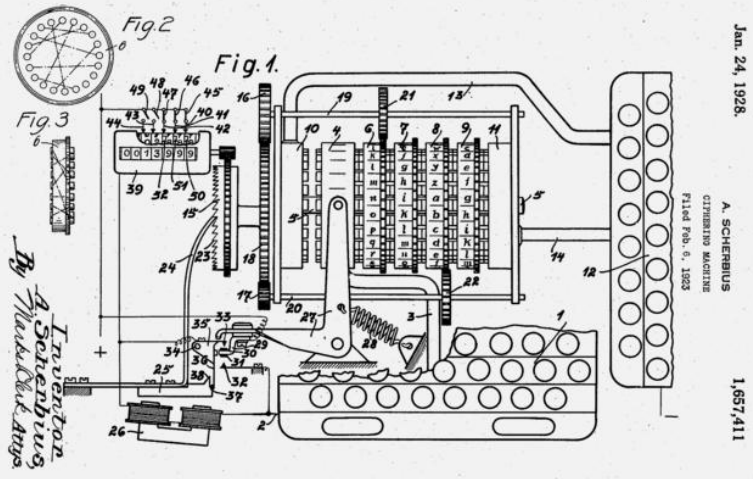
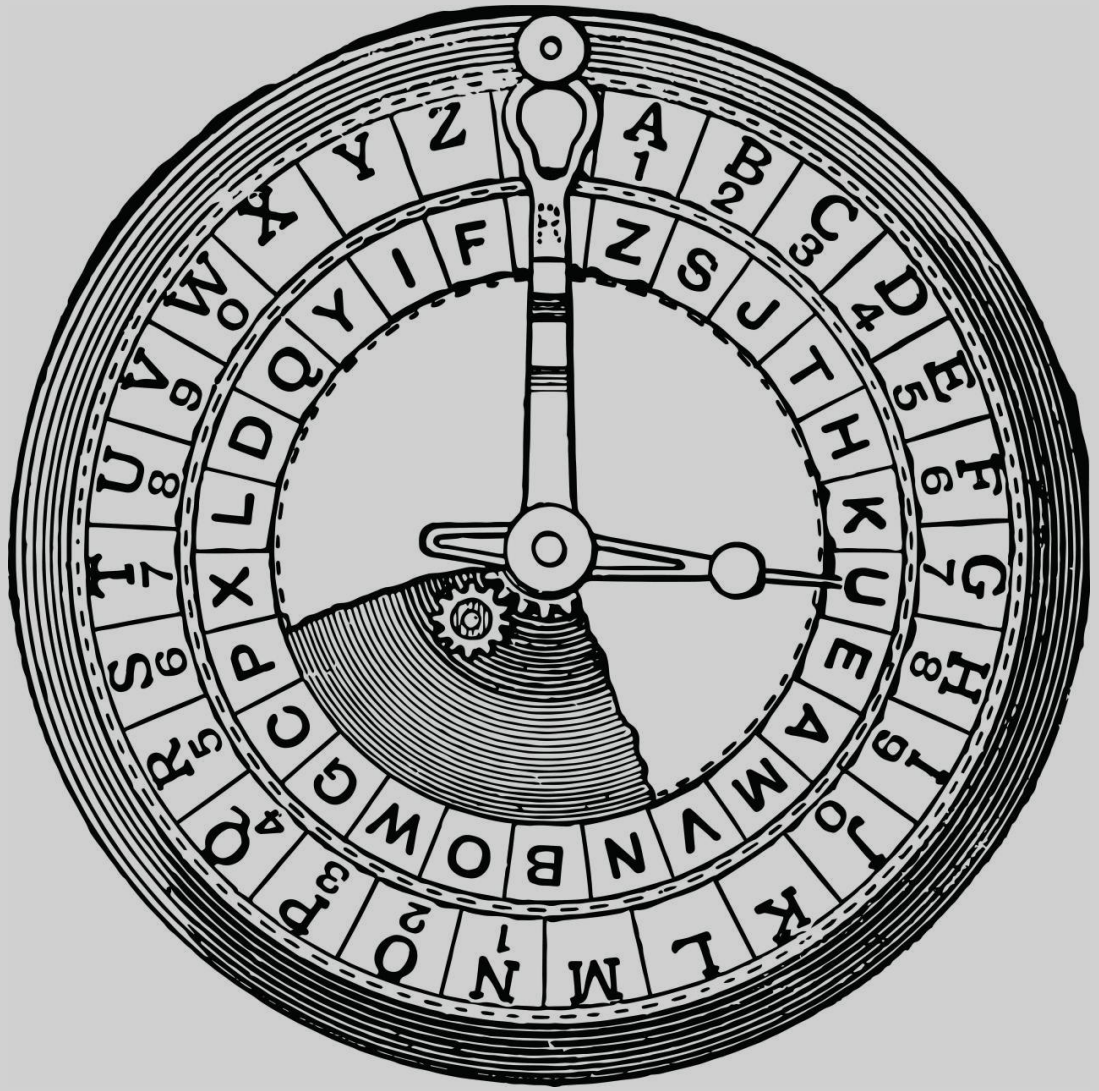
# Modos de Operação

Fundamentos da Criptografia

Nicollas Mocelin Sdroievski



# Nas últimas aulas



# Definição

- **Definição 3.25.** Seja $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ uma função chaveada, eficiente, e que preserva o tamanho da entrada. $F$ é uma **função pseudoaleatória** se para todo algoritmo probabilístico polinomial $D$ existe uma função negligenciável $\mathrm{negl}$ tal que

$$\left|\Pr\left[D^{F_k(\cdot)}(1^n) = 1\right] - \Pr\left[D^{f(\cdot)}(1^n) = 1\right]\right| \leq \mathrm{negl}(n)$$

Onde a primeira probabilidade é sobre a escolha uniforme de $k \in \{0,1\}^n$ e a aleatoriedade de $D$, e a segunda é sobre a escolha uniforme de $f \in \mathrm{Func}_n$ e a aleatoriedade de $D$

# A cifra da última aula

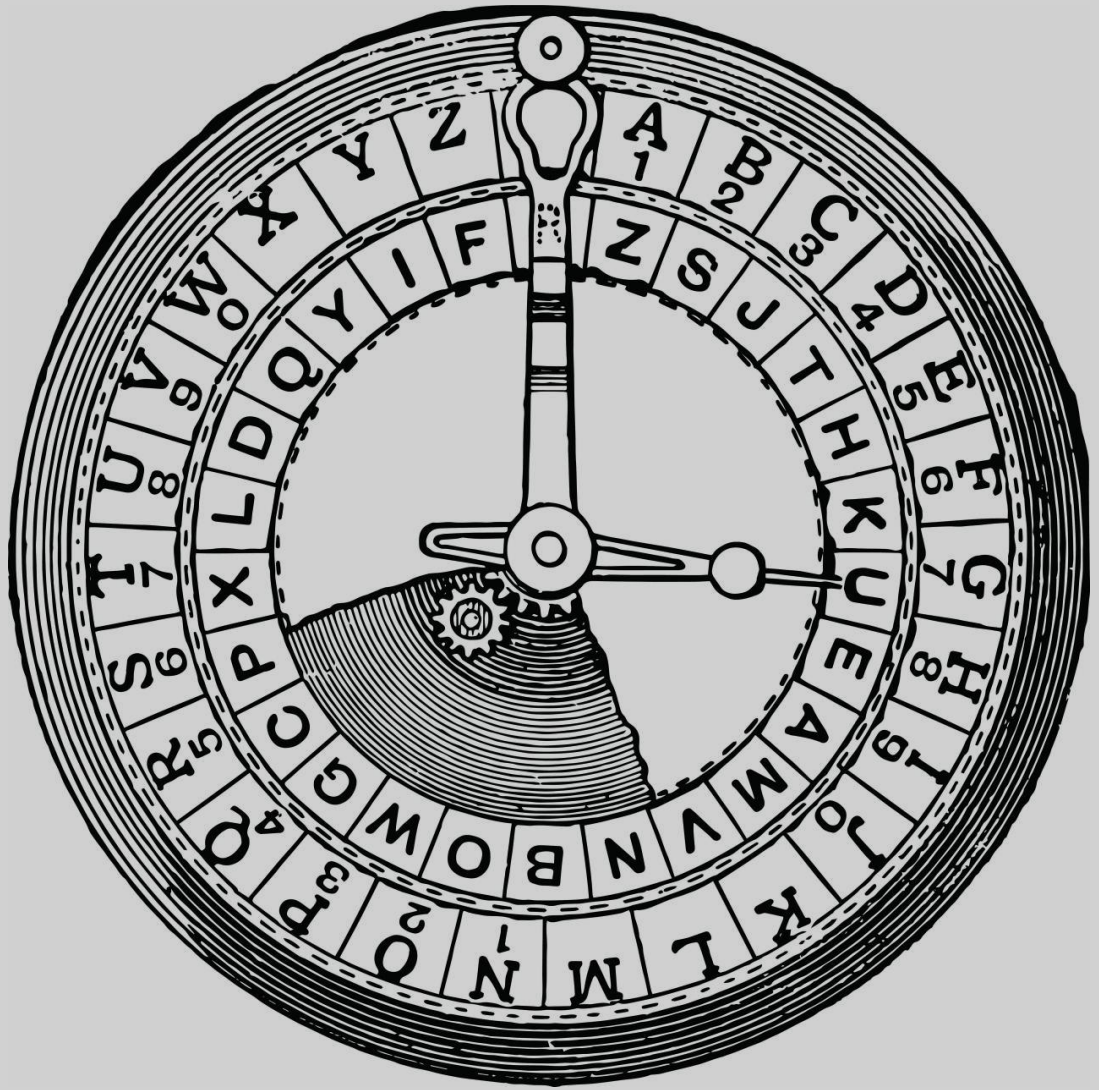
- **Construção 3.30.** Seja $F$ uma função pseudoaleatória. A cifra simétrica para mensagens de tamanho $n$ é definida a seguir
  - $\mathrm{Gen}$: na entrada $1^n$, escolhe $k \leftarrow \{0,1\}^n$ de maneira uniforme e retorna $k$
  - $\mathrm{Enc}$: na entrada $k, m \in \{0,1\}^n$, escolhe $r \leftarrow \{0,1\}^n$ de maneira uniforme e tem como saída

$$c := \langle r, F_k(r) \oplus m \rangle$$

  - $\mathrm{Dec}$: na entrada $k \in \{0,1\}^n$ e $c = \langle r, s \rangle$, tem como saída

$$m := F_k(r) \oplus s$$

# Permutações pseudoaleatórias ou cifras de bloco



# Ideia

- Seja $\mathrm{Perm}_n$ o conjunto de todas as permutações de $\{0,1\}^n$
- Imaginando $f \in \mathrm{Perm}_n$ como uma tabela de busca, cada par de linhas deve ser diferente
- Quantas permutações possíveis?
  - $2^n \cdot (2^n - 1) \cdot \ldots \cdot 1 = (2^n)!$
- Uma permutação (chaveada) pseudoaleatória deve ser indistinguível de uma permutação aleatória

# Permutações chaveadas

- Uma função chaveada $F$ é uma **permutação chaveada** se $\ell_{in} = \ell_{out}$ e, para todo $k \in \ell_{key}$, a função $F_k: \{0,1\}^{\ell in} \to \{0,1\}^{\ell in}$ é um-para-um (uma permutação)
- $\ell_{in}$ é chamada de **tamanho do bloco** de $F$
- A menos que indicado ao contrário, $\ell_{key} = \ell_{in} = n$
- $F$ é **eficiente** se existem algoritmos polinomiais para computar $\mathrm{F}_k(x)$ dados $k$ e $x$ **e** $F_k^{-1}(x)$, dados $k$ e $x$
  - Ou seja, $F$ deve ser eficientemente computável e eficientemente **invertível**

# Permutações pseudoaleatórias

- Definição de permutação pseudoaleatória é completamente análoga a de função pseudoaleatória
  - Porém deve ser apenas indistinguível de uma permutação aleatória
  - Escolha essencialmente “estética”
- **Proposição 3.27.** Se $F$ é uma permutação pseudoaleatória e, adicionalmente, $\ell_{in}(n) \geq n$, então $F$ também é uma função pseudoaleatória.
- **Ideia da demonstração:** só é possível diferenciar uma permutação aleatória de uma função aleatória ao encontrar uma colisão (o que ocorre com probabilidade negligenciável)

# Permutações pseudoaleatórias fortes

- Na definição de segurança de uma permutação pseudoaleatória, o distinguidor possui apenas oráculo para a função $F$
  - E não para sua inversa $F_k^{-1}$!
- Uma permutação chaveada $F_k$ pode ser indistinguível de uma permutação aleatória
  - Porém se tornar distinguível para adversários que têm acesso à $F_k^{-1}$
- Se algum protocolo necessita que o adversário tenha acesso à $F_k^{-1}$, precisamos de uma **permutação pseudoaleatória forte**

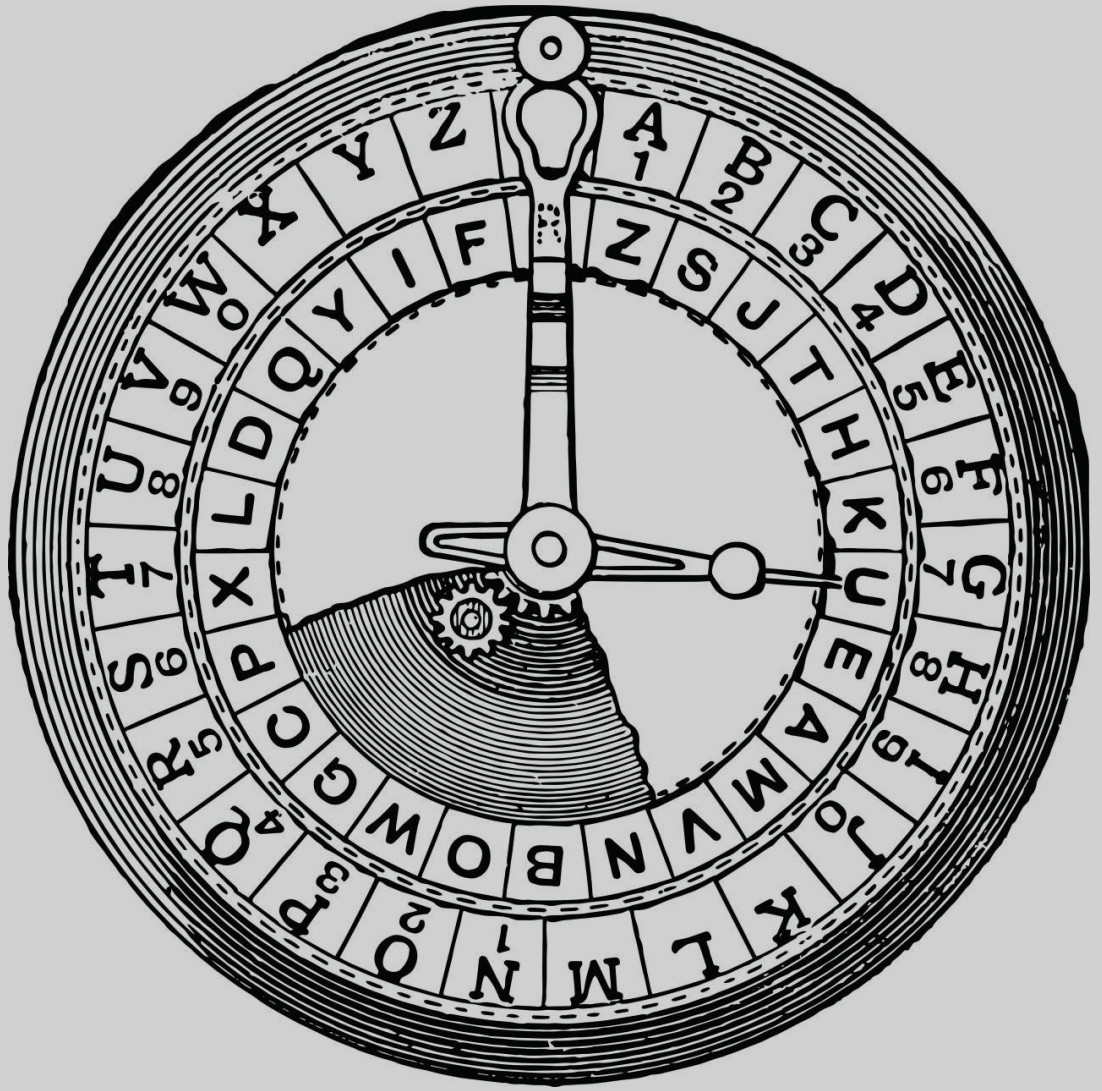
# Definição formal

- **Definição 3.28.** Seja $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ uma permutação chaveada eficiente e que preserva o tamanho da entrada. $F$ é uma *permutação pseudoaleatória forte* se para todo distinguidor probabilístico polinomial $D$, existe uma função negligenciável $\mathrm{negl}$ tal que:

$$\left|\Pr\left[D^{F_k(\cdot),\, F_k^{-1}(\cdot)}(1^n) = 1\right] - \Pr\left[D^{f(\cdot),\, f^{-1}(\cdot)}(1^n) = 1\right]\right| \leq \mathrm{negl}(n)$$

Onde a primeira probabilidade é sobre a escolha uniforme de $k \in \{0,1\}^n$ e a aleatoriedade de $D$, e a segunda é sobre a escolha uniforme de $f \in \mathrm{Perm}_n$ e a aleatoriedade de $D$.

# Modos de operação

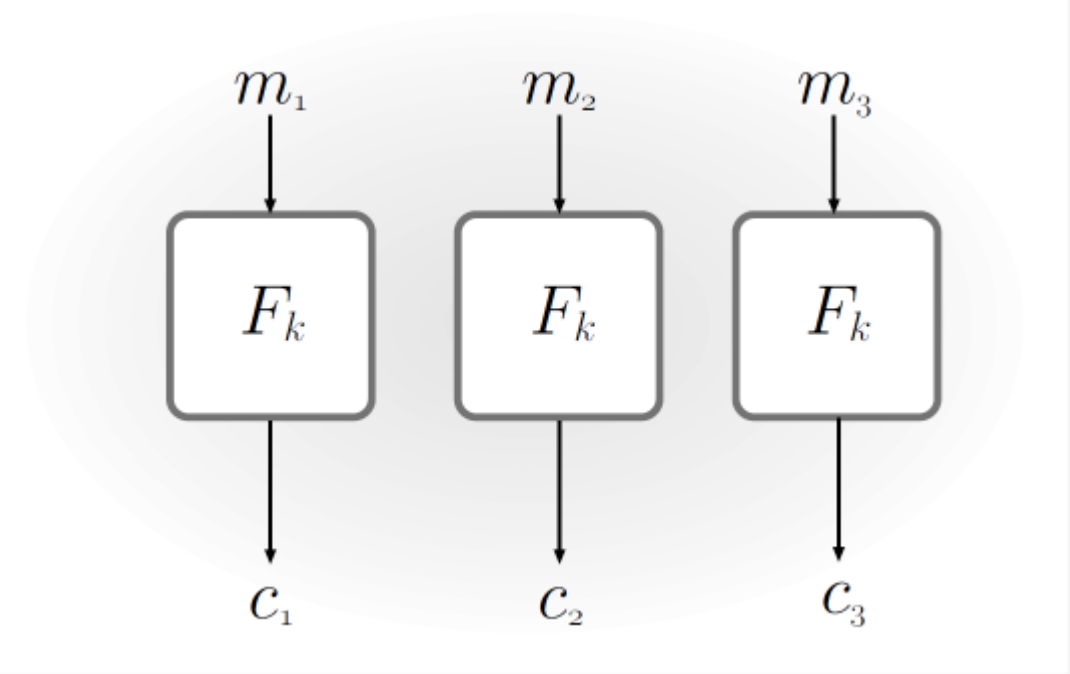


# Ideia geral

- Lembrando: uma cifra de bloco é uma permutação pseudoaleatória
- É possível construir uma cifra segura para múltiplas mensagens distintas de tamanho fixo $n$ usando uma cifra de bloco $F$
  - $\mathrm{Gen}(1^n)$ sorteia uma chave $k$ para $F$
  - $\mathrm{Enc}_k(m) = F_k(m)$
  - $\mathrm{Dec}_k(m) = F_k^{-1}(m)$
- Exercício para pós-graduação
- Modos de operação: permitem usar cifras de bloco para cifrar mensagens de tamanho arbitrário

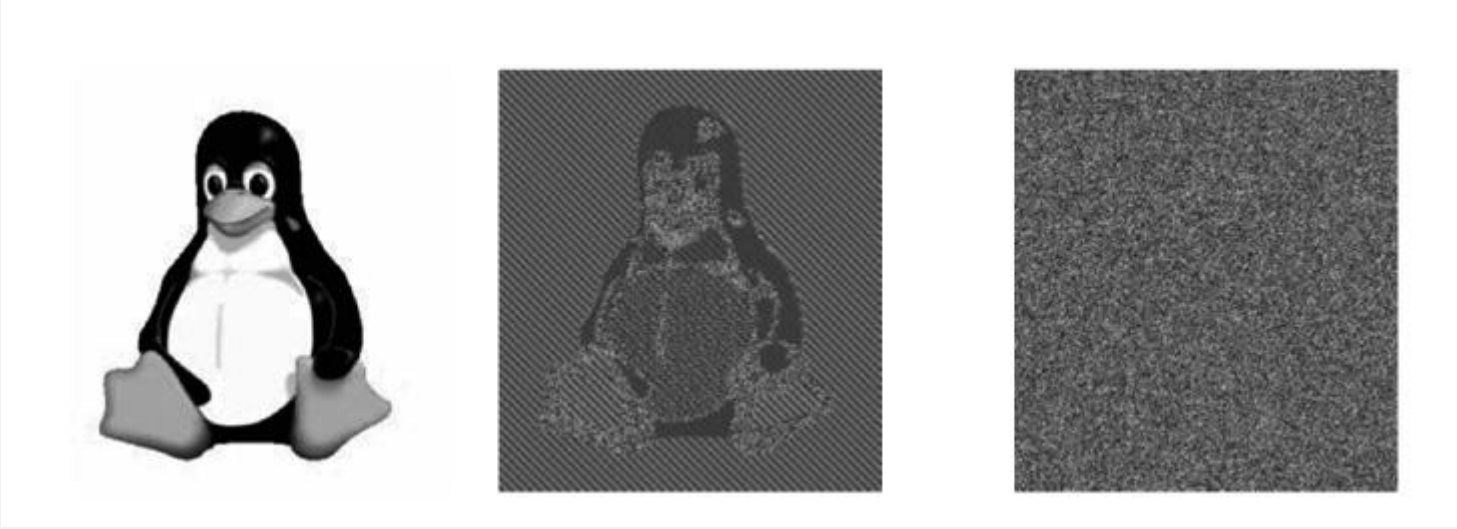
# ECB – Electronic Code Book

- Modo de operação básico e inseguro
- Aplica a cifra de bloco em cada “pedaço” da mensagem

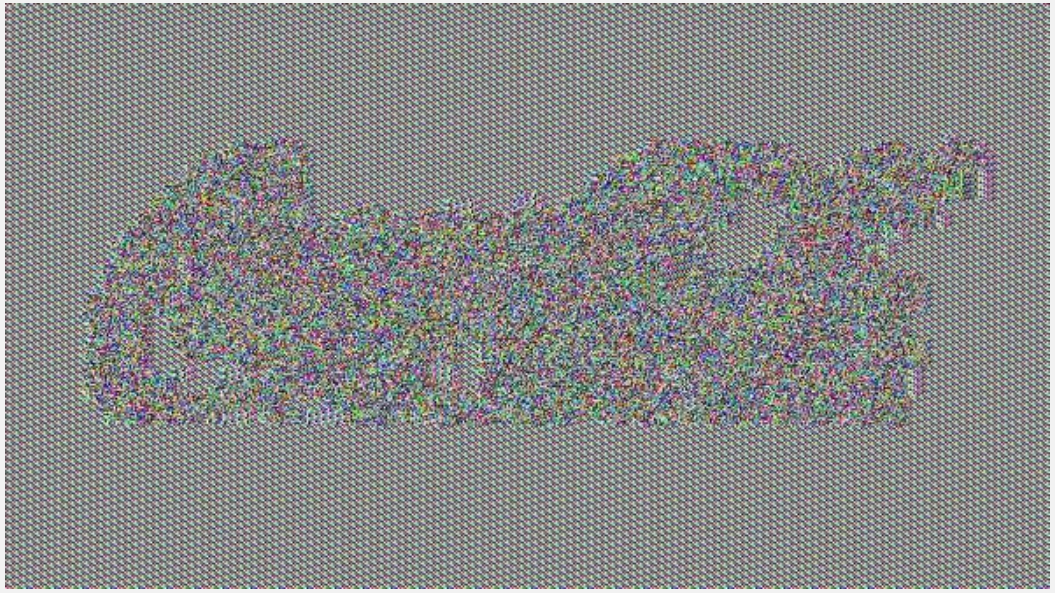


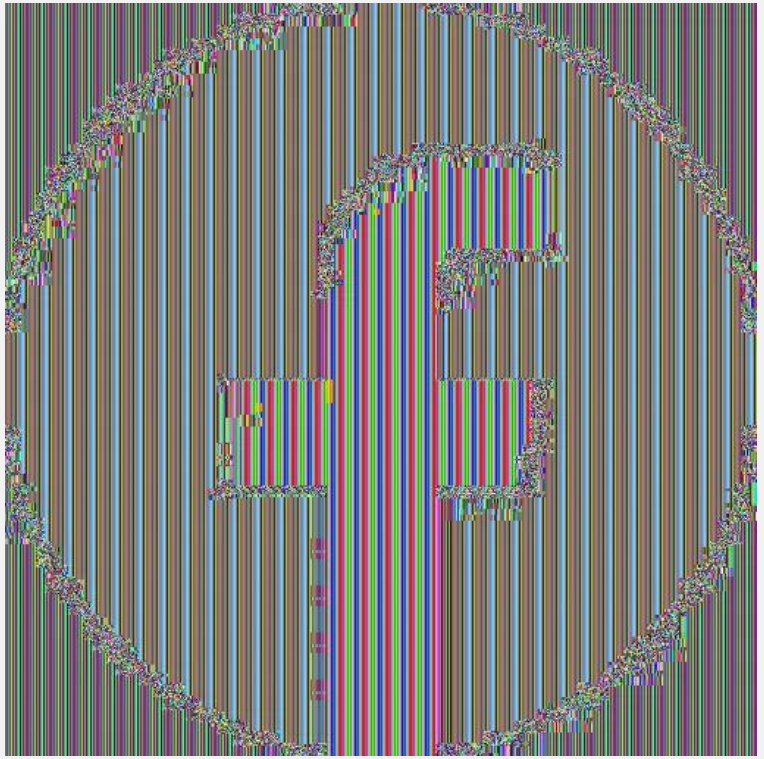
# Segurança

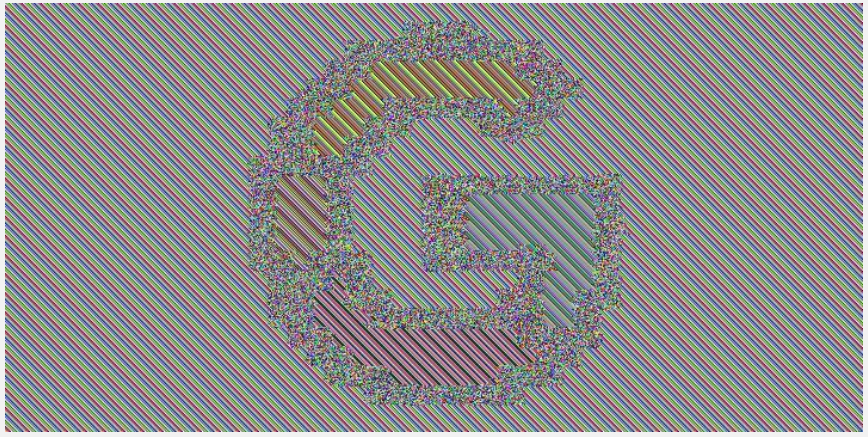
- ECB é determinístico
  - Não pode ser CPA-seguro
  - Nem mesmo EAV-seguro (exercício da lista)
  - Cifrar dois blocos iguais resulta em blocos de textos cifrados iguais

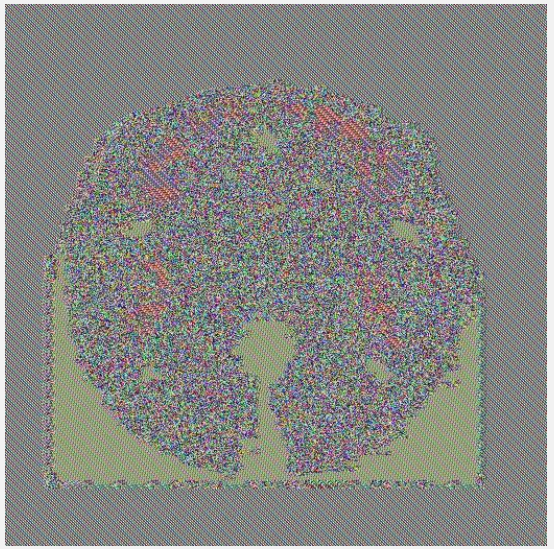


# Marcas e ECB



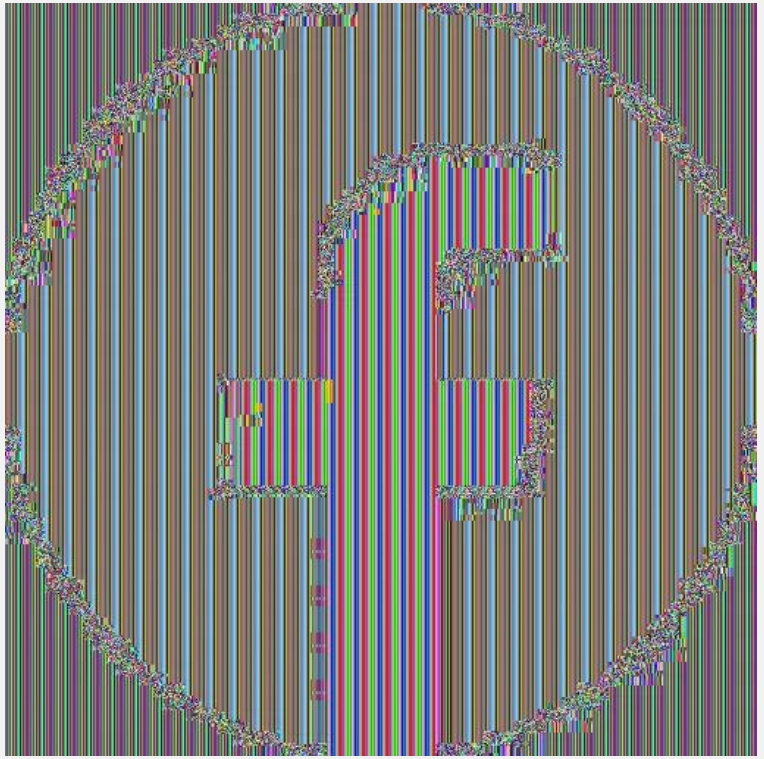






# Marcas e ECB



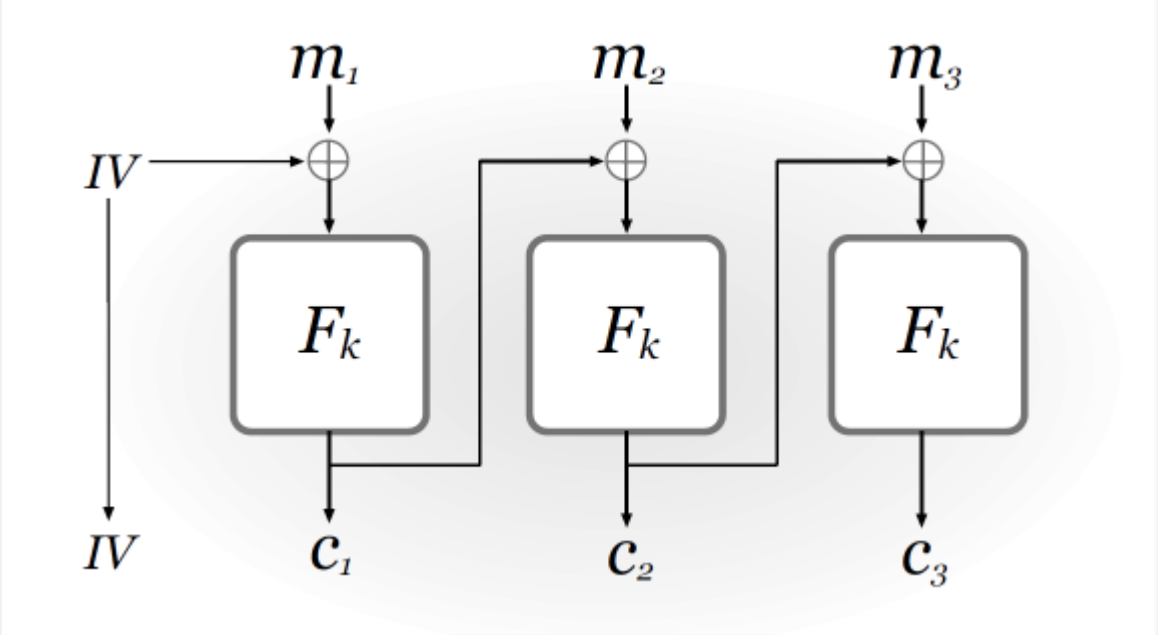







# CBC – Cipher Block Chaining

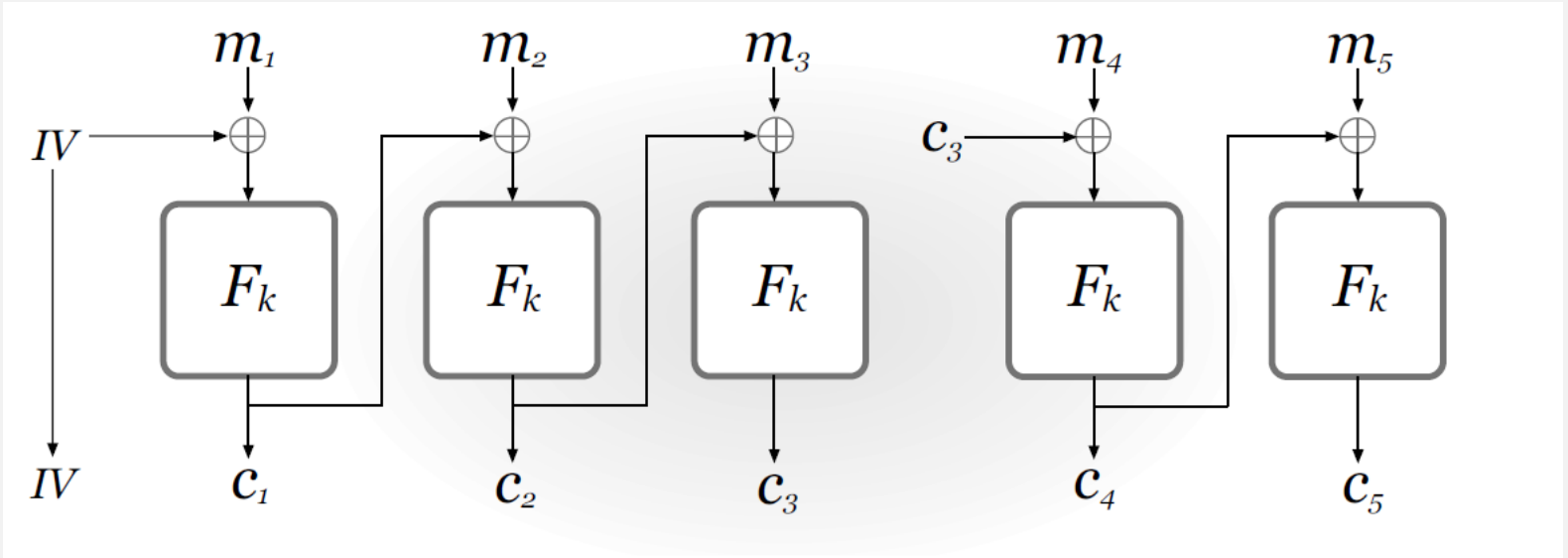
- Para cifrar a mensagem $m_i$, faz o XOR de $m_i$ com $c_{i-1}$ e então aplica a cifra de bloco
- Para a primeira mensagem, usa um vetor de inicialização aleatório $IV$



# Definição

- Para cifrar uma mensagem $(m_1, m_2, \ldots, m_\ell)$ onde cada $m_i$ tem tamanho $n$
  - $c_0 \coloneqq IV$ (uma string aleatória de tamanho $n$)
  - $c_1 \coloneqq F_k(c_0 \oplus m_1)$
  - $c_2 \coloneqq F_k(c_1 \oplus m_2)$
  - $\vdots$
  - $c_i \coloneqq F_k(c_{i-1} \oplus m_i)$
- Para decifrar $(c_0, c_1, \ldots, c_\ell)$
  - $m_i \coloneqq F_k^{-1}(c_i) \oplus c_{i-1}$
- CPA-seguro usando uma permutação pseudoaleatória

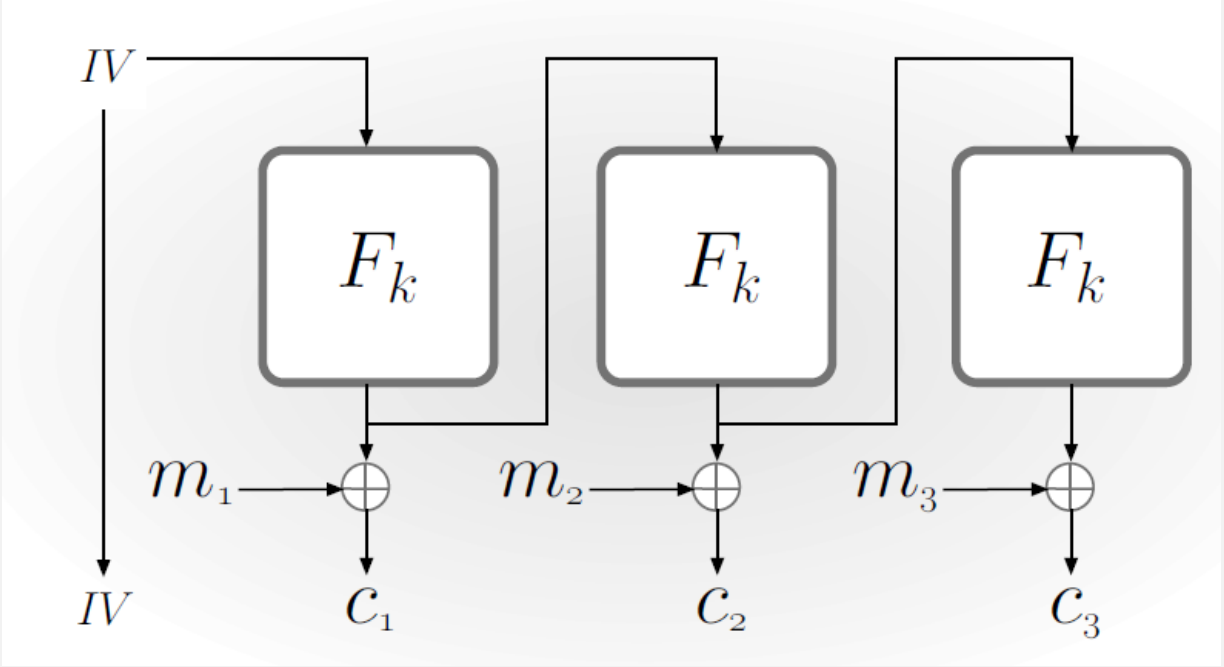
# Um problema – Chained CBC




- Vulnerável à um ataque "*chosen plaintext*"
- Assuma que o adversário sabe que $m_1 \in \{m, m'\}$ (porque ele mesmo escolheu) e observa $(IV, c_1, c_2, c_3)$
- O adversário pede que o oráculo de cifragem cifre a mensagem $(m_4, m_5)$ onde $m_4 = IV \oplus m \oplus c_3$ e recebe $(c_4, c_5)$
- É possível mostrar que $m_1 = m$ se e somente se $c_4 = c_1$

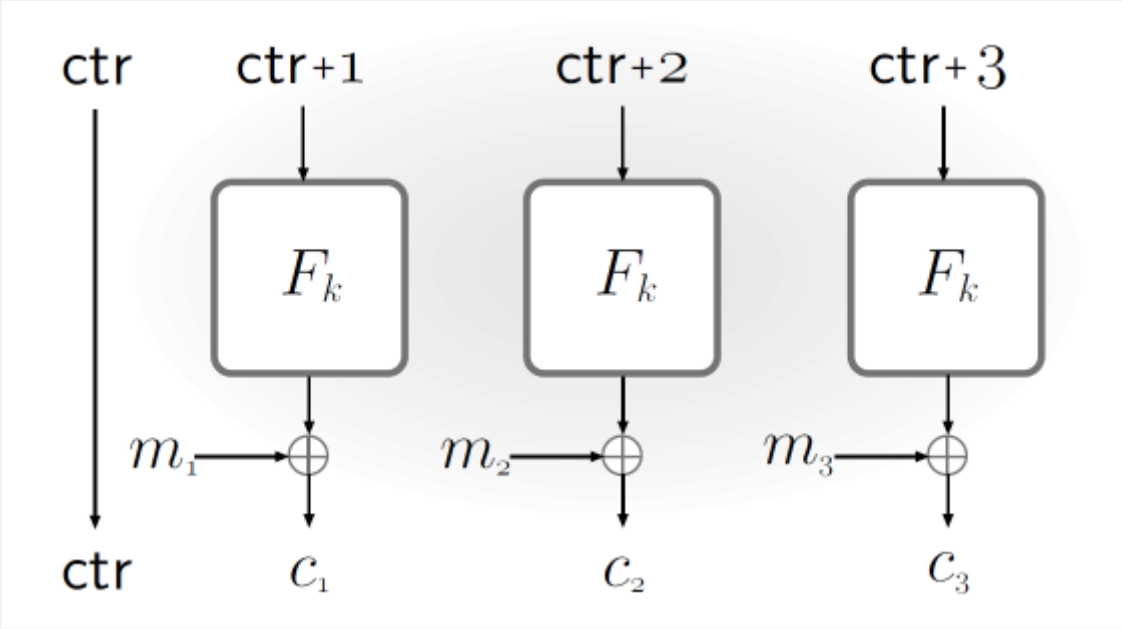
# OFB – Output Feedback

- Primeiro, é calculado um fluxo de strings pseudoaleatórias
  - $y_0 \coloneqq IV, y_i = F_k(y_{i-1})$
- Então, é feito o xor de cada string com um pedaço da mensagem
  - $c_i \coloneqq y_i \oplus m_i$
- Para decifrar:
  - $m_i \coloneqq c_i \oplus y_i$
- CPA-seguro se $F$ é uma PRP
- Pré-processável



# CTR – Counter

- Primeiro escolhe um valor aleatório ctr
- Usa $F_k(\text{ctr} + i)$ como "pad" para a mensagem $m_i$



# Algumas vantagens

- Completamente paralelizável
  - Cada bloco do fluxo pseudoaleatório pode ser comptuador de maneira independente
  - Necessita de apenas uma avaliação de $F$ para cifrar/decifrar um bloco
- Mais simples de analisar
- **Teorema 3.32.** Se $F$ é uma função pseudoaleatória, então o modo CTR é CPA-seguro.

# O experimento $\mathrm{PrivK}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n)$

Desafiante



$k \leftarrow \mathrm{Gen}(1^n)$

$m \in \mathcal{M}$

$\mathrm{Enc}_k(m)$ ⋮

$m_0, m_1 \in \mathcal{M}$ com $|m_0| = |m_1|$

$b \leftarrow \{0,1\}$

$c \leftarrow \mathrm{Enc}_k(m_b)$

$m \in \mathcal{M}$

$\mathrm{Enc}_k(m)$ ⋮

$b' =$ chute para $b$

Adversário $\mathcal{A}$
Tempo poli$(n)$



# Prova do Teorema 3.32

- Seja $\Pi$ a nossa cifra usando o modo CTR
- Como de costume, consideramos uma versão $\widetilde{\Pi}$ de $\Pi$ que troca $F$ por uma função verdadeiramente aleatória
- Seja $\mathcal{A}$ um algoritmo probabilístico polinomial qualquer
  - De maneira essencialmente idêntica à prova da aula anterior, existe uma função negligenciável $\mathrm{negl}$ tal que:

$$\left|\Pr\left[\mathrm{PrivK}^{\mathrm{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] - \Pr\left[\mathrm{PrivK}^{\mathrm{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1\right]\right| \leq \mathrm{negl}(n)$$

  - Só nos resta limitar $\Pr\left[\mathrm{PrivK}^{\mathrm{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1\right]$

# Segurança incondicional

- Vamos provar que

$$\Pr\left[\mathrm{PrivK}^{\mathrm{cpa}}_{\mathcal{A},\tilde{\Pi}}(n) = 1\right] < \frac{1}{2} + \frac{2q(n)^2}{2^n}$$

Onde $q(n)$ é o número de consultas de $\mathcal{A}$ ao oráculo de cifragem (e um limite para o número de blocos das mensagens cifradas)

- Ideia semelhante à da aula passada
  - Definimos um evento “ruim”, que permite ao adversário vencer
  - Se o evento “ruim” não acontece, a probabilidade de vitória de $\mathcal{A}$ é $1/2$
  - Mostramos que a probabilidade do evento “ruim” é baixa

# Continuação

- Seja $\ell^* \leq q(n)$ o número de blocos das mensagens $m_0, m_1$ que $\mathcal{A}$ produz para a fase de desafio
  - E seja $\mathrm{ctr}^*$ o valor inicial usado ao gerar o texto cifrado de desafio
  - Ou seja, para obter o texto cifrado de desafio, computamos $f(\mathrm{ctr}^* + 1), \ldots, f(\mathrm{ctr}^* + \ell^*)$
- Similarmente, seja $\ell_i \leq q(n)$ o número de blocos da $i$-ésima mensagem que $\mathcal{A}$ cifra com a ajuda do desafiante
  - E seja $\mathrm{ctr}_i$ o valor inicial usado ao gerar o $i$-ésimo texto cifrado
  - Ou seja, para obter o $i$-ésimo texto cifrado, computamos $f(\mathrm{ctr}_i + 1), \ldots, f(\mathrm{ctr}_i + \ell_i)$

# Dois casos

1. Não existem $i, j, j^*$ ($j \leq \ell_i$ e $j^* \leq \ell^*$) tal que $\mathrm{ctr_i} + j = \mathrm{ctr}^* + j^*$
   - Neste caso, os valores $f(\mathrm{ctr}^* + 1), \ldots, f(\mathrm{ctr}^* + \ell^*)$ são independentes do resto do experimento já que $f$ não foi aplicada a nenhum destes valores em nenhum outro momento.
   - Então, o texto cifrado foi obtido ao fazer o XOR de bits completamente uniformes com a mensagem $m_b$, igual ao one-time pad.
2. Existem $i, j, j^*$ ($j \leq \ell_i$ e $j^* \leq \ell^*$) tal que $\mathrm{ctr_i} + j = \mathrm{ctr}^* + j^*$
   - Chamamos esse evento de $\mathrm{Sobrepor}$
   - Neste caso, $\mathcal{A}$ poderia determinar qual mensagem foi cifrada, já que consegue deduzir $f(\mathrm{ctr}_i + j) = f(\mathrm{ctr}^* + j^*)$

# Probabilidade do evento Sobrepor

- Probabilidade maximizada quando $\ell^*$ e cada $\ell_i$ são os maiores possíveis
  - Assumimos que $\ell^* = \ell_i = q(n)$ para todo $i$
- Seja $\text{Sobrepor}_i$ o evento de que a $i$-ésima sequência $\text{ctr}_i + 1, \ldots, \text{ctr}_i + q(n)$ se sobrepõe à $\text{ctr}^* + 1, \ldots, \text{ctr}^* + q(n)$
- Pelo limitante da união:

$$\Pr[\text{Sobrepor}] \leq \sum_{i=1}^{q(n)} \Pr[\text{Sobrepor}_i]$$

# Probabilidade de $\mathrm{Sobrepor}_i$

- Fixando $\mathrm{ctr}^*$, o evento $\mathrm{Sobrepor}_i$ acontece exatamente quando $\mathrm{ctr}_i$ satisfaz

$$\mathrm{ctr}^* - q(n) + 1 \leq \mathrm{ctr}_i \leq \mathrm{ctr}^* + q(n) - 1$$

- Como existem $2q(n) - 1$ valores de $\mathrm{ctr}_i$ para os quais $\mathrm{Sobrepor}_i$ acontece e $\mathrm{ctr}_i$ é escolhido de maneira uniforme em $\{0,1\}^n$

$$\Pr[\mathrm{Sobrepor}_i] = \frac{2q(n) - 1}{2^n} < \frac{2q(n)}{2^n}$$

# Finalizando

- Pelo limitante da união de antes

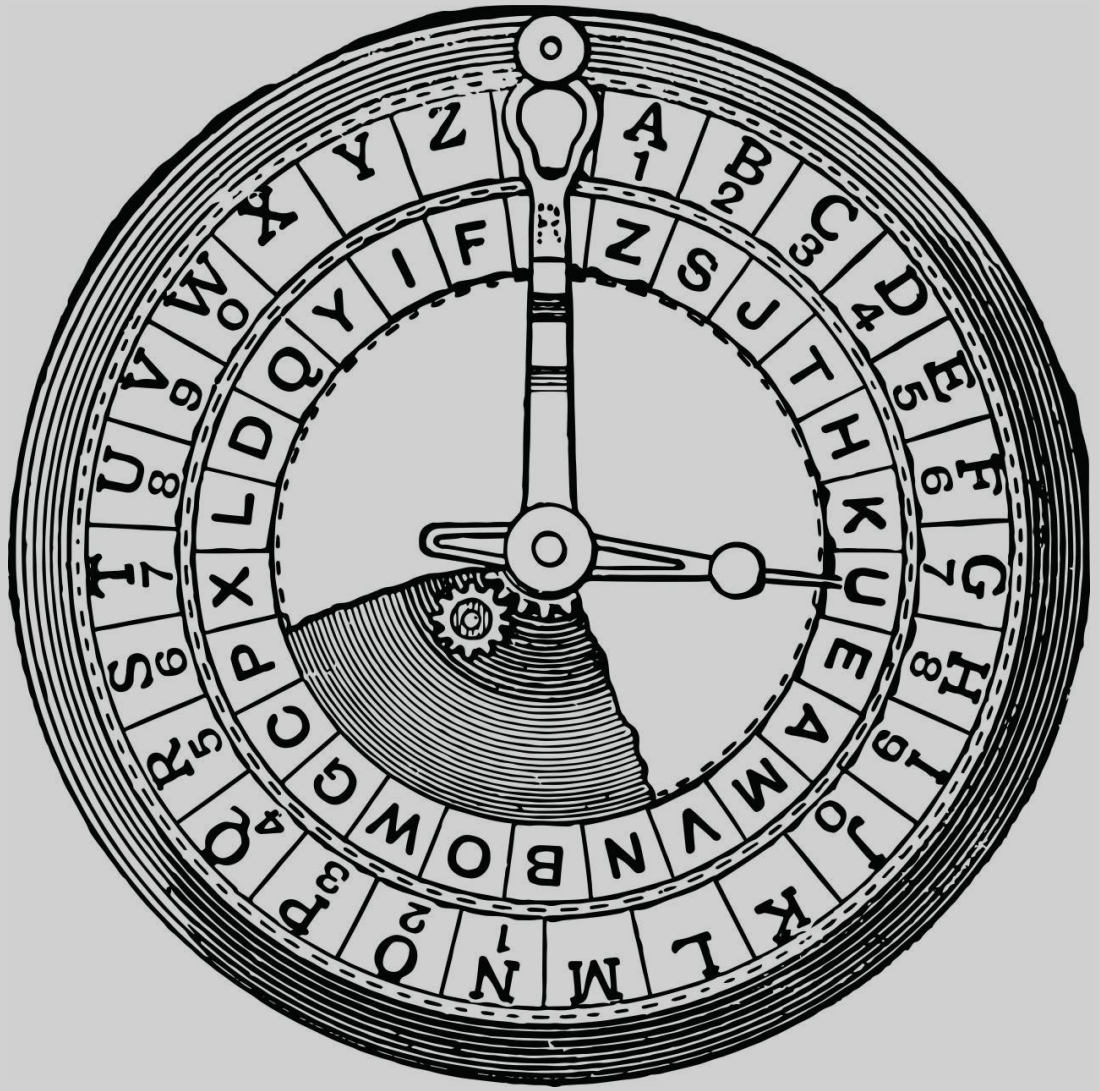
$$\Pr[\text{Sobrepor}] \leq q(n) \cdot \frac{2q(n)}{2^n} = \frac{2q(n)^2}{2^n}$$

- Pela lei da probabilidade total

$$\begin{aligned}
&\Pr\left[\text{PrivK}^{\text{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1\right] \\
&\leq \frac{2q(n)^2}{2^n} \cdot \Pr\left[\text{PrivK}^{\text{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1 \mid \text{Sobrepor}\right] + 1 \cdot \Pr\left[\text{PrivK}^{\text{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1 \mid \overline{\text{Sobrepor}}\right] \\
&\leq \frac{2q(n)^2}{2^n} + \frac{1}{2}
\end{aligned}$$

Como desejado

# Para fechar



# Para fechar

- Hoje
  - Permutações pseudoaleatórias (PRPs)
    - Primitiva muito comum na prática
  - Modos de operação
    - Permitem usar PRPs ou PRFs para cifrar mensagens de tamanho arbitrário
- Próxima aula
  - Segurança CCA
    - Tipo mais forte de segurança
    - Adversário tem acesso à um oráculo de decifragem (além de um de cifragem)
    - Ataques Padding-Oracle