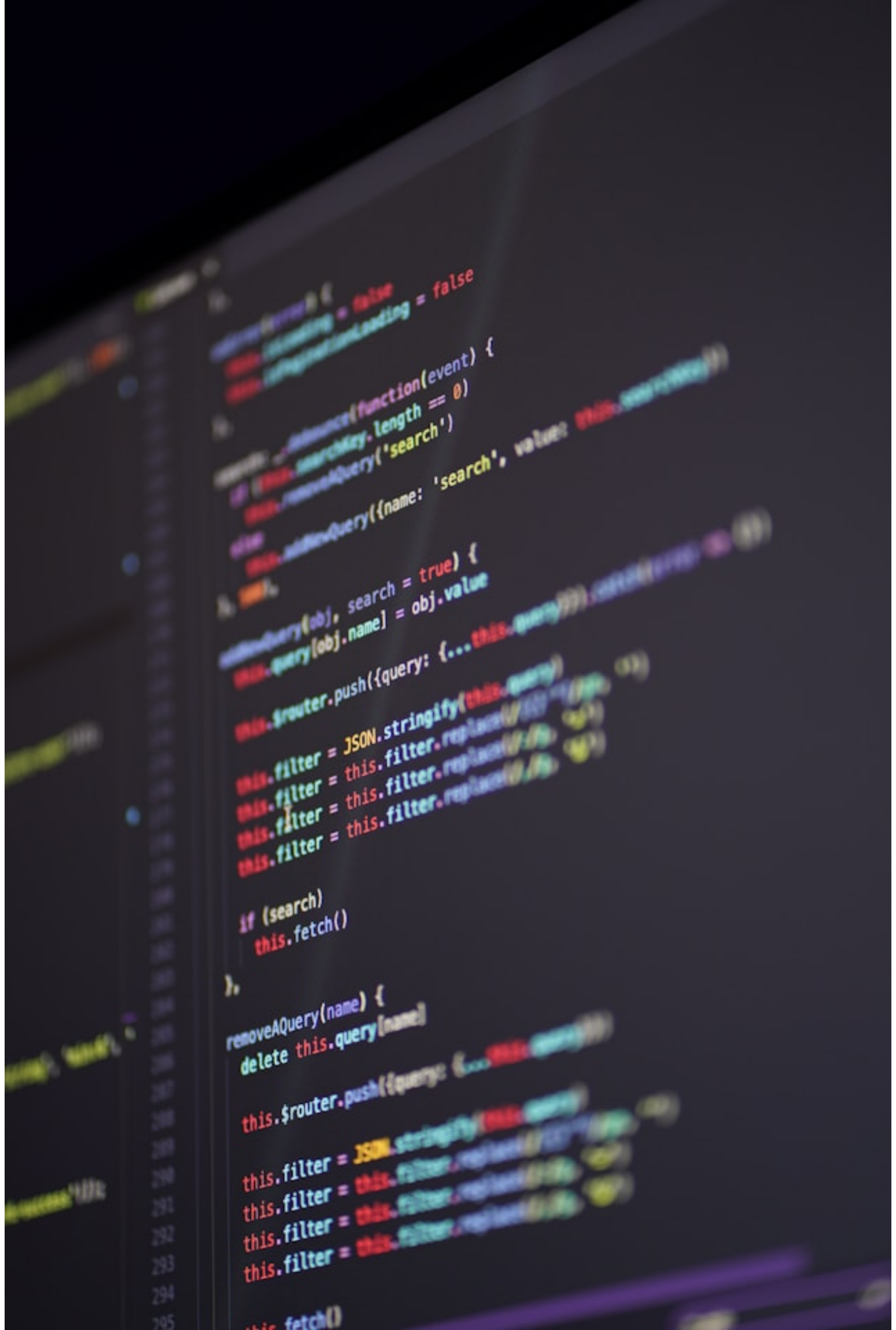


5954025 - Sistemas Distribuídos

Aula 13a - Segurança (Parte 1)

Prof. Dr. Denis M. L. Martins
DCM | FFCLRP | USP

Consequências das Aulas Anteriores

Gerenciamento de Transações Distribuídas

- **Sistemas Tightly Coupled (Fortemente Acoplados):** Interdependência substancial entre participantes. Transações requerem propriedades ACID e tipicamente usam comunicação síncrona.
- **Sistemas Loosely Coupled (Frouxamente Acoplados):** Interações baseadas em mensagens assíncronas e dados replicados localmente. Frequentemente aplicam *optimistic concurrency* (concorrência otimista) e tipicamente resultam em semântica BASE.
  - Em *big data settings*, a indisponibilidade é frequentemente mais custosa do que a inconsistência temporária, e o *overhead* de *locking* (necessário para consistência) tem um impacto severo na performance.

Consistência Eventual

Garantir a ordenação global de operações conflitantes é **custoso** e pode **degradar a escalabilidade**.

- **Consistência Eventual**: Na ausência de novas atualizações, todas as atualizações realizadas até aquele ponto são propagadas, de modo que as réplicas eventualmente terão os mesmos dados armazenados.
  - **Otimização:** Permite evitar a sincronização global, favorecendo a performance.
- **Transações BASE (NoSQL)**: Sistemas NoSQL de larga escala (como DynamoDB ou Bigtable) que abandonam a estrita consistência ACID frequentemente adotam o modelo BASE.

> Guia visual para consistência eventual em https://eda-visuals.boyney.io/visuals/eventual-consistency.

Modelo/Propriedades BASE

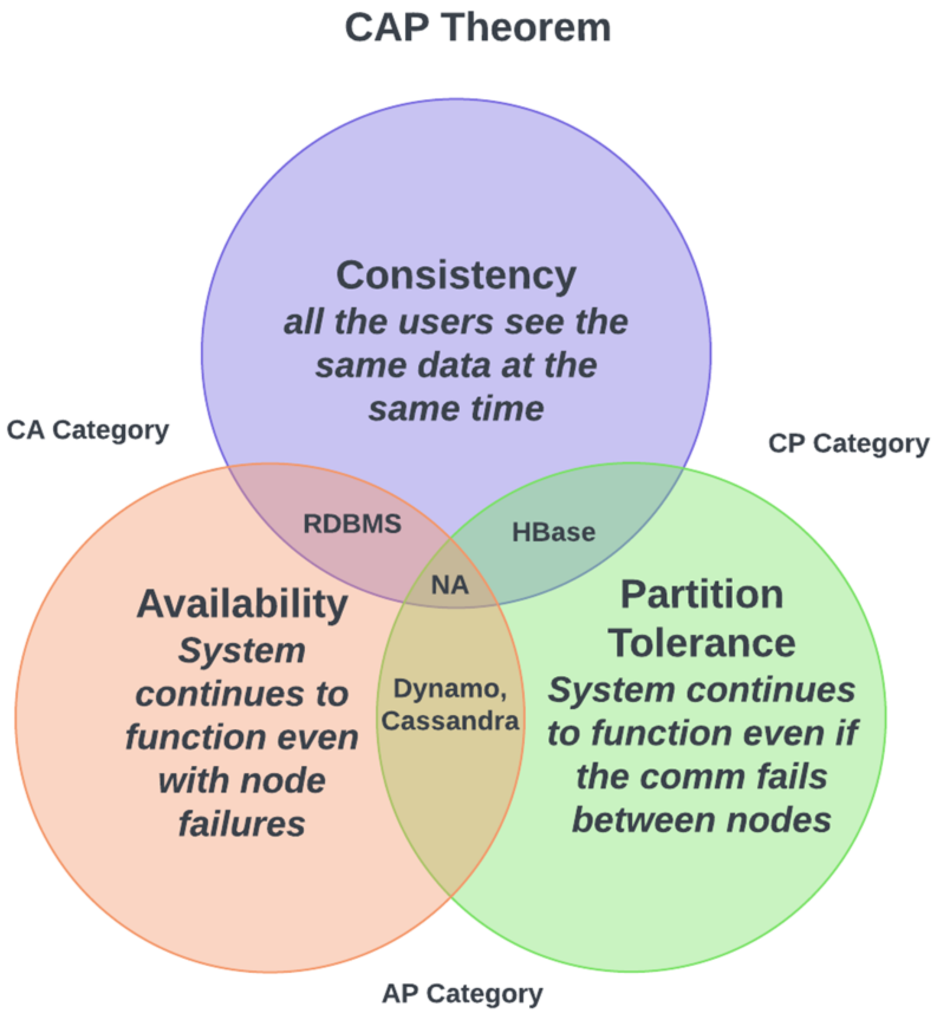
- **B**asically **A**vailable: Medidas garantem a disponibilidade sob todas as circunstâncias, se necessário, ao custo da consistência.
- **S**oft State: O estado pode evoluir (mudar) mesmo sem entrada externa, devido à propagação assíncrona de atualizações.
- **E**ventually Consistent: O banco de dados se tornará consistente ao longo do tempo, mas pode não ser consistente em qualquer momento, especialmente no commit da transação.

O Teorema CAP

O Teorema CAP afirma que é impossível para um sistema distribuído satisfazer simultaneamente as três propriedades:

- **Consistência** (Consistency): Todos os nós veem os mesmos dados ao mesmo tempo.
- **(Alta) Disponibilidade** (Availability): Toda requisição recebe uma resposta de sucesso ou falha.
- **Tolerância à Partição** (Partition tolerance): O sistema continua operando mesmo se falhas de rede impedirem a comunicação entre nós.

> Fonte da Imagem: Explaining the CAP Theorem and its Limitations.



Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Exemplos de Dilema CAP

1. **Escolha CP (Consistência + Tolerância à Partição):**
   Se houver uma partição de rede, o sistema deve garantir a consistência, o que significa que ele pode se tornar indisponível para algumas requisições até que a partição seja resolvida (Ex: Bancos de Dados Relacionais Distribuídos).

2. **Escolha AP (Disponibilidade + Tolerância à Partição):**
   Se houver uma partição de rede, o sistema deve permanecer disponível, mas pode retornar dados temporariamente inconsistentes (Ex: A maioria dos sistemas NoSQL de larga escala).

> **Observação**: **P** é obrigatório. Em sistemas reais, falhas de rede (partições) são inevitáveis. Portanto, o sistema deve sempre ser tolerante a partições.

Aula de Hoje: Segurança

Objetivos de Aprendizagem

- Entender os princípios de segurança em sistemas distribuídos.
- Compreender mecanismos de segurança.

Motivação

A natureza descentralizada desses sistemas, embora promova escalabilidade e resiliência, também os torna mais vulneráveis a falhas de segurança, interceptações e ataques coordenados.

- **Safety (Segurança Funcional/Operacional):** Proteção contra eventos não intencionais, naturais ou aleatórios (falhas).
  - Foco: Prevenir acidentes e falhas de sistema.
- **Security (Segurança Cibernética):** Proteção contra ataques ou ameaças deliberadas.
  - Foco: Defender-se de agentes maliciosos.

Princípios de Segurança

Para proteger sistemas distribuídos, cinco propriedades fundamentais devem ser garantidas:

- **Confidencialidade** (*Confidentiality*): proteção contra acesso não autorizado às informações.
  - Proteção contra divulgação indevida de informação.
- **Integridade** (*Integrity*): garantia de que os dados não foram modificados de forma maliciosa ou acidental.
  - Proteção contra modificação indevida de informação.
- **Disponibilidade** (*Availability*): serviços devem estar acessíveis mesmo sob falhas ou ataques.
  - Proteção contra negação indevida de uso/serviço.

> Esses princípios norteiam o *design* de protocolos e mecanismos de proteção distribuída, especialmente em ambientes como redes públicas e infraestruturas multitenant em nuvem.

Pilares Adicionais da Segurança

Autenticidade: Verificar a identidade real de um usuário ou dispositivo.

- *Mecanismo:* Senha, biometria, token físico.

Não-Repúdio: Impedir que uma entidade negue ter realizado uma ação.

- *Exemplo:* Assinatura digital em um contrato legal.

Ameaças

Os dados e serviços de um sistema computacional devem ser protegidos contra diferentes tipos de ameaças de segurança:

- **Interceptação**: uma entidade não autorizada obtém acesso aos serviços e/ou dados do sistema.
- **Interrupção**: serviços e/ou dados do sistema tornam-se indisponíveis
  - Exemplos: corrupção ou perda de dados, ataques de negação de serviço.
- **Modificação**: mudanças não autorizadas nos dados e serviços são efetuadas, de modo que estes não atendam a suas especificações originais.
- **Fabricação**: dados ou atividades, que de outro modo não existiriam, são gerados.
  - Exemplos: entrada de novos dados no sistema, captura e reenvio de mensagens antigas de modo a entrar no sistema

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Modelos de Ataque

Estratégico: Direcionado (*Targeted*): Especializado. Busca o "elo mais fraco" ou a vulnerabilidade específica.

- Exemplos: Espionagem industrial, chantagem direcionada.

Oportunista: Padronizado. Busca o "alvo mais fácil" ou qualquer falha disponível.

- Exemplos: Phishing, extorsão, criação de bots/zumbis (utilizados em ataques DDoS, spam, mineração de Bitcoin, etc.).

Mecanismos de Segurança

Monitoramento e Auditoria (Monitoring and auditing): Rastrear continuamente o acesso aos recursos do sistema.

- Verificar quais ações foram executadas por quais usuários.
- Analisar comportamento geral do sistema.

Criptografia (Encryption): Transformar dados em um formato que um atacante não consiga entender, ou que permita verificar se houve modificações.

- Técnica fundamental à segurança computacional.
- Provê os mecanismos necessários à implementação da confidencialidade
- Também utilizado para verificar a integridade dos dados (hash).

Mecanismos de Segurança (cont.)

Autenticação (Authentication): Verificar a *identidade* alegada de uma entidade.

- **Métodos**: Conhecimento (senha), Posse (token), Biometria (impressão digital).

Autorização (Authorization): Checar se uma entidade autenticada possui os *direitos adequados* para acessar determinados recursos.

- *Princípios:* **Menor Privilégio** (dar apenas o necessário) e **Separação de Privilégio**.
- **Controle de Acesso:** Modelos como RBAC (Baseado em Papéis) ou ABAC (Baseado em Atributos).

Prof. Dr. Denis M. L. Martins | martins.denis@usp.br

Confidencialidade vs. Privacidade (GDPR)

Confidencialidade é uma propriedade técnica, enquanto **Privacidade é um direito legal e ético**.

- **Confidencialidade:** Protege contra vazamento de dados.
  - Garante o **sigilo**.
  - Exemplo: Se um banco de dados armazena seu número de CPF, a confidencialidade é mantida se esse dado estiver **criptografado** e só puder ser descriptografado por funcionários autorizados do sistema.
- **Privacidade:** É o *direito* sobre o fluxo e uso dos seus próprios dados pessoais.
  - Garante o **controle**
  - Exemplo: O banco de dados ainda está criptografado (confidencialidade mantida). No entanto, o banco começa a vender seu histórico de transações para empresas de marketing sem o seu consentimento. **Neste caso, sua privacidade foi violada**, mesmo que os dados continuem tecnicamente confidenciais.

Alguns Ataques e Modelos de Confiança

- Ataque Sybil: Um atacante cria múltiplas identidades falsas para dominar a rede.
  - *Mitigação:* Exige um custo (computacional ou financeiro) para criar cada identidade (Ex: Proof-of-Work/Stake em Blockchains).
- Ataque Eclipse: Isolar um nó da rede, fazendo-o pensar que está isolado.
  - *Mitigação:* Usar fontes de informação diversas e centralizadas (Autoridade Certificadora).
- **Modelo de Responsabilidade Compartilhada:** A segurança é dividida entre o Provedor (Infraestrutura) e o Cliente (Dados/Aplicação).
  - *Atenção:* O cliente deve proteger os dados, mesmo que a infraestrutura seja do provedor.
- **Melhores Práticas:** Criptografia ponta a ponta (dados em trânsito E repouso) e Monitoramento Contínuo.

Criptografia

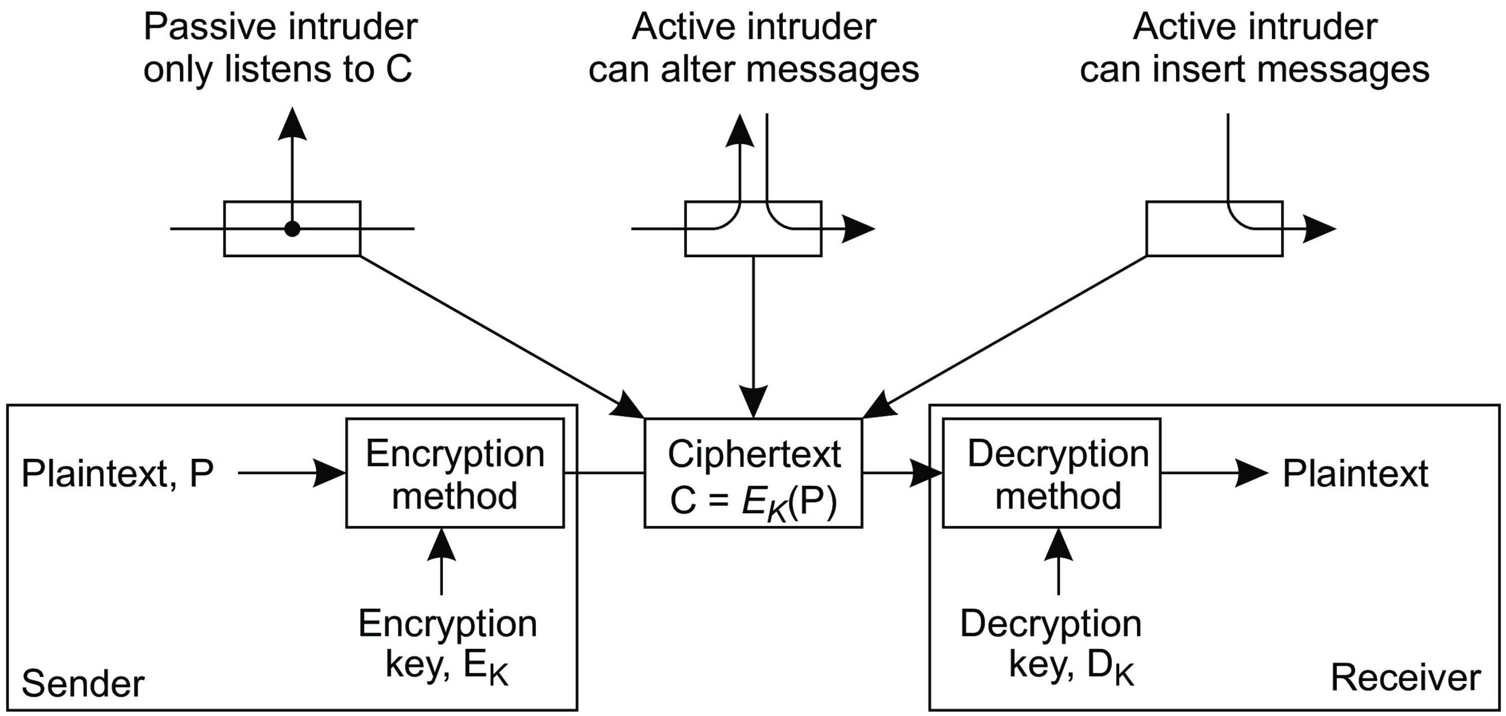
Prática e o estudo de métodos que promovem a comunicação segura entre entidades autorizadas na presença de comportamento adversário.

Criptografia

- **Texto aberto ou Plaintext (P):** Mensagem original.
- **Texto cifrado ou Ciphertext (C):** Versão criptografada, ilegível sem a chave.
- **Chave**: valor, geralmente secreto, parâmetro de algoritmo criptográfico.
- **Remetente e destinatário**: entidades autorizadas.
- **Adversário ou invasor**: entidade não autorizada.

> Na imagem: Máquina Enigma utilizada tanto para criptografar como para descriptografar códigos de guerra. Fonte da Imagem: Wikipedia.

Criptografia: Modelo Estruturante



- **Chave de criptografia (Encryption key):** Um *input* E_K para uma função de criptografia, resultando em: $C = E_K(P)$.
- **Chave de descriptografia (Decryption key):** Um *input* D_K para uma função de descriptografia, resultando em: $P = D_K(C)$.

Fonte da Imagem: Van Steen e Tanenbaum. Distributed Systems (livro), 4a edição.

Classes de Criptografia

- **Simétrica:** Usa uma única chave para criptografar e descriptografar. Rápida.
  - Se $P = D_K(E_K(P))$ então $D_K = E_K$.
  - Exemplo: Criptografia AES em um chat.
- **Assimétrica (Chave Pública/Privada):** Usa um par de chaves PK (chave pública) e SK (chave privada/secreta). Mais lenta, mas ideal para troca segura de chaves.
  - Se $P = D_K(E_K(P))$ então $D_K \neq E_K$
- **Híbrido:** Utiliza criptografia assimétrica para inicializar o estabelecimento de chaves simétricas para a criptografia dos dados.
- **Funções criptográficas hash**: Recebe dados longos e gera um *digest* (resumo) fixo.

> **Performance (Desempenho):** Hashing > Criptografia Simétrica > Criptografia Assimétrica. É possível calcular hashes em mais dados por segundo do que é possível criptografar; e é possível criptografar mais dados por segundo simetricamente do que assimetricamente.

Aspectos de Criptografia

Princípio Fundamental da Criptografia: "Se muitas pessoas inteligentes falharam em resolver um problema, é provável que ele não seja resolvido (em breve)." - *Kaufman, Perlman & Speciner, Network Security: Private Communication in a Public World, Second Edition, Prentice Hall Press, 2002.*

- Problema: Quebrar um algoritmo criptográfico específico.
- Se isso não aconteceu por muito tempo, provavelmente o algoritmo é forte.
- Muitos algoritmos criptográficos são criados sem uma prova de segurança matemática formal.

Exemplo Histórico de Criptografia: Cifra de César

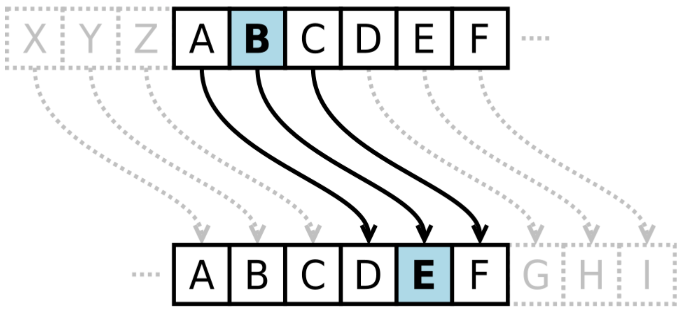
Princípio: Substituição por um deslocamento fixo no alfabeto.

- Origem: nomeada em homenagem a Júlio César de Roma (cerca de 100-44 AEC).
- Cada letra é movida um número constante N de posições.
- **Como Funciona:** Se o deslocamento for $N = 3$, 'A' vira 'D', 'B' vira 'E'.
- **Exemplo Prático:** Mensagem "OLA" com deslocamento 3 $\rightarrow$ "ROD".
- **Vulnerabilidade:** É trivial de quebrar por **força bruta** (testar todos os possíveis deslocamentos).

> As cifras antigas são ótimos exemplos didáticos, mas **extremamente fracas**. A fraqueza reside na previsibilidade e na falta de complexidade matemática. O ataque mais comum é a **análise de frequência** (contar quantas vezes cada letra aparece, por exemplo, 'E' em português).

> Para esta e outras cifras, veja também a demonstração interativa.

> Fonte da Imagem: Wikipedia.

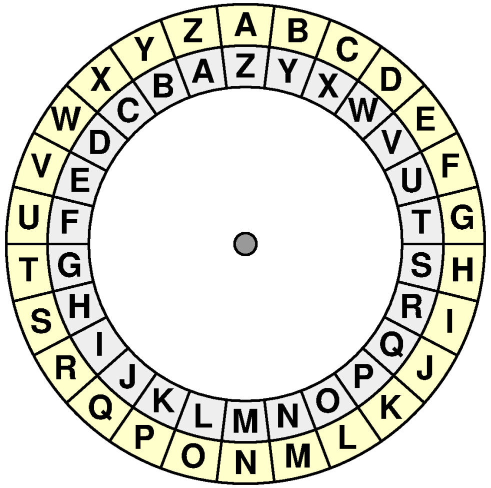


Exemplo Histórico de Criptografia: Cifra ATBASH

Princípio: Substituição reversa do alfabeto hebraico (origem: cerca de 600-500 AEC).

- Cada letra é mapeada para sua letra oposta no alfabeto.
- No alfabeto hebraico: Substituição do *aleph* (a primeira letra) pela *tav* (a última), *beth* (a segunda) pela *shin* (a penúltima),

```
Normal:  a b c d e f g h i j k l m n o p q r s t u v w x y z
Código:  Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
```



- **Exemplo Prático:** Mensagem "OLA" $\longrightarrow$ "L O Z".
- **Vulnerabilidade:** É um padrão fixo e previsível; a chave é sempre o próprio alfabeto.

> Fonte da Imagem: Aman Gondaliya @Medium.

Exemplo Histórico de Criptografia: Quadrado de Polybius

Princípio: Substituir letras por coordenadas numéricas em uma grade (5×5).

- Origem: Historiador grego Polybius (cerca de 204 – 122 AEC).
- Cada letra recebe um par de números (Linha, Coluna).
- **Como Funciona:** O alfabeto é preenchido em um quadrado. 'A' pode ser $(1, 1)$, 'B' pode ser $(1, 2)$, etc.
- **Exemplo Prático:** A letra 'S' pode ser representada por "43" (Linha 4, Coluna 3).
- **Vulnerabilidade:** É um sistema de mapeamento fixo e limitado; a chave é o próprio quadrado.

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

> Fonte da Imagem: Pixels.com.

Conclusão

- As cifras históricas são importantes por ilustrarem os princípios de substituição e transposição. No entanto, elas falham devido à sua **baixa entropia** (poucas possibilidades) e à **análise de frequência**.
- A criptografia moderna usa:
  - Matemática complexa (Teoria dos Números).
  - Funções unidirecionais (Hash Functions).
  - Chaves longas e aleatórias (AES, RSA).
- **Material Adicional**:
  - Aula sobre Segurança por Jens Lechtenborger.
  - Saltzer & Schroeder, The protection of information in computer systems, Proceedings of the IEEE 63(9), 1278-1308 (1975).

Dúvidas e Discussão

Questões

1. Qual o princípio fundamental por trás da Cifra de César?
2. Se você usa a cifra Atbash para codificar a palavra "SOL", qual será o resultado?
3. Em termos criptográficos, o que significa dizer que uma cifra é vulnerável à "Análise de Frequência"?
4. Qual a principal diferença entre um ataque de força bruta e uma análise de frequência?
5. Codifique a palavra "USP" usando um deslocamento de $N = 5$. Depois, decodifique o texto cifrado "YXU".