



SEGURANÇA DE SOFTWARE

Avelino Francisco Zorzo – Aula 03

Professores

MOISES BRANDALISE

Professor Convidado

Atua como Especialista em Segurança da Informação em uma instituição financeira. Na carreira, atuou no ramo da indústria por 10 anos no papel de líder técnico em infraestrutura de tecnologia e 3 anos como Analista de desenvolvimento de Sistemas em fábrica de software. Em segurança da informação, atuou na indústria da mídia por 6 anos como Analista e no segmento financeiro, por 4 anos como Especialista, além de 2 anos como Especialista em Proteção de dados pessoais, totalizando cerca de 25 anos de mercado.

AVELINO ZORZO

Professor PUCRS

Associado da Sociedade Brasileira de Computação (SBC) e da IEEE. Possui graduação em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (1986-1989), mestrado em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (1990-1994), doutorado em Ciência da Computação pela University of Newcastle Upon Tyne (1995-1999) e pós-doutorado na área de segurança no Cybercrime and Computer Security Centre da Newcastle University (2012-2013). Atualmente é professor titular da Escola Politécnica da Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Coordenador de Programas Profissionais da área de Computação da CAPES/MEC, avaliador de condições de ensino do Ministério da Educação, consultor ad hoc do CNPq, CAPES e da FAPERGS.

Ementa da disciplina

Estudo sobre os métodos e utilização de criptografia para transmissão e armazenamento. Estudo sobre protocolo de comunicação em navegadores (HTTPS) ou aplicativos de conversa (LibSignal). Estudo sobre segurança no desenvolvimento de software. Estudo sobre os problemas mais frequentes indicados pela OWASP. Estudo sobre métodos de autenticação e autorização.

Segurança de Software

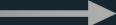
Por Avelino F. Zorzo - PUCRS

AULAS ANTERIORES

TEORIA



EXEMPLOS
PRÁTICOS

- 
1. INTRODUÇÃO A SEGURANÇA DE SOFTWARE
 2. MÉTODOS DE CRIPTOGRAFIA
 3. PROTOCOLOS DE COMUNICAÇÃO SEGURA
 4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE
 5. PROBLEMAS COMUNS DE SEGURANÇA INDICADOS PELA OWASP
 6. AUTENTICAÇÃO E AUTORIZAÇÃO

AGENDA



CONTEXTO ATUAL
E
SEGURANÇA



CRİPTOGRAFIA:
COMO FUNCIONA

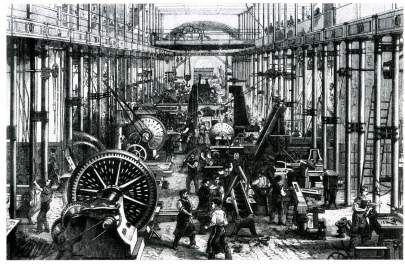


EXEMPLO PRÁTICO:
WHATSAPP

Parte 1 – Contexto Atual



- Contexto Atual
- Segurança
- Teoria vs prática
- Problemas práticos
- Segurança - Metas



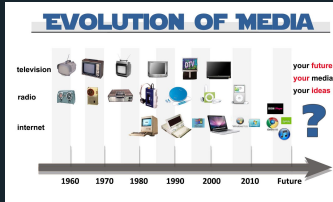
1. CONTEXTO ATUAL E SEGURANÇA

a) MUNDO EM MUDANÇAS



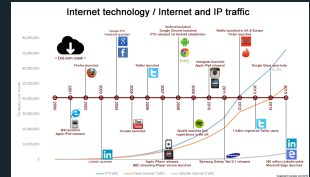


- Contexto Atual
- Segurança
- Teoria vs prática
- Problemas práticos
- Segurança - Metas



1. CONTEXTO ATUAL E SEGURANÇA

a) MUNDO EM MUDANÇAS





- Contexto Atual
- Segurança
- Teoria vs prática
- Problemas práticos
- Segurança - Metas

1. CONTEXTO ATUAL E SEGURANÇA

b) MUNDO FÍSICO vs MUNDO DIGITAL

Mundo Físico
(conhecido?)

Mundo Digital
(desconhecido)



Física

- Eletricidade
- Aeronáutica
-

Química

- Medicamentos
- Combustíveis
-

Biologia

- Alimentos
- Corpo humano
-



Internet

Ciência
de Dados

Algoritmo

Robôs

Inteligência
Artificial

Criptografia

Redes
 Sociais

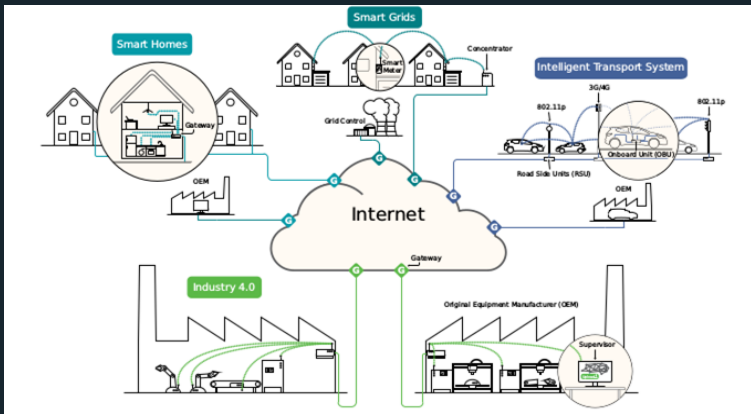




- Contexto Atual
- Segurança
- Teoria vs prática
- Problemas práticos
- Segurança - Metas

1. CONTEXTO ATUAL E SEGURANÇA

c) MUNDO DIGITAL – Smart*



Fonte: Boudguiga, et al., "Towards better availability and accountability for iot updates by means of a blockchain,"



- Contexto Atual
- Segurança
- Teoria vs prática
- Problemas práticos
- Segurança - Metas

1. CONTEXTO ATUAL E SEGURANÇA

d) MUNDO DIGITAL – Redes Sociais





- Contexto Atual
- Segurança
- Teoria vs prática
- Problemas práticos
- Segurança - Metas

1. CONTEXTO ATUAL E SEGURANÇA

e) MUNDO DIGITAL – Dados



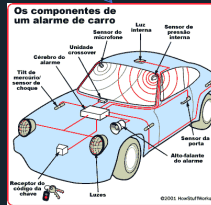
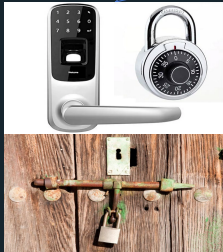
Segurança



- Contexto Atual
- **Segurança**
- Teoria vs prática
- Problemas práticos
- Segurança - Metas

1. CONTEXTO ATUAL E SEGURANÇA

f) SEGURANÇA NO MUNDO FÍSICO

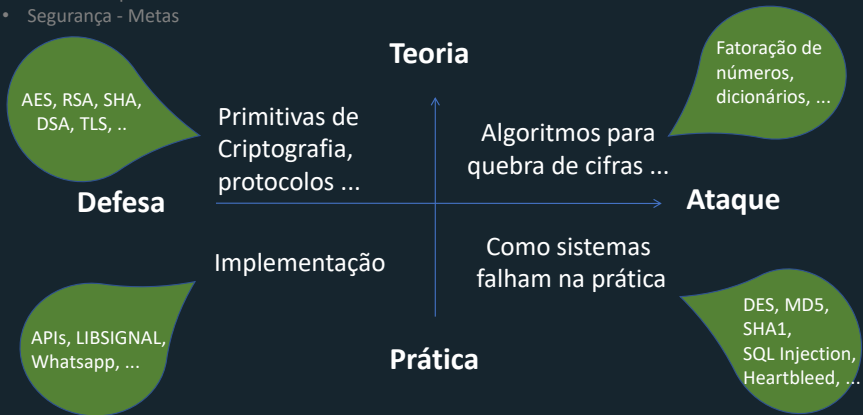




- Contexto Atual
- **Segurança**
- Teoria vs prática
- Problemas práticos
- Segurança - Metas

1. CONTEXTO ATUAL E SEGURANÇA

g) SEGURANÇA NO MUNDO DIGITAL



Teoria vs. Prática



- Contexto Atual
- Segurança
- Teoria vs prática
- Problemas práticos
- Segurança - Metas

1. CONTEXTO ATUAL E SEGURANÇA

h) SEGURANÇA NO MUNDO DIGITAL – teoria vs prática – grandes números

- Número de átomos no planeta 2^{170}
- Número de átomos no sol 2^{190}
- Número de átomos na galáxia 2^{223}
- Número de átomos no universo 2^{265}
- Tamanho de uma chave de 256 $\rightarrow 2^{256}$



- Contexto Atual
- Segurança
- Teoria vs prática
- Problemas práticos
- Segurança - Metas

1. CONTEXTO ATUAL E SEGURANÇA

i) SEGURANÇA NO MUNDO DIGITAL – teoria vs prática - tempo

- Tempo até a próxima era do gelo 2^{14} anos
- Tempo até o sol virar nova 2^{30} anos
- Idade do planeta Terra 2^{30} anos
- Idade do Universo 2^{34} anos
- Tempo para quebrar por força bruta uma chave de 256 bits $\rightarrow \sim 2^{192}$ anos
(Assumindo testar 1 bilhão de chaves em 1ms)

Dois problemas práticos – engenharia social programação*



- Contexto Atual
- Segurança
- Teoria vs prática
- **Problemas práticos**
- Segurança - Metas

1. CONTEXTO ATUAL E SEGURANÇA

j) ENGENHARIA SOCIAL – ataque simples no whatsapp

Site (legítimo)



Usuário



Criminoso





- Contexto Atual
- Segurança
- Teoria vs prática
- **Problemas práticos**
- Segurança - Metas



1. CONTEXTO ATUAL E SEGURANÇA

k) PROGRAMAÇÃO– Heartbleed - HTTPS

Funcionamento
normal do
heartbeat

Cliente
normal

Servidor me
envia uma
palavra de 6
letras se estiver
ai: **Grêmio**

Grêmio

Bob está conec-
tado. Usuário
Alice quer a
palavra de 6 letras
Grêmio. **Senha do
Bob xykzdp1**

Funcionamento
malicioso do
heartbeat

Cliente
malicioso

Servidor me
envia uma
palavra de 64000
letras se estiver
ai: **Inter**

Inter. Senha do
Bob xykzdp1
.....
.....

Bob está conec-
tado. Usuário
Mallory quer a
palavra de 64000
letras Inter. **Senha
do Bob xykzdp1**
.....

Segurança - Metas



- Contexto Atual
- Segurança
- Teoria vs prática
- Problemas práticos
- **Segurança - Metas**

1. CONTEXTO ATUAL E SEGURANÇA

I) SEGURANÇA - Metas

- 1. Privacidade:** sem vazar dados confidenciais
- 2. Autenticação:** sem se passar por outro
- 3. Integridade:** sem alteração
- 4. Não-repúdio:** não ser capaz de negar
5. ...

Parte 2-

Criptografia como funciona



- Terminologia
 - Criptografia Simétrica
 - Aritmética Modular
 - Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

a) TERMINOLOGIA

“Um texto cifrado não pode ser indistinguível de um texto aleatório.”

- Comunicação segura – receptor e remetente
- Mensagens e cifras – cifrar/decifrar

Remetente



Vídeo

Música

Texto

Fotos

...

010110
100100
100011
11...

Cifrar

chave (k ou pk)

10101001
11100110
1...

Decifrar

chave (k ou sk)

010110
100100
100011
11...

Vídeo

Música

Texto

Fotos

...



Receptor

Criptografia simétrica



- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

b) CRIPTOGRAFIA SIMÉTRICA

```
AES/CBC/NoPadding (128)
AES/CBC/PKCS5Padding (128)
AES/CTR/NoPadding (128)
AES/ECB/NoPadding (128)
AES/ECB/PKCS5Padding (128)
DES/CBC/NoPadding (56)
DES/CBC/PKCS5Padding (56)
DES/ECB/NoPadding (56)
DES/ECB/PKCS5Padding (56)
...
```

```
public static void main(String[] args)
```

```
// ... Exemplos NÃO SEGUROS
```

```
byte[] chave[] = { 01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16};
```

```
byte[] iv[] = { 11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26};
```

```
Cipher cifra = Cipher.getInstance("AES/CBC/PKCS5Padding");
```

```
SecretKeySpec skeySpec = new SecretKeySpec(chave, "AES");
```

```
IvParameterSpec ivp = new IvParameterSpec(iv);
```

```
cipher.init(Cipher.ENCRYPT_MODE, skeySpec, ivp);
```

```
byte[] encrypted = cipher.doFinal(bytesASeremCifrados);
```

```
// ...
```

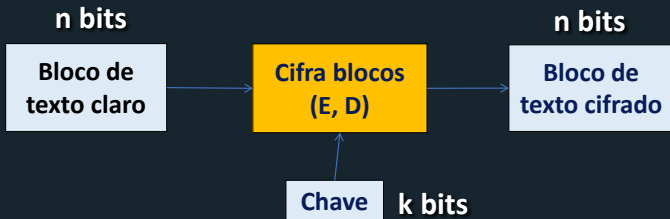
```
Cipher.ENCRYPT_MODE
Cipher.DECRYPT_MODE
```



- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

c) CRIPTOGRAFIA SIMÉTRICA: cifra de blocos



• Exemplos:

- **DES:** n=64 bits k=56 bits
- **3DES:** n=64 bits k=168 bits
- **AES:** n=128 bits k=128, 192, 256 bits

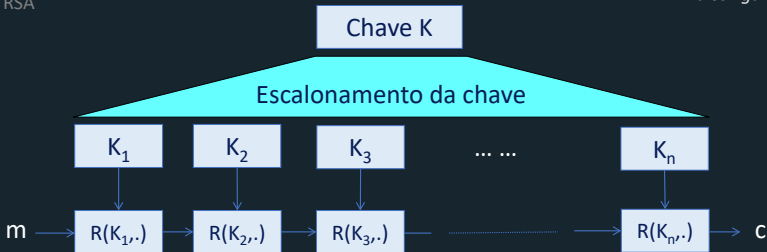


- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

c) CRIPTOGRAFIA SIMÉTRICA: cifra de blocos

“Um texto cifrado não pode ser indistinguível de um texto aleatório.”



- $R(K, \cdot)$ é chamada função da rodada
- **DES 16 rodadas**, 3DES 48 rodadas, AES-128 10 rodadas

Funcionamento do AES

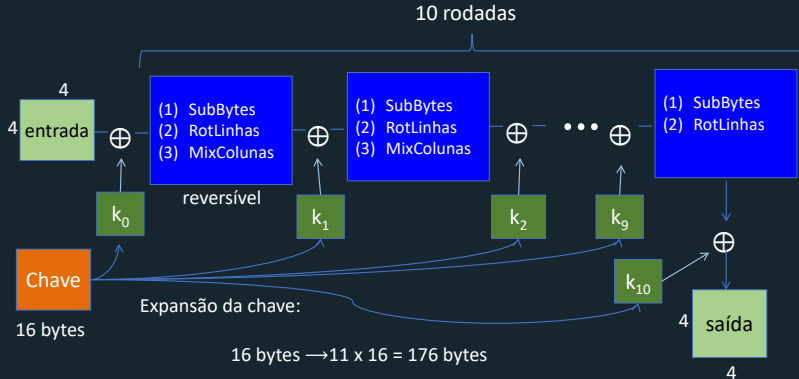


- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

d) CRIPTOGRAFIA SIMÉTRICA: AES

“A segurança deve estar na chave e não no algoritmo.”



Modos de operação e *padding*



- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

e) CRIPTOGRAFIA SIMÉTRICA: Modos de operação

- Um modo de operação define como uma cifra de bloco é aplicada para cifrar uma mensagem.
- Alguns exemplos de modos de operação
 - *Electronic code book mode* (ECB)
 - *Cipher Block Chaining mode* (CBC)
 - *Cipher feedback mode* (CFB)
 - *Output feedback mode* (OFB)
 - *Counter mode* (CTR)

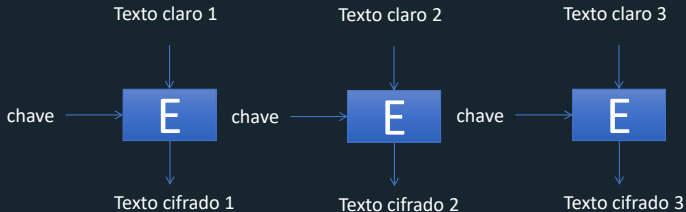


- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

e) CRIPTOGRAFIA SIMÉTRICA: Modos de Operação - ECB

- *Electronic Codebook (ECB)* - O modo de operação mais simples



Texto cifrado 1 = $E(\text{chave}, \text{Texto claro 1})$ Texto claro 1 = $D(\text{chave}, \text{Texto cifrado 1})$



- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

e) **CRIPTOGRAFIA SIMÉTRICA: Modos de Operação - ECB**

- ECB: operação determinística
 - mesmo bloco de texto claro na entrada → mesma saída
- Problemático na prática



- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

e) CRIPTOGRAFIA SIMÉTRICA: Modos de operação

“Não se deve passar qualquer informação a um atacante.”



Texto claro

Modo ECB



Texto cifrado

O que você realmente queria



Vazamento de
Informações

(Fonte: Wikipedia)

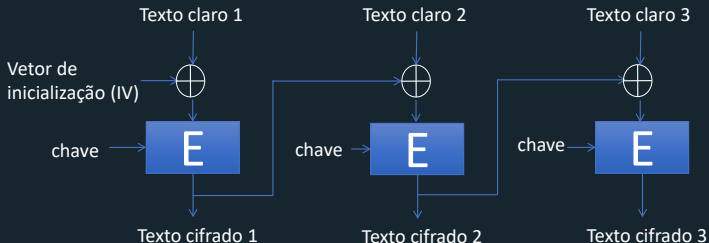


- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

f) CRIPTOGRAFIA SIMÉTRICA: Modos de Operação - CBC

• Cipher Block Chaining (CBC) - Um dos modos mais usados



Texto cifrado 1 = $E(\text{chave}, IV \oplus \text{Texto claro 1}) \Rightarrow \text{Texto claro 1} = D(\text{chave}, \text{Texto cifrado 1}) \oplus IV$

E o i-ésimo bloco?



- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

f) CRIPTOGRAFIA SIMÉTRICA: Modos de Operação - CBC

• Propriedades do CBC

- Devido ao IV aleatório, a mesma mensagem de entrada pode gerar diferentes saídas (ótimo!!)
- Se um bloco do texto claro mudar, então todos os blocos do texto cifrado subsequentes serão afetados
 - Será uma propriedade útil para gerar um código de autenticação de mensagem (*MAC*)
- Cifrar não pode ser paralelizada (ruim!!)

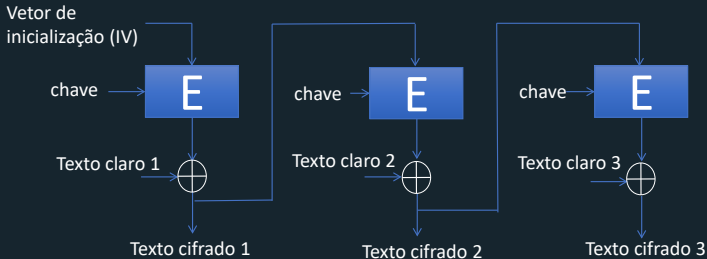


- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

g) CRIPTOGRAFIA SIMÉTRICA: Modos de Operação - CFB

- *Cipher feedback* (CFB): cifra de bloco \rightarrow cifra de fluxo





- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

g) CRIPTOGRAFIA SIMÉTRICA: Modos de Operação - CFB

- **Propriedades do CFB**
 - Somente a operação de cifrar é usada.
 - Não é possível cifrar paralelamente.
 - Mas é possível decifrar paralelamente
 - Por que e como?

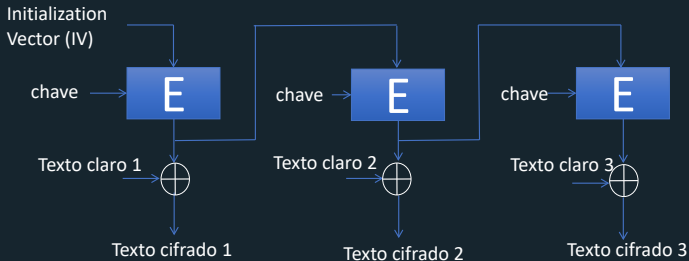


- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

h) CRIPTOGRAFIA SIMÉTRICA: Modos de Operação - OFB

- *Output feedback* (OFB): Essencialmente, uma cifra de fluxo





- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

h) CRIPTOGRAFIA SIMÉTRICA: Modos de Operação - OFB

- **Propriedades do OFB**
 - As operações de cifrar e decifrar são exatamente as mesmas (assim como na cifra de uso único – *one time pad*)

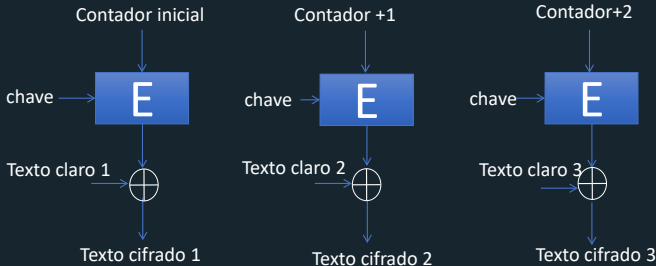


- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

i) CRIPTOGRAFIA SIMÉTRICA: Modos de Operação - CTR

- *Counter* (CTR): Se tornando muito popular (trocar CBC)





- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

i) CRIPTOGRAFIA SIMÉTRICA: Modos de operação - CTR

• Propriedades do CTR

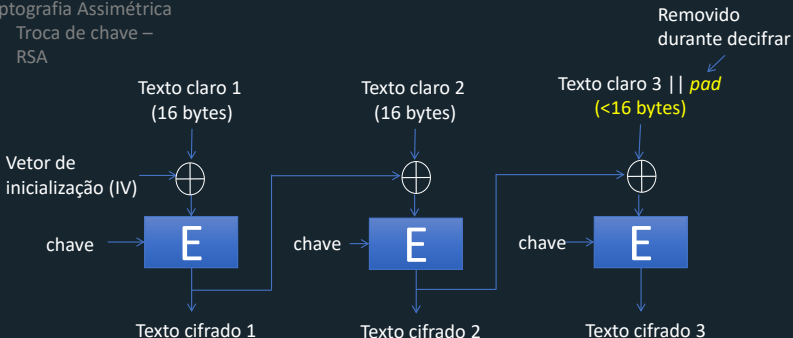
- Assim como o OFB, CTR é essencialmente uma cifra de fluxo
- As operações de cifrar e decifrar são as mesmas.
- É possível paralelizar as duas operações (uma grande vantagem sobre o CBC)



- Terminologia
- **Criptografia Simétrica**
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

j) CRIPTOGRAFIA SIMÉTRICA: Padding



PKCS7: para $n > 0$, n byte *pad* é: $n \ n \ n \ n \ .. \ n$

Pergunta: o que acontece se a mensagem é múltipla do tamanho do bloco?

Criptografia assimétrica



- Terminologia
- Criptografia Simétrica
- **Aritmética Modular**
- Criptografia Assimétrica
 - Troca de chave
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONA

k) ARITMÉTICA MODULAR

- N representa um número positivo inteiro.
- Notação: $Z_N = \{0, 1, 2, \dots, N-1\}$
- Exemplo: Seja $N = 12$
- Aritmética em Z_N funciona como esperado,
 - *e.g.* $x \cdot (y+z) = x \cdot y + x \cdot z$ em Z_N

$$\begin{array}{r} 17 \overline{) 12} \\ \underline{12} \\ 5 \end{array}$$

$9 + 8 = 5$ em Z_{12}

$5 \times 7 = 11$ em Z_{12}

$5 - 7 = 10$ em Z_{12}



- Terminologia
- Criptografia Simétrica
- **Aritmética Modular**
- Criptografia Assimétrica
 - Troca de chave
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

1) Problema do Logaritmo Discreto

- Dado Z_p^* , um gerador g de Z_p^* e um elemento x de Z_p^* , encontrar o logaritmo discreto base g de a em Z_p^*
 - $\text{Dlog}_g a = x \text{ mod } p \rightarrow g^x = a \text{ mod } p$
- Não existe algoritmo polinomial para resolver Dlog para grandes números
- Para números pequenos é fácil, e.g. $\text{Dlog}_3 6 = x \text{ mod } 7$

Em $Z_7 \rightarrow 3^x = 6 \rightarrow 3^3 = 6$

$3^x = 6? \text{ } p = 7; Z_7^* = \{1, 2, 3, 4, 5, 6\}; g = 3; \{3^0, 3^1, 3^2, 3^3, 3^4, 3^5\} = \{1, 3, 2, 6, 4, 5\}$



- Terminologia
- Criptografia Simétrica
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave
 - RSA

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

m) CRIPTOGRAFIA ASSIMÉTRICA: Troca de chaves – Diffie Hellman

Escolha um grande número primo p

Escolha um g que seja um gerador de p

p e g são conhecidos por todos

Escolha um número aleatório a

$A = g^a \pmod{p}$

$A = g^a \pmod{p}$

$p =$

124.325.339.146.889.384.540.494.091.085.456.630.
009.856.882.741.872.806.181.731.279.018.491.820.
800.119.460.022.367.403.769.795.008.250.021.191.
767.583.423.221.479.185.609.066.059.226.301.250.
167.164.084.041.279.837.566.626.881.119.772.675.
984.258.163.062.926.954.046.545.485.368.458.404.
445.166.682.380.071.370.274.810.671.501.916.789.
361.956.272.226.105.723.317.679.562.001.235.501.
455.748.016.154.805.420.913

(309 dígitos - **pequeno**)

$B^a \pmod{p}$

23221
74810

439
828

Alice



- Terminologia
- Criptografia Simétrica
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - **RSA**

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

n) CRIPTORAFIA ASSIMÉTRICA: RSA

Exemplo: estes números são pequenos e não tem segurança

1. Número (2048 bits) \rightarrow ~600 dígitos
2. $p =$
3. 15.766.490.741.353.360.780.957.737.305.079.164.312.124.731.579.065.684.765.007.471.684.718.609.079.347.
615.185.097.793.328.538.026.867.056.470.609.470.632.937.458.431.667.423.756.361.602.670.890.202.687.340.
274.965.847.429.289.285.030.265.671.446.302.371.944.740.069.219.188.845.554.267.036.319.750.563.448.118.
4. 437.989.490.986.516.330.899.872.140.560.137.115.456.861.132.345.688.653.525.709.731.912.407.252.306.681.
946.430.405.288.546.624.004.118.828.698.717.592.226.661.852.180.908.522.610.022.873.108.039.715.606.017.
1 311.308.303.457.499.116.263.663.550.964.008.395.825.118.212.184.564.005.835.724.283.050.585.467.665.432.
5. 807.031.954.002.009.204.352.218.565.598.875.683.478.349.768.844.880.672.908.137.547.142.897.653.200.065.
.771.412.724.723.277.395.478.654.009.813.600.118.795.398.476.206.576.124.437.204.687.762.749.399
6. Mantenha secreta a chave privada $SK = \{23, 5, 11\}$



- Terminologia
- Criptografia Simétrica
- Aritmética Modular
- Criptografia Assimétrica
 - Troca de chave –
 - **RSA**

2. CRIPTOGRAFIA – COMO FUNCIONANÇA

o) CRIPTOGRAFIA ASSIMÉTRICA: RSA exemplo

- Cifrar $\rightarrow C = M^{pk} \bmod N$
- Decifrar $\rightarrow M = C^{sk} \bmod N$

- Dada a mensagem $M = 8$

- Cifrar:

$$C = 8^7 \bmod 55 = 2,097,152 \bmod 55 = 2$$

- Decifrar:

$$M = 2^{23} \bmod 55 = 8,388,608 \bmod 55 = 8$$

Parte 3- Exemplo Prático: Whatsapp

*Baseado em: <https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf>



a) INTRODUÇÃO

- Introdução e Termos
 - Tipo de Chaves
 - Estabelecimento de Sessão
 - Troca de Mensagens
 - Transmissão de Mídias e Anexos
- Aplicativo para comunicação móvel
- 1 bilhão 2016 → 2 bilhões em 2022
- Segurança → Abril 2016 → EE2E (End-to-End Encryption)
 - Proteger mensagens, chamadas e arquivos em comunicação individuais e em grupo de atacantes e do próprio WhatsApp.
- Signal Protocol, desenvolvido pela Open Whisper Systems, é a base para a criptografia *end-to-end*.



- Introdução e Termos
 - Tipo de Chaves
 - Estabelecimento de Sessão
 - Troca de Mensagens
 - Transmissão de Mídias e Anexos
- ECDH (Elliptic Curve Diffie Helman)
 - Protocolo de troca de chaves.
- HKDF (HMAC based Extract-and-Expand Key Derivation Function).
 - Função para derivar Root Key e Chain Key a partir de um `master_secret`.
- Curve25519
 - Curva elíptica: $y^2 = x^3 + 486662x^2 + x \mod 2^{255}-19$
 - Uma das mais rápidas e menos suscetível a geradores de números aleatórios fracos.



- Introdução e Termos
 - **Tipo de Chaves**
 - Estabelecimento de Sessão
 - Troca de Mensagens
 - Transmissão de Mídias e Anexos
-
- Chaves assimétricas usando Curve25519
 - Identity Key Pair
 - Par de chaves gerado no momento da instalação. Assinar as Signed Pre Keys.
 - Signed Pre Key
 - Par de chaves gerado no momento da instalação e assinado pela Identity Key. Usada na criação do `master_secret`.
 - One-Time Pre Keys
 - Uma fila de pares de chaves de uso único. Gerado no momento da instalação e reabastecido conforme necessário. Usado na criação do `master_secret`.



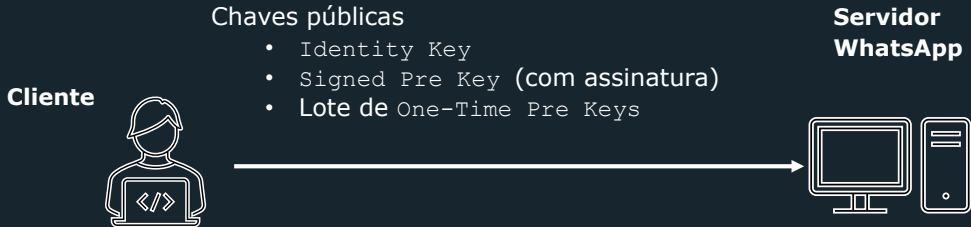
- Introdução e Termos
 - **Tipo de Chaves**
 - Estabelecimento de Sessão
 - Troca de Mensagens
 - Transmissão de Mídias e Anexos
-
- Chaves simétricas para sessões
 - Root Key
 - 32 bytes derivada do `master_secret`. Derivar Chain Keys
 - Chain Key
 - 32 bytes derivada da Root Key. Derivar Message Keys
 - Message Key
 - 80 bytes derivada da Chain Key. 32 bytes AES-256 (criptografia de mensagens), 32 bytes para HMAC-SHA256 (autenticação de mensagens), 16 bytes para o IV (inicialização aleatória). Criptografar e autenticar uma mensagem (uso único).



- Introdução e Termos
- Tipo de Chaves
- **Estabelecimento de Sessão**
- Troca de Mensagens
- Transmissão de Mídias e Anexos

3. EXEMPLO PRÁTICO - WHATSAPP

d) INSTALAÇÃO

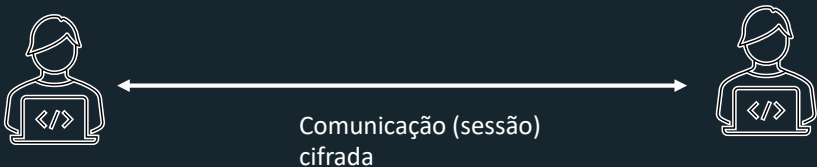


WhatsApp não tem acesso às chaves privadas do usuário. Armazenadas no dispositivo do cliente.

Armazena as chaves públicas associadas com o identificador do usuário



- Introdução e Termos
 - Tipo de Chaves
 - **Estabelecimento de Sessão**
 - Troca de Mensagens
 - Transmissão de Mídias e Anexos
-
- A fim de comunicar com outro usuário do whatsapp, o cliente (Iniciador) precisa estabelecer uma sessão cifrada.
 - Uma vez que a sessão é estabelecida, ela é mantida até que ocorra algum evento externo, como reinstalação do aplicativo ou troca de equipamento.





- Introdução e Termos
- Tipo de Chaves
- Estabelecimento de Sessão
- Troca de Mensagens
- Transmissão de Mídias e Anexos

1. O iniciador requisita as chaves **públicas**: Identity Key, Signed Pre Key, e One-Time Pre Key do destinatário ao servidor.
2. Servidor retorna as chaves requisitadas. A One-Time Pre Key é usada somente uma vez, então é removida do servidor depois de requisitada.

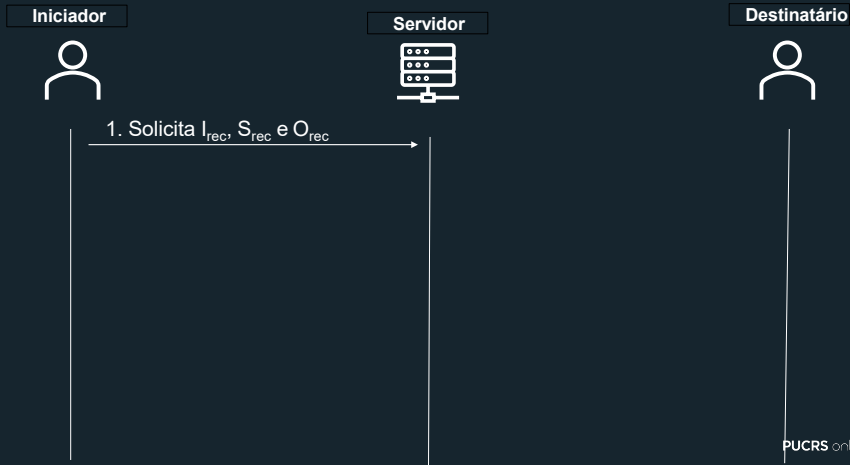




- Introdução e Termos
- Tipo de Chaves
- **Estabelecimento de Sessão**
- Troca de Mensagens
- Transmissão de Mídias e Anexos

3. EXEMPLO PRÁTICO - WHATSAPP

d) INICIADOR

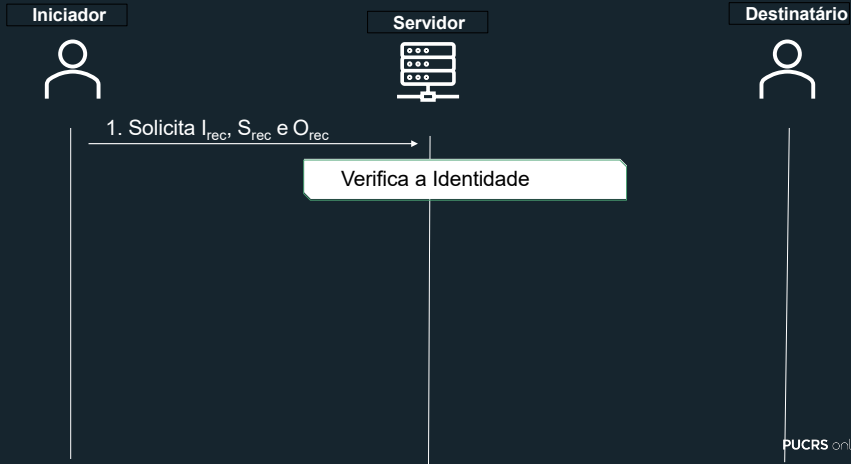




- Introdução e Termos
- Tipo de Chaves
- **Estabelecimento de Sessão**
- Troca de Mensagens
- Transmissão de Mídias e Anexos

3. EXEMPLO PRÁTICO - WHATSAPP

d) INICIADOR

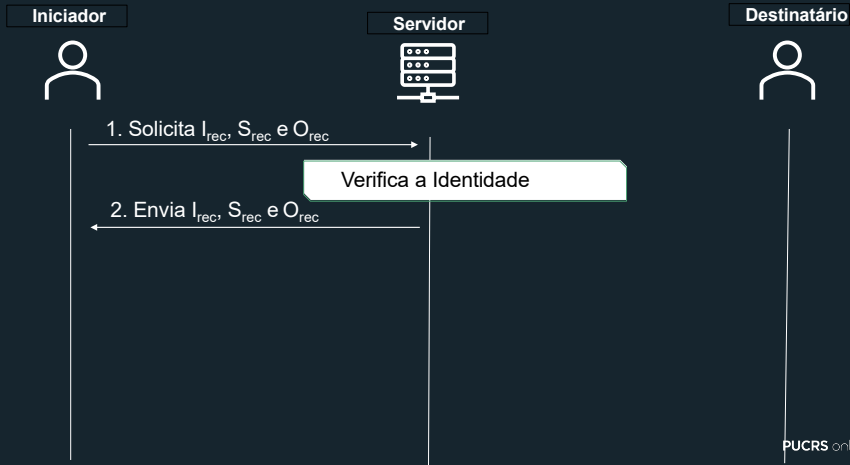




- Introdução e Termos
- Tipo de Chaves
- **Estabelecimento de Sessão**
- Troca de Mensagens
- Transmissão de Mídias e Anexos

3. EXEMPLO PRÁTICO - WHATSAPP

d) INICIADOR





- Introdução e Termos
- Tipo de Chaves
- **Estabelecimento de Sessão**
- Troca de Mensagens
- Transmissão de Mídias e Anexos

3. O iniciador salva as chaves do destinatário

- Identity Key como $I_{\text{recipient}}$
- Signed Pre Key como $S_{\text{recipient}}$
- One-Time Pre Key como $O_{\text{recipient}}$

4. O iniciador gera um par de chaves efêmeras na Curve25519

- $E_{\text{initiator}}$

5. O iniciador carrega sua Identity Key como $I_{\text{initiator}}$



- Introdução e Termos
- Tipo de Chaves
- **Estabelecimento de Sessão**
- Troca de Mensagens
- Transmissão de Mídias e Anexos

6. O iniciador calcula um `master_secret`

- `master_secret =`

$$\text{ECDH}(I_{\text{initiator}}, S_{\text{recipient}}) || \text{ECDH}(E_{\text{initiator}}, I_{\text{recipient}}) || \\ \text{ECDH}(E_{\text{initiator}}, S_{\text{recipient}}) || \text{ECDH}(E_{\text{initiator}}, O_{\text{recipient}})$$

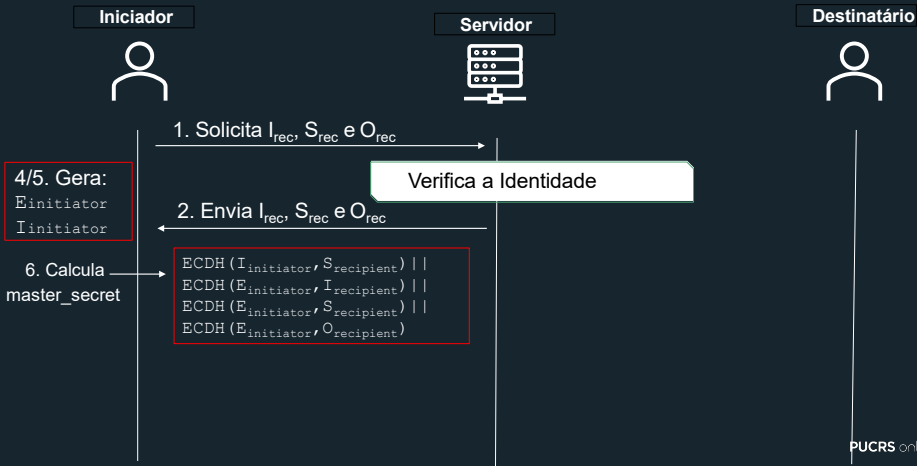
7. O iniciador usa HKDF para criar uma Root Key e Chain Keys a partir do `master_secret`.



- Introdução e Termos
- Tipo de Chaves
- **Estabelecimento de Sessão**
- Troca de Mensagens
- Transmissão de Mídias e Anexos

3. EXEMPLO PRÁTICO - WHATSAPP

d) INICIADOR





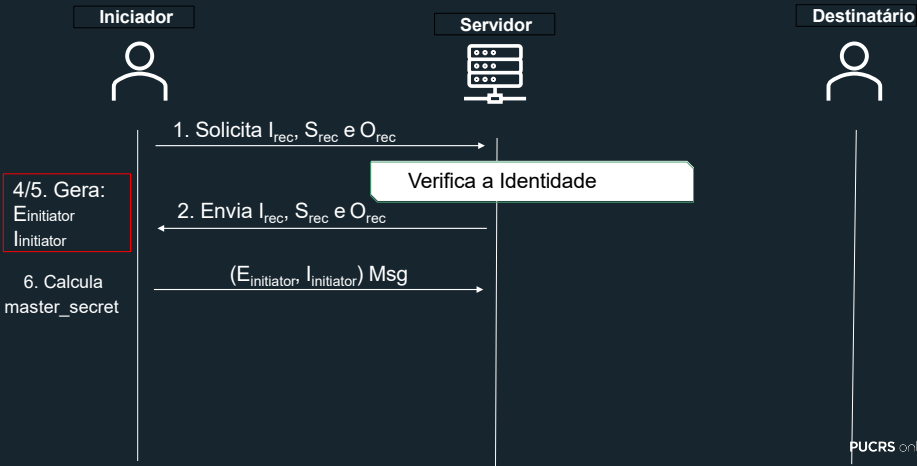
- Introdução e Termos
 - Tipo de Chaves
 - Estabelecimento de Sessão
 - Troca de Mensagens
 - Transmissão de Mídias e Anexos
-
- Depois de contruir a sessão, o iniciador pode enviar mensagens para o destinatário imediatamente, mesmo que ele esteja *offline*.
 - O iniciador envia todas informações necessárias (inclui suas chaves **públicas** $E_{initiator}$ e $I_{initiator}$) para o destinatário construir uma sessão correspondente
 - Estas informações vão no cabeçalho de todas as mensagens enviadas.



- Introdução e Termos
- Tipo de Chaves
- **Estabelecimento de Sessão**
- Troca de Mensagens
- Transmissão de Mídias e Anexos

3. EXEMPLO PRÁTICO - WHATSAPP

d) INICIADOR





- Introdução e Termos
 - Tipo de Chaves
 - **Estabelecimento de Sessão**
 - Troca de Mensagens
 - Transmissão de Mídias e Anexos
-
- Quando o destinatário recebe uma mensagem que inclui informações de configuração de sessão:
 1. Destinatário calcula o `master_secret` correspondente usando suas chaves privadas e as chaves públicas do iniciador.
 2. Destinatário apaga a `One-Time Pre Key` usada pelo iniciador.
 3. Destinatário usa HKDF para derivar uma `Root Key` correspondente e `Chain Keys` a partir do `master_secret`.



f) TROCA DE MENSAGENS

- Introdução e Termos
 - Tipo de Chaves
 - Estabelecimento de Sessão
 - **Troca de Mensagens**
 - Transmissão de Mídias e Anexos
-
- Cada mensagem é criptografada com uma Message Key única.
 - Para criptografia, AES256 no modo CBC.
 - Para autenticação, HMAC-SHA256 é usado.
 - Message Keys são efêmeras, não são usadas novamente.
 - A Message Key é derivada da Chain Key do remetente que é atualizada a cada mensagem enviada.
 - ECDH é realizado a cada mensagem recebida para criar uma nova Chain Key.



g) DOUBLE-RATCHET

- Introdução e Termos
 - Tipo de Chaves
 - Estabelecimento de Sessão
 - **Troca de Mensagens**
 - Transmissão de Mídias e Anexos
-
- Mecanismo de “ratcheting” sobre a Chain Key para obter novas Chain Keys;
 - Quando um usuário envia uma mensagem:
 - Calcula *hash* da Chain Key para obter a próxima Chain Key.
 - Quando o usuário recebe uma mensagem:
 - Uma nova Chain Key e Root Key são calculadas.



- Introdução e Termos
- Tipo de Chaves
- Estabelecimento de Sessão
- Troca de Mensagens
- Transmissão de Mídias e Anexos

h) DOUBLE-RATCHET: cálculo da *message key* de uma *chain key*

- A primeira fase de “ratchet” é conhecida como o “Hash Ratchet”, onde é feito o hash da Chain Key com HMAC-SHA256 para obter uma nova Chain Key.
- Nesta fase, a Chain Key é usada para derivar a Message Key de uso único:
 - $\text{Message Key} = \text{HMAC-SHA256}(\text{Chain Key}, 0x01)$
- Posteriormente, a Chain Key é atualizada assim:
 - $\text{Chain Key} = \text{HMAC-SHA256}(\text{Chain Key}, 0x02)$
- Significa que uma Message Key armazenada não pode ser usada para derivar valores atuais ou passados da Chain Key.



- Introdução e Termos
- Tipo de Chaves
- Estabelecimento de Sessão
- Troca de Mensagens
- Transmissão de Mídias e Anexos

i) DOUBLE-RATCHET: cálculo da *chain key* de uma *root key*

- Cada vez que uma mensagem é transmitida, uma chave pública efêmera é enviada junto com ela.
- A segunda fase, conhecida como “DH Ratchet”, ocorre quando um usuário recebe uma mensagem.
- Uma nova Chain Key e Root Key são calculadas da seguinte forma:
 - $\text{ephemeral_secret} = \text{ECDH}(E_{\text{initiator}}, E_{\text{recipient}})$.
 - $\text{Chain Key, Root Key} = \text{HKDF}(\text{Root Key}, \text{ephemeral_secret})$.
- A nova Chain Key é usada para criar a próxima Message Key de uso único.

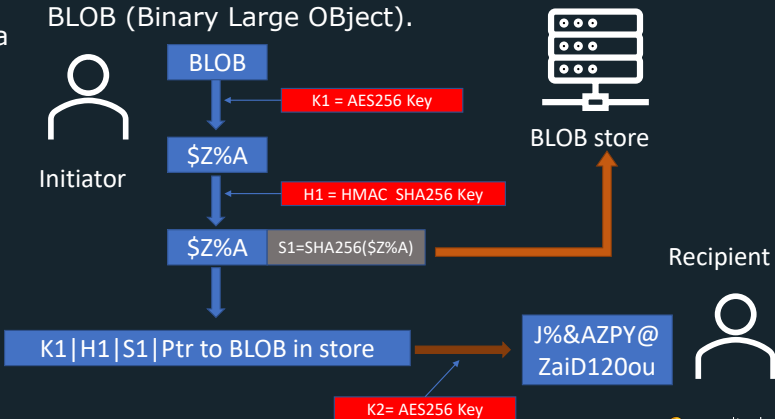


- Introdução e Termos
- Tipo de Chaves
- Estabelecimento de Sessão
- Troca de Mensagens
- **Transmissão de Mídias e Anexos**

3. EXEMPLO PRÁTICO - WHATSAPP

j) **ENVIO**

1. O remetente gera uma chave AES efêmera de 32 bytes, e uma chave HMAC-SHA256 efêmera de 32 bytes.



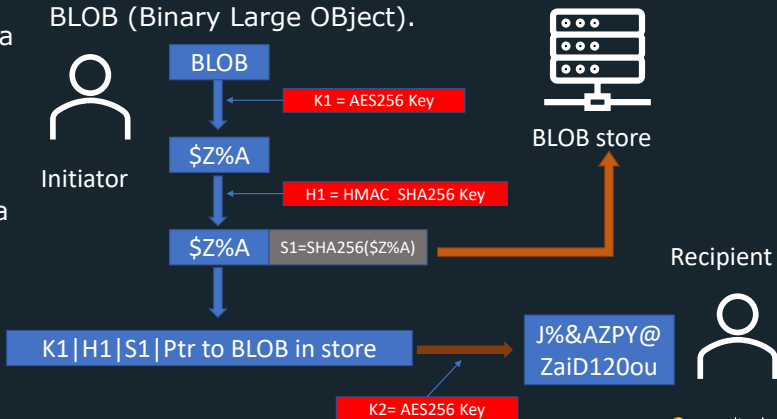


- Introdução e Termos
- Tipo de Chaves
- Estabelecimento de Sessão
- Troca de Mensagens
- Transmissão de Mídias e Anexos

3. EXEMPLO PRÁTICO - WHATSAPP

j) ENVIO

2. O remetente cifra o anexo com a chave AES256 no modo CBC com um IV aleatório, em seguida, acrescenta um MAC do texto cifrado usando HMAC-SHA256



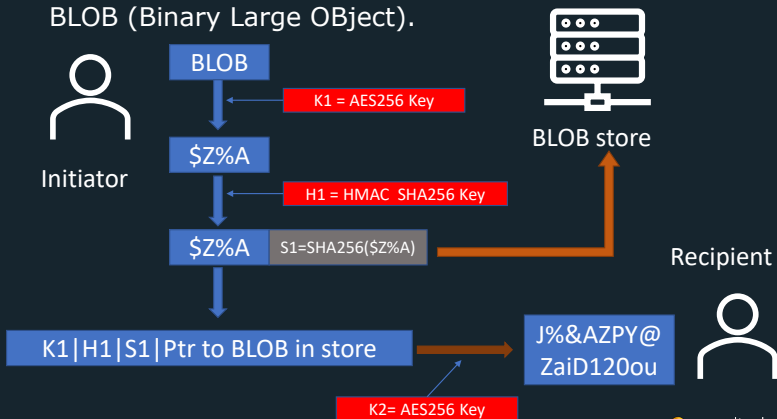


- Introdução e Termos
- Tipo de Chaves
- Estabelecimento de Sessão
- Troca de Mensagens
- **Transmissão de Mídias e Anexos**

3. EXEMPLO PRÁTICO - WHATSAPP

j) **ENVIO**

3. O remetente envia o anexo cifrado com K1 a um servidor de BLOB.





- Introdução e Termos
- Tipo de Chaves
- Estabelecimento de Sessão
- Troca de Mensagens
- Transmissão de Mídias e Anexos

3. EXEMPLO PRÁTICO - WHATSAPP

j) ENVIO

4. O remetente transmite uma mensagem cifrada normal ao destinatário que contém a chave de criptografia, a chave HMAC, um hash SHA256 do BLOB criptografado, e um ponteiro para o BLOB no servidor de BLOB.

BLOB (Binary Large Object).



Initiator

BLOB

K1 = AES256 Key

\$Z%A

H1 = HMAC SHA256 Key

\$Z%A

S1=SHA256(\$Z%A)

K1|H1|S1|Ptr to BLOB in store

K2= AES256 Key

Whatsapp store



BLOB store

Recipient

J%&AZPY@
ZaiD120ou

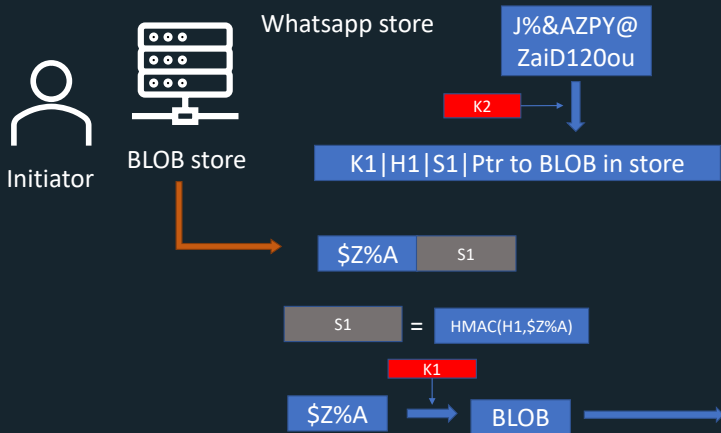




- Introdução e Termos
- Tipo de Chaves
- Estabelecimento de Sessão
- Troca de Mensagens
- **Transmissão de Mídias e Anexos**

3. EXEMPLO PRÁTICO - WHATSAPP

k) RECEBIMENTO



5. O destinatário decifra a mensagem, recupera o BLOB criptografado a partir do servidor de BLOB, verifica o *hash*, verifica o MAC, e decifra o BLOB com K1.



- Bom nível de segurança para as mensagens trocadas pelos usuários.
- Mais → transmissão para grupos.
- A biblioteca do Signal Protocol é *open source*
 - <https://github.com/signalapp/libsignal>

Conclusão

*Baseado em: <https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf>



CONTEXTO ATUAL
E
SEGURANÇA



CRİPTOGRAFIA:
COMO FUNCIONA

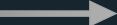


EXEMPLO PRÁTICO:
WHATSAPP

TEORIA



EXEMPLOS
PRÁTICOS

- 
1. INTRODUÇÃO A SEGURANÇA DE SOFTWARE
 2. MÉTODOS DE CRIPTOGRAFIA
 3. PROTOCOLOS DE COMUNICAÇÃO SEGURA
 4. SEGURANÇA NO DESENVOLVIMENTO DE SOFTWARE
 5. PROBLEMAS COMUNS DE SEGURANÇA INDICADOS PELA OWASP
 6. AUTENTICAÇÃO E AUTORIZAÇÃO

PUCRS online  **UOL** edtech.

PUCRS online  **uol** edtech.