

Criptografia Aplicada - parte 1


A comunicação em redes abertas, como a Internet, está sujeita a diversos tipos de ataques: escuta clandestina (eavesdropping), adulteração de mensagens, falsificação de identidade e negação de autoria. A criptografia é a principal ferramenta para mitigar esses riscos, fornecendo proteção tanto para dados em trânsito (e.g., transmissão de senhas, transações financeiras) quanto para dados em repouso (e.g., armazenamento em bancos de dados e dispositivos móveis).





Importância da Criptografia para a Segurança da Informação

A criptografia é a principal ferramenta para mitigar riscos de segurança, fornecendo proteção tanto para dados em trânsito (e.g., transmissão de senhas, transações financeiras) quanto para dados em repouso (e.g., armazenamento em bancos de dados e dispositivos móveis).

 Referência: Cap. 1, p. 21–23 — Tanenbaum & Wetherall destacam a necessidade da criptografia como componente essencial da segurança em redes de computadores, reforçando que ela viabiliza comunicação segura mesmo em ambientes hostis.

Conceitos Básicos da Segurança da Informação

O livro aponta um conjunto de propriedades fundamentais que a criptografia deve prover para garantir segurança:

Confidencialidade

- Assegura que somente partes autorizadas possam acessar a informação.
- Exemplo: no e-commerce, o número do cartão de crédito é cifrado para que apenas o servidor da instituição financeira possa decifrá-lo.
- No livro: p. 23 — descrita como requisito de impedir que terceiros leiam informações transmitidas em canais abertos.

Integridade

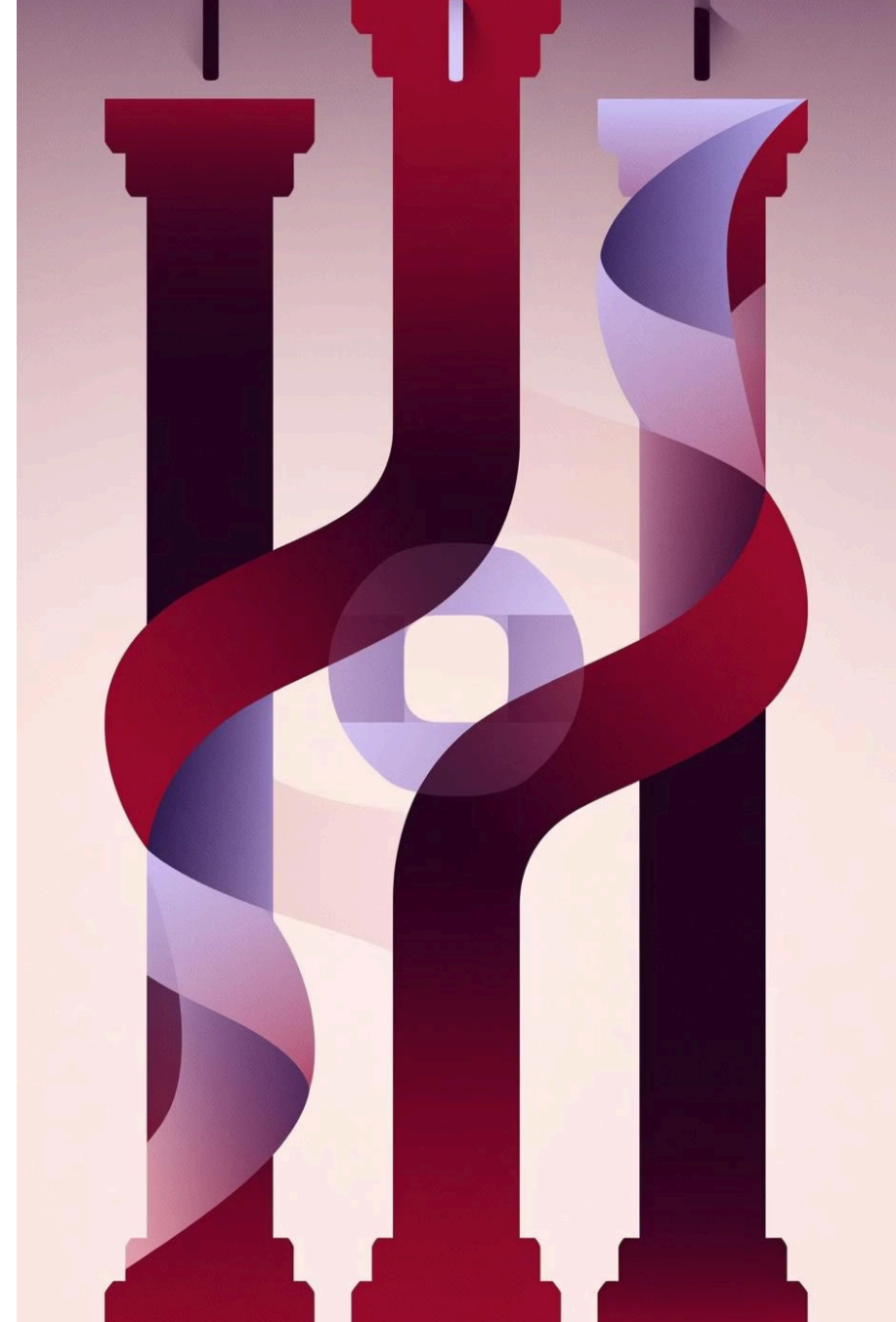
- Garante que os dados não sejam modificados, seja de forma accidental ou maliciosa, durante o armazenamento ou transmissão.
- Exemplo: funções de hash aplicadas em downloads de software para verificar se o arquivo não foi adulterado.
- No livro: p. 23 — discutida como essencial para assegurar que o conteúdo original não sofra alterações indetectáveis.

Autenticação

- Processo de verificar a identidade de uma entidade (usuário, sistema ou dispositivo).
- Exemplo: login em sistemas bancários via certificado digital ou token de autenticação.
- No livro: p. 23–24 — tratada como mecanismo para confirmar quem está participando da comunicação.

Não Repúdio (Irretratabilidade)

- Impede que um participante negue a autoria de uma ação ou mensagem enviada.
- Exemplo: assinaturas digitais aplicadas a contratos eletrônicos ou ordens de pagamento.
- No livro: p. 24 — apresentado como requisito associado às assinaturas digitais, garantindo responsabilidade legal.



Tipos de Criptografia

Criptografia Simétrica

A criptografia simétrica, também chamada de criptografia de chave secreta, utiliza a mesma chave tanto para o processo de cifragem quanto para o de decifragem. Isso significa que o emissor e o receptor devem compartilhar uma chave secreta previamente acordada.

 Referência: Cap. 2, p. 34–36 — onde Tanenbaum apresenta o conceito de cifra simétrica e a necessidade do compartilhamento da chave.



Exemplos Clássicos e Modernos de Criptografia Simétrica

1

DES (Data Encryption Standard)

- Criado pela IBM e adotado como padrão pelo NIST em 1977.
- Trabalha com blocos de 64 bits e chave de 56 bits.
- Vulnerável a ataques de força bruta devido ao tamanho reduzido da chave.
- 📖 p. 42–45: detalhamento do funcionamento do DES.

2

3DES (Triple DES)

- Extensão do DES que aplica o algoritmo três vezes com chaves diferentes (ou duas chaves reutilizadas).
- Considerado mais seguro que o DES, mas mais lento.
- 📖 p. 47: apresentação do 3DES como evolução do DES.

3

AES (Advanced Encryption Standard)

- Selecionado pelo NIST em 2001 para substituir o DES.
- Baseado no algoritmo Rijndael, utiliza blocos de 128 bits e chaves de 128, 192 ou 256 bits.
- Mais eficiente e seguro contra ataques conhecidos.
- 📖 p. 48–50: descrição do AES e justificativas para sua escolha como padrão.

4

RC4

- Algoritmo de fluxo, muito usado no passado em protocolos como SSL e WEP.
- Rapidez na cifragem de dados em fluxo contínuo.
- Vulnerabilidades graves foram descobertas, tornando-o obsoleto em sistemas modernos.
- 📖 p. 53: descrição do RC4 e seu uso em protocolos de rede.

Vantagens e Desvantagens da Criptografia Simétrica

Vantagens

- Rapidez: operações matemáticas menos complexas, permitindo cifragem em alta velocidade.
- Baixo custo computacional: adequado para grandes volumes de dados.
- Eficiência em hardware e software.
- 📖 p. 36 — Tanenbaum destaca a eficiência das cifras simétricas para transmissão de grandes quantidades de informação.

Desvantagens

- Distribuição de chaves: principal problema, pois todos os participantes precisam ter acesso à chave secreta antes da comunicação.
- Escalabilidade: em redes com muitos usuários, o número de chaves necessárias cresce rapidamente $(N(N-1)/2)$ para N usuários).
- Menor flexibilidade: não oferece mecanismos de autenticação ou não repúdio sozinha.
- 📖 p. 36–38 — discussão sobre os problemas de distribuição e gerenciamento de chaves.

Aplicações Práticas: Criptografia de discos e arquivos (BitLocker, VeraCrypt), VPNs (IPSec), Sistemas de backup.

Criptografia Assimétrica

A criptografia assimétrica, também chamada de criptografia de chave pública, utiliza um par de chaves distintas:

Chave Pública

Pode ser livremente distribuída e usada para cifrar mensagens.

Chave Privada

Mantida em segredo e usada para decifrar mensagens ou assinar digitalmente documentos. Esse modelo resolve o problema da distribuição de chaves, comum nas cifras simétricas.

📖 Referência: Cap. 9, p. 292–295 — introdução ao conceito de criptografia de chave pública e comparação com a criptografia simétrica.



Exemplos de Criptografia Assimétrica



RSA (Rivest–Shamir–Adleman)

- Baseado na dificuldade da fatoração de números grandes.
- Usado tanto para cifragem de dados quanto para assinaturas digitais.
- Amplamente adotado em protocolos de segurança (TLS, certificados digitais, VPNs).
- 📖 p. 295–302 — descrição do funcionamento do RSA, incluindo geração de chaves e operações de cifrar/decifrar.



ElGamal

- Baseado no problema do logaritmo discreto em corpos finitos.
- Oferece segurança probabilística, pois incorpora aleatoriedade no processo de cifragem.
- Utilizado em sistemas de assinatura digital (ex.: DSS – Digital Signature Standard).
- 📖 p. 303–305 — explicação do ElGamal e sua aplicação em assinaturas digitais.



ECC (Elliptic Curve Cryptography)

- Baseada na dificuldade do logaritmo discreto em curvas elípticas.
- Proporciona o mesmo nível de segurança que RSA, mas com chaves muito menores (e.g., ECC 256 bits \approx RSA 3072 bits).
- Muito usada em dispositivos móveis e IoT, onde há limitação de processamento e energia.
- 📖 p. 308–310 — introdução à ECC e vantagens em termos de eficiência e segurança.

Vantagens e Desvantagens da Criptografia Assimétrica

Vantagens

- Distribuição de chaves simplificada: basta publicar a chave pública.
- Oferece autenticação e não repúdio via assinaturas digitais.
- Mais escalável em grandes redes, já que não é necessário compartilhar segredos prévios.
- 📖 p. 295–296 — destaque para a resolução do problema da distribuição de chaves.

Desvantagens

- Baixa performance: operações matemáticas complexas tornam-na lenta.
- Ineficiente para grandes volumes de dados: geralmente é usada apenas para troca de chaves ou assinatura, não para criptografar arquivos inteiros.
- 📖 p. 297–298 — discussão sobre limitações de desempenho em comparação com cifras simétricas.

Aplicações Práticas: Certificados digitais (HTTPS, S/MIME), Assinaturas eletrônicas (contratos digitais), Troca segura de chaves (TLS/SSL).





Funções de Hash

Uma função de hash criptográfica é uma função matemática unidirecional que transforma uma entrada de tamanho arbitrário em uma saída de tamanho fixo, chamada resumo (digest).

- A função deve ser determinística: a mesma entrada sempre gera a mesma saída.
- É usada como mecanismo de verificação de integridade e em protocolos de autenticação.

📖 Referência: Cap. 11, p. 371–373 — definição de funções de hash e introdução ao seu papel na criptografia.

Exemplos de Funções de Hash

MD5 (Message Digest 5)

- Produz um hash de 128 bits.
- Muito utilizado no passado, mas atualmente considerado inseguro devido à existência de colisões práticas.
- 📖 p. 375–376 — descrição do MD5 e suas vulnerabilidades.

SHA-1 (Secure Hash Algorithm 1)

- Produz um hash de 160 bits.
- Foi amplamente usado em certificados digitais e assinaturas eletrônicas.
- Tornou-se obsoleto após demonstrações de colisões viáveis (ataque SHAttered em 2017).
- 📖 p. 377–378 — explicação sobre o SHA-1 e seus problemas de segurança.

SHA-2 (Secure Hash Algorithm 2)

- Família de funções (SHA-224, SHA-256, SHA-384, SHA-512).
- Baseadas em melhorias sobre o SHA-1, com maior resistência a colisões.
- Atualmente recomendada para aplicações seguras.
- 📖 p. 378–380 — descrição da família SHA-2 e suas variantes.

Propriedades Criptográficas Essenciais das Funções de Hash



Efeito Avalanche

Pequenas alterações na entrada geram grandes mudanças no resultado.

📖 p. 373 — exemplo prático de avalanche em funções de hash.



Resistência à Colisão

Deve ser computacionalmente inviável encontrar duas entradas diferentes que resultem no mesmo hash.

📖 p. 374 — discussão sobre a importância da resistência à colisão.



Resistência à Inversão (pré-imagem)

Dado o valor do hash, deve ser impossível recuperar a entrada original.

📖 p. 374–375 — análise da propriedade de pré-imagem e sua relevância em segurança.

Aplicações Práticas: Verificação de integridade (sha256sum), Armazenamento de senhas (com salt), Assinaturas digitais, Blockchain.

Protocolos Criptográficos

TLS/SSL

TLS/SSL protege a comunicação fim-a-fim (cliente–servidor) acima do TCP, oferecendo confidencialidade e integridade para protocolos de aplicação como HTTP (origem do "HTTPS"). O livro descreve SSL/TLS como um conjunto de protocolos amplamente empregado pelos navegadores e servidores Web, com TLS padronizado na RFC 5246.

Arquitetura em camadas

SSL/TLS não é um único protocolo:

- Protocolo de Registro (Record Protocol): fornece confidencialidade (cifra simétrica) e integridade (MAC/HMAC) para os dados da aplicação.
- Protocolos de controle em cima do Registro: Handshake, Change Cipher Spec e Alert. Essa organização é mostrada e explicada no texto (Figura 17.2).

Handshake: visão geral e fases

O Handshake autentica as partes (normalmente o servidor), negocia o conjunto de cifras e estabelece chaves para a conexão segura.

Fase 1 — Capacidades de segurança

Cliente envia `client_hello` com versão, random, ID de sessão, lista de conjuntos de cifras e métodos de compressão. O servidor responde com `server_hello` escolhendo versões/métodos e um único conjunto de cifras da lista do cliente.

Fase 3 — Ações do cliente

Cliente (opcionalmente) envia `certificate`, depois `client_key_exchange` com o material para `pre_master_secret`; se enviou certificado, prova posse da chave privada em `certificate_verify`.

Depois do `pre_master_secret`, as partes derivam o `master_secret` e, a partir dele, o `key_block` com os parâmetros necessários: segredos MAC de escrita (cliente/servidor), chaves de escrita e IVs (quando modo CBC). O livro detalha as fórmulas para SSLv3 (MD5/SHA encadeados) e o encadeamento do `key_block`.

Fase 2 — Autenticação do servidor e troca de parâmetros

Servidor envia `certificate` (cadeia X.509) e, conforme o método, `server_key_exchange`; pode solicitar autenticação do cliente via `certificate_request`; finaliza com `server_hello_done`.

Fase 4 — Ativação da cifra e verificação mútua

Ambos trocam `change_cipher_spec` (muda do estado pendente para o atual) e `finished` (hash das mensagens do Handshake autenticado pelo segredo), encerrando a negociação.