Trabalho 3
Gerador\Verificador de Assinaturas

Lucas Resende Silveira Reis, 18/0144421

[1]Dep. Ciência da Computação – Universidade de Brasília (UnB)
CiC 0201 - SEGURANÇA COMPUTACIONAL

toy.lcdv@hotmail.com

Abstract. *This corresponds to project 3, on encryption, decipherment with RSA.*

Resumo. *Este corresponde ao trabalho 3, sobre cifração, decifração com a RSA.*

1. Introdução

O RSA (Rivest–Shamir–Adleman) é um dos algoritmos mais amplamente utilizados em criptografia assimétrica. Desenvolvido por Ron Rivest, Adi Shamir e Leonard Adleman em 1977, o RSA é fundamental para a segurança da comunicação digital e transações online. Baseado na dificuldade computacional de fatorar grandes números primos, o RSA utiliza um par de chaves, pública e privada, para cifrar e decifrar informações. A segurança do algoritmo repousa na complexidade de se determinar os fatores primos de um grande número, tornando o RSA uma escolha robusta para a implementação de assinaturas digitais, troca segura de chaves e autenticação em sistemas de segurança informática.

1.1. Funcionalidade

O algoritmo RSA, concebido por Ron Rivest, Adi Shamir e Leonard Adleman em 1977, é um dos pilares da criptografia assimétrica. Essa técnica utiliza um par de chaves, uma pública, conhecida por todos, e outra privada, mantida em segredo. A geração das chaves envolve a escolha aleatória de dois números primos grandes, p e q, e o cálculo de $n = pq$ e $\phi(n) = (p-1)(q-1)$. A partir desses valores, são determinados e e d de maneira que $ed \equiv 1 \pmod{\phi(n)}$.

A criptografia RSA é amplamente empregada na internet, sendo aplicada em diversos contextos, como e-mails e transações online. O processo de encriptação envolve elevar a mensagem à potência de e módulo n, enquanto a decriptação utiliza a potência de d módulo n. A segurança do RSA baseia-se na dificuldade computacional de fatorar números primos grandes, garantindo a confidencialidade e autenticidade das comunicações digitais.

A assinatura em RSA é um processo essencial para garantir a autenticidade e integridade das mensagens. Para assinar uma mensagem, utiliza-se a biblioteca de hash, como o hashlib, para criar um resumo criptográfico da mensagem. Em seguida, esse resumo é cifrado utilizando a chave privada do remetente e a função de encriptação RSA.

Na fase de verificação, o destinatário realiza a decriptação da assinatura utilizando a chave pública do remetente. O resultado é comparado com o resumo criptográfico

original, obtido a partir da mensagem recebida. Se as duas informações coincidirem, a assinatura é considerada válida, atestando a origem da mensagem e sua integridade. Esse processo confere confiança nas transmissões digitais, assegurando que as mensagens não foram adulteradas e foram realmente enviadas pelo remetente autenticado.

2. Resultados

Se o código é executado com os imputs presentes no README estes serão os resultados:

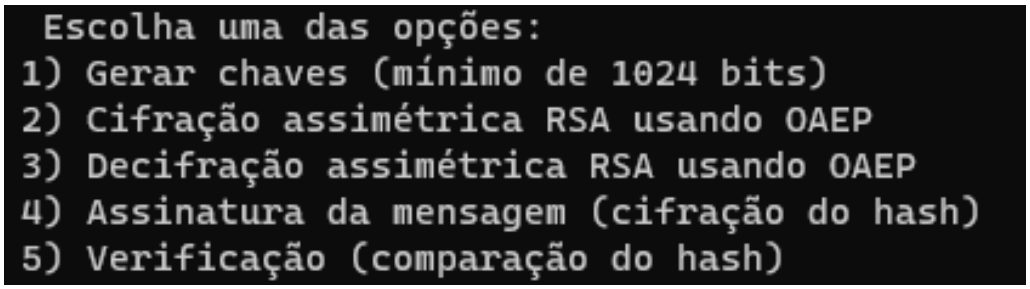

Figura 1. Menu de escolhas

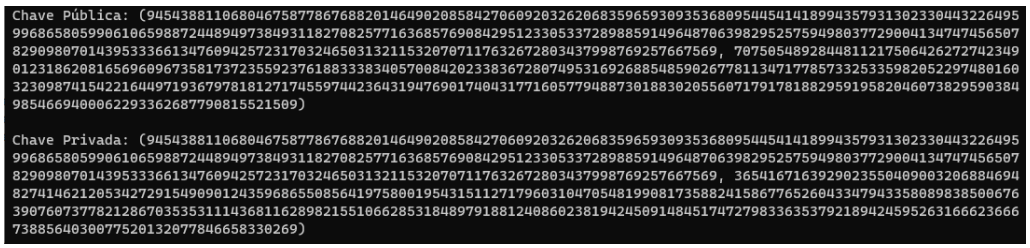

Figura 2. Resultado 1

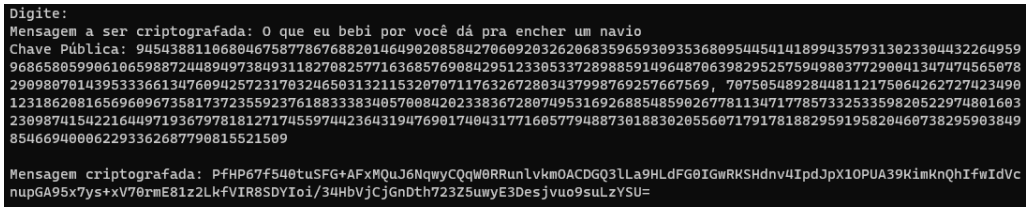

Figura 3. Resultado 2

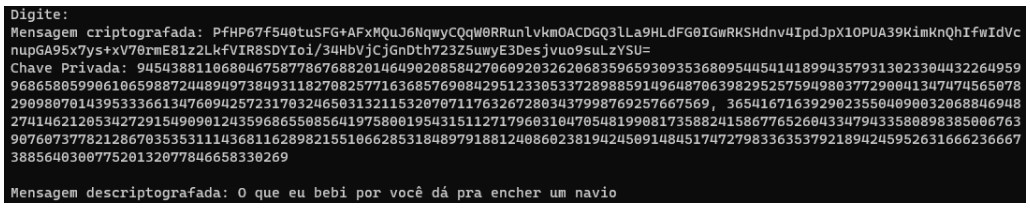

Figura 4. Resultado 3

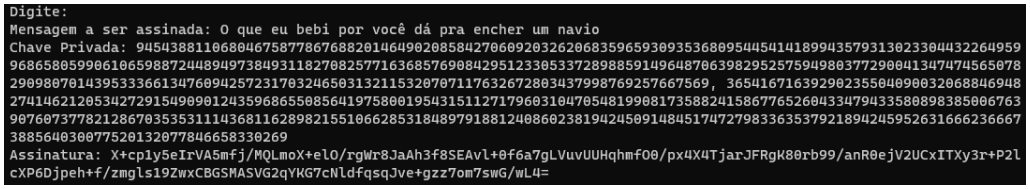

Figura 5. Resultado 4

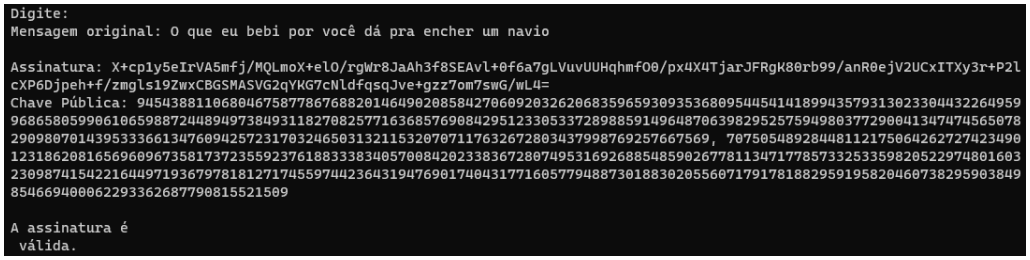

Figura 6. Resultado 5

3. Conclusão

Em conclusão, o algoritmo RSA, combinando técnicas de criptografia assimétrica e assinatura digital, desempenha um papel crucial na segurança da comunicação digital. Ao utilizar pares de chaves pública e privada, o RSA possibilita a transmissão segura de informações, garantindo autenticidade e integridade. A assinatura digital, baseada em hash e encriptação RSA, oferece uma robusta verificação de autenticidade das mensagens. Dessa forma, o RSA se torna uma peça fundamental na arquitetura de segurança online, protegendo transações, comunicações e dados sensíveis contra ameaças cibernéticas.

Referências

[1] Criptografia - Criptografia RSA - Fábrica de Noobs
https://www.youtube.com/watch?v=GAR1Ur_2IGk

[2] RSA (sistema criptográfico)
https://pt.wikipedia.org/wiki/RSA_(sistema_criptogr%C3%A1fico)

[3] Optimal asymmetric encryption padding
https://en.wikipedia.org/wiki/Optimal_asymmetric_encryption_padding

[4] Teste de primalidade de Miller-Rabin https://pt.wikipedia.org/wiki/Teste_de_primalidade_de_Miller-Rabin

[5] https://docs.python.org/3/library/hashlib.html
https://docs.python.org/3/library/hashlib.html