any polynomial $q(x)$ in $F[x]$, there exist unique polynomials $d(x)$ and $r(x)$ such that:*

$$q(x) = d(x) \cdot P(x) + r(x)$$

where $\deg(r(x)) < \deg(P(x))$.

This theorem forms the basis for division in polynomial rings and is essential for algorithms like the Euclidean algorithm used to compute greatest common divisors.

Euclidean Algorithm for Polynomials [1] PolynomialGCD$a(x), b(x)$ $b(x) \neq 0$ $r(x) \leftarrow a(x) \bmod b(x)$ $a(x) \leftarrow b(x)$ $b(x) \leftarrow r(x)$ **return** $a(x)$

2 Finite Fields in Cryptography

Finite fields, particularly $\mathbb{F}_{2^n}$, are extensively used in cryptography due to their properties that facilitate efficient computation and provide security.

2.1 Construction of $\mathbb{F}_{2^n}$

To construct $\mathbb{F}_{2^n}$, we use an irreducible polynomial of degree n over $\mathbb{F}_2$. For example, to construct $\mathbb{F}_{2^8}$, which is used in AES, we can use the irreducible polynomial:

$$P(x) = x^8 + x^4 + x^3 + x + 1$$

Lemma 1 *The polynomial $x^8 + x^4 + x^3 + x + 1$ is irreducible over $\mathbb{F}_2$.*

Elements of $\mathbb{F}_{2^8}$ can be represented as polynomials of degree less than 8 with coefficients in $\mathbb{F}_2$, modulo $P(x)$.

3 Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES) is a symmetric block cipher algorithm widely adopted for secure data encryption. Established by the U.S. National Institute of Standards and Technology (NIST) in 2001, AES replaced the older Data Encryption Standard (DES).

3.1 AES Variants

AES supports three key sizes, each with a fixed block size of 128 bits:

- **AES-128**: 128-bit key, 10 rounds
- **AES-192**: 192-bit key, 12 rounds
- **AES-256**: 256-bit key, 14 rounds

3.2 State Representation

In AES, the 128-bit block is represented as a 4x4 matrix of bytes, called the state:

$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}$$

3.3 Round Structure

Each AES round consists of several operations:

1. **SubBytes**: Non-linear substitution using an S-box
2. **ShiftRows**: Cyclic shifting of state rows
3. **MixColumns**: Column mixing (omitted in final round)
4. **AddRoundKey**: XOR with round key

3.3.1 SubBytes Operation

The SubBytes step provides non-linearity in the cipher. It operates on each byte of the state independently, using an S-box derived from the multiplicative inverse over $\mathbb{F}_{2^8}$, followed by an affine transformation.

Let $x = \langle C_7, C_6, C_5, C_4, C_3, C_2, C_1, C_0 \rangle$ be an 8-bit input. The S-box transformation is denoted as:

$$S(C_7 C_6 C_5 C_4 C_3 C_2 C_1 C_0) = \text{8-bit output}$$

The S-box is constructed as follows:

1. For each byte b, compute its multiplicative inverse in $\mathbb{F}_{2^8}$: b^{-1}
2. Apply the affine transformation:
$$b' = Ab^{-1} + c$$
   where A is a fixed 8x8 matrix and c is a fixed 8-bit vector.

3.3.2 ShiftRows Operation

In the ShiftRows step, each row of the state is cyclically shifted to the left:

- Row 0 is not shifted
- Row 1 is shifted 1 byte to the left
- Row 2 is shifted 2 bytes to the left
- Row 3 is shifted 3 bytes to the left

3.3.3 MixColumns Operation

In the MixColumns step, each column of the state is treated as a polynomial over $\mathbb{F}_{2^8}$ and multiplied by a fixed polynomial:

$$c(x) = 03x^3 + 01x^2 + 01x + 02 \pmod{x^4 + 1}$$

This operation can be represented as a matrix multiplication:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} s_{0,j} \\ s_{1,j} \\ s_{2,j} \\ s_{3,j} \end{bmatrix}$$

This operation ensures diffusion, spreading the influence of each input bit over the entire ciphertext.

3.4 Key Expansion

The AES key expansion algorithm generates a series of round keys from the cipher key. For AES-128, it generates 11 128-bit round keys.

AES-128 Key Expansion [1] ExpandKey$key[16]$ $W[44]$ 44 32-bit words $i = 0$ to 3 $W[i] \leftarrow (key[4i], key[4i+1], key[4i+2], key[4i+3])$ $i = 4$ to 43 $temp \leftarrow W[i-1]$ $i \bmod 4 = 0$ $temp \leftarrow SubWord(RotWord(temp)) \oplus Rcon[i/4]$ $W[i] \leftarrow W[i-4] \oplus temp$ **return** W

Where:

- SubWord applies the S-box to each byte of the input word
- RotWord performs a cyclic permutation on the input word
- Rcon[i] is the round constant for round i

4 Security Considerations

The security of AES relies on several factors:

1. **Confusion**: Provided by the SubBytes operation
2. **Diffusion**: Achieved through ShiftRows and MixColumns
3. **Key size**: Larger key sizes (192, 256 bits) provide increased security against brute-force attacks
4. **Number of rounds**: More rounds increase resistance to cryptanalysis

As of 2024, AES remains secure against known attacks when properly implemented. However, ongoing research in quantum computing may pose future challenges to its security.