Comprehensive Cryptographic Standards Resume

Nathan Rayburn

November 10, 2024

Randomness Standards and PRNGs

[left=0pt]**True Random Number Generators (TRNG):** Utilizes physical sources such as thermal noise and quantum effects. Suitable for generating secure non-deterministic random values8:14†source. **Pseudorandom Number Generators (PRNG):** Deterministic sequences generated from a seed:

- – **Blum-Blum-Shub (BBS):** Secure if factoring large numbers is hard.

$$x_{i+1} = x_i^2 \mod n, \quad \text{where } n = pq \text{ and } p, q \equiv 3 \pmod 4$$

  – **Mersenne Twister:** Non-cryptographic; has a period of $2^{19937} - 1$8:21†source.

- **Cryptographic PRNGs (CPRNGs):** Meets unpredictability requirements:
  - **Hash_DRBG:** Utilizes hash functions like SHA-256; NIST SP800-90A standard.
  - **HMAC_DRBG:** Based on HMAC, providing stronger security but with slightly lower efficiency8:41†source.
  - **CTR_DRBG:** Uses block ciphers in counter mode; supports AES with 128, 192, or 256-bit keys8:42†source.

Symmetric Cryptography Standards

Block Ciphers

- **AES (Advanced Encryption Standard):**
  - **Block size:** 128 bits; **Key sizes:** 128, 192, and 256 bits.
  - **Standards:** NIST FIPS PUB 197, ISO/IEC 18033-39:9†source.
  - **Rounds:** 10, 12, or 14 based on key size.
- **Camellia:** Block size: 128 bits; Key sizes: 128, 192, 256 bits. ISO/IEC 18033-39:10†source.

Modes of Operation and IV/Nonce Usage

- **ECB (Electronic Codebook):** No IV; not recommended due to pattern leakage9:35†source.
- **CBC (Cipher Block Chaining):** Uses an IV; secure when a unique random IV is used each time:

$$C_i = E_k(P_i \oplus C_{i-1}), \quad C_0 = \text{IV}$$

- **CTR (Counter Mode):** Requires a nonce and counter, ensuring confidentiality but needing MAC for integrity:

$$C_i = P_i \oplus E_k(\text{Nonce}||\text{CTR}_i)$$

Hash Functions

- **SHA-2:** Provides 224, 256, 384, and 512-bit digests; NIST FIPS 180. Uses Merkle-Damgård construction9:21†source.
- **SHA-3:** Based on a sponge construction, providing resistance to length extension attacks. Supports SHAKE128, SHAKE2569:23†source.

Asymmetric Cryptography Standards

- **RSA:**
  - **Key Generation:** Select primes p and q:

$$n = p \cdot q, \quad \phi(n) = (p-1)(q-1)$$

  - **Keys:** Public key (n, e), private key $d \equiv e^{-1} \pmod{\phi(n)}$.
  - **Security Warning:** Avoid small exponents (e.g., $e = 3$) to prevent attacks.
- **Elliptic Curve Cryptography (ECC):**
  - **Common Curves:** Curve25519, P-256 for key exchange; Ed25519 for signatures7:27†source.
  - **ECDSA (Elliptic Curve Digital Signature Algorithm):** Signature generation:

$$r = (kG)_x \mod n, \quad s = k^{-1}(H(m) + d \cdot r) \mod n$$

Authenticated Encryption and MACs

MAC Schemes

- **HMAC:** Hash-based MAC; uses hash functions like SHA-256 to ensure message integrity:

$$HMAC(K, M) = \text{Hash}((K \oplus \text{opad})||\text{Hash}((K \oplus \text{ipad})||M))$$

  Standardized in RFC 21049:52†source.
- **Poly1305:** Fast MAC used with ChaCha20-Poly1305 authenticated encryption:

$$\text{MAC} = (Acc + s) \mod 2^{128}$$

Authenticated Encryption Schemes

- **Encrypt-then-MAC:**
  - **Description:** First encrypts the message, then computes the MAC on the ciphertext.
  - **Formula:**

$$C = \text{Encrypt}(P), \quad T = \text{MAC}(C)$$

    Transmit (C, T).
  - **Advantages:** Provides integrity and confidentiality. Preferred because the MAC authenticates both the ciphertext and the message.
  - **Recommendation:** Used in protocols like IPsec.
- **MAC-then-Encrypt:**
  - **Description:** Computes the MAC on the plaintext, then encrypts both the plaintext and MAC together.
  - **Formula:**

$$T = \text{MAC}(P), \quad C = \text{Encrypt}(P||T)$$

  - **Drawbacks:** Vulnerable to padding oracle attacks and may expose integrity information about plaintext if the encryption leaks details.
  - **Use Case:** Previously used in SSL/TLS, now deprecated due to vulnerabilities.

- **Encrypt-and-MAC:**
  - **Description:** Independently encrypts the message and computes the MAC on the plaintext.
  - **Formula:**
  $$C = \text{Encrypt}(P), \quad T = \text{MAC}(P)$$
  Transmit (C, T).
  - **Drawbacks:** Does not securely bind the ciphertext and MAC, making it vulnerable to attacks where the MAC can be tampered with separately.
  - **Recommendation:** Not recommended for new applications due to weaker security guarantees.

Galois/Counter Mode (GCM)

- **Overview:** AES-GCM combines AES in counter mode with a Galois field multiplication-based MAC for authenticated encryption.
- **IV Requirements:** Recommended 96-bit IV. Reusing IVs under the same key breaks confidentiality and integrity.
- **Counter Initialization:** For a 96-bit IV:
$$\text{Counter}_0 = \text{IV} || 00000001_{\text{32-bit}}$$
- **Message Size Constraints:** Up to $2^{32} - 2$ blocks per message; 2^{32} messages per key.

Digital Signatures

- **RSA-PSS (Probabilistic Signature Scheme):** RSA-based scheme for probabilistic signing, with random padding for improved security. Standard in PKCS #1 v2.27:38†source.
- **EdDSA (Edwards-curve Digital Signature Algorithm):** Deterministic signature scheme optimized for twisted Edwards curves like Ed255197:53†source.

References and Standards

- NIST FIPS PUB 197 - AES Standard
- NIST FIPS 180 - SHA-2 and SHA-3
- NIST SP800-90A - DRBG Standards
- RFC 2104 - HMAC Standard
- PKCS #1 v2.2 - RSA-PSS
- RFC 7539 - Poly1305 MAC