Criptografia

Cifras de fluxo

Roadmap

- Cifra one-time pad
- Como é gerado o fluxo de chaves
- Geradores de números pseudoaleatório

Universidade Federal do Ceará
Campus de Crateús

Ciência da Computação
Sistemas de Informação

emerson@crateus.ufc.br

Cifra de fluxo

*Uma **cifra de fluxo** é aquela que opera sobre um fluxo de dados cifrando um **bit** por vez à medida que são transmitidos ou armazenados.*

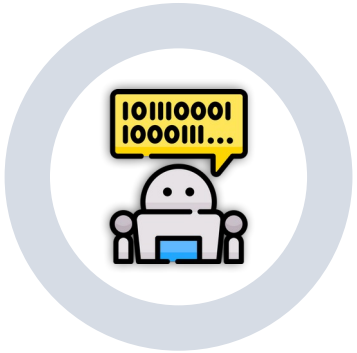
Como é possível cifrar um bit individual? O texto cifrado é obtido combinando um bit de um **fluxo de chave** a um bit de **texto simples** por meio da operação lógica XOR.

0	0	0
0	**1**	**1**
1	**0**	**1**
1	1	0

- Note que a função é perfeitamente balanceada.
- Ou seja, observando um valor de saída, há exatamente de chance para qualquer valor dos bits de entrada.

Cifra de fluxo

Cifra one-time pad



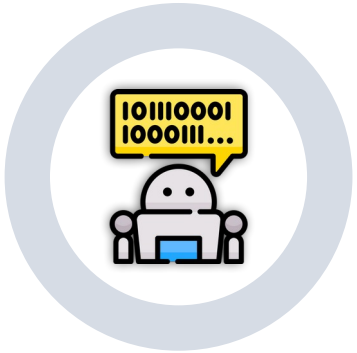
- O tipo mais simples e mais seguro de cifra de fluxo é o chamado ***one-time pad*** (também conhecido como cifra de uso único ou chave de uso único.

> *A essência da cifra **one-time pad** é a mesma de uma cifra de Vigenère, exceto que, em vez de repetir a chave, simplesmente escolhe-se uma chave tão longa quanto o texto claro.*

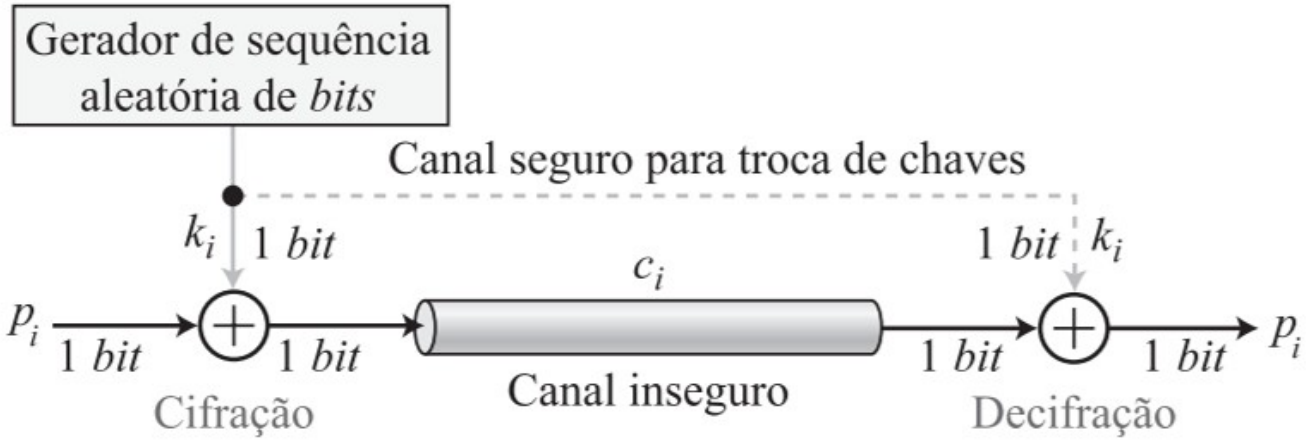
- Uma cifra ***one-time pad*** usa um fluxo de chaves escolhido aleatoriamente para **cada cifração,** e, em seguida, é descartada.
- Tanto o algoritmo de cifração como o de decifração utilizam uma única operação de **OU-exclusivo**.

Cifra de fluxo

Cifra one-time pad



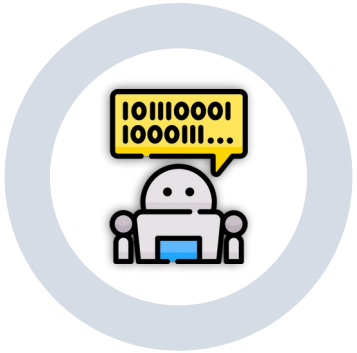
- Esse sistema foi introduzido por um engenheiro da AT&T chamado Gilbert Vernam em 1918.
- Ele construiu uma máquina eletromecânica que cifrava automaticamente a mensagem.



> *Seu sistema funciona com dados binários (bits) em vez de letras. Esta foi a primeira vez que a criptografia e a transmissão foram automatizadas.*

Cifra de fluxo

Cifra one-time pad

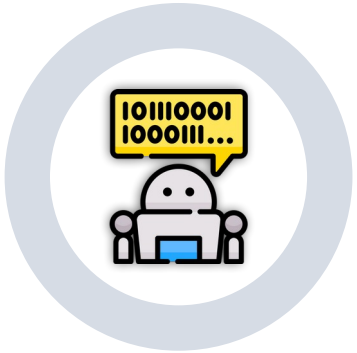


- A **one-time pad** é uma **cifra ideal**. **Ela é perfeita!**
- Não há uma forma pela qual um adversário seja capaz de adivinhar a **chave** ou **estatísticas** do texto claro ou do texto cifrado.
- Também **não existe qualquer relação** entre o texto claras e o texto cifrado.
- Em outras palavras, a mensagem cifrada é um fluxo de bits **verdadeiramente aleatórios**, mesmo se o texto às claras contiver alguns padrões.
- Um atacante não é capaz de quebrar a cifra a não ser que ela teste todas as sequências aleatórias de chave possíveis, que serão , em que o tamanho do texto claro é bits

*O **one-time pad** é o único criptossistema que é **provado** ser completamente seguro.*

Cifra de fluxo

Cifra one-time pad



- O problema das ***one-time pad*** é que ela é muito difícil de usar — é necessário uma *one-time pad* diferente para cada mensagem que enviar.
- Assim, então antes de enviar qualquer mensagem, ambos os lados têm que garantir que tenham todas as ***one-time pad*** e que elas estejam em ordem.
- Se eles planejam enviar mensagens repetidamente, isso significa que terão que ter muitas ***one-time pad***!

> *Em resumo, as **one-time pad** são boas na **teoria**, mas são incrivelmente difíceis de usar na prática.*

Cifra de fluxo

- As cifras de fluxo produzem um fluxo pseudoaleatório de bits **chamado fluxo de chaves**.
- O **texto simples**, o **texto cifrado** e o **fluxo de chaves** consistem em bits individuais, como no exemplo abaixo.

Cifração

`11001100` texto simples

`01101100` fluxo de chaves

`10100000` texto cifrado

Decifração

`10100000` texto cifrado

`01101100` fluxo de chaves

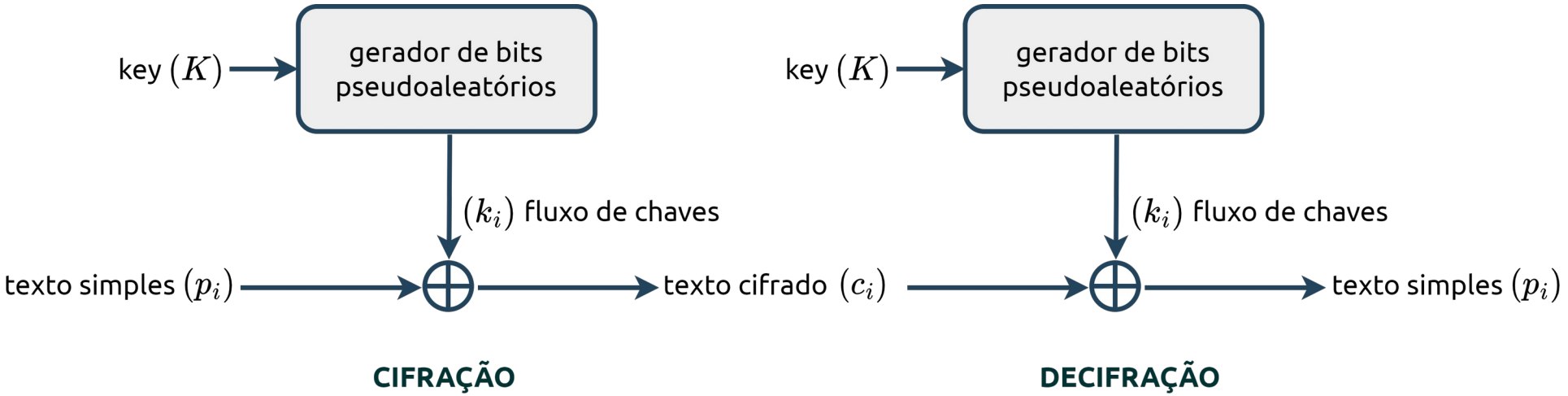
`11001100` texto simples

*Note que as funções de **cifração** e **decifração** são as mesmas, porque ambas fazem a mesma coisa, ou seja, bits XOR com o fluxo de chaves.*

Cifra de fluxo

Qual é exatamente a natureza do fluxo de chave?

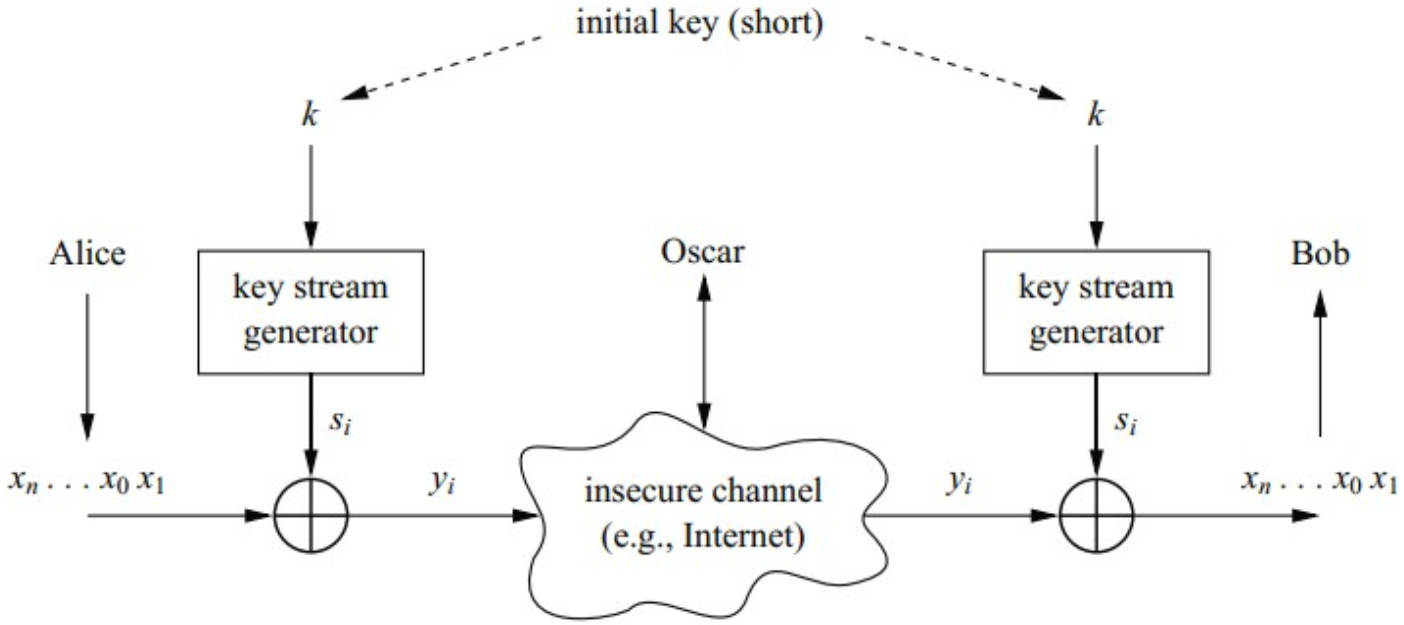
- O fluxo de chave deve ser fornecido a ambos os usuários **antecipadamente** por meio de algum canal **independente** e **seguro**. Isso introduz problemas logísticos.
- Portanto, o gerador de fluxo de bits deve ser implementado como um **procedimento algorítmico**, para que o fluxo de bits criptográfico possa ser produzido por ambos os usuários.



Cifra de fluxo

Cifras de fluxo prática

- Nessa abordagem, o **gerador de fluxo de bits** é um algoritmo controlado por **chave**.
- Os dois usuários precisam apenas compartilhar a **chave geradora**, e cada um pode produzir o **fluxo de chave**.
- Para tornar a **cifra prática** é preciso substituir o fluxo de chave **verdadeiramente** aleatório por um gerador de números pseudoaleatórios, onde a chave serve como uma semente



Cifra de fluxo

Como é gerado o fluxo de chaves?

- A geração **fluxo de chaves** — ou seja, os valores , é a questão central para a **segurança** das cifras de fluxo.
- Os bits de fluxo de chave, **,** não são os próprios bits de chave.

*Gerar o fluxo de chave é basicamente o **objetivo** das cifras de fluxo.*

*Um requisito central para os bits do fluxo de chave deve ser que eles apareçam como uma **sequência aleatória**.*

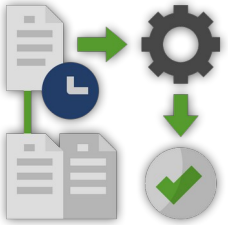
Caso contrário, um invasor poderia adivinhar os bits e fazer a decifração sozinho..

*Para gerar esse fluxo de bits, precisamos de um **Geradores de Números Pseudoaleatório***

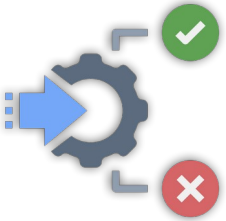
Cifra de fluxo

Geradores de números pseudoaleatório (*Pseudorandom Number Generators* - PRNG)

- Em essência, existem duas estratégias fundamentais para gerar bits aleatórios ou números aleatórios.



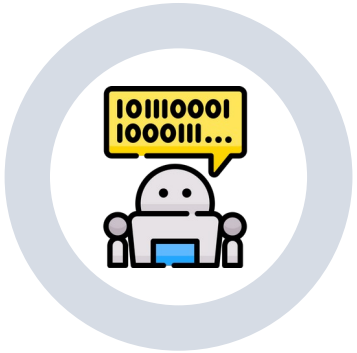
- A primeira calcula bits de forma **determinística** usando um algoritmo.
- Essa técnica é conhecida como geradores de números pseudoaleatórios (PRNG) ou geradores de bits aleatórios determinísticos (DRBG).



- A segunda é produzir bits de forma **não determinística** usando alguma fonte física que produz algum tipo de saída aleatória.
- Essa técnica é conhecida como geradores de números aleatórios verdadeiros (TRNG).

Princípios de geração de números pseudoaleatório

Requisitos do gerador



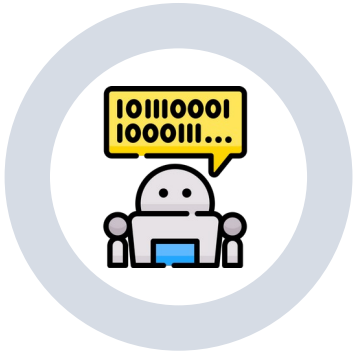
- Há dois requisitos distintos e não necessariamente compatíveis para uma sequência de números aleatórios: **aleatoriedade** e **imprevisibilidade**.

Aleatoriedade

- A preocupação na geração de uma sequência de números **supostamente aleatórios** tem sido que a sequência de números seja **aleatória** em algum sentido estatístico bem definido.
  - **Distribuição uniforme**. A distribuição dos bits na sequência deve ser uniforme; ou seja, a frequência de ocorrência de **uns** e **zeros** deve ser aproximadamente igual.
  - **Independência.** Nenhuma subsequência na sequência pode ser inferida a partir das outras.

Princípios de geração de números pseudoaleatório

Requisitos do gerador



- Há dois requisitos distintos e não necessariamente compatíveis para uma sequência de números aleatórios: **aleatoriedade** e **imprevisibilidade**.

Imprevisibilidade

- Em algumas aplicações criptográficas, o requisito não é apenas que a sequência de números sejam **estatisticamente aleatórios**, mas que os membros sucessivos da sequência sejam **imprevisíveis**.
- A ideia aqui é a mesmo, deve-se ter cuidado para que um oponente não seja capaz de **prever** futuras elementos da sequência com base nos elementos anteriores.

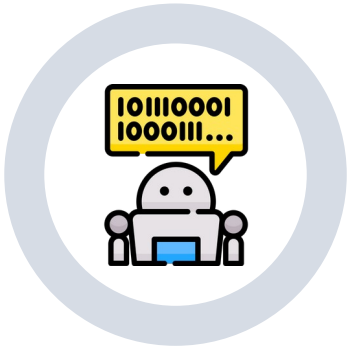
Princípios de geração de números pseudoaleatório

Requisitos do gerador

Sementes

- A semente que serve como entrada para o **PRNG** deve ser segura.
- Como o PRNG é um **algoritmo determinístico**, se o adversário puder deduzir a semente, então a saída também pode ser determinada.
- Portanto, a semente deve ser imprevisível (aleatória).

Projetos de algoritmos de PRNG



- Há duas categorias principais de PRNG.

- **Algoritmos desenvolvidos para o propósito específico**
  - São usados em várias aplicações de PRNG.
  - Ou são projetados especificamente para uso em cifras de fluxo.

- **Algoritmos baseados em algoritmos criptográficos existentes**
  - Cifras de bloco simétricas
  - Cifras assimétricas
  - Funções hash e códigos de autenticação de mensagem

Fim!

[Aula 06] Cifras de fluxo