


# Criptografia Aplicada

# 1. INTRODUÇÃO

# 1.1 Importância da Criptografia para a Segurança da Informação

- 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).
-  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.

## 1.2 Conceitos Básicos da Segurança da Informação

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

## 1.2.1 Confidencialidade

- Definição: assegura que somente partes autorizadas possam acessar a informação.
- Exemplo prático: no envio de dados via e-commerce, o número do cartão de crédito é cifrado de forma 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.

## 1.2.2 Integridade

- Definição: garante que os dados não sejam modificados, seja de forma acidental ou maliciosa, durante o armazenamento ou transmissão.
- Exemplo prático: 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.

## 1.2.3 Autenticação

- Definição: processo de verificar a identidade de uma entidade (usuário, sistema ou dispositivo).
- Exemplo prático: 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.


## 1.2.4 Não Repúdio (Irretratabilidade)

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





## **2. TIPOS DE CRIPTOGRAFIA**

## 2.1 Criptografia Simétrica (Cap. 2)

- 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

- 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.


# Exemplos Clássicos e Modernos

- 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

- 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: softwares como BitLocker (Windows) e VeraCrypt usam AES para proteger dados armazenados.
- VPNs: protocolos como IPSec utilizam criptografia simétrica (AES, 3DES) para criar túneis seguros.
- Sistemas de backup: criptografia de grandes volumes de dados em servidores.
-  p. 50–52 — exemplos de aplicações reais de cifras simétricas em sistemas de comunicação e armazenamento.

## 2.2 Criptografia Assimétrica (Cap. 9)


- 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 Clássicos e Modernos

- 1. 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.
- 2. 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.


# Exemplos Clássicos e Modernos

- 3. ECC (Elliptic Curve Cryptography – Criptografia de Curvas Elípticas)
- 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

- 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: usados em protocolos HTTPS, e-mails seguros (S/MIME) e autenticação em VPNs.
- Assinaturas eletrônicas: contratos digitais, documentos oficiais e sistemas de governo eletrônico.
- Troca segura de chaves: utilizada em conjunto com criptografia simétrica em protocolos como TLS/SSL.
-  p. 310–312 — exemplos de uso em certificação digital e protocolos de segurança.

## 2.3 Funções de Hash (Cap. 11)

- 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


- 1. 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.
- 2. 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.

# Exemplos de Funções de Hash


- 3. 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

- 3. 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: garantir que um arquivo baixado não foi alterado (uso de sha256sum).
- Armazenamento de senhas: sistemas não armazenam senhas em claro, apenas seus hashes (com salt).
- Assinaturas digitais: antes de assinar, documentos são convertidos em hashes para reduzir o custo computacional.
- Blockchain: cada bloco contém o hash do anterior, garantindo a imutabilidade da cadeia.
-  p. 381–383 — exemplos de uso prático em assinaturas digitais e protocolos de segurança.

# **3. PROTOCOLOS CRIPTOGRÁFICOS**

## 3.1 TLS/SSL (Cap. 7 e 8)

- 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.

# Sessão × Conexão (e reuso)

- O material distingue sessão (parâmetros criptográficos negociados pelo Handshake e potencialmente reutilizáveis) e conexão (associação ponto-a-ponto transitória que usa os parâmetros da sessão). O estado de sessão inclui, por exemplo, identificador de sessão, especificação de cifra, segredo mestre (master\secret, 48 bytes) e se a sessão é retomável; o estado de conexão inclui aleatórios do cliente/servidor, segredos MAC de escrita, chaves de escrita e IVs.

# 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. As mensagens e parâmetros principais (como client\hello, server\hello, certificate, server\key\exchange, certificate\request, certificate\verify, client\key\exchange, change\cipher\spec, finished) e seus campos estão na descrição das quatro fases do processo (com os nonces para mitigar replay).

# Handshake: visão geral e fases

- 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 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`.



# Handshake: visão geral e fases

- 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.
- 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.

# Handshake: visão geral e fases

- Métodos de troca de chaves (exemplos):
  - RSA: cliente gera um pre-master-secret de 48 bytes, cifra com a chave pública do servidor (do certificado) e envia; o servidor decifra com a chave privada.
  - Diffie–Hellman (estático ou efêmero): cliente e servidor trocam valores DH e ambos calculam o pre-master-secret.
- Conjuntos de cifras (cipher suites): cada suite nomeia método de troca de chaves e especificação de cifra (algoritmo de dados em massa + algoritmo de hash para MAC). O TLS herda quase todos os métodos do SSLv3 (com exceções como Fortezza).

# Derivação de segredos e chaves

- 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`.
- No TLS, essas funções são unificadas numa PRF (Pseudorandom Function) e o MAC passa a ser HMAC (MD5 ou SHA-1), com escopo ligeiramente diferente (inclui, por exemplo, a versão). O texto mostra a forma geral da PRF e do cálculo do HMAC/PRF.

# Protocolo de Registro (Record)

- O Record fragmenta ( $\leq 2^{14}$  bytes), (opcionalmente) comprime, calcula MAC/HMAC com número de sequência e metadados do fragmento, cifra (cifra simétrica escolhida na suite) e anexa cabeçalho antes de enviar via TCP. Também define os tipos de conteúdo (change\cipher\spec, alert, handshake, application\data). O material cobre ainda o preenchimento para cifras de bloco e o formato do cabeçalho do registro.
- Serviços oferecidos pelo Record: confidencialidade (cifra simétrica) e integridade (MAC/HMAC), ambos baseados em chaves que o Handshake negociou.

# Alertas, Change Cipher Spec e fechamento

- Change Cipher Spec é uma mensagem de 1 byte que ativa a especificação de cifra negociada (muda do estado pendente para o atual). Alert transporta avisos/erros (p. ex., unknown\ca, decrypt\error, insufficient\security etc.). Em HTTPS, o fechamento correto envolve o envio de close\notify pelo TLS antes de encerrar o TCP; a ausência desse alerta pode indicar problema/ataque e deve gerar aviso.


# Como TLS/SSL combina simétrica, assimétrica e hash (na prática)

- Assimétrica (RSA/DH): empregada no Handshake para autenticação (certificados X.509) e/ou estabelecimento de segredo inicial (pre\master\secret).
- Hash/HMAC: usado para integridade e autenticação de mensagens no Record (HMAC no TLS) e para verificar o Handshake (finished, certificate\verify).
- Simétrica: usada para criptografar os dados da aplicação (alta performance) após a troca de chaves.

## 3.2 HTTPS – HTTP sobre TLS/SSL

- O HTTPS (HyperText Transfer Protocol Secure) nada mais é do que o HTTP encapsulado dentro do TLS/SSL.
- Enquanto o HTTP puro é transmitido em texto claro, o HTTPS garante que toda a troca de informações entre cliente e servidor ocorra em um canal seguro.
- O navegador inicia uma conexão TCP com o servidor e, em seguida, dispara o handshake TLS/SSL. Só depois disso o protocolo HTTP começa a ser usado dentro do túnel seguro.
- Esse processo protege contra espionagem (eavesdropping), modificação de dados (man-in-the-middle) e garante a autenticidade do servidor por meio de certificados digitais.

# Serviços de Segurança Oferecidos


- 1. Confidencialidade
- A comunicação é cifrada com algoritmos simétricos (como AES ou 3DES), negociados durante o Handshake.
- Garante que informações sensíveis (senhas, números de cartão de crédito, dados pessoais) não sejam legíveis por terceiros.
-  p. 437 — Tanenbaum destaca a importância da cifra simétrica para o tráfego de dados de aplicação.




# Serviços de Segurança Oferecidos

- 2. Integridade
- As mensagens HTTP são acompanhadas de códigos de autenticação de mensagem (HMAC), garantindo que não tenham sido alteradas durante a transmissão.
- 📖 p. 431–432 — explicação sobre o uso de HMAC no Record Protocol do TLS, herdado por HTTPS.


# Serviços de Segurança Oferecidos

- 3. Autenticação
- O servidor apresenta um certificado digital X.509, assinado por uma Autoridade Certificadora (CA), que prova sua identidade ao cliente.
- Opcionalmente, também é possível autenticar o cliente (em sistemas corporativos e bancos).
-  p. 420–423 — detalhamento do uso de certificados e do campo certificate no Handshake TLS.

# Serviços de Segurança Oferecidos

- 4. Não Repúdio
- Quando combinado com assinaturas digitais e registros de transação, HTTPS pode dar suporte a sistemas de não repúdio (ex.: comprovar que uma transação de pagamento foi realmente feita).
-  p. 424 — menção ao papel das assinaturas digitais no TLS/SSL.


# Aplicações Concretas

- Transações bancárias: login em contas, transferências e pagamentos online.
- Comércio eletrônico (e-commerce): compras em sites como Amazon, Mercado Livre, etc.
- Acesso a sistemas corporativos e governamentais: portais de serviços, intranets, sistemas de declaração de impostos.
- Proteção de dados pessoais: redes sociais, cadastros, formulários e comunicações privadas.
-  p. 438 — o livro cita o uso massivo em comércio eletrônico e portais que precisam garantir segurança ao usuário.

# Exemplo Prático em Sala


- 1. Inspeção de certificado em navegador
- Acesse um site seguro: <https://www.bb.com.br> ou <https://www.ufrj.br>.
- Clique no cadeado ao lado da URL → “Exibir certificado”.
- Analise os campos:
  - Emissor (CA): a autoridade certificadora responsável.
  - Sujeito: o domínio e, muitas vezes, a organização responsável.
  - Período de validade: datas de início e expiração do certificado.
  - Algoritmo de chave pública: geralmente RSA ou ECC.

# Exemplo Prático em Sala

- 2. Discussão em grupo
- O que acontece se um certificado está expirado ou emitido por uma CA não confiável?
- Como os navegadores avisam ao usuário (ex.: alerta “Não seguro”)?
- Relacionar com ataques reais: phishing e ataques man-in-the-middle usando certificados falsos.
-  Referência: Cap. 17, p. 437–438 — o livro recomenda observar o uso de certificados para validação do servidor em HTTPS.


## **4. APLICAÇÕES PRÁTICAS E CASOS REAIS**

## 4.1 VPNs (Virtual Private Networks)


- As VPNs criam túneis criptografados através de redes públicas (como a Internet), garantindo comunicação segura entre duas redes privadas ou entre usuário e empresa.
- Objetivo: confidencialidade, integridade e autenticação do tráfego.
- Protocolos utilizados:
  - IPSec: oferece dois modos (transporte e túnel), usa criptografia simétrica (AES, 3DES) para dados e assimétrica para troca de chaves.
  - SSL/TLS VPNs: permitem acesso seguro a aplicações via navegador.
  - Aplicações: acesso remoto de funcionários, interconexão entre filiais de empresas.
-  Referência: Cap. 8, p. 273–275 — discussão sobre VPNs baseadas em IPSec e SSL/TLS.




## 4.2 E-mails Seguros

- A proteção de e-mails requer garantir confidencialidade, integridade, autenticação e não repúdio.
- Protocolos principais:
- PGP (Pretty Good Privacy): combina simétrica (para cifrar a mensagem), assimétrica (para trocar a chave de sessão) e funções de hash (para assinaturas digitais).
- S/MIME (Secure/Multipurpose Internet Mail Extensions): padrão baseado em certificados X.509, amplamente usado em clientes de e-mail corporativos.
- Aplicações: comunicação corporativa, governo eletrônico, envio de documentos sigilosos.
-  Referência: Cap. 8, p. 281–284 — seção sobre segurança em e-mail com PGP e S/MIME.

## 4.3 Blockchain e Criptomoedas

- Embora o livro não trate diretamente de blockchain (a edição é de 2014), ele aborda os fundamentos aplicados nesse contexto:
- Hashing: cada bloco contém o hash do anterior, garantindo imutabilidade e integridade da cadeia.
- Assinaturas digitais (ECDSA): garantem que apenas o dono da chave privada pode autorizar transações.
- Criptografia simétrica: usada em algumas camadas de privacidade em sistemas distribuídos.
- Aplicações: Bitcoin, Ethereum, contratos inteligentes.
-  Referência indireta: Cap. 11, p. 371–380 (funções de hash) e Cap. 9, p. 308–310 (ECC), que são a base dos mecanismos usados no blockchain.

## 4.4 Autenticação e Controle de Acesso

- A criptografia é usada em sistemas de autenticação para proteger credenciais e validar identidades.
- Senhas protegidas com hash: em sistemas operacionais e bancos de dados.
- Protocolos de autenticação:
  - Kerberos: usa chaves simétricas e servidores de autenticação para validar usuários em redes corporativas.
  - Certificados digitais: autenticação baseada em chaves públicas, amplamente usada em VPNs e HTTPS.
  - Aplicações: login em sistemas bancários, redes corporativas, portais governamentais.
-  Referência: Cap. 8, p. 268–272 — seção sobre protocolos de autenticação (Kerberos, certificados).

## 5. Conclusão e Atividade

# Caso Real: Vazamento de Dados da Equifax (2017)

- Uma das maiores agências de crédito dos EUA sofreu um vazamento que expôs dados de 147 milhões de pessoas.
- Investigação mostrou que a criptografia não foi corretamente aplicada em partes do sistema e que conexões HTTPS estavam mal configuradas.
- Impactos: roubo de identidades, perda de confiança dos clientes, multas milionárias.

# Caso Real: Vazamento de Dados da Equifax (2017)

- 1. Onde houve falhas de segurança? – Relacionar com os conceitos estudados (TLS, HTTPS, hash de senhas, gestão de chaves).
- 2. Como a criptografia poderia ter evitado ou reduzido o impacto do ataque?
- 3. Que boas práticas as empresas devem adotar para proteger dados sensíveis?
- 4. Como equilibrar segurança, custo e usabilidade em sistemas que lidam com milhões de usuários?