1.Cifras

Algoritmo matemático que realiza a transformação dos dados em formato legível (***texto claro***) para uma forma ilegível (***texto cifrado***) e vice-versa. A cifra utiliza a chave de criptografia como parâmetro para determinar como os dados serão embaralhados ou substituídos durante o processo de criptografia. É importante mencionar que diferentes cifras podem ser utilizadas na criptografia simétrica, como o algoritmo DES, AES, RC4, entre outros.

Nas **cifras de transposição**, nenhuma carta é substituída; elas são simplesmente reorganizadas. Os algoritmos de criptografia de bloco de criptografia modernos, como AES e o legado 3DES, ainda usam a transposição como parte do algoritmo.

As **cifras de substituição** substituem uma letra para outra. Em sua forma mais simples, as cifras de substituição mantêm a frequência de letras da mensagem original. A cifra de César foi uma simples cifra de substituição. Porque a mensagem inteira dependeu da mesma mudança de chave única, a cifra César é referida como uma cifra de substituição monoalfabética. Também é bastante fácil de rachar. Por esta razão, as cifras polialfabéticas, como a cifra Vigenère, foram inventadas.

2.Criptografia e criptoanálise

A criptologia é a ciência de criar e violar os códigos secretos. A criptologia combina duas disciplinas separadas:

- **Criptografia:** O desenvolvimento e uso de códigos
- **Criptoanálise:** A quebra desses códigos

Enquanto houver criptografia, haverá criptoanálise. A criptoanálise é a prática e o estudo de determinar o significado de informações criptografadas (quebrando o código), sem acesso à chave secreta compartilhada. Isso também é conhecido como codebreaking. Vários métodos são usados na criptoanálise, como:

2.1 Método de força bruta

O atacante tenta todas as principais teclas sabendo que, eventualmente, um deles funcionará.

2.2 Método de cifra

O atacante tem o texto cifrado de várias mensagens criptografadas, mas nenhum conhecimento do texto simples subjacente.

2.3 Método de texto simples

O atacante tem acesso ao texto cifrado de várias mensagens e sabe algo sobre o texto simples subjacente a esse texto cifrado.

2.4 Método Chosen-plaintext

O atacante escolhe quais dados o dispositivo de criptografia criptografa e observa a saída do cifra de texto.

2.5 Método chosen-ciphertext

O atacante pode escolher diferentes textos cifrados a descriptografos e tem acesso ao texto simples descriptografado.

2.6 Método Meet-in-the-Middle

O atacante conhece uma parte do texto simples e do cifrado correspondente.

3.Gerenciamento de chaves

O **gerenciamento de chaves** é frequentemente considerado a parte mais difícil do projeto de um criptosistema. Muitos criptossistemas falharam devido a erros de gerenciamento de chave e todos os algoritmos de criptografia modernos procedimentos de gerenciamento de chaves. Na prática, a maioria dos ataques a sistemas criptográficos são direcionados ao nível de gerenciamento de chaves, e não ao algoritmo criptográfico em si.

3.1 Comprimento de chave

Também chamado de tamanho da chave, é a medida em bits

3.2 Keyspace

Este é o número de possibilidades que podem ser geradas por um comprimento de chave específico. O espaço da chave de um algoritmo é o conjunto de todos os valores de chave possíveis. Uma chave que tem ***n*** bits produz um keyspace que tem 2^n valores-chave possíveis. Ao adicionar um pouco à chave, o ***keyspace*** é efetivamente duplicado.

Ao adicionar um bit ao comprimento da chave, o espaço de tecla dobra, e um atacante precisa do dobro de tempo para pesquisar o espaço de tecla. **Chaves mais longas são mais seguras**; no entanto, eles também consomem mais recursos. Deve-se ter cuidado ao escolher chaves mais longas, pois lidar com elas pode adicionar uma carga implicar ao processador em produtos da extremidade inferior.

Quase todo algoritmo tem algumas chaves fracas em seu ***keyspace*** que permitem a um invasor quebrar a criptografia por meio de um atalho. Chaves fracas mostram as regularidades na criptografia. Vários tipos de chaves criptográficas podem ser gerados:

3.2.1 Chaves simétricas

Pode ser trocado entre dois roteadores que suportam uma VPN

3.2.2 Chaves assimétricas

São usados em aplicações HTTPS seguras

3.2.3 Chaves hash

São usados em geração de chaves simétricas e assimétricas, assinaturas digitais e outros tipos de aplicações

3.2.4 Assinaturas digitais

São usados quando se conecta a um site seguro

4.Criptografia simétrica

A criptografia simétrica envolve três componentes principais:

4.1 Texto claro (*plaintext*)

O texto claro refere-se aos dados originais que se deseja proteger. Pode ser qualquer forma de informação, como mensagens de texto, arquivos, documentos, imagens, entre outros. Antes de serem criptografados, esses dados estão em formato legível e compreensível.

4.2 Chave simétrica

A ***chave simétrica*** é um valor secreto, compartilhado entre o remetente e o destinatário da mensagem. É com base nessa chave que ocorre a transformação dos dados originais em formato ilegível. A chave simétrica é utilizada tanto para ***criptografar*** (***processo de transformação do texto claro em texto cifrado***) quanto para ***descriptografar*** (***processo de reconverter o texto cifrado em texto claro***) os dados. É crucial que a chave simétrica seja mantida em segredo e seja conhecida apenas pelas partes autorizadas.

4.3 Texto cifrado

O texto cifrado é o resultado da aplicação do algoritmo de criptografia a chave simétrica e ao texto claro. Após a criptografia, os dados originais são transformados em uma forma ilegível e aparentemente aleatória. O texto cifrado é o que é transmitido ou armazenado de forma segura, uma vez que não pode ser compreendido por terceiros que não possuam a chave correta para desfazer a criptografia.

A criptografia simétrica utiliza o texto claro, a chave simétrica e o texto cifrado para proteger a confidencialidade dos dados. O remetente utiliza a chave simétrica para criptografar o texto claro, gerando o texto cifrado, que pode ser transmitido ou armazenado com segurança. O destinatário, por sua vez, utiliza a mesma chave simétrica para descriptografar o texto cifrado e recuperar o texto claro original. Ela pode utilizar ***cifras de fluxo*** ou ***cifras de bloco***.

Os algoritmos simétricos usam a mesma chave pré-compartilhada para criptografar e descriptografar dados. Uma chave pré-compartilhada, também chamada de chave secreta, é conhecida pelo remetente e pelo receptor antes que qualquer comunicação criptografada possa ocorrer.

4.4 DES

O algoritmo ***DES*** (***Data Encryption Standard***) é um algoritmo de criptografia simétrica de blocos que foi amplamente utilizado durante muitos anos. O DES opera em blocos de dados de tamanho fixo de 64 bits e utiliza uma estrutura de rede Feistel, composta por 16 rounds de operações. Cada round consiste em uma série de etapas, incluindo permutação, substituição e combinação linear. Uma característica importante do DES é o uso de uma chave de 56 bits. Durante a criptografia, a chave é expandida para gerar 16 subchaves de 48 bits cada uma, uma para cada round. Essas subchaves são derivadas por meio de um processo de permutação e rotação da chave original.

4.5 Triple DES

O algoritmo ***3DES*** (***Triple Data Encryption Standard***), também conhecido como TDEA, é uma extensão do algoritmo DES (Data Encryption Standard) que visa aumentar a segurança usando múltiplas aplicações do DES em sequência. O 3DES opera em blocos de dados de tamanho fixo de 64 bits, assim como o DES, e utiliza uma estrutura de rede Feistel. Ele consiste em três etapas principais: criptografia, descriptografia e criptografia novamente (ou seja, EDE - Encrypt, Decrypt, Encrypt).

4.6 Blowfish

O algoritmo Blowfish é um algoritmo de criptografia simétrica de blocos projetado por Bruce Schneier em 1993. É um algoritmo de chave variável, o que significa que pode trabalhar com chaves de tamanho variável, de 32 bits a 448 bits. **O Blowfish opera em blocos de dados de tamanho fixo (normalmente 64 bits) e utiliza uma estrutura de rede Feistel**, que consiste em repetir uma série de transformações em cada bloco de dados. O algoritmo é dividido em duas partes principais: a etapa de inicialização e a etapa de criptografia/descriptografia. Uma característica importante do Blowfish é que ele é um algoritmo rápido e eficiente em termos de desempenho, tornando-o adequado para uma ampla gama de aplicações.

4.7 Twofish

O algoritmo Twofish é um algoritmo de criptografia simétrica de blocos que foi finalista no concurso ***AES*** (***Advanced Encryption Standard***) em 2000. Ele foi projetado por Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall e Niels Ferguson. O Twofish é considerado um algoritmo altamente seguro e eficiente. **O Twofish opera em blocos de dados de tamanho fixo, normalmente de 128 bits, e usa uma estrutura de rede Feistel**, semelhante ao algoritmo Blowfish. O processo de criptografia/descriptografia envolve quatro partes principais: expansão de chaves, permutação, substituição e combinação linear. O Twofish é altamente considerado por sua segurança e resistência a ataques criptográficos conhecidos. Ele suporta tamanhos de chave de 128, 192 e 256 bits, o que o torna uma escolha flexível para diferentes níveis de segurança.

4.8 RC4

O algoritmo ***RC4*** (***Rivest Cipher 4***) é um algoritmo de criptografia de fluxo, também conhecido como cifra de fluxo. Ele é amplamente utilizado em diversas aplicações, como protocolos de segurança, redes sem fio e sistemas de segurança de dados. O RC4 opera gerando uma sequência pseudoaleatória de bytes, conhecida como keystream, que é combinada com o texto claro por meio de uma operação XOR (OU exclusivo). O algoritmo possui duas partes principais: a etapa de inicialização e a geração do keystream. É importante mencionar que o RC4 foi amplamente utilizado no passado, mas foram descobertos alguns problemas de segurança relacionados ao seu uso em determinadas situações.

4.9 RC6

O algoritmo ***RC6*** (***Rivest Cipher 6***) é um algoritmo de criptografia simétrica de blocos, desenvolvido por Ron Rivest em 1997. Ele é projetado para oferecer segurança, eficiência e flexibilidade em termos de tamanho de chave. O RC6 opera em blocos de dados de tamanho fixo, normalmente de 128 bits, e **é baseado em uma estrutura de rede Feistel**, assim como o algoritmo Blowfish. O processo de criptografia/descriptografia envolve várias etapas, incluindo a expansão de chaves, a permutação, a substituição e a combinação linear. **O RC6 é considerado um algoritmo seguro e eficiente**, projetado para resistir a diversos ataques criptográficos conhecidos. Ele suporta tamanhos de chave de 128, 192 e 256 bits, o que proporciona flexibilidade em termos de níveis de segurança e requisitos de aplicação.

5.Algoritmos assimétricos

Os algoritmos assimétricos, também chamados algoritmos de chave pública, são projetados para que a chave usada para criptografia seja diferente da chave usada para descriptografia. A chave de descriptografia não pode, em uma quantidade razoável de tempo, ser calculada a partir da chave de criptografia e vice-versa.

Algoritmos assimétricos usam uma chave pública e uma chave privada. Ambas as chaves são capazes do processo de criptografia, mas a chave emparelhada complementar é necessária para descriptografar. O processo também é reversível. Os dados criptografados com a chave pública requerem a chave privada para descriptografar. Algoritmos assimétricos alcançam confidencialidade e autenticidade usando este processo.

Exemplos de protocolos que usam algoritmos de chave assimétrica incluem: IKE, SSL, PGP e SSH.

5.1 Componentes

A criptografia assimétrica, também conhecida como criptografia de chave pública, utiliza um par de chaves distintas para criptografar e descriptografar dados.

Esse par de chaves consiste em uma chave pública e uma chave privada. Para combinar eficiência e segurança, é comum utilizar a criptografia assimétrica em conjunto com a criptografia simétrica. Isso é conhecido como criptografia híbrida. Nesse caso, a criptografia assimétrica é usada para estabelecer uma chave de sessão segura, que é então usada para criptografar os dados de forma eficiente com a criptografia simétrica. A implementação adequada dessa abordagem requer a sincronização adequada dos algoritmos, o gerenciamento correto das chaves e a proteção das chaves de sessão.

5.2 Geração do par de chaves

O primeiro passo é gerar o par de chaves, composto pela chave pública e pela chave privada. Essas chaves são matematicamente relacionadas, mas computacionalmente inviáveis de serem derivadas uma da outra. **A chave pública é divulgada publicamente, enquanto a chave privada é mantida em sigilo pelo proprietário**.

5.3 Criptografia, transmissão e descriptografia

Para enviar uma mensagem segura para um destinatário, o remetente utiliza uma das chaves (pública ou privada) para criptografar a mensagem. Isso é feito aplicando uma função matemática específica à mensagem original, juntamente com a chave usada. O resultado é a mensagem criptografada, que só pode ser descriptografada com a outra chave correspondente.

A mensagem criptografada é transmitida e somente o destinatário que possui a chave correspondente poderá descriptografá-la.

Ao receber a mensagem criptografada, o destinatário utiliza a chave correspondente para descriptografar a mensagem, o que resulta na recuperação da mensagem original. Os algoritmos assimétricos **são substancialmente mais lentos** que os algoritmos simétricos. Seu design é baseado em problemas computacionais, como fatorar números extremamente grandes ou calcular logaritmos discretos de números extremamente grandes.

5.4 RSA

O algoritmo RSA, desenvolvido por Ron Rivest, Adi Shamir e Leonard Adleman em 1977, é um dos algoritmos de criptografia assimétrica mais amplamente utilizados. Ele se baseia na dificuldade de fatorar grandes números inteiros para fornecer segurança na troca de informações. **O tamanho das chaves seguras no algoritmo RSA depende da aplicação e do período de segurança desejado**. Normalmente, são utilizadas chaves com tamanho de 2048 bits ou mais, consideradas seguras para a maioria dos cenários.

5.5 DSA

O algoritmo DSA (Digital Signature Algorithm) é um algoritmo de assinatura digital, uma forma de criptografia assimétrica, desenvolvido pelo Governo dos Estados Unidos. Ele é amplamente utilizado para garantir a autenticidade e integridade de mensagens e documentos digitais. O tamanho das chaves seguras no algoritmo DSA é geralmente de 1024 bits ou mais, dependendo dos requisitos de segurança. O algoritmo DSA oferece um mecanismo eficiente e seguro para a geração de assinaturas digitais, garantindo a autenticidade e integridade de mensagens e documentos digitais.

5.6 Diffie-Hellman

O algoritmo Diffie-Hellman é um protocolo de troca de chaves que permite que duas partes estabeleçam uma chave secreta compartilhada, mesmo que estejam se comunicando por um canal inseguro. Ele foi desenvolvido por Whitfield Diffie e Martin Hellman em 1976 e é amplamente utilizado em criptografia de chave pública. **A segurança do algoritmo Diffie-Hellman está baseada na dificuldade do problema do logaritmo discreto**. O algoritmo Diffie-Hellman é uma ferramenta fundamental na área de criptografia, permitindo a troca segura de chaves em ambientes inseguros, contribuindo para a confidencialidade e privacidade das comunicações.

5.7 ECC

O algoritmo ECC (Elliptic Curve Cryptography) é um sistema criptográfico que se baseia na teoria das curvas elípticas sobre corpos finitos. Nesse algoritmo, a chave pública e a chave privada são geradas com base em operações matemáticas na curva elíptica. A curva elíptica utilizada é definida por uma equação matemática específica. Uma característica importante das curvas elípticas é que elas possuem um grupo aditivo associado a elas. Essa propriedade permite realizar operações matemáticas, como adição e multiplicação, entre pontos na curva. Essas operações são utilizadas na geração de chaves e no processo de criptografia e descriptografia. No contexto do ECC, a segurança é baseada na complexidade do problema matemático conhecido como "***problema do logaritmo discreto***". Em geral, uma chave ECC de 256 bits é considerada segura o suficiente para a maioria das aplicações, enquanto um nível de segurança semelhante com outros algoritmos requer chaves de 2048 bits ou mais.

5.8 ElGamal

O algoritmo ElGamal é um algoritmo de criptografia assimétrica que combina a criptografia de chave pública e a troca de chaves de Diffie-Hellman. Ele é amplamente utilizado para garantir a confidencialidade das comunicações e proteger a privacidade dos dados. O tamanho das chaves seguras no algoritmo ElGamal geralmente é de 2048 bits ou mais, dependendo dos requisitos de segurança. O algoritmo ElGamal oferece uma abordagem segura e eficiente para criptografia de chave pública, fornecendo confidencialidade e privacidade na comunicação.