# Trabalho de sistemas seguros

## Chaves Assimétricas

Na primeira parte do trabalho, geramos as chaves utilizando o algoritmo RSA

Salvamos a chave privada em um arquivo chamado private\_rsa:

b'-----BEGIN RSA PRIVATE KEY-----\nMIICXAIBAAKBgQDTyJMmniMTkxhGTVGo8qUFnBmJWTIpKRZ06qhpOzPdjjhSq8bC\npgvfc+vmlnHQnPDt1WBWBQiZkAq2unlBm27sCkY3xsor8xCUh32J2NLz3u0Z0b5V\np8bH5+pgOP0Ib6GrU9VteJUZaFQntiFmV/GCzpwj+TRUWo3WneGAVcoLbQIDAQAB\nAoGAHF8xxXnoeZIabaqkjrVpAaoiqqmSJHz+Zu3b33MgtOT9ayfbkg6UJxycxRCx\nETFBPp94DzYP66CmKsoq/ezQjFyjBnpcOHoYFBkPmwrZe7VLoSKh7tmzyP3IYgmJ\nYwbMVvp/KVsXdkUTjMRVVoyFrBSdVOp636ODPHb8xA4K6IkCQQDZ+2Tlfa8eZA7s\nSGb42e1n7gLGpIDDnuXMOYq2/qFuwQ1nHsiv6CpHjK85o9lZD1quoa4lqEUe9WJo\nL4pl+m61AkEA+Lhs44JNsArkuu7N2fkGCK6OwE8CTuRa3TOXTw6Tb1E3kB49NQ+n\nDj1tOFo29jPVJb811lo3vJClxZcpyuXk2QJARaG3tlNyyEuANNxgAGhlHQj73Aaa\nl2K/14AeHpyyZtXGeUVtKDwKnXaEVu5PNNLhzd0Z6y+YfO439D4TgMYB7QJAFTng\nCRaAvMphNNbWeIi5DBqhkiXsA4GXg5M4kYWj7XcX79O004dvcVaqsIl6cU6BHUSR\nOqbeZNhPXWqGJYuzGQJBAJoAjBQMico6Gc6I2z4r2oeER4Lcue7Y2mk0bubXsXAV\ni+mg+xZcAtu4kLvKsbEJjojxYQTYKie4qOXfJmaXkBk=\n-----END RSA PRIVATE KEY-----'

Salvamos a chave pública em um arquivo chamado public\_rsa:

b'-----BEGIN PUBLIC KEY-----\nMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDTyJMmniMTkxhGTVGo8qUFnBmJ\nWTIpKRZ06qhpOzPdjjhSq8bCpgvfc+vmlnHQnPDt1WBWBQiZkAq2unlBm27sCkY3\nxsor8xCUh32J2NLz3u0Z0b5Vp8bH5+pgOP0Ib6GrU9VteJUZaFQntiFmV/GCzpwj\n+TRUWo3WneGAVcoLbQIDAQAB\n-----END PUBLIC KEY-----'

Expoente chave privada:

19923392860567262758559720792637907717416711631320753714535344566076888106015215977716143372292955509117389782152766994170733775673302088663443376304174877378886246782332626405335511324926148449978164415141285892450405129708262210615902386059834933463188037909954844899488816597739298151268208730617921857673

Expoente chave pública:

65537

Módulo:

148719427990887747305757956842405599029168146038539620853612332563795212369819096162371694887663478481850436988860262592553020240473954154131505857552516774781190502053095479614791863028720825460010463410562148262478178826030097400037211095862530437584456000921697068752355141884375603168041211241847648291693

Módulo de primeiro fator (chave privada):

11416644138952714317177797483492061644075299666278602617058754552691090556666462448381151564166151396044883489347771809647201895827350411553718169633451701

Módulo de segundo fator (chave pública):

13026544944452500147974569662921529480542862343443060137310557852509757686979523178708878977657363530135655541851203182976004152900571789560074604095333593

Para a resolução da segunda questão fizemos uma API que recebe um arquivo binário criptografa, salva num arquivo e retorna o id salvo, para que futuramente ele possa ser de criptografado, para comodidade, fizemos a API retornar o conteúdo criptografado, porém em um cenário real isso não aconteceria.

{

"encryptedContent": "b'3\\x82\\x99H\\xd9\\xfd\\xbf\\x020c+!\\xfd\\xf3,\_\\xf7\\r\\x18\\xccr\\xf7D8{\\xd5m2\\xe7\\xa0{\\x06'",

"encryptedName": "1217a2c9-aa4f-459b-a837-270aee9cfb77"

}

E para a de criptografia expusemos uma API que recebe um UUID referente ao encryptedName da API anterior por parâmetro de caminho e retorna o conteúdo descriptografado

CONTEUDO SUPER SECRETO