

Segurança da Informação

Aula6 – Criptografia

Curso de Ciência da Computação

Prof. Dr. Rodrigo Xavier de Almeida Leão



CRIPTOGRAFIA

- Criptografia é a prática de proteger a comunicação contra acesso não autorizado ou interceptação, convertendo-a em um código ou cifra.
- Algoritmos e protocolos matemáticos para transformar texto simples (dados não criptografados) em texto cifrado (dados criptografados) que só podem ser decifrados tendo acesso à chave usada para criptografar os dados



Os objetivos básicos da criptografia são (KATZ; LINDELL, 2007), (MENEZES; OORSCHOT; VANSTONE 2001), (FIARRESGA, 2010):

- Sigilo: proteção dos dados contra divulgação não autorizada.
- Autenticação: garantia que a entidade se comunicando é aquela que ela afirma ser.
- Integridade: garantia de que os dados recebidos estão exatamente como foram enviados por uma entidade autorizada.
- Não repúdio: garantia que não se pode negar a autoria de uma mensagem.
- Anonimato: garantia de não rastreabilidade de origem de uma mensagem.

A criptografia é muito mais do que sigilo e garantia de confidencialidade. Pense em variadas aplicações: autenticação de mensagens, assinatura digital, protocolos para troca de chaves secretas, protocolos de autenticação, leilões e eleições eletrônicos, dinheiro digital.

QUEBRAS DE CRIPTOGRAFIA

O algoritmo RC4 já foi pivô de problemas no WEP (*Wired Equivalent Privacy*), protocolo de segurança utilizado em redes Wi-Fi em 2001. Já em 2013, o TLS/SSL teve uma grave falha de segurança divulgada relacionada ao mesmo algoritmo RC4 (SOPHOS, 2013).

O DES (*Data Encryption Standard*) teve sua efetividade invalidada em 1997, quando uma mensagem cifrada com o algoritmo foi quebrado pela primeira vez. Os custos dos equipamentos que realizavam o ataque de força bruta foram diminuindo, ao mesmo tempo em que o tempo para a quebra também foi sendo drasticamente reduzida. Em 1998, um equipamento com custo de US\$ 250 mil quebrou uma chave de 56 bits em aproximadamente dois dias (NOMIYA, 2010).

<https://learn.microsoft.com/pt-br/defender-for-identity/security-assessment-weak-cipher>

O que são cifras fracas?

A criptografia depende de cifras para criptografar nossos dados. Por exemplo, RC4 (Rivest Cipher 4 também conhecido como ARC4 ou ARCFOUR que significa Alegado RC4) é um. Embora o RC4 seja notável por sua simplicidade e velocidade, várias vulnerabilidades foram descobertas desde o lançamento original do RC4, tornando-o inseguro. RC4 é especialmente vulnerável quando o início do fluxo de chaves de saída não é descartado, ou quando chaves não aleatórias ou relacionadas são usadas.

 Filtrar por título

Documentação do Microsoft
Defender para Identidade

▸ Visão geral

▾ Implantar

Guia de instalação rápida

[Learn](#) /



Documentação do Microsoft Defender para Identidade

O serviço de nuvem Microsoft Defender para Identidade ajuda a proteger ambientes híbridos empresariais contra vários tipos de ameaças avançadas, internas ou ataques cibernéticos direcionados.

A segurança de sistemas criptográficos depende de uma série de fatores, tais como (NAKAMURA; GEUS, 2007):

- Geração de chaves: sem uma geração aleatória de chaves, o algoritmo utilizado pode revelar padrões que diminuem o espaço de escolha das chaves, o que facilita a sua descoberta.
- Mecanismo de troca de chaves: as chaves precisam ser distribuídas e trocadas para o estabelecimento das comunicações seguras e, para tanto, protocolos como o *Diffie-Hellman* são utilizados. Esse protocolo será discutido nas próximas aulas.
- Taxa de troca das chaves: quanto maior a frequência de troca automática das chaves, maior será a segurança, pois isso diminui a janela de oportunidade de ataques, pois, caso uma chave seja quebrada, em pouco tempo ela já não é mais útil para a comunicação.
- Tamanho da chave: são diferentes para a criptografia de chave privada ou simétrica e para a criptografia de chaves públicas ou assimétricas.

O primeiro uso documentado da criptografia foi em torno de 1900 a.C., no Egito, quando um escriba usou hieróglifos fora do padrão numa inscrição. Já entre 600 a.C. e 500 a.C., os hebreus utilizavam a cifra de substituição simples, que era fácil de ser revertida com o uso de cifragem dupla para obter o texto original. A Cifra de César é um dos exemplos mais clássicos de criptografia, em que as substituições eram feitas com as letras do alfabeto avançando três casas.

Com a Cifra de Vigenère a evolução consistiu no uso de diferentes valores de deslocamento para as substituições.


```
def criptografar(mensagem):
    mensagem_criptografada = ""
    for char in mensagem:
        # Verifica se o caractere é uma letra
        if char.isalpha():
            # Calcula a posição da letra criptografada no alfabeto
            posicao = ord(char) + 3
            # Se a posição ultrapassar 'z' ou 'Z', retorna ao início do alfabeto
            if char.islower() and posicao > ord('z'):
                posicao -= 26
            elif char.isupper() and posicao > ord('Z'):
                posicao -= 26
            mensagem_criptografada += chr(posicao)
        else:
            # Se o caractere não for uma letra, mantém inalterado
            mensagem_criptografada += char
    return mensagem_criptografada

# Exemplo de uso
mensagem_original = "Exemplo de mensagem a ser criptografada!"
mensagem_criptografada = criptografar(mensagem_original)
print("Mensagem original:", mensagem_original)
print("Mensagem criptografada:", mensagem_criptografada)
```

```
def descriptografar(mensagem_criptografada):
    mensagem_descriptografada = ""
    for char in mensagem_criptografada:
        # Verifica se o caractere é uma letra
        if char.isalpha():
            # Calcula a posição da letra original no alfabeto
            posicao = ord(char) - 3
            # Se a posição for menor que 'a' ou 'A', retorna ao final do alfabeto
            if char.islower() and posicao < ord('a'):
                posicao += 26
            elif char.isupper() and posicao < ord('A'):
                posicao += 26
            mensagem_descriptografada += chr(posicao)
        else:
            # Se o caractere não for uma letra, mantém inalterado
            mensagem_descriptografada += char
    return mensagem_descriptografada

# Exemplo de uso
mensagem_criptografada = "Hankpsh gh phrjwldj d vhu flupsrjdugd!"
mensagem_descriptografada = descriptografar(mensagem_criptografada)
print("Mensagem criptografada:", mensagem_criptografada)
print("Mensagem descriptografada:", mensagem_descriptografada)
```

O primeiro uso documentado da criptografia foi em torno de 1900 a.C., no Egito, quando um escriba usou hieróglifos fora do padrão numa inscrição. Já entre 600 a.C. e 500 a.C., os hebreus utilizavam a cifra de substituição simples, que era fácil de ser revertida com o uso de cifragem dupla para obter o texto original. A Cifra de César é um dos exemplos mais clássicos de criptografia, em que as substituições eram feitas com as letras do alfabeto avançando três casas.

Com a Cifra de Vigenère a evolução consistiu no uso de diferentes valores de deslocamento para as substituições.

```
1 def descriptografar(mensagem_criptografada):
2     mensagem_descriptografada = ""
3     for char in mensagem_criptografada:
4         # Verifica se o caractere é uma letra
5         if char.isalpha():
6             # Calcula a posição da letra original no alfabeto
7             posicao = ord(char) - 3
8             # Se a posição for menor que 'a' ou 'A', retorna ao
9             # final do alfabeto
10            if char.islower() and posicao < ord('a'):
11                posicao += 26
12            elif char.isupper() and posicao < ord('A'):
13                posicao += 26
14            mensagem_descriptografada += chr(posicao)
15        else:
16            # Se o caractere não for uma letra, mantém inalterado
17            mensagem_descriptografada += char
18    return mensagem_descriptografada
19
20 # Exemplo de uso
21 mensagem_criptografada = "Hankpsh gh phrjwldj d vhu flupsrjdugd!"
22 mensagem_descriptografada = descriptografar(mensagem_criptografada)
```

AI just now ⌵ Debug ⚡ Basic

The error in the code snippet is asking you to combine the two `if` branches using the logical `or` operator. This means that you can merge the conditions into one `if` statement by using `or` to combine them.

Here is the corrected code snippet with the explanation:

Python

Copy → Insert

```
if (char.islower() and posicao <
ord('a')) or (char.isupper() and posicao <
ord('A')):
    posicao += 26
```

By using the `or` operator, you can check both conditions in a single `if` statement.

Rerun ▾

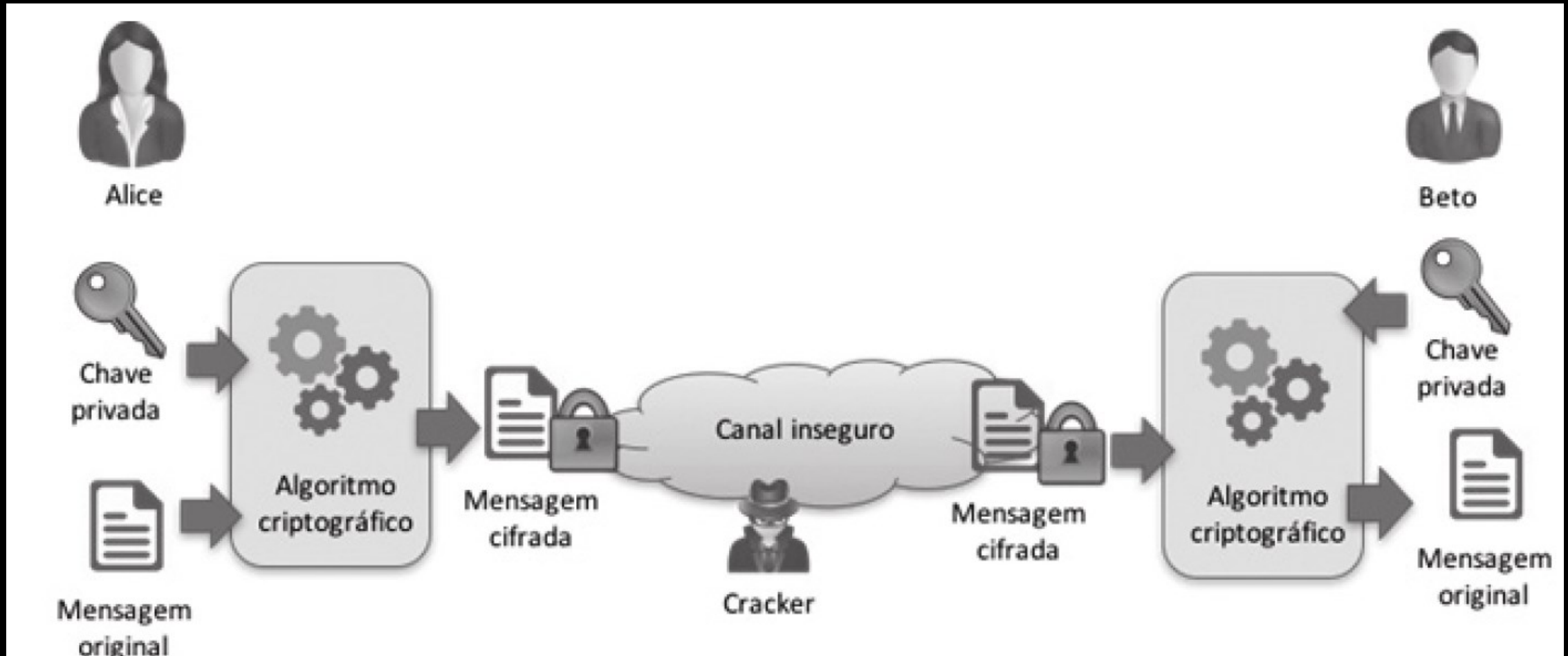
Type message...

CRIPTOGRAFIA SIMÉTRICA – CHAVE PRIVADA

- Uma única chave é usada tanto para criptografar quanto para descriptografar os dados. As chaves simétricas são compartilhadas entre as partes que desejam se comunicar de forma segura.
- Algoritmos comuns de criptografia simétrica incluem AES (Advanced Encryption Standard) e DES (Data Encryption Standard).

○ AES (*Advanced Encryption Standard*), também conhecido por Rijndael, é considerado o padrão de criptografia, substituindo o DES (*Data Encryption Standard*). O AES é uma cifra de blocos de 128 bits.

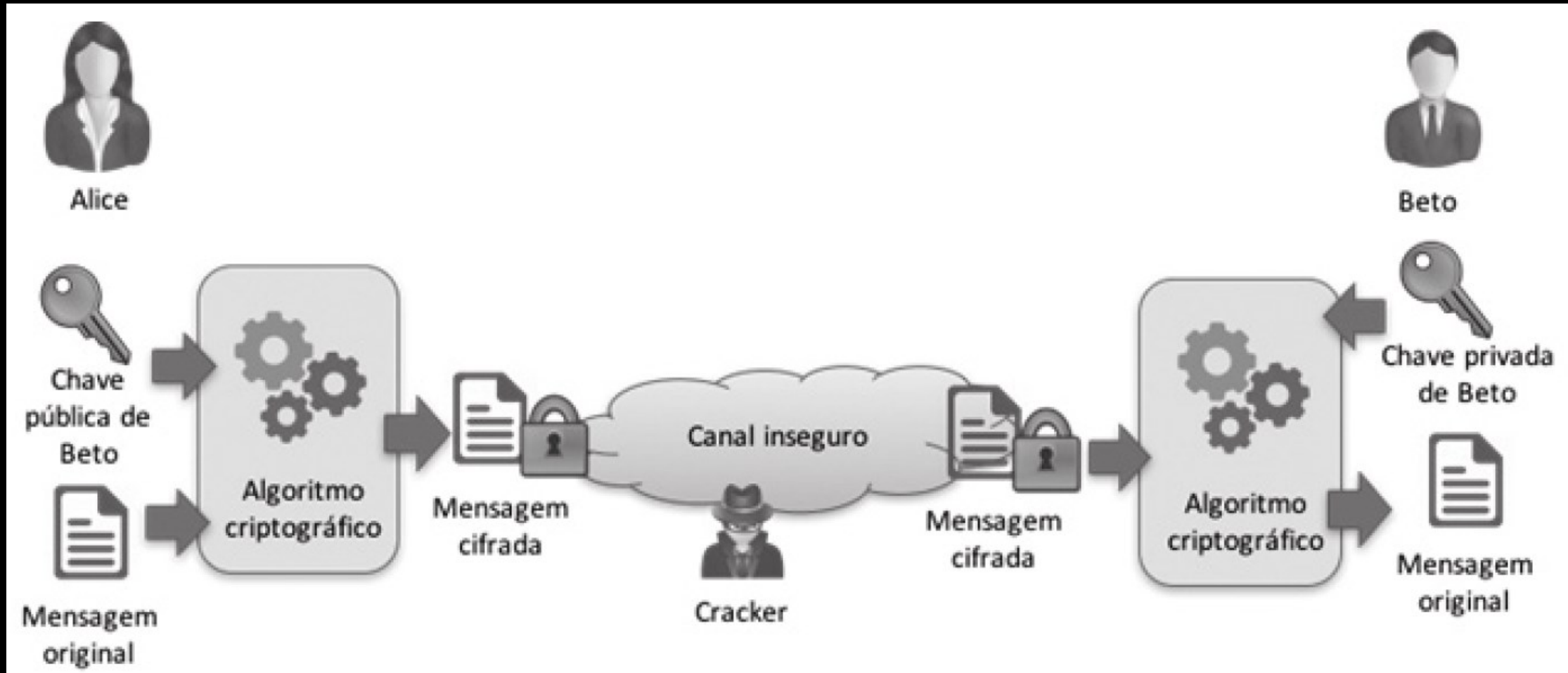
CRİPTOGRAFIA SIMÉTRICA – CHAVE PRIVADA



CRIPTOGRAFIA ASSIMÉTRICA – CHAVE PÚBLICA

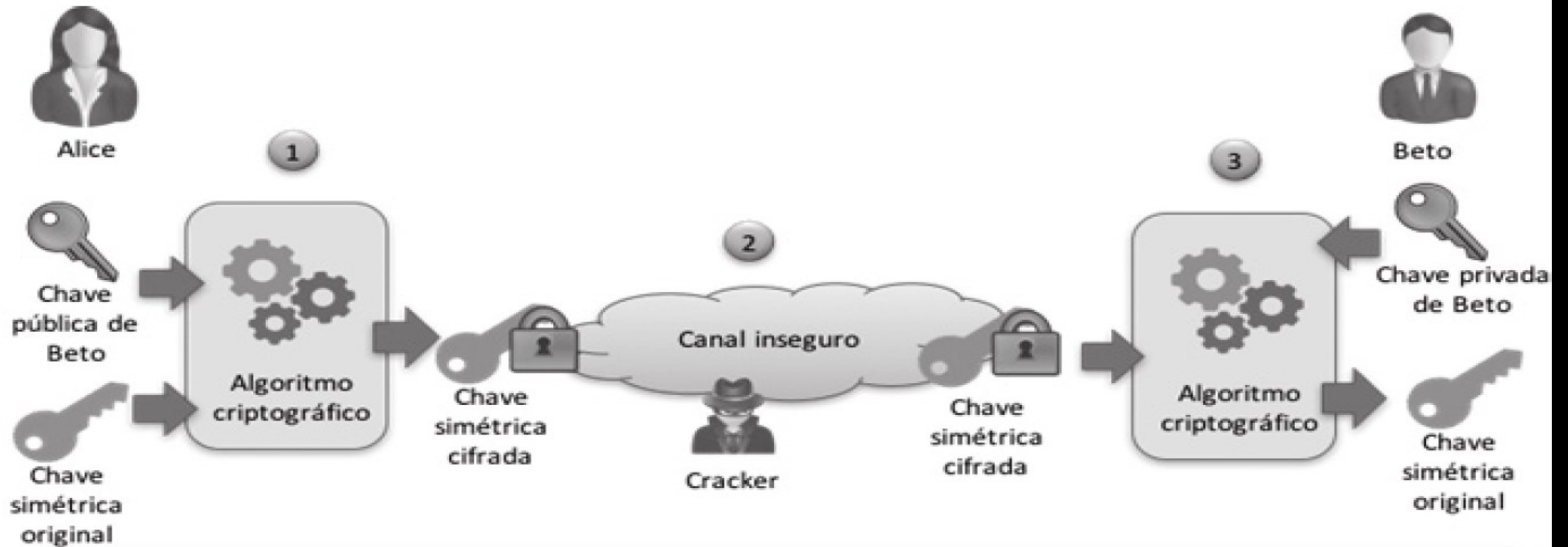
- Um par de chaves é utilizado: uma chave pública e uma chave privada.
- A chave pública é compartilhada livremente e usada para criptografar dados, enquanto a chave privada é mantida em segredo e usada para descriptografar os dados.
- Algoritmos de criptografia assimétrica incluem RSA (Rivest-Shamir-Adleman) e ECC (Elliptic Curve Cryptography)

CRİPTOGRAFIA ASSIMÉTRICA – CHAVE PÚBLICA



CRIPTOGRAFIA ASSIMÉTRICA – CHAVE PÚBLICA

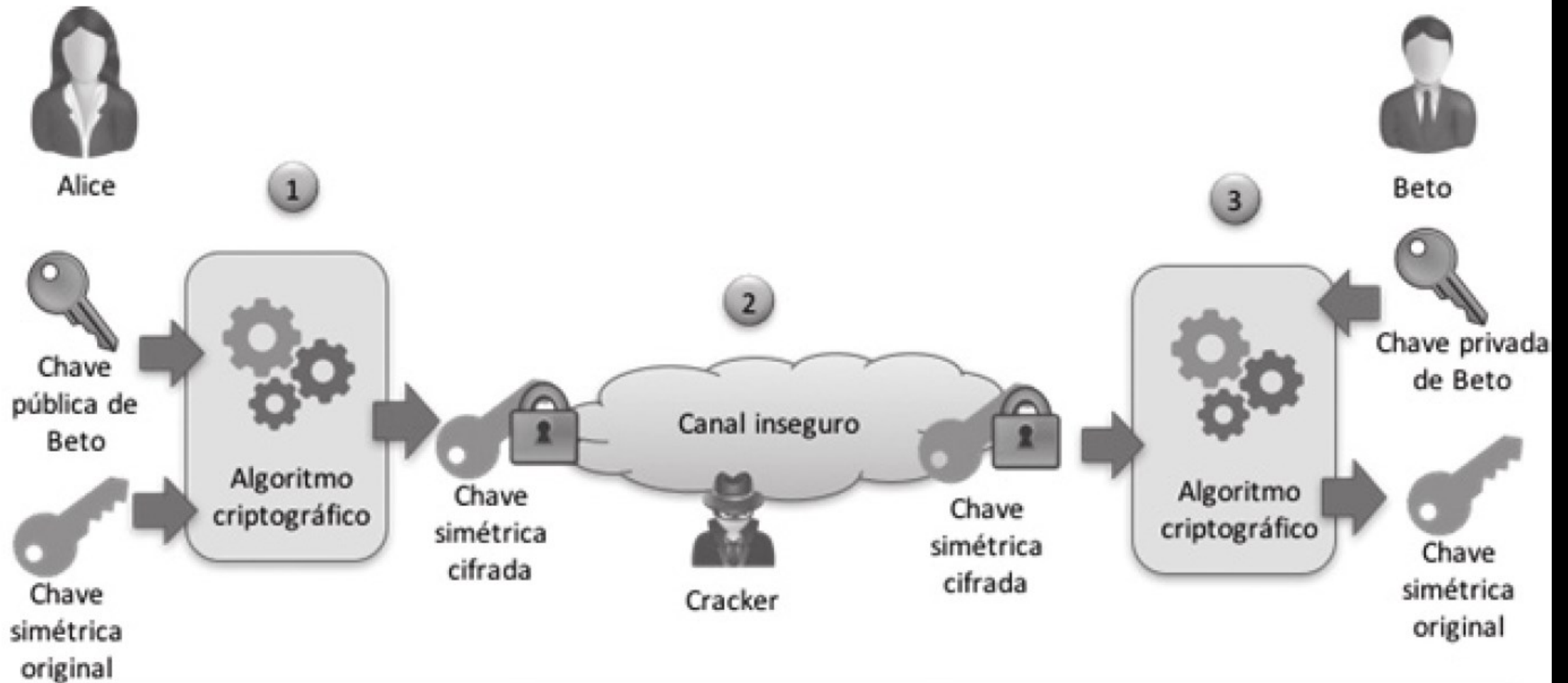
Figura 3.4 | Criptografia de chave pública sendo utilizada para a troca de chave privada compartilhada



Esse mecanismo de uso em conjunto da criptografia de chave pública com a criptografia de chave privada é bastante comum em várias aplicações, como é o caso do SSL (*Secure Sockets Layer*, para transmissões seguras), por exemplo, que iremos discutir com mais detalhes na próxima aula.

CRIPTOGRAFIA ASSIMÉTRICA – CHAVE PÚBLICA

Figura 3.4 | Criptografia de chave pública sendo utilizada para a troca de chave privada compartilhada



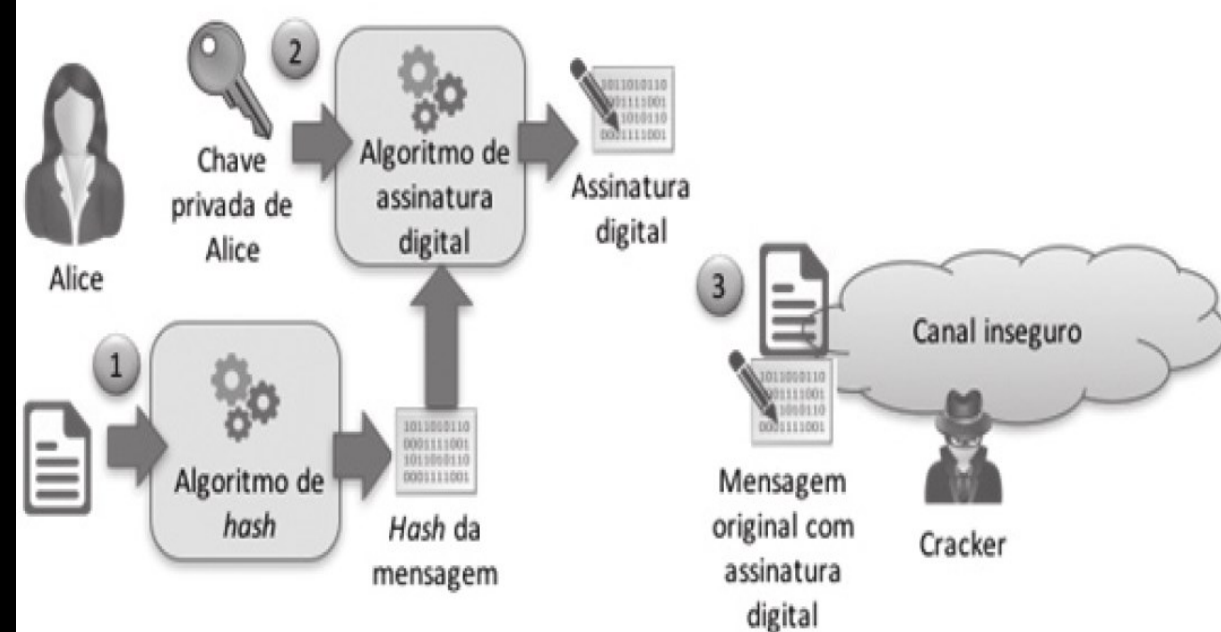
SSL/TLS: Protocolos de segurança usados para proteger comunicações online, como conexões HTTPS, e-mails seguros e VPNs.

- **PGP/GPG:** Protocolos de criptografia de e-mail que garantem a privacidade e a autenticidade das mensagens.
- **Criptomoedas:** Tecnologias como o Bitcoin utilizam criptografia para garantir a segurança das transações e a integridade do sistema.
- **Hashing:** Uma técnica relacionada à criptografia, mas não reversível. Os algoritmos de hash produzem uma "impressão digital" única de um conjunto de dados, usada para verificar a integridade dos dados

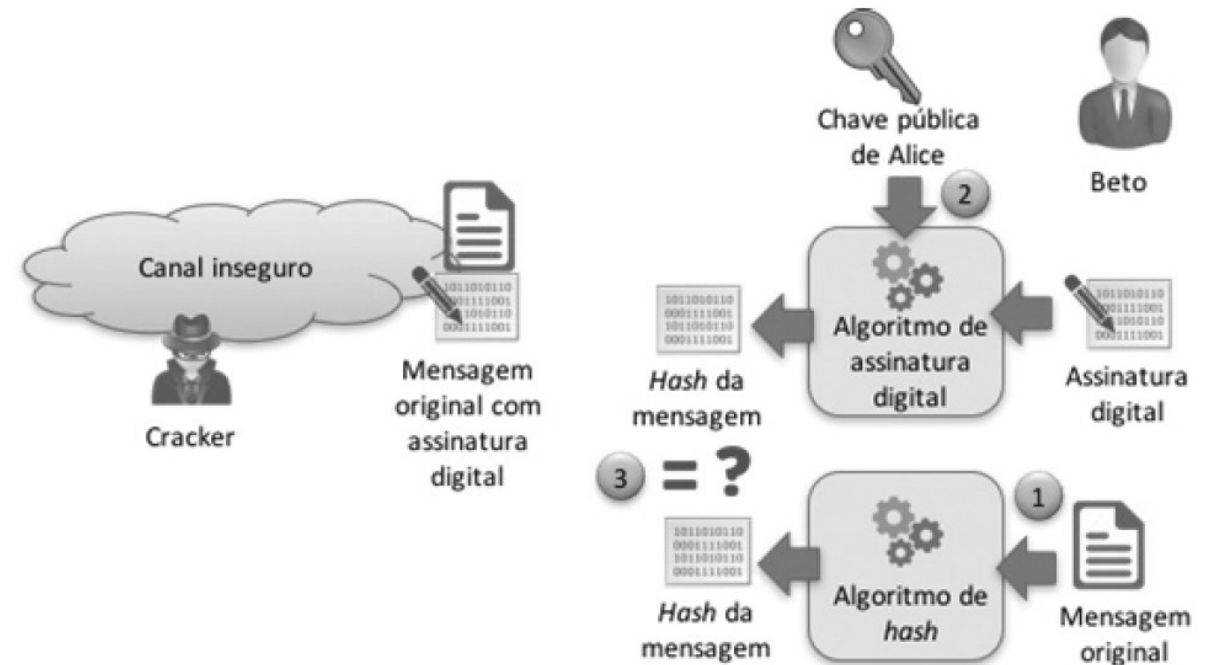


- Existem vários protocolos e técnicas que utilizam a criptografia para diferentes finalidades:
- **SSL/TLS:** Protocolos de segurança usados para proteger comunicações online, como conexões HTTPS, e-mails seguros e VPNs.
- **PGP/GPG:** Protocolos de criptografia de e-mail que garantem a privacidade e a autenticidade das mensagens.
- **Criptomoedas:** Tecnologias como o Bitcoin utilizam criptografia para garantir a segurança das transações e a integridade do sistema.
- **Hashing:** Uma técnica relacionada à criptografia, mas não reversível. Os algoritmos de hash produzem uma "impressão digital" única de um conjunto de dados, usada para verificar a integridade dos dados

ASSINATURA DIGITAL



- 1 Um *hash* da mensagem original é gerado.
- 2 Alice utiliza a sua chave privada para assinar digitalmente o *hash* da mensagem original.
- 3 A assinatura digital é enviada para Beto, junto com a mensagem original.



- 1 Um *hash* da mensagem original é gerado.
- 2 Beto utiliza a chave pública de Alice sobre a assinatura digital de Alice, o que gera o *hash*.
- 3 Caso o *hash* gerado a partir da mensagem original seja igual ao *hash* gerado a partir da assinatura digital, então a mensagem teve origem pela Alice.

Diffie-Hellman

O Diffie-Hellman, criado em 1976 por Whitfield Diffie e Martin Hellman, foi o primeiro método criptográfico para troca de chaves; este permite que duas entidades que não possuem conhecimento prévio uma da outra possam compartilhar uma chave secreta mesmo com o uso de um canal inseguro.

Matematicamente, o Diffie-Hellman utiliza o cálculo de logaritmos discretos em um campo infinito para gerar e estabelecer uma chave secreta compartilhada, a partir de uma informação prévia comum que não é crítica, no caso de ser comprometida.

Os passos gerais são:

- Um número (n_1) é compartilhado por Alice e Beto, o qual pode ser capturado por um terceiro.
- Alice e Beto escolhem, cada um, um número que não é compartilhado com ninguém. Alice escolhe n_2 e Beto escolhe n_3 .
- Alice e Beto realizam, cada um, um cálculo entre n_1 , que é compartilhado entre eles, n_2 e n_3 , respectivamente. São gerados assim n_4 e n_5 .
- Alice e Beto trocam entre si os números n_4 e n_5 , que são números gerados a partir dos cálculos entre n_1 , que é compartilhado entre ambos, e n_2 e n_3 , que são números privados de Alice e Beto, respectivamente. Os números n_4 e n_5 são irreversíveis para n_2 e n_3 , que são os números privados e secretos.
- Alice e Beto fazem um cálculo, respectivamente, entre n_2 e n_5 , e entre n_3 e n_4 , o que gera uma chave n_6 , que é comum aos dois.
- O número n_6 é a chave comum, secreta e compartilhada, que é utilizada na comunicação segura.

RSA

Os pesquisadores do MIT, Ron Rivest, Adi Shamir e Leonard Adleman, publicaram o algoritmo RSA em 1978, com o uso de exponenciação modular do produto de dois números primos muito grandes para cifragem e decifragem, além da assinatura digital.

O algoritmo RSA é composto por 3 partes:

1. Geração de chaves pública e privada.
2. Cifragem.
3. Decifragem.

De uma forma bastante geral, a geração das chaves pública e privada é feita a partir de dois números primos, que passam por uma série de cálculos até que se chegue às chaves pública e privada.

1. Geração das Chaves:

Escolha dois números primos grandes distintos, p e q .

Calcule o produto $n = p \times q$. Este é o módulo da chave pública e da chave privada.

Calcule a função totiente de Euler $\phi(n) = (p - 1) \times (q - 1)$.

Escolha um número inteiro e (chave pública) tal que $1 < e < \phi(n)$ e e seja coprimo com $\phi(n)$ (ou seja, $\text{mdc}(e, \phi(n)) = 1$).

Calcule o inverso multiplicativo de e módulo $\phi(n)$ para obter d (chave privada). Isso pode ser feito usando o algoritmo de Euclides estendido.

As chaves pública e privada são então (e, n) e (d, n) , respectivamente.

2. Criptografia:

Para criptografar uma mensagem M , converta-a para um número inteiro m que seja menor que n .

Calcule $C = m^e \mod n$. C é a mensagem criptografada.

3. Descriptografia:

Para descriptografar a mensagem criptografada C , calcule $M = C^d \mod n$.

1. Geração das Chaves:

Escolhemos dois números primos: $p = 61$ e $q = 53$.

Calculamos $n = p \times q = 61 \times 53 = 3233$.

Calculamos $\phi(n) = (p - 1) \times (q - 1) = 60 \times 52 = 3120$.

Escolhemos um expoente de criptografia público e . Vamos usar $e = 17$.

Calculamos o inverso multiplicativo de e módulo $\phi(n)$. Ou seja, encontramos d tal que $d \times e \equiv 1 \pmod{\phi(n)}$.

Usando o algoritmo de Euclides estendido ou outras técnicas, descobrimos que $d = 2753$.

Então, as chaves pública e privada são $(e, n) = (17, 3233)$ e $(d, n) = (2753, 3233)$, respectivamente.

2. Criptografia:

Digamos que queremos criptografar a mensagem $M = 123$.

Calculamos $C = M^e \pmod{n} = 123^{17} \pmod{3233} = 855$.

3. Descriptografia:

Para descriptografar a mensagem criptografada $C = 855$:


Calculamos $M = C^d \pmod{n} = 855^{2753} \pmod{3233}$.

Após a computação, descobrimos que $M = 123$, que é a mensagem original.

5. Cálculo do inverso multiplicativo de e módulo $\phi(n)$:

Agora, precisamos encontrar um número inteiro d tal que $d \times e \equiv 1 \pmod{\phi(n)}$. Em outras palavras, d é o inverso multiplicativo de e módulo $\phi(n)$. Para calcular d , podemos usar o algoritmo de Euclides estendido ou outras técnicas. No entanto, para simplificar, usaremos a função `pow()` em Python. No Python, a função `pow(x, y, z)` calcula $x^y \pmod{z}$. Portanto, podemos usar essa função para calcular d diretamente. Em nosso exemplo, $e = 17$ e $\phi(n) = 3120$. Portanto, d é o inverso multiplicativo de 17 módulo 3120.

python

 Copy code

```
d = pow(e, -1, phi_n)
```

A função `pow(17, -1, 3120)` retorna $d = 2753$.

Portanto, o valor de d é 2753, que é a chave privada no exemplo dado.