HORIZONSHIELD

Securing Your Digital Future, Today.

FOUNDATION SERIES · BOOK 15 OF 16

Applied Cryptography Guide

Algorithms, Protocols, PKI & Post-Quantum — 2026

Practical cryptography for security practitioners. Symmetric and asymmetric encryption, TLS 1.3 internals, PKI management, key management best practices, hashing, digital signatures, and post-quantum cryptography readiness planning.

ALGORITHMS	PROTOCOLS	TOPICS	LEVEL
30+	15	PKI, TLS, PQC	Intermediate

Contents

APPLIED CRYPTOGRAPHY 2026

CHAPTER 01

Cryptographic Primitives

Understanding what algorithms to use — and what to absolutely avoid — is the most practical cryptography knowledge for a security practitioner.

ALGORITHMS | AVOID LIST | KEY SIZES | CURRENT STANDARD

Current Standard vs. Deprecated Algorithms

Primitive	Use This (2026 Standard)	Never Use	Minimum Key Size	Reason for Deprecation
Symmetric encryption	AES-256-GCM (authenticated encryption)	DES, 3DES, RC4, RC2, Blowfish (key≤64bit)	256-bit key	DES: 56-bit key broken in 1997. 3DES: Sweet32 birthday attack. RC4: multiple biases.
Asymmetric encryption	RSA-4096, ECC P-384, X25519	RSA<2048, DSA<2048, ElGamal (obsolete)	RSA: 4096-bit; ECC: 384-bit	RSA-1024 factored; NIST deprecated 2048 below RSA for long-term use post-2030.
Key exchange	ECDHE (P-384), X25519	RSA key exchange (static), DH-1024, DH-768	ECDHE: 384-bit; X25519: 255-bit	Static RSA: no forward secrecy. Weak DH broken by Logjam attack.
Hashing	SHA-256, SHA-384, SHA-512, SHA-3	MD5, SHA-1	256-bit output minimum	MD5: full collision attack (Wang, 2004). SHA-1: SHAttered collision (Google, 2017).

Primitive	Use This (2026 Standard)	Never Use	Minimum Key Size	Reason for Deprecation
Password hashing	Argon2id (preferred), bcrypt (acceptable), scrypt	MD5, SHA-1, SHA-256 (all too fast)	Argon2id: 64MB RAM, 3 iterations minimum	Fast hashes: GPU can compute billions per second. Password hashing must be slow.
Digital signatures	ECDSA P-384, Ed25519, RSA-PSS-4096	RSA-PKCS1v1.5, DSA (any key size)	Ed25519: 255-bit; ECDSA: 384-bit	PKCS1v1.5 padding oracle attacks. DSA: vulnerable to nonce reuse (PS3 key extracted).
MAC / Message authentication	HMAC-SHA256, HMAC-SHA384, Poly1305	HMAC-MD5, CRC32, custom MACs	256-bit key minimum	HMAC-MD5: collision resistance issues though not directly exploitable yet.

AUTHENTICATED ENCRYPTION — AES-GCM IS MANDATORY

Unauthenticated encryption (AES-CBC without MAC, AES-ECB, AES-CTR) is vulnerable to padding oracle attacks, BEAST, and CRIME. Always use authenticated encryption: AES-256-GCM or ChaCha20-Poly1305. These provide BOTH confidentiality AND integrity — without a separate MAC step.

CHAPTER 04

TLS 1.3 — Protocol Deep Dive

TLS 1.3 removes every weak cipher and protocol feature that enabled the attacks against TLS 1.2. Understand the differences.

TLS 1.3 | FORWARD SECRECY | CIPHER SUITES | HANDSHAKE

TLS 1.3 vs TLS 1.2 — Key Differences

Feature	TLS 1.2	TLS 1.3	Security Impact
Handshake round trips	2 RTT (plus optional 0-RTT with session resumption)	1 RTT (0-RTT for session resumption)	Faster — but 0-RTT has replay attack risk
Forward secrecy	Optional — RSA key exchange still allowed (no FS)	Mandatory — ECDHE always used	PFS means past traffic cannot be decrypted if key compromised
Cipher suites	~80 suites including many weak/export ciphers	5 suites — ALL strong (AES-GCM and ChaCha20-Poly1305 only)	Eliminates entire classes of cipher-downgrade attacks
RSA key exchange	Allowed — enables massive passive decryption if server key compromised	Removed completely	NSA bulk decryption using static RSA is no longer possible
SHA-1 support	Allowed in certificate chains and HMAC	Removed entirely	SHA-1 collision attacks (SHAttered) no longer viable
RC4 support	Deprecated but may be negotiated in misconfigured deployments	Removed — not negotiable	RC4 biases enabling plaintext recovery eliminated

Feature	TLS 1.2	TLS 1.3	Security Impact
Compression	Supported — enabled CRIME attack	Removed	CRIME attack not possible in TLS 1.3
Renegotiation	Allowed — enabled renegotiation attacks	Removed	Entire renegotiation attack class eliminated

```
# Verify TLS configuration on your servers:
testssl.sh --protocols --ciphers --headers target.com

# Key things to verify:
# □ TLS 1.3 supported
# □ TLS 1.2 supported (for compatibility)
# □ TLS 1.1 and TLS 1.0 NOT offered
# □ SSLv3 NOT offered
# □ HSTS header present (Strict-Transport-Security: max-age=31536000; includeSubDomains)
# □ Forward secrecy: ECDHE in cipher suites
# □ RC4, DES, 3DES, EXPORT ciphers NOT offered
# □ Certificate uses SHA-256 or SHA-384 (not SHA-1)

# nginx TLS 1.3 configuration:
ssl_protocols TLSv1.3 TLSv1.2;
ssl_ciphers ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305;
ssl_prefer_server_ciphers off;  # Let client choose in TLS 1.3
ssl_session_timeout 1d;
ssl_session_cache shared:MozSSL:10m;
add_header Strict-Transport-Security 'max-age=63072000' always;
```

POST-QUANTUM CRYPTOGRAPHY — ACT NOW FOR LONG-LIVED DATA

NIST finalised first post-quantum cryptography standards in August 2024: ML-KEM (formerly CRYSTALS-Kyber) for key exchange; ML-DSA (CRYSTALS-Dilithium) for digital signatures. "Harvest now, decrypt later" attacks are a current threat — adversaries collecting encrypted traffic today to decrypt when quantum computers arrive. Begin crypto-agility planning immediately for any data with 10+ year secrecy requirements.

HS

H O R I Z O N S H I E L D

Securing Your Digital Future, Today.

Applied Cryptography Guide 2026

Part of the HorizonShield Security Series — 16 comprehensive professional cybersecurity manuals covering every domain of modern enterprise security.