K. J. Somaiya College of Engineering, Mumbai-77

Batch: **Roll No.: 1811**

Experiment No.: PKCS V1.5

Grade: AA/AB/BB/BC/CC/CD/DD

Signature of faculty in-charge with date

Title: Public-Key Cryptosystems (PKCSv1.5)

Objective: Design a secure system in line with the public-key standard (PKCSv1.5) using virtual lab.

Expected Outcome of Experiment:

CO	Outcome
CO2	Identify the principles of cryptographic techniques and Apply various cryptographic algorithms for securing systems

Books/ Journals/ Websites referred:

1. http://cse29-iiith.vlabs.ac.in/exp9/Further%20Reading.html?domain=Computer%20Science&lab=Cryptography%20Lab
2. https://medium.com/asecuritysite-when-bob-met-alice/whats-so-special-about-pkcs-1-v1-5-and-the-attack-that-just-won-t-go-away-51ccf35d65b7
3. https://en.wikipedia.org/wiki/PKCS

Abstract:

In cryptography, PKCS stands for "Public Key Cryptography Standards". These are a group of public-key cryptography standards devised and published by RSA Security LLC, starting in the early 1990s. The company published the standards to promote the use of the cryptography techniques to which they had patents, such as the RSA algorithm, the Schnorr signature algorithm and several others. Though not industry standards (because the company retained control over them), some of the standards in

recent years have begun to move into the "standards-track" processes of relevant standards organizations such as the IETF and the PKIX working-group.

Related Theory:

Algorithm: Encryption using PKCS#1v1.5

Input : Recipient's RSA public key (n, e); k = |n| bytes; Data 'D' of length |D| bytes with |D| <= k-11

Output : Encrypted data block of length k bytes.

1. Form the k-byte padded message block EB
EB = 00 || 02 || PS || 00 || D
where || denotes concatenation and PS is a string of (k-|D|-3) non-zero randomly generated bytes(i.e., at least 8 random bytes)

2. Encrypt EB with the RSA Algorithm
C = RSA(EB)

3. Output C

RSA Algorithm

- Key Generation (at A)
- Select two large primes p, q such that p is not equal to q
- Compute n = p * q
- Compute phi(n) = (p-1) * (q-1)
- Select 'e' such that gcd (e, phi(n)) = 1
- Compute d = e-1 mod phi(n)
- A's Public key is (e, n); A's Private key is d

Encryption
Any party B wishing to send a message M to party A encrypts M using RSA as:

C = Me mod n

Decryption
Party A decrypts 'C', received from party B, using his private key d as:

$$M = Cd \bmod n$$

PKCS V1.5:

PKCS#1 v1.5 aims to address the issues of small messages in RSA. With this we pad the message with random string (r):

x = 0x00 || 0x02 || r || 0x00 || m

and then the cipher becomes:
c = x^e (mod N)

The length of random value (r) is k-Len-3 bytes, and where k is the number of bytes in modulus (N), and Len is the number of bytes in the message. In this way we pad short messages, in order for the value of x to be large enough to make sure it is not possible to easily reverse the cipher for short messages. The receiver then derives the value of x. It then detects the 0x0 and 0x2 at the start of the value, and then removes the padding.

Unfortunately PKCS#1 v1.5 is sussible to Bleichenbacher’s attack and has been the core of many attacks on SSL:

Let’s say that Eve is attacking the server. In the message she sends, there’s padding of the pre-shared key (as it is much smaller than the public modulus — n).

In PKCS#1 v1.5 padding we then have two bytes at the start:

0x00 0x02

Eve then captures the cipher in the handshake and which contains the SSL pre-shared key (M):

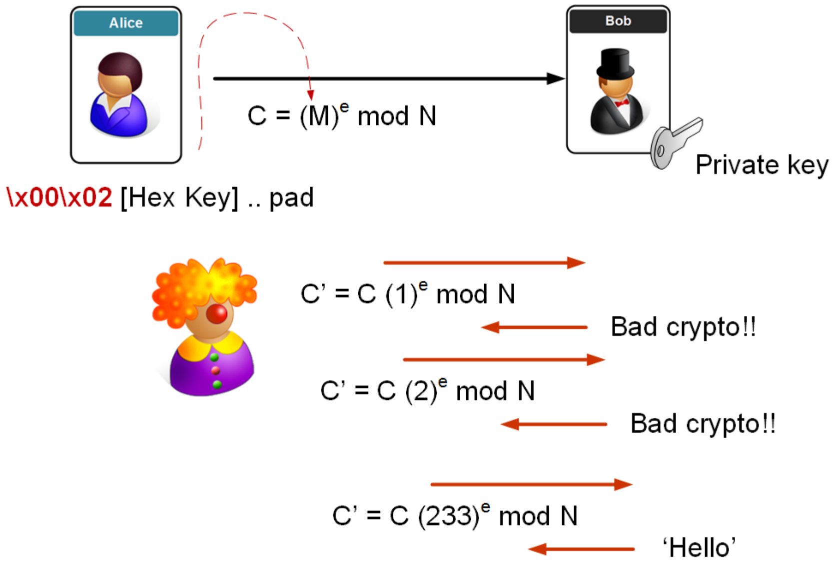
C=M^e (mod N)

She then plays it back to the server, but adds an ‘s’ value (where she multiplies the cipher (c) by s to the power of e (mod N)):

C′=C×(se) (mod N)

where e and N are the known public key elements. The server decrypts and gets:
M′=(C(s^e))^ d (modN)=C^d×s^{ed} (modN)=M×s (modN)
M=C′s

When the server reads this, the first two bytes are likely to be incorrect, so it responds to say "Bad Crypto!". Eve then keeps trying with different s values, until the server gives her a positive response, and she's then on her way to finding the key. As we have 16 bits at the start, it will take us between 30,000 (1 in 215 which is 1-in-32,728) and 130,000 attempts (1 in 217 which is 1-in-131,073) to get a successful access.
We use padding to make sure that M (the pre-shared key) is the same length as the modulus (n). As M is only 48 bytes, we need to pad to give a length equal to n (256 bytes for a 2048-bit key).



In the end we just need to divide the original cipher by the value we have found, and we get M. In this case we capture the cipher with M (which starts with \x00\x02), and then play it back with the addition of s^e, and then detect when there is a success in the cipher code:

Implementation Details:

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Computer Science & Engineering > Cyptography Lab > Experiments

- Introduction
- Theory
- Objective
- Manual
- Experiment
- Quizzes
- Further Readings
- Feedback

Public-Key Cryptosystems (PKCSv1.5)

Plaintext (string):

My name is Soumya

encrypt

Ciphertext (hex):

decrypt

Decrypted Plaintext (string):

Status:

RSA private key

1024 bit | 1024 bit (e=3) | 512 bit | 512 bit (e=3) | Generate | bits = 512

Modulus (hex):

C4E3F7212602E1E396C0B6623CF11D26204ACE3E7D26685E037AD2507DCE8
2FC
28F2D5F8A67FC3AFAB89A6D818D1F4C28CFA548418BD9F8E7426789A67E7
3E41

Public exponent (hex, F4=0x10001):

10001

Private exponent (hex):

7cd1745aec69096129b1f42da52ac9eae0afebbe0bc2ec89253598dcf454960e
3e5e4ec9f8c87202b986601dd167253ee3fb3fa047e14f1dfd5ccd37e931b29d

P (hex):

f0e4dd1eac5622bd3932860fc749bbc48662edabdf3d2826059acc0251ac0d3
b

Q (hex):

d13cb38fbcd06ee9bca330b4000b3dae5dae12b27e5173e4d888c325cda61ab
3

D mod (P-1) (hex):

b3d5571197fc31b0eb6b4153b425e24c033b054d22b9c8282254fe69d8c8c593

D mod (Q-1) (hex):

968ffe89e50d7b72585a79b65cfdb9c1da0963cceb56c3759e57334de5a0ac3f

1/Q mod P (hex):

d9bc4f420e93adad9f007d0e5744c2fe051c9ed9d3c9b65f439a18e13d6e3908

Community Links
Sakshat Portal
Outreach Portal
FAQ : Virtual Labs

Contact Us
Phone: General Information : 011-26582050
Email: support@vlab.co.in

Follow Us

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Computer Science & Engineering > Cyptography Lab > Experiments

Introduction
Theory
Objective
Manual
Experiment
Quizzes
Further Readings
Feedback

Public-Key Cryptosystems (PKCSv1.5)

Plaintext (string):

My name is Soumya

encrypt

Ciphertext (hex):

b410b18acc79cebb46427112933e7e8781c01ed897d4b3ea797e6e5e03e7f7f0
a26745099eb83648103baa2f9095b0b5c5c97cf2bf8305a0012fec4b4ac069ae

decrypt

Decrypted Plaintext (string):

Status:

Encryption Time: 8ms

RSA private key

1024 bit | 1024 bit (e=3) | 512 bit | 512 bit (e=3) | Generate | bits = 512

Modulus (hex):

C4E3F7212602E1E396C0B6623CF11D26204ACE3E7D26685E037AD2507DCE8
2FC
28F2D5F8A67FC3AFAB89A6D818D1F4C28CFA548418BD9F8E7426789A67E7
3E41

Public exponent (hex, F4=0x10001):

10001

Private exponent (hex):

7cd1745aec6909612 9b1f42da52ac9eae0afebbe0bc2ec89253598dcf454960e
3e5e4ec9f8c87202b98601dd167253ee3fb3fa047e14f1dfd5ccd37e931b29d

P (hex):

f0e4dd1eac5622bd393286 0fc749bbc48662edabdf3d2826059acc0251ac0d3
b

Q (hex):

d13cb38fbcd06ee9bca330b4000b3dae5dae12b27e5173e4d888c325cda61ab
3

D mod (P-1) (hex):

b3d5571197fc31b0eb6b4153b425e24c033b054d22b9c8282254fe69d8c8c593

D mod (Q-1) (hex):

968ffe89e50d7b72585a79b65cfdb9c1da0963cceb56c3759e57334de5a0ac3f

1/Q mod P (hex):

d9bc4f420e93adad9f007d0e5744c2fe051c9ed9d3c9b65f439a18e13d6e3908

Community Links
Sakshat Portal
Outreach Portal
FAQ : Virtual Labs

Contact Us
Phone: General Information : 011-26582050
Email: support@vlab.co.in

Follow Us

Computer Science & Engineering > Cyptography Lab > Experiments

Public-Key Cryptosystems (PKCSv1.5)

Plaintext (string):

My name is Soumya

encrypt

Ciphertext (hex):

b410b18acc79cebb46427112933e7e8781c01ed897d4b3ea797e6e5e03e7f7f0
a26745099eb83648103baa2f9095b0b5c5c97cf2bf8305a0012fec4b4ac069ae

decrypt

Decrypted Plaintext (string):

My name is Soumya

Status:

Decryption Time: 14ms

RSA private key

1024 bit | 1024 bit (e=3) | 512 bit | 512 bit (e=3) | Generate | bits = 512

Modulus (hex):

C4E3F7212602E1E396C0B6623CF11D26204ACE3E7D26685E037AD2507DCE8
2FC
28F2D5F8A67FC3AFAB89A6D818D1F4C28CFA548418BD9F8E7426789A67E7
3E41

Public exponent (hex, F4=0x10001):

10001

Private exponent (hex):

7cd1745aec69096129b1f42da52ac9eae0afebbe0bc2ec89253598dcf454960e
3e5e4ec9f8c87202b98601dd167253ee3fb3fa047e14f1dfd5ccd37e931b29d

P (hex):

f0e4dd1eac5622bd393286 0fc749bbc48662edabdf3d2826059acc0251ac0d3
b

Q (hex):

d13cb38fbcd06ee9bca330b4000b3dae5dae12b27e5173e4d888c325cda61ab
3

D mod (P-1) (hex):

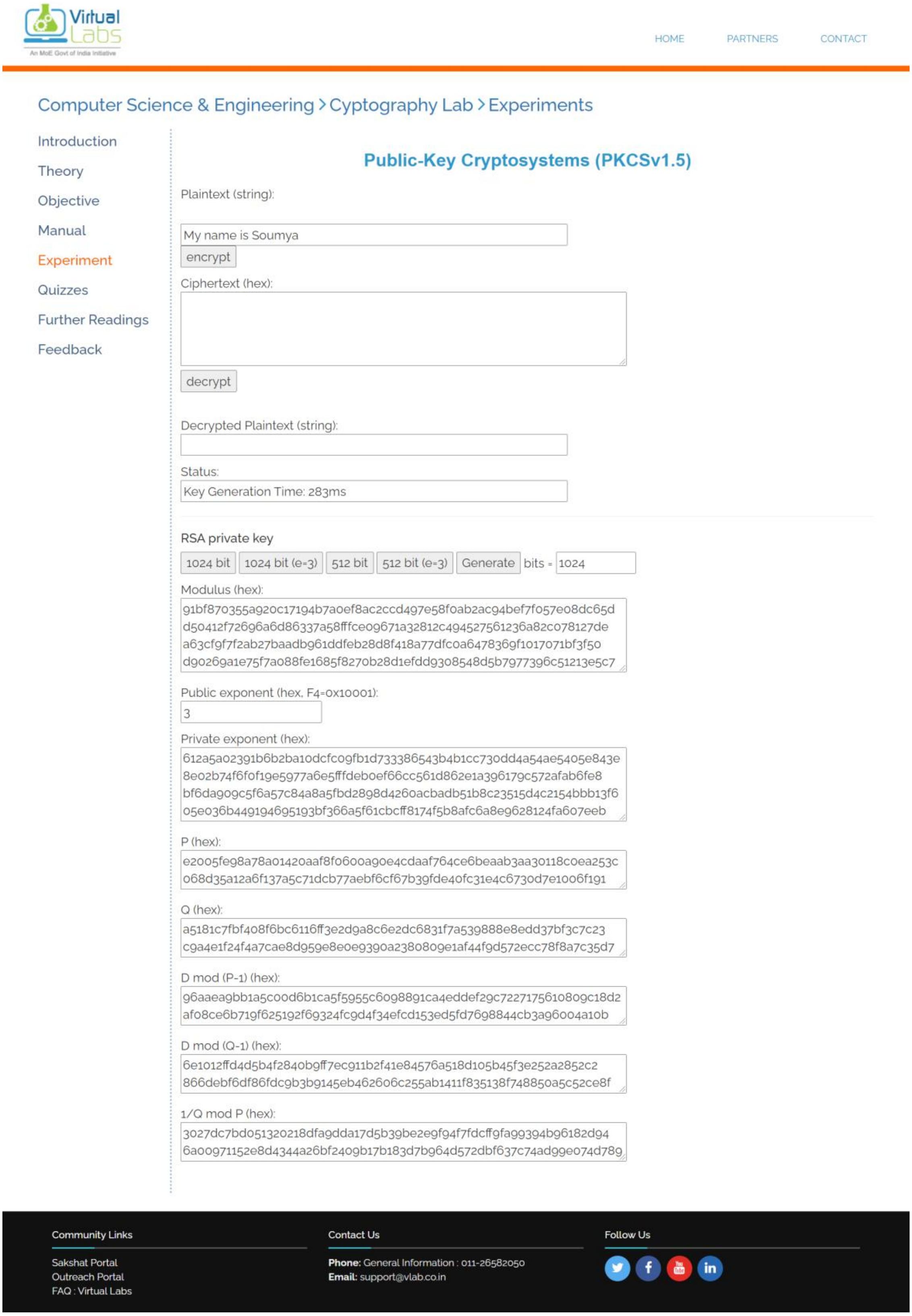
b3d5571197fc31b0eb6b4153b425e24c033b054d22b9c8282254fe69d8c8c593

D mod (Q-1) (hex):

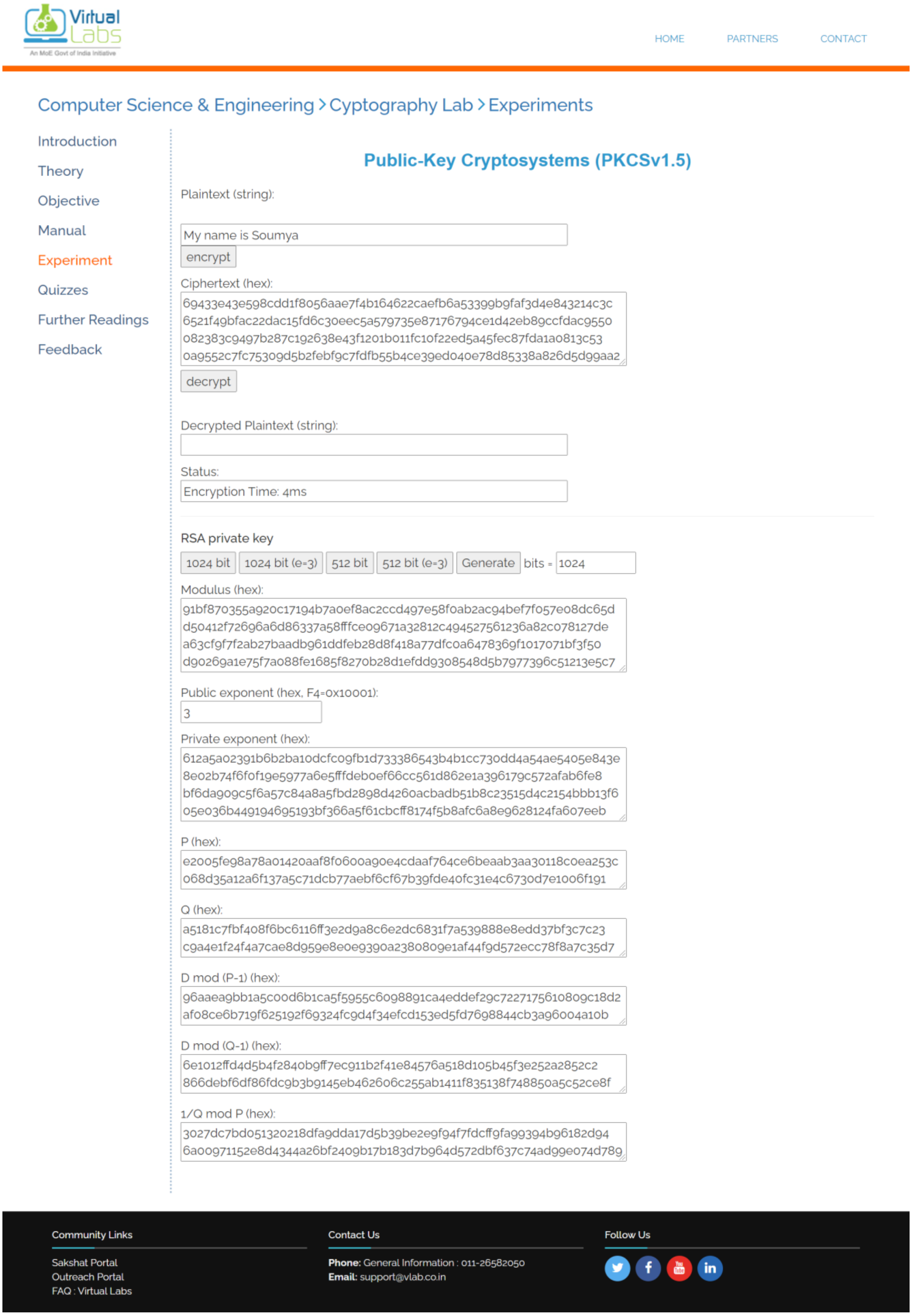
968ffe89e50d7b72585a79b65cfdb9c1da0963cceb56c3759e57334de5a0ac3f

1/Q mod P (hex):

d9bc4f420e93adad9f007d0e5744c2fe051c9ed9d3c9b65f439a18e13d6e3908



Virtual Labs
An MoE Govt of India Initiative
HOME
PARTNERS
CONTACT
Computer Science & Engineering > Cyptography Lab > Experiments
Introduction
Theory
Objective
Manual
Experiment
Quizzes
Further Readings
Feedback
Public-Key Cryptosystems (PKCSv1.5)
Plaintext (string):
My name is Soumya
encrypt
Ciphertext (hex):
decrypt
Decrypted Plaintext (string):
Status:
Key Generation Time: 283ms
RSA private key
1024 bit
1024 bit (e=3)
512 bit
512 bit (e=3)
Generate
bits = 1024
Modulus (hex):
91bf870355a920c17194b7a0ef8ac2ccd497e58f0ab2ac94bef7f057e08dc65d
d50412f72696a6d86337a58fffce09671a32812c494527561236a82c078127de
a63cf9f7f2ab27baadb961ddfeb28d8f418a77dfc0a6478369f1017071bf3f50
d90269a1e75f7a088fe1685f8270b28d1efdd9308548d5b7977396c51213e5c7
Public exponent (hex, F4=0x10001):
3
Private exponent (hex):
612a5a02391b6b2ba10dcfc09fb1d733386543b4b1cc730dd4a54ae5405e843e
8e02b74f6f0f19e5977a6e5fffdeb0ef66cc561d862e1a396179c572afab6fe8
bf6da909c5f6a57c84a8a5fbd2898d4260acbadb51b8c23515d4c2154bbb13f6
05e036b449194695193bf366a5f61cbcff8174f5b8afc6a8e9628124fa607eeb
P (hex):
e2005fe98a78a01420aaf8f0600a90e4cdaaf764ce6beaab3aa30118c0ea253c
068d35a12a6f137a5c71dcb77aebf6cf67b39fde40fc31e4c6730d7e1006f191
Q (hex):
a5181c7fbf408f6bc6116ff3e2d9a8c6e2dc6831f7a539888e8edd37bf3c7c23
c9a4e1f24f4a7cae8d959e8e0e9390a238080e1af44f9d572ecc78f8a7c35d7
D mod (P-1) (hex):
96aaea9bb1a5c00d6b1ca5f5955c6098891ca4eddef29c722717561080c18d2
af08ce6b719f625192f69324fc9d4f34efcd153ed5fd7698844cb3a96004a10b
D mod (Q-1) (hex):
6e1012ffd4d5b4f2840b9ff7ec911b2f41e84576a518d105b45f3e252a2852c2
866debf6df86fdc9b3b9145eb462606c255ab1411f835138f748850a5c52ce8f
1/Q mod P (hex):
3027dc7bd051320218dfa9dda17d5b39be2e9f94f7fdcff9fa99394b96182d94
6a00971152e8d4344a26bf2409b17b183d7b964d572dbf637c74ad99e074d789
Community Links
Sakshat Portal
Outreach Portal
FAQ : Virtual Labs
Contact Us
Phone: General Information : 011-26582050
Email: support@vlab.co.in
Follow Us


Virtual Labs
An MoE Govt of India Initiative
HOME
PARTNERS
CONTACT
Computer Science & Engineering > Cyptography Lab > Experiments
Introduction
Theory
Objective
Manual
Experiment
Quizzes
Further Readings
Feedback
Public-Key Cryptosystems (PKCSv1.5)
Plaintext (string):
My name is Soumya
encrypt
Ciphertext (hex):
69433e43e598cdd1f8056aae7f4b164622caefb6a53399b9faf3d4e843214c3c
6521f49bfac22dac15fd6c30eec5a579735e87176794ce1d42eb89ccfdac9550
082383c9497b287c192638e43f1201b011fc10f22ed5a45fec87fda1a0813c53
0a9552c7fc75309d5b2febf9c7fdfb55b4ce39ed040e78d85338a826d5d99aa2
decrypt
Decrypted Plaintext (string):
Status:
Encryption Time: 4ms
RSA private key
1024 bit
1024 bit (e=3)
512 bit
512 bit (e=3)
Generate
bits = 1024
Modulus (hex):
91bf870355a920c17194b7a0ef8ac2ccd497e58f0ab2ac94bef7f057e08dc65d
d50412f72696a6d86337a58fffce09671a32812c494527561236a82c078127de
a63cf9f7f2ab27baadb961ddfeb28d8f418a77dfc0a6478369f1017071bf3f50
d90269a1e75f7a088fe1685f8270b28d1efdd9308548d5b7977396c51213e5c7
Public exponent (hex, F4=0x10001):
3
Private exponent (hex):
612a5a02391b6b2ba10dcfc09fb1d733386543b4b1cc730dd4a54ae5405e843e
8e02b74f6f0f19e5977a6e5fffdeb0ef66cc561d862e1a396179c572afab6fe8
bf6da909c5f6a57c84a8a5fbd2898d4260acbadb51b8c23515d4c2154bbb13f6
05e036b449194695193bf366a5f61cbcff8174f5b8afc6a8e9628124fa607eeb
P (hex):
e2005fe98a78a01420aaf8f0600a90e4cdaaf764ce6beaab3aa30118c0ea253c
068d35a12a6f137a5c71dcb77aebf6cf67b39fde40fc31e4c6730d7e1006f191
Q (hex):
a5181c7fbf408f6bc6116ff3e2d9a8c6e2dc6831f7a539888e8edd37bf3c7c23
c9a4e1f24f4a7cae8d959e8e0e9390a238080e1af44f9d572ecc78f8a7c35d7
D mod (P-1) (hex):
96aaea9bb1a5c00d6b1ca5f5955c6098891ca4eddef29c7227175610809c18d2
af08ce6b719f625192f69324fc9d4f34efcd153ed5fd7698844cb3a96004a10b
D mod (Q-1) (hex):
6e1012ffd4d5b4f2840b9ff7ec911b2f41e84576a518d105b45f3e252a2852c2
866debf6df86fdc9b3b9145eb462606c255ab1411f835138f748850a5c52ce8f
1/Q mod P (hex):
3027dc7bd051320218dfa9dda17d5b39be2e9f94f7fdcff9fa99394b96182d94
6a00971152e8d4344a26bf2409b17b183d7b964d572dbf637c74ad99e074d789
Community Links
Sakshat Portal
Outreach Portal
FAQ : Virtual Labs
Contact Us
Phone: General Information : 011-26582050
Email: support@vlab.co.in
Follow Us

Virtual Labs
An MoE Govt of India Initiative

HOME PARTNERS CONTACT

Computer Science & Engineering > Cyptography Lab > Experiments

- Introduction
- Theory
- Objective
- Manual
- Experiment
- Quizzes
- Further Readings
- Feedback

Public-Key Cryptosystems (PKCSv1.5)

Plaintext (string):

My name is Soumya

encrypt

Ciphertext (hex):

69433e43e598cdd1f8056aae7f4b164622caefb6a53399b9faf3d4e843214c3c
6521f49bfac22dac15fd6c30eec5a579735e87176794ce1d42eb89ccfdac9550
082383c9497b287c192638e43f1201b011fc10f22ed5a45fec87fda1a0813c53
0a9552c7fc75309d5b2febf9c7fdfb55b4ce39ed040e78d85338a826d5d99aa2

decrypt

Decrypted Plaintext (string):

My name is Soumya

Status:

Decryption Time: 30ms

RSA private key

1024 bit | 1024 bit (e=3) | 512 bit | 512 bit (e=3) | Generate | bits = 1024

Modulus (hex):

91bf870355a920c17194b7a0ef8ac2ccd497e58f0ab2ac94bef7f057e08dc65d
d50412f72696a6d86337a58fffce09671a32812c494527561236a82c078127de
a63cf9f7f2ab27baadb961ddfeb28d8f418a77dfc0a6478369f1017071bf3f50
d90269a1e75f7a088fe1685f8270b28d1efdd9308548d5b7977396c51213e5c7

Public exponent (hex, F4=0x10001):

3

Private exponent (hex):

612a5a02391b6b2ba10dcfc09fb1d733386543b4b1cc730dd4a54ae5405e843e
8e02b74f6f0f19e5977a6e5fffdeb0ef66cc561d862e1a396179c572afab6fe8
bf6da909c5f6a57c84a8a5fbd2898d4260acbadb51b8c23515d4c2154bbb13f6
05e036b449194695193bf366a5f61cbcff8174f5b8afc6a8e9628124fa607eeb

P (hex):

e2005fe98a78a01420aaf8f0600a90e4cdaaf764ce6beaab3aa30118c0ea253c
068d35a12a6f137a5c71dcb77aebf6cf67b39fde40fc31e4c6730d7e1006f191

Q (hex):

a5181c7fbf408f6bc6116ff3e2d9a8c6e2dc6831f7a539888e8edd37bf3c7c23
c9a4e1f24f4a7cae8d959e8e0e9390a2380809e1af44f9d572ecc78f8a7c35d7

D mod (P-1) (hex):

96aaea9bb1a5c00d6b1ca5f5955c6098891ca4eddef29c7227175610809c18d2
af08ce6b719f625192f69324fc9d4f34efcd153ed5fd7698844cb3a96004a10b

D mod (Q-1) (hex):

6e1012ffd4d5b4f2840b9ff7ec911b2f41e84576a518d105b45f3e252a2852c2
866debf6df86fdc9b3b9145eb462606c255ab1411f835138f748850a5c52ce8f

1/Q mod P (hex):

3027dc7bd051320218dfa9dda17d5b39be2e9f94f7fdcff9fa99394b96182d94
6a00971152e8d4344a26bf2409b17b183d7b964d572dbf637c74ad99e074d789

Community Links

Sakshat Portal
Outreach Portal
FAQ : Virtual Labs

Contact Us

Phone: General Information : 011-26582050
Email: support@vlab.co.in

Follow Us

1. Enlist all the Steps followed and various options explored

Step 1: Read the lab introduction, theory and manual to understand how to perform the experiment.
Step 2: Enter the input text to be encrypted in the 'Plaintext' area
Step 3: Select keysize of public key from RSA Private key section by clicking on one of the key button.
Step 4: Click on encrypt button to generate a ciphertext.

2. Explain your program logic, classes and methods used.

The logic for the RSA algorithm is given as follows:

1. Choose two distinct prime numbers p and q.

2. Find n such that n = pq.
n will be used as the modulus for both the public and private keys.

3. Find the totient of n, $\phi(n)$
$\phi(n)=(p-1)(q-1)$.

4. Choose an e such that $1 < e < \phi(n)$, and such that e and $\phi(n)$ share no divisors other than 1 (e and $\phi(n)$ are relatively prime). e is kept as the public key exponent.

5. Determine d (using modular arithmetic) which satisfies the congruence relation
$de \equiv 1 \pmod{\phi(n)}$.

The following logic is used for encryption using PKCS V1.5

1. Form the k-byte padded message block EB
EB = 00 || 02 || PS || 00 || D
where || denotes concatenation and PS is a string of (k-|D|-3) non-zero randomly generated bytes (i.e., at least 8 random bytes)

2. Encrypt EB with the RSA Algorithm
C = RSA(EB)

3. Output C

3. Explain the Importance of the approach followed by you

- This approach helps us understand the basic mathematical foundations of cryptography, to gain insightful experience by working with fundamental cryptographic applications and to train in the art of design and analysis of information security protocols.
- We can easily generate some plaintext, encrypt it with the given key size and check if we get back the original text after decryption. This helps us verify the working of the algorithm.

Conclusion:-

Hence successfully performed a virtual lab to design a secure system in line with the public-key standard (PKCSv1.5) which uses RSA algorithm.