Cryptography: Comprehensive Summary

Mathematical Notes Collection

October 28, 2025

Contents

1 Introduction to Cryptography

Cryptography is the science of secure communication in the presence of adversaries. It encompasses techniques for protecting information confidentiality, ensuring data integrity, authenticating parties, and providing non-repudiation.

1.1 Basic Concepts

Definition 1.1 (Cryptography)**.** *Cryptography is the practice and study of techniques for secure communication in the presence of third parties called adversaries.*

Definition 1.2 (Cryptosystem)**.** *A cryptosystem is a tuple* (P, C, K, E, D) *where:*

- P *is the set of plaintexts*
- C *is the set of ciphertexts*
- K *is the set of keys*
- E *is the set of encryption functions*
- D *is the set of decryption functions*

1.2 Security Goals

1. **Confidentiality**: Information is accessible only to authorized parties
2. **Integrity**: Information cannot be modified without detection
3. **Authentication**: Verification of identity of communicating parties
4. **Non-repudiation**: Prevention of denial of participation in communication

1.3 Types of Cryptography

1. **Symmetric Cryptography**: Same key for encryption and decryption
2. **Asymmetric Cryptography**: Different keys for encryption and decryption
3. **Hash Functions**: One-way functions for data integrity
4. **Digital Signatures**: Authentication and non-repudiation

2 Mathematical Foundations

2.1 Number Theory

2.1.1 Modular Arithmetic

Definition 2.1 (Modular Arithmetic)**.** *For integers* a*,* b*, and* $n > 0$*:*

$$a \equiv b \pmod{n} \iff n \mid (a - b)$$

Theorem 2.2 (Properties of Modular Arithmetic)**.** *For integers* a*,* b*,* c*, and* $n > 0$*:*

1. $(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$
2. $(a \cdot b) \bmod n = ((a \bmod n) \cdot (b \bmod n)) \bmod n$
3. $(a^b) \bmod n = ((a \bmod n)^b) \bmod n$

2.1.2 Greatest Common Divisor

Definition 2.3 (GCD)**.** *The greatest common divisor of integers a and b is the largest integer d such that $d \mid a$ and $d \mid b$.*

Algorithm 1 Extended Euclidean Algorithm

Require: Integers a, b with $a \geq b \geq 0$
Ensure: Integers d, x, y such that $d = \gcd(a, b) = ax + by$
if $b = 0$ **then**
  return $(a, 1, 0)$
else
  $(d', x', y') \leftarrow \text{ExtendedEuclidean}(b, a \bmod b)$
  $d \leftarrow d'$
  $x \leftarrow y'$
  $y \leftarrow x' - \lfloor a/b \rfloor \cdot y'$
  return (d, x, y)
end if

2.1.3 Modular Inverse

Definition 2.4 (Modular Inverse)**.** *For integers a and n, the modular inverse of a modulo n is an integer x such that:*

$$ax \equiv 1 \pmod{n}$$

Proposition 2.5 (Existence of Modular Inverse)**.** *The modular inverse of a modulo n exists if and only if* $\gcd(a, n) = 1$.

2.2 Finite Fields

Definition 2.6 (Finite Field)**.** *A finite field $\mathbb{F}_q$ is a field with q elements where $q = p^n$ for prime p and integer $n \geq 1$.*

Theorem 2.7 (Finite Field Properties)**.** *For finite field $\mathbb{F}_q$:*

1. *The multiplicative group $\mathbb{F}_q^*$ is cyclic*
2. *For any $a \in \mathbb{F}_q$: $a^q = a$*
3. *For any $a \in \mathbb{F}_q^*$: $a^{q-1} = 1$*

2.3 Elliptic Curves

Definition 2.8 (Elliptic Curve)**.** *An elliptic curve over field K is defined by the Weierstrass equation:*

$$y^2 = x^3 + ax + b$$

where $a, b \in K$ and $\Delta = -16(4a^3 + 27b^2) \neq 0$.

Definition 2.9 (Elliptic Curve Group Law)**.** *For points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ on elliptic curve E:*

$$P + Q = \begin{cases} \mathcal{O} & \text{if } P = -Q \\ (x_3, y_3) & \text{otherwise} \end{cases} \tag{1}$$

where:

$$x_3 = \lambda^2 - x_1 - x_2 \tag{2}$$
$$y_3 = \lambda(x_1 - x_3) - y_1 \tag{3}$$
$$\lambda = \begin{cases} \frac{y_2-y_1}{x_2-x_1} & \text{if } P \neq Q \\ \frac{3x_1^2+a}{2y_1} & \text{if } P = Q \end{cases} \tag{4}$$

3 Symmetric Cryptography

3.1 Block Ciphers

Definition 3.1 (Block Cipher). *A block cipher is a deterministic algorithm operating on fixed-length groups of bits (blocks) with an unvarying transformation specified by a symmetric key.*

3.1.1 Data Encryption Standard (DES)

Definition 3.2 (DES). *DES is a 64-bit block cipher with 56-bit key using Feistel network structure.*

Algorithm 2 DES Encryption

```
1: Split 64-bit plaintext into L_0 and R_0 (32 bits each)
2: for i = 1 to 16 do
3:     L_i ← R_{i−1}
4:     R_i ← L_{i−1} ⊕ F(R_{i−1}, K_i)
5: end for
6: Ciphertext ← R_16||L_16
```

3.1.2 Advanced Encryption Standard (AES)

Definition 3.3 (AES). *AES is a 128-bit block cipher with key sizes of 128, 192, or 256 bits using substitution-permutation network.*

Algorithm 3 AES Round Function

1: **SubBytes**: Apply S-box to each byte
2: **ShiftRows**: Cyclically shift rows
3: **MixColumns**: Mix columns using matrix multiplication
4: **AddRoundKey**: XOR with round key

3.2 Stream Ciphers

Definition 3.4 (Stream Cipher). *A stream cipher is a symmetric key cipher where plaintext digits are combined with a pseudorandom cipher digit stream (keystream).*

3.2.1 Linear Feedback Shift Register (LFSR)

Definition 3.5 (LFSR). *An LFSR is a shift register whose input bit is a linear function of its previous state.*

Algorithm 4 LFSR Generation

Initialize register with seed value
for each clock cycle **do**
  Output least significant bit
  Compute feedback bit as XOR of selected taps
  Shift register right and insert feedback bit
end for

3.3 Modes of Operation

3.3.1 Electronic Codebook (ECB)

Definition 3.6 (ECB Mode). *In ECB mode, each plaintext block is encrypted independently using the same key.*

3.3.2 Cipher Block Chaining (CBC)

Definition 3.7 (CBC Mode). *In CBC mode, each plaintext block is XORed with the previous ciphertext block before encryption.*

Algorithm 5 CBC Encryption

$C_0 \leftarrow IV$ (initialization vector)
for $i = 1$ to n **do**
  $C_i \leftarrow E_K(P_i \oplus C_{i-1})$
end for

3.3.3 Counter (CTR) Mode

Definition 3.8 (CTR Mode). *In CTR mode, a counter is encrypted to produce a keystream, which is XORed with the plaintext.*

4 Asymmetric Cryptography

4.1 RSA Cryptosystem

Definition 4.1 (RSA). *RSA is an asymmetric cryptosystem based on the difficulty of factoring large integers.*

Algorithm 6 RSA Key Generation

Choose two large primes p and q
Compute $n = pq$ and $\phi(n) = (p-1)(q-1)$
Choose e such that $\gcd(e, \phi(n)) = 1$
Compute d such that $ed \equiv 1 \pmod{\phi(n)}$
Public key: (n, e), Private key: (n, d)

Theorem 4.2 (RSA Correctness). *For RSA cryptosystem with public key (n, e) and private key (n, d):*

$$(m^e)^d \equiv m \pmod{n}$$

for any message m with $0 \leq m < n$.

Algorithm 7 RSA Encryption/Decryption

1: **Encryption**: $c \equiv m^e \pmod{n}$
2: **Decryption**: $m \equiv c^d \pmod{n}$

4.2 Diffie-Hellman Key Exchange

Definition 4.3 (Diffie-Hellman). *Diffie-Hellman is a method for securely exchanging cryptographic keys over a public channel.*

Algorithm 8 Diffie-Hellman Key Exchange

1: Alice and Bob agree on prime p and generator g
2: Alice chooses secret a, sends $A = g^a \bmod p$ to Bob
3: Bob chooses secret b, sends $B = g^b \bmod p$ to Alice
4: Alice computes $K = B^a \bmod p$
5: Bob computes $K = A^b \bmod p$
6: Shared secret: $K = g^{ab} \bmod p$

4.3 Elliptic Curve Cryptography (ECC)

Definition 4.4 (Elliptic Curve Discrete Logarithm Problem). *Given elliptic curve E, point P, and point $Q = kP$, find integer k.*

Algorithm 9 ECDH Key Exchange

1: Alice and Bob agree on elliptic curve E and base point G
2: Alice chooses secret a, sends $A = aG$ to Bob
3: Bob chooses secret b, sends $B = bG$ to Alice
4: Alice computes $K = aB$
5: Bob computes $K = bA$
6: Shared secret: $K = abG$

5 Hash Functions

5.1 Properties of Hash Functions

Definition 5.1 (Cryptographic Hash Function). *A cryptographic hash function $H : \{0,1\}^* \to \{0,1\}^n$ should satisfy:*

1. ***Preimage resistance**: Given h, hard to find m such that $H(m) = h$*
2. ***Second preimage resistance**: Given m, hard to find $m' \neq m$ such that $H(m) = H(m')$*
3. ***Collision resistance**: Hard to find any m, m' such that $H(m) = H(m')$*

5.2 SHA Family

5.2.1 SHA-1

Definition 5.2 (SHA-1). *SHA-1 produces a 160-bit hash value and operates on 512-bit blocks.*

Algorithm 10 SHA-1 Processing

Pad message to multiple of 512 bits
Initialize hash values h_0, h_1, h_2, h_3, h_4
for each 512-bit block **do**
    Break block into 16 32-bit words
    Extend to 80 words using recurrence relation
    Process through 80 rounds with different functions
    Update hash values
end for
Output 160-bit hash

5.2.2 SHA-256

Definition 5.3 (SHA-256)**.** *SHA-256 produces a 256-bit hash value and operates on 512-bit blocks.*

5.3 Merkle-Damgård Construction

Definition 5.4 (Merkle-Damgård)**.** *The Merkle-Damgård construction builds a collision-resistant hash function from a collision-resistant compression function.*

Algorithm 11 Merkle-Damgård Construction

Pad message M to length multiple of block size
Initialize $h_0 = IV$
for each block M_i **do**
    $h_i = f(h_{i-1}, M_i)$
end for
Output h_n

6 Digital Signatures

6.1 RSA Signatures

Algorithm 12 RSA Signature Generation

Compute hash $h = H(m)$ of message m
Generate signature $s = h^d \bmod n$
Send (m, s)

6.2 Elliptic Curve Digital Signature Algorithm (ECDSA)

7 Key Management

7.1 Key Distribution

Definition 7.1 (Key Distribution Problem)**.** *The key distribution problem is how to securely share cryptographic keys between communicating parties.*

Algorithm 13 RSA Signature Verification

1: Compute hash $h = H(m)$ of received message m
2: Compute $h' = s^e \bmod n$
3: Accept if $h = h'$, reject otherwise

Algorithm 14 ECDSA Signature Generation

1: Compute hash $h = H(m)$ of message m
2: Choose random k with $1 \leq k < n$
3: Compute $R = kG = (x_R, y_R)$
4: Compute $r = x_R \bmod n$
5: Compute $s = k^{-1}(h + rd) \bmod n$
6: Signature: (r, s)

7.1.1 Key Distribution Centers (KDC)

7.2 Public Key Infrastructure (PKI)

Definition 7.2 (PKI). *A PKI is a framework for managing digital certificates and public-private key pairs.*

Definition 7.3 (Digital Certificate). *A digital certificate binds a public key to an identity and is signed by a Certificate Authority (CA).*

7.3 Perfect Forward Secrecy

Definition 7.4 (Perfect Forward Secrecy). *Perfect forward secrecy ensures that compromise of long-term keys does not compromise past session keys.*

8 Protocols and Applications

8.1 Transport Layer Security (TLS)

8.2 Secure Shell (SSH)

Definition 8.1 (SSH). *SSH is a cryptographic network protocol for secure remote login and file transfer.*

8.3 IPSec

Definition 8.2 (IPSec). *IPSec is a suite of protocols for securing Internet Protocol communications.*

Algorithm 15 ECDSA Signature Verification

1: Compute hash $h = H(m)$ of message m
2: Compute $u_1 = hs^{-1} \bmod n$ and $u_2 = rs^{-1} \bmod n$
3: Compute $R = u_1 G + u_2 Q = (x_R, y_R)$
4: Accept if $r = x_R \bmod n$, reject otherwise

Algorithm 16 KDC Protocol

1: Alice requests session key with Bob from KDC
2: KDC generates random session key K_{AB}
3: KDC sends $E_{K_A}(K_{AB})$ to Alice
4: KDC sends $E_{K_B}(K_{AB})$ to Bob
5: Alice and Bob use K_{AB} for secure communication

Algorithm 17 TLS Handshake

1: Client sends ClientHello with supported cipher suites
2: Server sends ServerHello with chosen cipher suite
3: Server sends certificate and ServerHelloDone
4: Client verifies certificate and sends ClientKeyExchange
5: Both parties compute master secret and session keys
6: Client and server send ChangeCipherSpec and Finished

8.4 PGP/GPG

Definition 8.3 (PGP). *Pretty Good Privacy (PGP) is a data encryption and decryption program providing cryptographic privacy and authentication.*

9 Advanced Topics

9.1 Quantum Cryptography

Definition 9.1 (Quantum Key Distribution). *Quantum key distribution uses quantum mechanics to guarantee secure key exchange.*

9.1.1 BB84 Protocol

Algorithm 18 BB84 Protocol

1: Alice generates random bit string and basis choices
2: Alice sends qubits encoded in chosen bases
3: Bob measures qubits in random bases
4: Alice and Bob announce basis choices
5: Alice and Bob discard bits where bases don't match
6: Alice and Bob perform error correction and privacy amplification

9.2 Homomorphic Encryption

Definition 9.2 (Homomorphic Encryption). *Homomorphic encryption allows computation on encrypted data without decrypting it.*

Definition 9.3 (Fully Homomorphic Encryption). *Fully homomorphic encryption supports arbitrary computation on encrypted data.*

9.3 Multi-Party Computation

Definition 9.4 (Secure Multi-Party Computation). *Secure multi-party computation allows parties to jointly compute a function over their inputs while keeping those inputs private.*

9.4 Zero-Knowledge Proofs

Definition 9.5 (Zero-Knowledge Proof). *A zero-knowledge proof is a method by which one party (the prover) can prove to another party (the verifier) that they know a value x without conveying any information apart from the fact that they know the value x.*

9.4.1 Properties of Zero-Knowledge Proofs

Definition 9.6 (Completeness). *If the statement is true, the honest verifier will be convinced by an honest prover.*

Definition 9.7 (Soundness). *If the statement is false, no cheating prover can convince the honest verifier that it is true, except with some small probability.*

Definition 9.8 (Zero-Knowledge). *If the statement is true, no verifier learns anything other than the fact that the statement is true.*

9.4.2 Interactive Proof Systems

Definition 9.9 (Interactive Proof System). *An interactive proof system is a protocol between a prover and verifier where the prover convinces the verifier of the truth of a statement through multiple rounds of interaction.*

Algorithm 19 Interactive Zero-Knowledge Proof Protocol

1: Prover commits to secret using commitment scheme
2: Verifier sends random challenge c
3: Prover responds r based on challenge and secret
4: Verifier checks response against commitment
5: Repeat k rounds until desired confidence level

9.4.3 Non-Interactive Zero-Knowledge Proofs

Definition 9.10 (Non-Interactive Zero-Knowledge Proof). *A non-interactive zero-knowledge proof requires no interaction between prover and verifier, using a common reference string or random oracle.*

Theorem 9.11 (Fiat-Shamir Transformation). *Any public-coin interactive proof can be converted to a non-interactive proof using the Fiat-Shamir heuristic with a random oracle.*

9.4.4 Succinct Non-Interactive Arguments of Knowledge (SNARKs)

Definition 9.12 (SNARK). *A SNARK is a non-interactive zero-knowledge proof system where the proof size and verification time are sublinear in the size of the computation being proven.*

Definition 9.13 (Arithmetic Circuit). *An arithmetic circuit is a directed acyclic graph where each node performs an arithmetic operation (addition or multiplication) on field elements.*

9.4.5 zk-SNARKs

Definition 9.14 (zk-SNARK). *A zk-SNARK is a zero-knowledge Succinct Non-interactive Argument of Knowledge that provides privacy and succinctness.*

1. **QAP (Quadratic Arithmetic Program)**: Convert circuit to polynomial constraints

Algorithm 20 SNARK Construction

Convert computation to arithmetic circuit
Generate proving key and verification key
Prover computes witness for circuit satisfiability
Prover generates proof using proving key
Verifier checks proof using verification key

2. **Trusted Setup**: Generate proving and verification keys
3. **Proof Generation**: Create proof of circuit satisfiability
4. **Proof Verification**: Verify proof without revealing witness

9.4.6 zk-STARKs

Definition 9.15 (zk-STARK)**.** *A zk-STARK is a zero-knowledge Scalable Transparent Argument of Knowledge that requires no trusted setup.*

Algorithm 21 zk-STARK Construction

Convert computation to algebraic intermediate representation
Generate execution trace and constraints
Apply FRI (Fast Reed-Solomon Interactive Oracle Proofs)
Generate proof without trusted setup
Verify proof using public randomness

9.4.7 Bulletproofs

Definition 9.16 (Bulletproof)**.** *A Bulletproof is a non-interactive zero-knowledge proof protocol that allows a prover to convince a verifier that a committed value lies within a given range.*

Theorem 9.17 (Range Proof)**.** *Bulletproofs can prove that a committed value v satisfies $0 \leq v < 2^n$ for any positive integer n.*

9.4.8 Applications of Zero-Knowledge Proofs

1. **Authentication**: Prove identity without revealing credentials
2. **Blockchain Privacy**: Private transactions in cryptocurrencies
3. **Compliance**: Prove regulatory compliance without revealing sensitive data
4. **Decentralized Identity**: Anonymous credentials and selective disclosure
5. **Private Computation**: Prove computation results without revealing inputs
6. **Supply Chain**: Prove product authenticity without revealing trade secrets

9.4.9 Commitment Schemes

Definition 9.18 (Commitment Scheme)**.** *A commitment scheme allows a party to commit to a chosen value while keeping it hidden until they reveal it.*

Algorithm 22 Pedersen Commitment

1: Choose generators g, h of group G with prime order p
2: To commit to value v: $C = g^v h^r$ where r is random
3: To open: reveal (v, r)
4: Verification: check $C = g^v h^r$

9.4.10 Sigma Protocols

Definition 9.19 (Sigma Protocol). *A Sigma protocol is a three-move interactive proof system: commitment, challenge, response.*

Algorithm 23 Schnorr Identification Protocol

1: Prover chooses random k, sends $R = g^k$
2: Verifier sends random challenge c
3: Prover sends $s = k + cx$ where x is secret key
4: Verifier checks $g^s = R \cdot y^c$ where $y = g^x$ is public key

9.4.11 Proof Composition

Definition 9.20 (Proof Composition). *Proof composition allows combining multiple zero-knowledge proofs into a single proof.*

1. **AND Composition**: Prove multiple statements simultaneously
2. **OR Composition**: Prove at least one of several statements
3. **Proof Recursion**: Use proofs to verify other proofs

9.4.12 Trusted Setup

Definition 9.21 (Trusted Setup). *A trusted setup is a ceremony where secret parameters are generated and then destroyed, leaving only public parameters.*

Definition 9.22 (Universal Setup). *A universal setup can be used for any circuit of a given size, rather than requiring a new setup for each circuit.*

9.4.13 Post-Quantum Zero-Knowledge

Definition 9.23 (Post-Quantum Zero-Knowledge). *Post-quantum zero-knowledge proofs are resistant to attacks by quantum computers.*

1. **Lattice-based**: Using lattice problems for security
2. **Code-based**: Using error-correcting codes
3. **Multivariate**: Using systems of multivariate equations
4. **Isogeny-based**: Using elliptic curve isogenies

10 Cryptanalysis

10.1 Attack Types

1. **Ciphertext-only attack**: Attacker has only ciphertexts
2. **Known-plaintext attack**: Attacker has ciphertexts and corresponding plaintexts
3. **Chosen-plaintext attack**: Attacker can choose plaintexts and obtain ciphertexts
4. **Chosen-ciphertext attack**: Attacker can choose ciphertexts and obtain plaintexts

10.2 Statistical Attacks

10.2.1 Frequency Analysis

Definition 10.1 (Frequency Analysis). *Frequency analysis studies the frequency of letters or groups of letters in ciphertext to break substitution ciphers.*

10.3 Mathematical Attacks

10.3.1 Linear Cryptanalysis

Definition 10.2 (Linear Cryptanalysis). *Linear cryptanalysis exploits linear approximations of the cipher to recover the key.*

10.3.2 Differential Cryptanalysis

Definition 10.3 (Differential Cryptanalysis). *Differential cryptanalysis studies how differences in input pairs affect differences in output pairs.*

10.4 Side-Channel Attacks

Definition 10.4 (Side-Channel Attack). *Side-channel attacks exploit information leaked through physical implementation of cryptographic systems.*

1. **Timing attacks**: Exploit timing variations in operations
2. **Power analysis**: Exploit power consumption patterns
3. **Electromagnetic analysis**: Exploit electromagnetic emissions
4. **Cache attacks**: Exploit cache access patterns

11 Security Models and Proofs

11.1 Security Definitions

Definition 11.1 (Semantic Security). *A cryptosystem is semantically secure if no efficient algorithm can distinguish between encryptions of different messages.*

Definition 11.2 (Chosen-Plaintext Attack (CPA) Security). *A cryptosystem is CPA-secure if it remains secure even when the attacker can obtain encryptions of chosen plaintexts.*

11.2 Random Oracle Model

Definition 11.3 (Random Oracle). *A random oracle is an ideal hash function that returns truly random responses to unique queries.*

11.3 Provable Security

Theorem 11.4 (Security Reduction)**.** *If problem A is hard and cryptosystem C can be broken, then problem A can be solved efficiently.*

12 Implementation Considerations

12.1 Secure Random Number Generation

Definition 12.1 (Cryptographically Secure PRNG)**.** *A cryptographically secure pseudorandom number generator produces output that is computationally indistinguishable from true randomness.*

12.2 Constant-Time Implementation

Definition 12.2 (Constant-Time Algorithm)**.** *A constant-time algorithm takes the same amount of time to execute regardless of input values.*

12.3 Memory Management

1. **Secure memory allocation**: Prevent memory leaks of sensitive data
2. **Memory wiping**: Clear sensitive data from memory
3. **Memory protection**: Prevent unauthorized access to sensitive memory regions

13 Standards and Regulations

13.1 Cryptographic Standards

1. **FIPS 140-2**: Security requirements for cryptographic modules
2. **Common Criteria**: International standard for security evaluation
3. **NIST Guidelines**: Recommendations for cryptographic implementations

13.2 Export Controls

Definition 13.1 (Export Control)**.** *Export controls restrict the export of cryptographic software and hardware to certain countries.*

14 Applications

14.1 Digital Currency

Definition 14.1 (Cryptocurrency)**.** *Cryptocurrency is a digital currency secured by cryptography, typically using blockchain technology.*

14.2 Blockchain

Definition 14.2 (Blockchain)**.** *A blockchain is a distributed ledger maintained by a network of nodes using cryptographic techniques.*

14.3 Secure Communication

1. **Email encryption**: PGP, S/MIME
2. **Messaging**: Signal, WhatsApp
3. **VoIP**: SRTP, ZRTP

14.4 Authentication Systems

1. **Multi-factor authentication**: TOTP, HOTP
2. **Biometric authentication**: Fingerprint, face recognition
3. **Smart cards**: EMV, contactless payments

15 Future Directions

15.1 Post-Quantum Cryptography

Definition 15.1 (Post-Quantum Cryptography). *Post-quantum cryptography refers to cryptographic algorithms resistant to attacks by quantum computers.*

15.1.1 Lattice-Based Cryptography

Definition 15.2 (Lattice Problem). *A lattice problem involves finding short vectors in high-dimensional lattices.*

15.1.2 Code-Based Cryptography

Definition 15.3 (Code-Based Cryptography). *Code-based cryptography uses error-correcting codes to construct cryptographic primitives.*

15.2 Lightweight Cryptography

Definition 15.4 (Lightweight Cryptography). *Lightweight cryptography provides security for resource-constrained devices.*

15.3 Attribute-Based Encryption

Definition 15.5 (Attribute-Based Encryption). *Attribute-based encryption allows fine-grained access control based on attributes.*

16 Key Theorems and Results

Theorem 16.1 (Shannon's Perfect Secrecy). *A cryptosystem has perfect secrecy if and only if the key is at least as long as the message and used only once.*

Theorem 16.2 (Goldwasser-Micali Security). *The Goldwasser-Micali cryptosystem is semantically secure under the quadratic residuosity assumption.*

Theorem 16.3 (ElGamal Security). *The ElGamal cryptosystem is semantically secure under the decisional Diffie-Hellman assumption.*

Proposition 16.4 (Hash Function Security). *A hash function is collision-resistant if and only if it is second preimage resistant and preimage resistant.*

17 Conclusion

Cryptography is a fundamental field that provides the mathematical and algorithmic foundations for secure communication and data protection. The field encompasses:

- **Theoretical foundations**: Number theory, algebra, probability theory
- **Symmetric cryptography**: Block ciphers, stream ciphers, modes of operation
- **Asymmetric cryptography**: RSA, Diffie-Hellman, elliptic curves
- **Hash functions**: SHA family, Merkle-Damgård construction
- **Digital signatures**: RSA, ECDSA, DSA
- **Key management**: Distribution, PKI, perfect forward secrecy
- **Security protocols**: TLS, SSH, IPSec
- **Advanced topics**: Quantum cryptography, homomorphic encryption, zero-knowledge proofs

The field continues to evolve with:

- **Post-quantum cryptography**: Preparing for quantum computing threats
- **Lightweight cryptography**: Securing IoT and embedded devices
- **Privacy-preserving techniques**: Homomorphic encryption, secure multi-party computation
- **Quantum cryptography**: Quantum key distribution and quantum-resistant algorithms

Cryptography remains essential for protecting information in our digital world, balancing security requirements with practical implementation constraints while adapting to emerging threats and technological advances.