Member-only story

How to use RSA Asymmetric Encryption and Decryption in Java.

Manoj Ahirwar · Follow

Published in AWS in Plain English · 2 min read · Jan 1

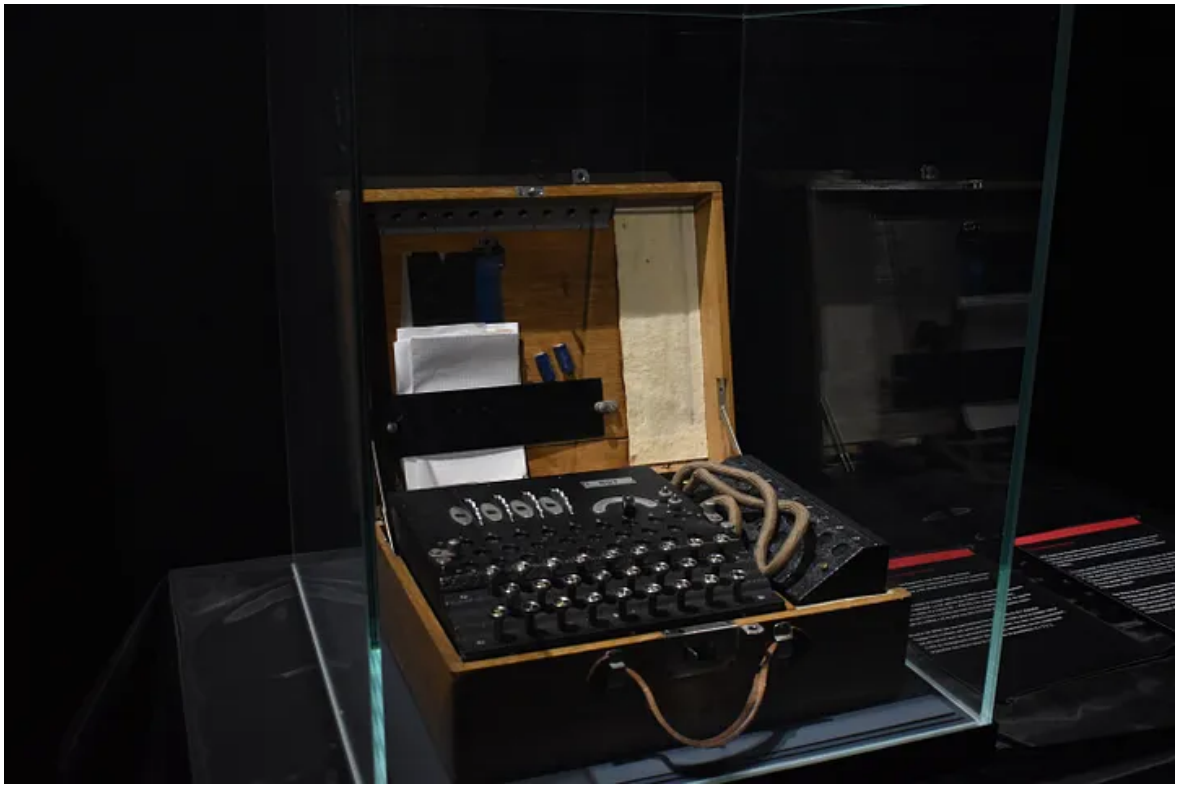

Photo by Mauro Sbicego on Unsplash

RSA (Rivest–Shamir–Adleman) is a popular asymmetric encryption algorithm that is widely used to secure sensitive data, such as passwords and

financial transactions.

Asymmetric encryption uses a pair of keys — a public key and a private key to encrypt and decrypt data.

Here's how to use RSA asymmetric encryption and decryption using Java.

Generate a public/private key pair

The first step in using RSA encryption is to generate a public/private key pair. In Java, you can use the `KeyPairGenerator` class to generate a key pair. Here's an example of how to do this:

```
KeyPairGenerator keyGen = KeyPairGenerator.getInstance("RSA");
keyGen.initialize(2048); // initialize with key size 2048
KeyPair keyPair = keyGen.generateKeyPair();
PrivateKey privateKey = keyPair.getPrivate();
PublicKey publicKey = keyPair.getPublic();
```

Encrypt the data

To encrypt data using RSA, you can use the `Cipher` class. First, you'll need to initialize the cipher with the public key and the encryption mode (e.g., `Cipher.ENCRYPT_MODE`). Then, you can use the `doFinal` method to encrypt the data. Here's an example of how to do this:

```
// data to be encrypted
String data = "This is the data to be encrypted";

// encrypt the data
Cipher cipher = Cipher.getInstance("RSA");
cipher.init(Cipher.ENCRYPT_MODE, publicKey);
byte[] encryptedData = cipher.doFinal(data.getBytes());
```

Decrypt the data

To decrypt the data, you'll need to use the private key and the decryption mode (e.g., `Cipher.DECRYPT_MODE`). Then, you can use the `doFinal` method to decrypt the data. Here's an example of how to do this:

```
// decrypt the data
cipher.init(Cipher.DECRYPT_MODE, privateKey);
byte[] decryptedData = cipher.doFinal(encryptedData);

// convert the decrypted data back to a string
String decryptedString = new String(decryptedData);
System.out.println("Decrypted data: " + decryptedString);
```

. . .

That's it! With these few lines of code, you can use RSA asymmetric encryption and decryption in your Java applications. I hope this helps.

More content at PlainEnglish.io.

Sign up for our free weekly newsletter. Follow us on Twitter, LinkedIn, YouTube, and Discord.

Looking to scale awareness and adoption for your tech startup? Check out Circuit.

Java Programming Encryption Code API