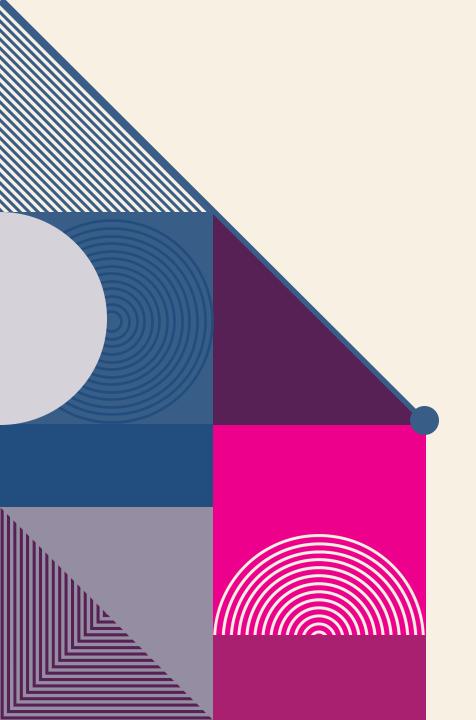
GERADOR/ VERIFICADOR DE ASSINATURAS

Marcelo Marques Rodrigues (221018960) Arthur Delpino Barbabella (221002094)



APRESENTAÇÃO

- 1. RSA
- 2. OAEP
- 3. Assinatura Digital
- 4. Verificação
- 5. Implementação em Python



RSA

- Algoritmo de criptografia assimétrica robusto
- Cifragem e assinatura digital
- Utiliza chave pública (n,e) e privada (n,d)
- Decifração: M = C^d mod n
- Cifração: C = M^e mod n
- Em que:
 - o n é o módulo público (p x q)
 - o M é a mensagem original



GERAÇÃO DAS CHAVES

- Dois números primos grandes, **p** e **q**
- Calcula-se $\mathbf{n} = \mathbf{p} \times \mathbf{q}$, que é o módulo público
- Calcula-se a função totiente: $\phi(n) = (p 1) \times (q 1)$
- Escolhe-se um expoente público e pequeno e coprimo de φ(n) (comum: 65537)
- Calcula-se o inverso modular de e mod φ(n), obtendo o expoente privado d
- A chave pública é (n, e) e a privada é (n, d)



TESTE DE PRIMALIDADE

- Algoritmo de Miller-Rabin (teste probabilístico)
- Serve pra verificar se p e q são provavelmente primos
- Sendo submetido um número **n** ao teste:
 - Decompõe **n-1** em **2^s** x d
 - Repete **k vezes**:
 - Escolhe um α aleatório entre 2 e n-2
 - Calcula $\mathbf{x} = \mathbf{\alpha}^{\mathbf{d}} \mathbf{mod} \mathbf{n}$
 - Se x = 1 ou x = n 1, α não é testemunha de composição
 - Caso contrário, executa **s-1** vezes:
 - Calcula $x = x^2 \mod n$
 - Se x = n 1, rodada bem-sucedida e testa outro α Se nunca ocorrer, então α é testemunha de composição



OAEP

- Esquema de **preenchimento** (padding)
- Acrescenta aleatoriedade à cifragem RSA
- Monta um **DB** (Data Block)
 IHash || Padding || 0x01 || M
- Monta um EM (Encoded Message):
 0x00 || maskedSeed || maskedDB
- Em que:

```
maskedDB = DB (XOR) MGF1(seed)
maskedSeed = seed (XOR) MGF1(maskedDB)
Padding = len_chave - len_M - 2 * len_lHash - 2
```



RSA COM OAEP

Aplica cifração com chave pública (n, e) no EM:

 $C = EM^e \mod n$

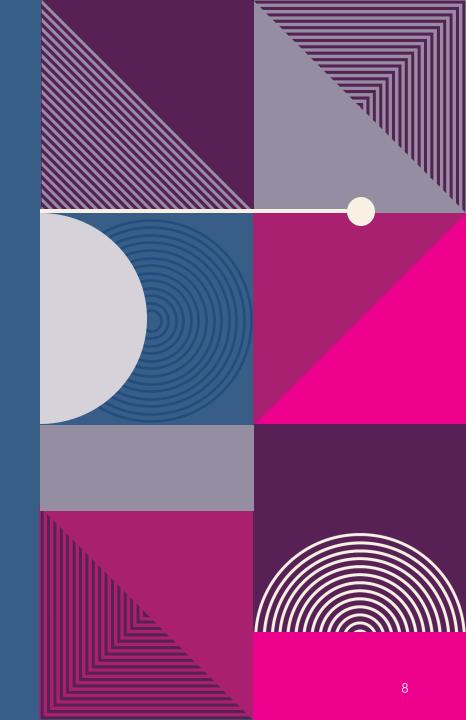
• Decifra com a chave privada (n, d):

 $EM = C^d \mod n$

- OAEP contorna aspecto determinístico do RSA
- Diferentes **seeds**, diferentes **EMs**, diferentes **Cs**

DECODIFICAÇÃO OAEP

- Extrair maskedSeed e maskedDB de EM:
 0x00 || maskedSeed || maskedDB
 - Primeiro byte é 0x00
 - \circ MaskedSeed \rightarrow próximos hash_length bytes
 - o MaskedDB → bytes restantes
- Recuperar o seed e o DB:
 - o seed = maskedSeed (XOR) MGF1(maskedDB)
 - o **DB** = maskedDB (XOR) MGF1(seed)

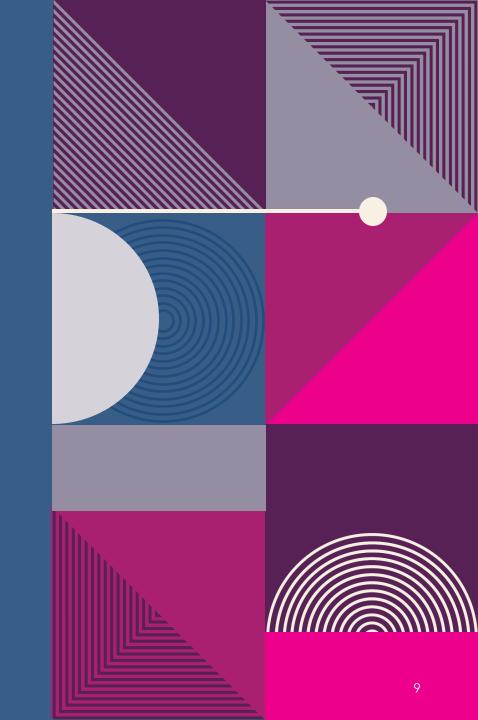


DECODIFICAÇÃO OAEP

• Extrair a mensagem de DB:

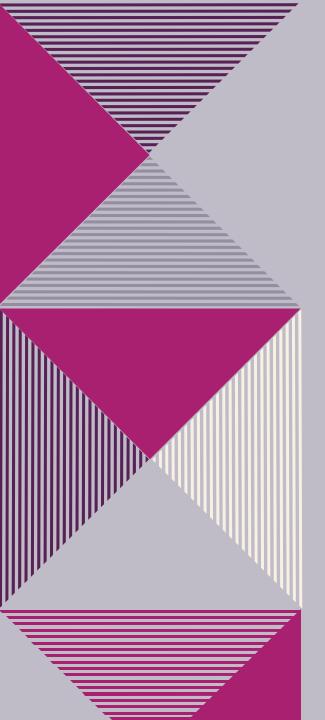
| IHash | | Padding | | 0x01 | M

- o Primeiros 32 bytes são do **IHash** (SHA-3)
- o Ignora os bits 0 do Padding
- o Encontra o **delimitador** de byte **0x01**
- o Tudo após o delimitador é a mensagem M





ASSINATURA E VERIFICAÇÃO



NAVIGATING Q&A SESSIONS

- Know your material in advance
- Anticipate common questions
- Rehearse your responses

Maintaining composure during the Q&A session is essential for projecting confidence and authority. Consider the following tips for staying composed:

- Stay calm
- Actively listen
- Pause and reflect
- Maintain eye contact