

CENG 471- TERM PROJECT

STAGE 3



May 19, 2019

Hasan yenıada

220201024

**Table of Contents**

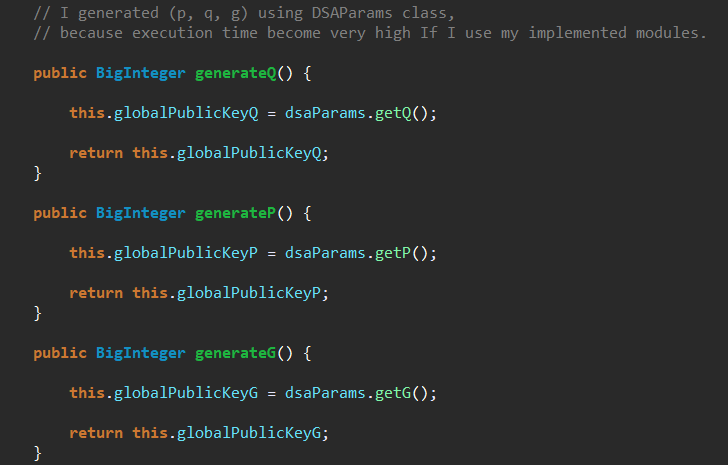
1. **Introduction3**
2. **First Step – Shared Global Public Key Values Generation3**
3. **Second Step – Public/Private Key Generation of Users4**
4. **Third Step – DSA Digital Signature Creation by Sender Alice** **5**
5. **Fourth Step – DSA Signature Verification by Receiver Bob** **6**
6. **Comments of Requirements** **7**
7. **References** **7**
8. Introduction

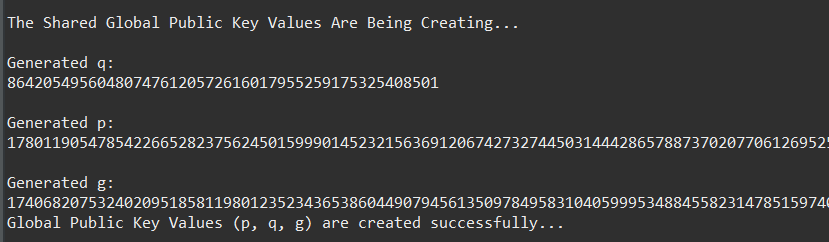
In that third stage of term project, it is asked for us to implement the following scenario:

The sender would like to send a document after he/she sign it. Then, the receiver should verify the sender’s signature. However, it is also asked for us to implement solution to satisfy secrecy, identification, integrity and non-repudiation requirements. I will explain one by one how to implement the given scenario and satisfy the requirements in that report.

1. FIRST STEP – SHARED GLOBAL PUBLIC KEY VALUES GENERATION(P, Q, G)

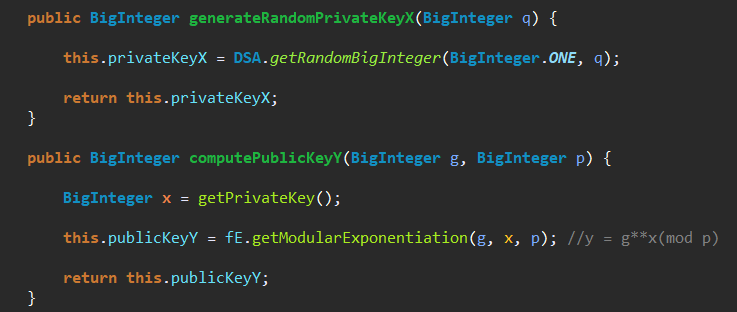
In the first step, we should generate shared global public keys p, q and g. I have used DSAParams class from java.security library. Because if I generate them using the modules which I implemented for the first and second stage, the execution time of them become very high, so I generated them like that:





1. SECOND STEP – PUBLIC/PRIVATE KEY GENERATION OF USERS

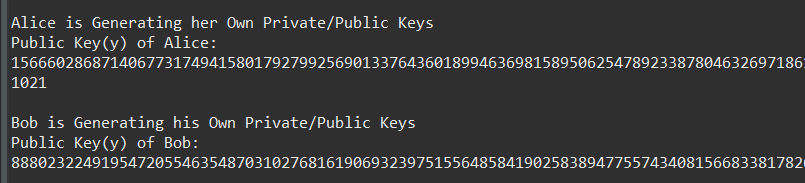
In the second step, the Sender Alice and the Receiver Bob generates their own random private keys X, and then compute their public keys Y, implementation of public and private key generation is:



**Private Key** = x is the random BigInteger such that: x < q

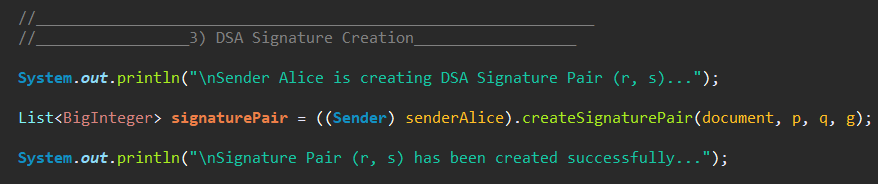
**Public Key** = y is computed as: y = g^^x (mod p)

**One example execution:**

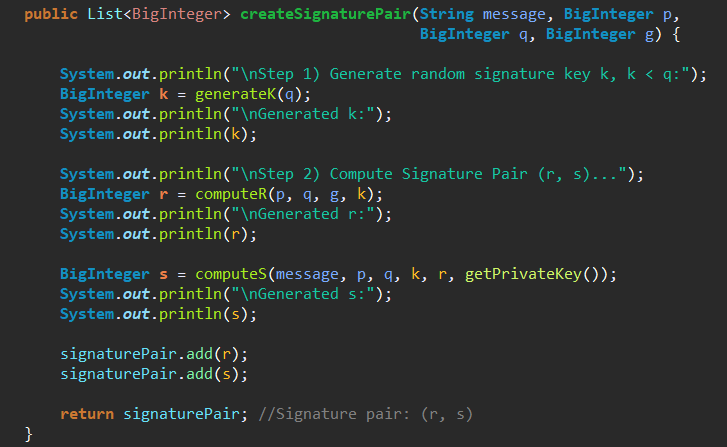


1. THIRD STEP – DSA DIGITAL SIGNATURE CREATION BY SENDER ALICE

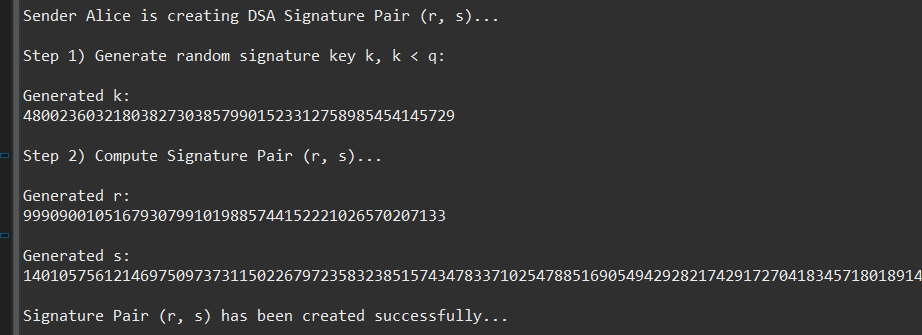
After generating shared global public keys (p, q, g) and the privatee/public keys of the users, now we can create digital signature for the document, the implementation:



Alice creates signature pair like that:

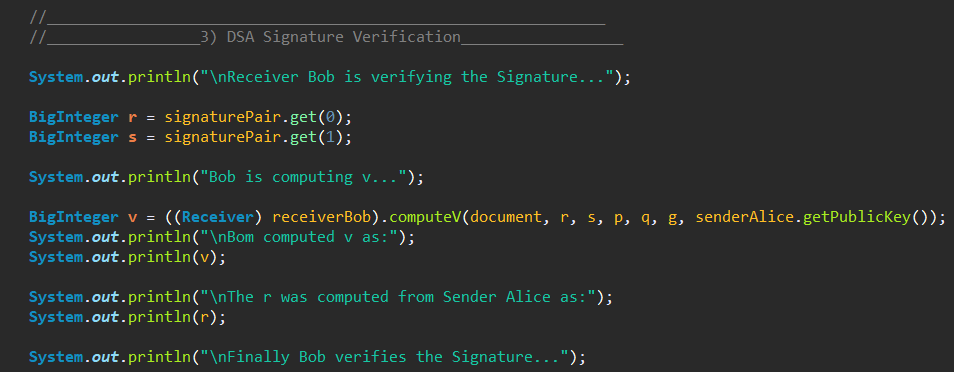


**One execution result:**

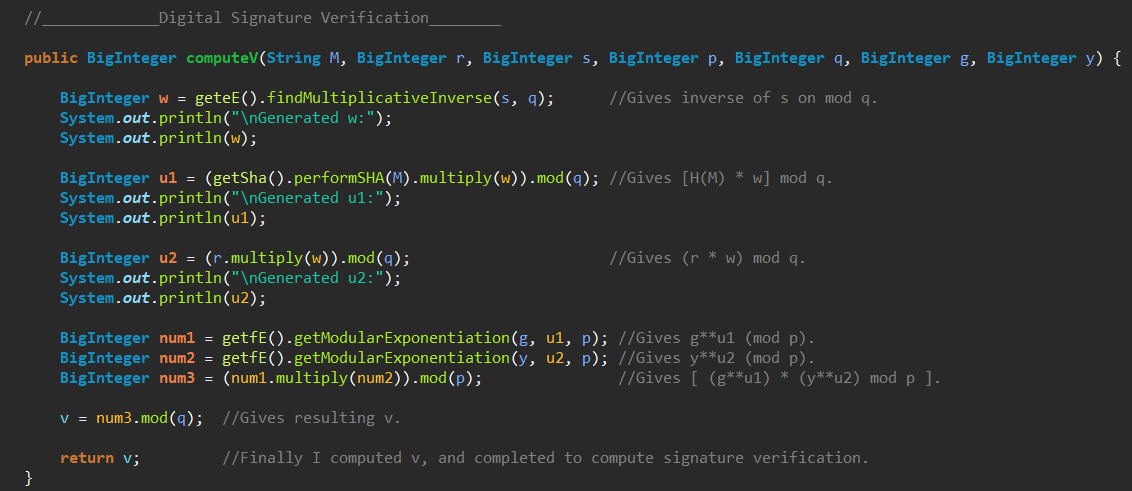


1. FOURTH STEP – DSA DIGITAL SIGNATURE VERIFICATION BY RECEIVER BOB

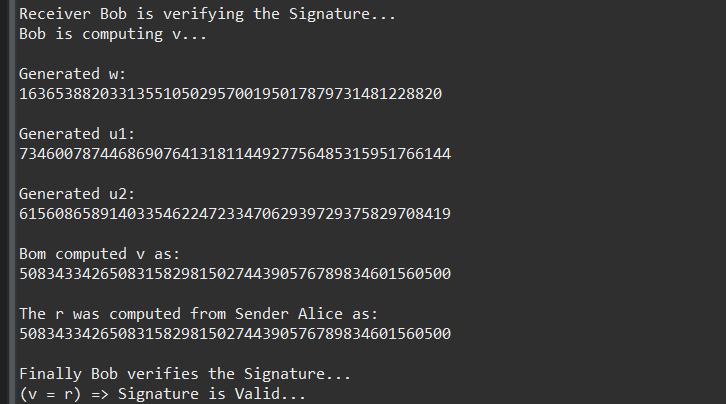
Up to now, the sender Alice created digital signature pair, and now the receiver Bob will use that signature pair (r, s), shared global public key values (p, q, g) and the public key of the Alice, then compute the v from them and compare that v with the r and returns signature is valid or not, here is the implementation:



Bob takes all necessary datas as argument and computes v like that:



**Validation result is computed as follows:**



Note that the resulting v and r are identical so the created Signature is valid.

Now let’s comment on the secrecy, identification, integrity and non-repudiation requirements.

1. COMMENTS ON THE REQUIREMENTS

**Data integrity:**

Data integrity is satisfied with SHA 256 Hash Function, if any change occur during transmission of the message to receiver, the Hash of the received message should give the identical value with the hassh of the sent message, because of the collision free property of the hash functions.

**Identification:**

Identification is satisfied with DSA, the sender signs the message with DSA and Bob verifies that Signature to identify whether received data is come from sender or not.

**Secrecy:**

DSA is just a digital signature scheme and cannot be used for encryption or key exchange, so to satisfy secrecy or confidentiality, we should add encryotion to that scenario.

**Non-Repudiation:**

Non-repudiation is satisfied with DSA, the valid signature can only be created by sender Alice, so she cannot be refuse her action later on, and non repudiation is satisfied.

1. REFERENCES

* <http://www.java2s.com/Code/Java/Security/Digital-Signature-Algorithm-DSA.htm>