**Universidade de Brasília**
Ciência da computação
Segurança Computacional

# Cifra de Bloco e modo operação CTR

Talles Marcelo de Mesquita Caldeira Mat:20/0060295

**13 de novembro de 2023**

# 1 Pré-requisitos

1. Versão do Python 3.x.x >=
2. Acesso ao prompt de comando(cmd)

# 2 Introdução

A criptografia, uma das pedras angulares da segurança digital, tem evoluído significativamente ao longo dos anos, adaptando-se às crescentes demandas de um mundo cada vez mais conectado. No cerne dessa evolução está a criptografia assimétrica, um método que revolucionou a forma como as informações são protegidas no ciberespaço.

Conceitos Básicos de Criptografia A criptografia, em sua essência, é a arte de codificar mensagens para garantir sua confidencialidade e integridade. Desde tempos antigos, variadas técnicas foram desenvolvidas para proteger informações importantes. No entanto, foi com o advento dos computadores e da internet que a criptografia ganhou uma nova dimensão, evoluindo para métodos mais complexos e seguros.

Criptografia Assimétrica Entre esses métodos, a criptografia assimétrica se destaca. Diferente da criptografia simétrica, que usa a mesma chave para cifrar e decifrar uma mensagem, a criptografia assimétrica utiliza um par de chaves - uma pública e uma privada. Essa abordagem não apenas reforça a segurança, mas também facilita a troca segura de chaves e informações em um ambiente aberto como a internet. A chave pública pode ser compartilhada livremente para cifrar mensagens, enquanto a chave privada, mantida em segredo, é usada para decifrá-las.

## 2.1 RSA

Um dos pilares dessa tecnologia é o RSA, nome derivado dos sobrenomes de seus inventores - Rivest, Shamir e Adleman. Introduzido em 1977, o RSA permanece como um dos algoritmos mais populares e confiáveis de criptografia de chave pública. Baseado na dificuldade matemática de fatorar grandes números, o RSA é amplamente utilizado para segurança de dados, autenticação e assinaturas digitais.

## 2.2 OAEP

Junto ao RSA, o Optimal Asymmetric Encryption Padding (OAEP) representa um avanço significativo na segurança de algoritmos de criptografia assimétrica. O OAEP melhora a segurança ao adicionar um elemento de aleatoriedade ao processo de cifragem, mitigando diversos tipos de ataques criptográficos. Essa técnica de padding é essencial para assegurar que o RSA permaneça resiliente contra as ameaças emergentes no mundo digital.

# 3 Desenvolvimento

## 3.1 Geração de Chaves RSA

A robustez e a segurança do algoritmo RSA residem fundamentalmente no processo de geração de chaves, um processo que envolve conceitos avançados de matemática e teoria dos números. Este processo pode ser dividido em várias etapas críticas, cada uma desempenhando um papel vital na garantia da segurança do sistema criptográfico.

## 3.2 Seleção de Números Primos

A primeira etapa na geração de chaves RSA é a seleção de dois grandes números primos, geralmente denominados p e q. Estes números devem ser grandes e, idealmente, devem ter comprimentos semelhantes para maximizar a segurança. A seleção de primos grandes e aleatórios é crucial porque a principal vulnerabilidade do RSA é a possibilidade de fatorar o produto desses dois números. Quanto maiores e mais aleatórios forem os primos, mais difícil será para um atacante realizar a fatoração e comprometer o sistema.

## 3.3 Cálculo do Módulo n

Após a escolha dos números primos, o próximo passo é calcular o módulo n, que é simplesmente o produto de p e q. O módulo n é usado como parte da chave pública e da chave privada no RSA. O tamanho de n (em bits) determina o tamanho da chave. A segurança do RSA baseia-se na premissa de que, embora seja fácil calcular n a partir de p e q, é extremamente difícil fazer o caminho inverso sem conhecer os valores originais de p e q.

### 3.4 Função Totiente de Euler

A Função Totiente de Euler, (n), é um conceito crucial na geração da chave privada no RSA. Para o módulo n, que é o produto de dois números primos, (n) é calculado como (p1)×(q1). Esta função é usada para garantir que o expoente escolhido para a chave pública e o cálculo da chave privada sejam realizados de maneira que a cifragem e a decifragem sejam operações inversas uma da outra. A seleção de um expoente público "e"que seja coprimo com (n) é fundamental para a validade do algoritmo.

### 3.5 Expoente Público "e"e Privado d

O expoente público "e"é escolhido dentro de certos parâmetros. Comumente, o valor 65537 é usado por ser um grande número primo que oferece um bom equilíbrio entre segurança e eficiência computacional. O expoente privado "d"é então calculado como o inverso multiplicativo de "e"módulo (n). O cálculo de "d"deve ser feito de forma que d×e seja congruente a 1 módulo (n), o que é essencial para a operação de decifragem funcionar corretamente.

A combinação dessas etapas resulta na formação de um par de chaves: a chave pública (e,n) e a chave privada (d,n)

### 3.6 OAEP e Segurança do RSA

O RSA, embora robusto, enfrenta desafios de segurança, especialmente quando utilizado em sua forma mais básica. É aqui que o Optimal Asymmetric Encryption Padding (OAEP) desempenha um papel crucial, elevando significativamente a segurança do RSA.

### 3.7 Mecanismo de Padding OAEP

O OAEP é uma técnica de padding que adiciona aleatoriedade e complexidade ao processo de criptografia RSA, tornando-o mais resistente a uma série de ataques criptográficos, incluindo ataques de texto cifrado escolhido. O processo começa com a adição de um padding aleatório à mensagem original, aumentando assim sua complexidade. Isso é seguido por uma etapa de mascaramento, onde o padding aleatório e a mensagem são misturados através de funções hash, resultando em uma sequência que é então criptografada. Esta abordagem impede que um atacante deduza qualquer informação útil da mensagem criptografada, mesmo que ele possua conhecimento parcial sobre a mensagem original ou o padding.

### 3.8 Processos de Cifração e Decifração

No contexto do RSA com OAEP, o processo de cifração envolve primeiro a aplicação do OAEP na mensagem original, resultando em uma mensagem com padding. Esta mensagem é então convertida em um número inteiro e cifrada usando a chave pública do destinatário. A mensagem criptografada pode ser transmitida com segurança e só pode ser decifrada pelo detentor da chave privada correspondente.

Para a decifragem, o processo é invertido. A mensagem criptografada é primeiramente decifrada usando a chave privada, resultando na mensagem com padding OAEP. Em seguida, o padding OAEP é cuidadosamente removido, revelando a mensagem original. Este processo assegura que apenas o destinatário pretendido, que possui a chave privada, possa acessar o conteúdo da mensagem.

### 3.9 Assinatura Digital RSA

Criação e Verificação de Assinaturas O processo de assinatura digital no RSA começa com a geração de um hash da mensagem usando uma função hash segura, como SHA-256. Este hash, que atua como um resumo da mensagem, é então criptografado com a chave privada do remetente, criando a assinatura digital.

Para verificar uma assinatura, o receptor primeiro decifra a assinatura usando a chave pública do remetente, revelando o hash da mensagem original. Este hash é então comparado com um novo hash gerado a partir da mensagem recebida. Se os hashes coincidirem, isso confirma que a mensagem veio do remetente declarado e não foi alterada durante o trânsito.

## 4 Conclusão

### 4.1 Simplificações na codificação

No código apresentado, algumas simplificações foram feitas para viabilizar a implementação do trabalho. Por exemplo, embora o código realize a geração de chaves RSA e inclua o mecanismo de OAEP para o padding, detalhes mais complexos como a gestão eficiente de chaves e a implementação de protocolos completos de

segurança para assegurar a integridade dos dados durante o transporte não são abordados. Além disso, a problematização ao ficar convertendo e validando string/bytes, que após uma analise profunda podemos perceber que a implementação em C ou mesmo Java, seria mais fácil ao trabalhar com esses tipos de dados. Entretanto, tornou-se um pouco menos complexo ao conhecer um pouco mais da linguagem python sobre a qual não possuía domínio.

Um dos maiores desafios é garantir a segurança contra adversários cada vez mais sofisticados, que pode eventualmente quebrar algoritmos como o RSA, e nosso caso ainda mais fácil quando verificamos a implementação do padding podendo ser quebrado se analisado o padrão dele gerado. Além disso, a complexidade na implementação correta e segura dessas tecnologias, especialmente em sistemas de grande escala, não deve ser subestimada. Erros de implementação podem comprometer a segurança, mesmo que os algoritmos subjacentes sejam teoricamente seguros.

Em resumo, enquanto o RSA e o OAEP continuam sendo pilares na segurança digital, os desafios na implementação e as exigências de um ambiente digital em rápida mudança requerem uma vigilância constante e adaptações regulares às novas realidades de segurança.

# Referências


[1] David M. Burton. *Elementary Number Theory*. 7th. McGraw-Hill Education, 2010.

[2] Jonathan Katz e Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman e Hall/CRC, 2007.

[3] Alfred J. Menezes, Paul C. van Oorschot e Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[4] *Post-Quantum Cryptography*. Accessed: [your access date here]. 2023.

[5] R.L. Rivest, A. Shamir e L. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. Vol. 21. 2. 1978, pp. 120–126.