Encryption Methods Comparison

Encryption Type	AES (Advanced Encryption Standard)	RSA (Rivest–Shamir–Adleman)	AES/RSA Hybrid Encryption
Encryption Type	Symmetric block cipher	Asymmetric public-key cipher	Hybrid symmetric-asymmetric
Key	Single shared key	Public key for encryption, Private key for decryption	AES key encrypts data; RSA Public key encrypts AES key
Speed	Very fast (hardware accelerated)	Very slow (exponential complexity)	Fast (bulk data uses AES; RSA only encrypts the key)
Security Assessment	Secure against brute-force (With sufficiently large keys); Quantum vulnerable (Grover's algo reduces effective key size)	Secure against brute-force; Vulnerable to quantum computing (Shor's algorith)	Inherits secuirty of both; weakest link principle applies
Use Case	Encrypting large files, disk encryption, VPN tunnels, databases	Key exchange, Digital Signatures, Authentication (SSH keys), Certificates	TLS/SSL, PGP, secure messaging
Typical Key Size	128/192/256 bits	2048/3072/4096 bits	AES-256 for data + RSA-2048/4096 for session key exchange
Advantages	High speed, low ressource usage, standard efficieny	Solves the key distribution problem (no need to share secret key beforehand)	Combines AES speed with RSA's secure key exchange
Disadvantages	Secure key distribution required	Impractical for large data	More complex implementation than using just one method

Encryption Terminology

Encryption	Use Case
Initialization Vector	Random nonce ensuring unique ciphertext for identical plaintext/key pairs
Operation Modes	Block cipher modes: GCM (authenticated), CBC (chained), ECB (insecure)
Block Chain	CBC mode: each block XORed with previous ciphertext block
Nonce	Number used once to prevent replay attacks and ensure protocol freshness

Cryptographic Primitives Explained

GCM (Galois/Counter Mode)

- **What:** Authenticated encryption mode. Encrypts *and* generates an authentication tag in one pass.
- **Why it matters:** Provides confidentiality, integrity, and authenticity. Detects tampered ciphertext before decryption (preventing attacks). Parallelizable and fast. **Best practice:** Use this for all new AES implementations. In code: `AES/GCM/NoPadding`.

CBC (Cipher Block Chaining)

- **What:** Each plaintext block is XORed with the previous ciphertext block before encryption.
- **Why it matters:** Older, widely supported, but slower (sequential) and vulnerable to padding oracle attacks if implemented incorrectly. Requires a separate HMAC for integrity. **Avoid if possible**; migrate to GCM.

ECB (Electronic Codebook)

- **What:** Each block encrypted independently with no chaining or randomization.
- **Why it matters: INSECURE.** Identical plaintext blocks produce identical ciphertext, leaking patterns (see "ECB penguin"). **Never use** for structured data. Only safe for encrypting random keys.

XOR (Exclusive OR)

- **What:** Bitwise operation: `0^0=0`, `0^1=1`, `1^0=1`, `1^1=0`.
- **Why it matters:** Reversible property `((A XOR B) XOR B) = A` makes it perfect for combining plaintext with keystreams in stream ciphers and chaining modes like CBC. Fundamental building block.

Ciphertext

- **What:** The encrypted output—pseudorandom binary data.
- **Why it matters:** Must be indistinguishable from random noise. **Never decrypt without verifying authentication** (if using AEAD). Treat as a raw byte array; encoding/decoding errors often indicate tampering.