

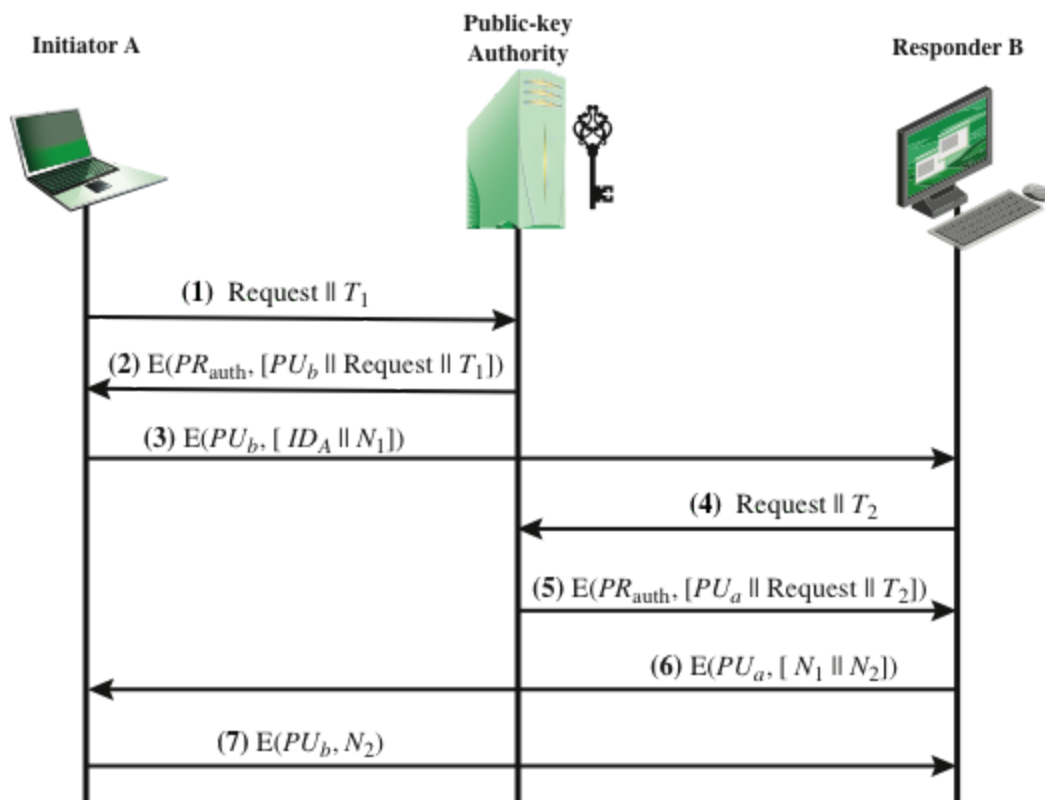
PUBLIC KEY DISTRIBUTION AUTHORITY

SIL765 - Assignment 2

Submitted By:
Mehak
2018MCS2143

PROBLEM STATEMENT

You are required to (a) build a PKDA, and (b) build clients that wish to confidentially send messages suitably encrypted with public key of receiver, but only after they know the other client's public key in a secure manner. You are required to (a) build a, and (b) build clients that wish to send messages suitably encrypted with public key of receiver but of course only after they know each other's public key in a secure manner. Specifically use the scheme described below.



To do so, you will need to:

Assume:

- that clients already (somehow) know the public key of the distribution authority, PKDA,
- that the clients have their corresponding private keys with themselves, and
- that PKDA has the public keys of all the clients,

Messages between PKDA and clients are encrypted using RSA algorithm and PKDA's private key,

Encrypted messages are sent/received between clients once they have each other client's public key, and finally

Find a way to generate and encode "current time" and "nonces".

Use the above to ensure client A can send 3 messages to B, viz Hi 1, Hi 2, and Hi 3. Client B in turn responds with Got-it 1, Got-it 2, etc. to messages received from A.

INTRODUCTION

A public key infrastructure (PKI) is a set of roles, policies, and procedures needed to create, manage, distribute, use, store & revoke digital certificates and manage public-key encryption. The purpose of a PKI is to facilitate the secure electronic transfer of information for a range of network activities such as e-commerce, internet banking and confidential email. It is required for activities where simple passwords are an inadequate authentication method and more rigorous proof is required to confirm the identity of the parties involved in the communication and to validate the information being transferred. PKI provides assurance of public key. It provides the identification of public keys and their distribution. PKDA provides the facility to share the public keys between clients securely.

SYSTEM SPECIFICATION

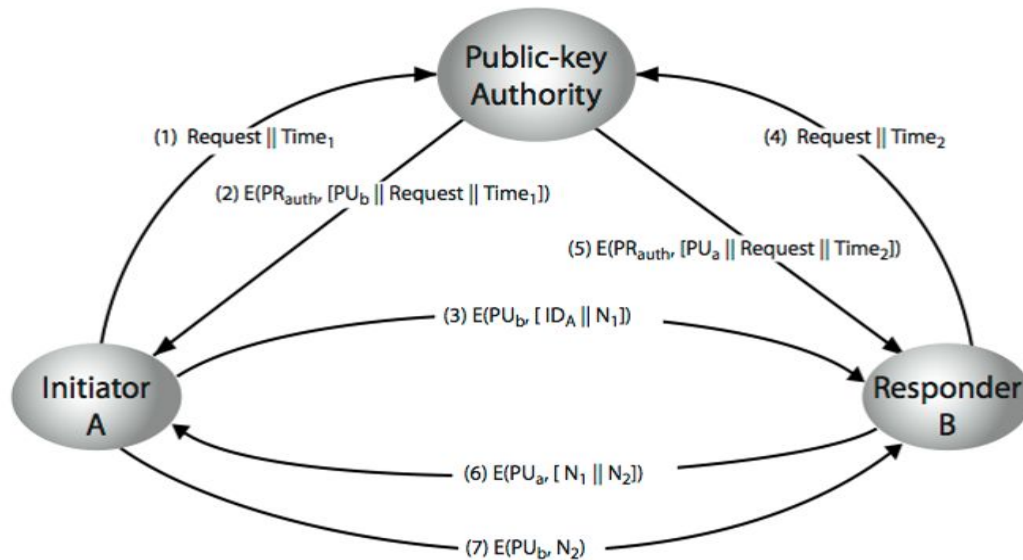
Language Used : Python

Assumptions :

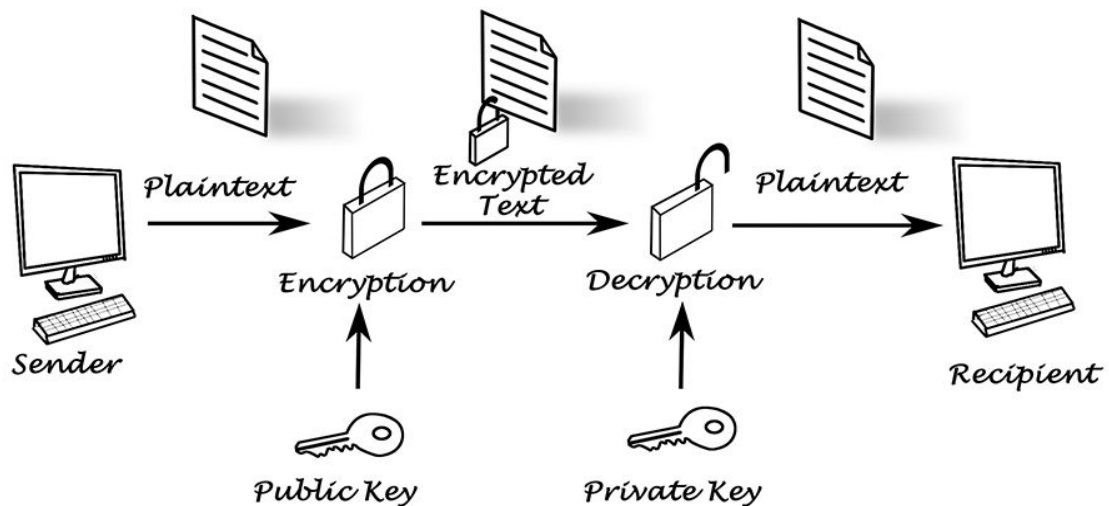
1. PKDA has public keys of Client A and Client B.
2. Clients A and B have their private keys and public key of PKDA.

PROCEDURE

1. Key Exchange (Communication between clients and PKDA)



2. Communication between clients using public key encryption



IMPLEMENTATION

1. Communication between PKDA and Client A

```
meh@meh-Len:~/Documents/nss$ python server.py  
Server is listening on 38556  
New client connected: ('127.0.0.1', 38557)  
Request received by client (encoded): QZxpZW50IEEgfHwgQ2xpZW50IEIgfHwgMTU1MDg2MzMzMQ==  
  
Request received by client (decoded): Client A || Client B || 1550863331  
Response sent by PKDA (original): -----BEGIN PUBLIC KEY-----  
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBggQC4pbNlX8VVX7mHpGmq+d3o7Bz  
qKkpUNK/MxNP9Y8NyD0e4aUiDW0tXjORH/AhtnqX1fF9UdkI5QDt8Nrkd3d4WV3  
600LVkP/LrP7dwrV3gFFvYh8hyImYX2ctqemyqX40BaIdNMYYTz4QIZuvNDXBIAm3  
BsWhOxkdLzpr7t//TwIDAQAAB  
-----END PUBLIC KEY-----Client A || Client B || 1550863331  
Response sent by PKDA (encrypted): [REDACTED]X[REDACTED][REDACTED]KJ[REDACTED]0k[REDACTED]mIo+[REDACTED]?o2[REDACTED]v[REDACTED]  
[REDACTED]'d'Vq[REDACTED]{[REDACTED]+GC[REDACTED]w[P[REDACTED]+Aa}[REDACTED]c[REDACTED];[REDACTED]-[REDACTED](REDACTED)0000H003[REDACTED]_0[REDACTED]K0000z?0[REDACTED]PM0;"000t03g00Bh0[REDACTED]b_nV;[REDACTED]  
&z2[REDACTED]  
[REDACTED]  
f[REDACTED]<q:[REDACTED]0000030B0U*[REDACTED]Cz.[REDACTED]JoZr[REDACTED][0q0  
  
#N[REDACTED]R0 0[N]YXJ00#Ax0d[nN:00d0K<;w[REDACTED]0000s i[REDACTED]D-00  
00j00000hd0q[REDACTED]VI000v0+0vh0000拈 00%-00P0c00*0[X0(0L[REDACTED]000  
00000@00  
[REDACTED]0-00[REDACTED]M0r[REDACTED]  
@I0nN000[REDACTED]Se`000100S000(H0L000
```

```
meh@meh-Len:~/Documents/nss$ python client_A.py
My id: Cli A
Sending request to PKDA (original): Cli A || Client B || 1551202309
Sending request to PKDA (encoded): Q2xpIEEgfHwgQ2xpZW50IEIgfHwgMTU1MTIwMjMwOQ==

Response from PKDA (encrypted): 0b
00B0"|000r0h4T@+000%|0TTT @Y0

Jf00ohjOT00_000?0R=00

inBT=0l00n6TU0;050T0040TKF 00*jrl>0SUA}|0A200000U0000000TPP=TB0o0EG00B0+00z{00
0w0tBS\N0oe0-@0TW40(000T000000Y#001000/ZU00/0100u0k7G0h-py%00r0MYL00+[]|000CE000
|h=00sd?e&Z=kQP0#-0Wk 000-00(80900)00000:00NY|00 L[0000000;_0000Z000Nk.*XN00W!4000^00,An00000Q7)|0z000j0tkKl:002B)00|000
Ah00QV0a0-0U00 CZ00CZ0H0-> 00Te00LCg00000p0cEf0

Key-----BEGIN PUBLIC KEY-----
MIGfMA0GCsqGsIB3DQEBAQUAA4GNADCBiQKBgcqSDepPy2GVJaUG8nCSPjglxNHj
RpGvApj1xdVlzssqw9gAddmJKcdTowJI0fo0c39ggX0L2TPcz6AXTZhd1JhZ0ixW
vXLsaLKgu+n7cQNttQYoVRjjfotYcmYDIqoLwZoJTf4WQ+LT0ZhAv+nFuHTQH9qF
0MRrLjZbhaTFk6RFHQIDAQAAB
-----END PUBLIC KEY-----Cli A || Client B || 1551202309
Message sent to B (original): Cli A || 8a86e2713e3e4498a6dd0eae0cc2794a
Message sent to B (encrypted): %X0000StIOIM0
000j<00+L00000|000 050000e0,$010 -0"f(.00N.0p000B5Ot<0_l0i0Bzu00 00d00NM0:000080BB-60NS0o>-f0l0000sr
}0004S000ck0.0054ok稍00~00600000 000PV,zSO0c0BU00#00RY0`00mSK00 0#000000000vh000,0)00Z3s6700Z00W00B-K 4#0%
Message received from B (decrypted): 8a86e2713e3e4498a6dd0eae0cc2794a||c5c8fee1cbc4412dbc26e39ab3dbdfcl
Message sent to B (original): c5c8fee1cbc4412dbc26e39ab3dbdfcl
Message sent to B (encrypted): a0V000d0/000000 0*f0t00RoY0$-000030b x{H0S0Y00EE00 000000HX:]000K00000sn000-00JuSQI0000QJO
```


2. Communication between Server and Client B

```
New client connected: ('127.0.0.1', 38558)
Request received by client (encoded): Q2xpZW50IEIgfHwgQ2xpZW50IEEgfHwgMTU1MDg2MzM4NQ==

Request received by client (decoded): Client B || Client A || 1550863385
Response sent by PKDA (original): -----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDBq6fjm/L0mXD0pLSw2qjYXsgb
nIyTNMw4XmuG9Q1xzU/6faJrEA09ZVXUztzLHoKbc0N0PtNEptrNlVMJJtt+sfmZ
vtDPHY8QcA0TFYgTivohFLt0SoLSduuWDA/d5egUQhKP6bPsuJUDJxypP0/8efYn
nYRacY00BK2/kdzmQIDAQAB
-----END PUBLIC KEY-----Client B || Client A || 1550863385
Response sent by PKDA (encrypted): {0000^~0500_0000b0|0000E00e0.100cTG0000!0000!005h:0000Y0_0_00-0U0n000d[000Xna
```

```
meh@meh-Len:~/Documents/nss$ python client_B.py
My id: Client B
Message received from A (encrypted): %X0000s00stI0iM0_
000j<00+l00000]000_050000e0,$0]0_0"{.000.0p000B50t<0_l0i0BZu000d00nM0e0000800B-6000o>0-f0100005r
Message received from A (decrypted): Cli A || 8a86e2713e3e4498a6dd0eae0cc2794a
Sending request to PKDA (original): Client B || Client A || 1551202309
Sending request to PKDA (encoded): Q2xpZW50IEIgfHwgQ2xpZW50IEEgfHwgMTU1MTIwMjMwOQ==

Response from PKDA (encrypted): n00_000\00050W000t00f0000040000|00000
}000000
0Y000z000050V0500S00e0s0W0000LSL00#0080000W0c000000]0000:000h0$z00000m0060<09W000]5s0>08c00000000`s0000:000000000
c0r0?ey000300000W000S00BV0:0r00000Mx0sh300b00a,0000d9|1u000?0000~l000^0000k,00rS0Z'0e)00000i0000K000$Af0000_f0u000
Response from PKDA (decrypted): -----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQJCqJdzDdzFQCw09DFypFgcoJyz
B9dvyZ4BTquW9E46M7CykfVUooaQkzf8L0mQ6zbEFHopoy+c710x167X6YQ0A22
HW2mgUo7sixtDNRq+F044fPKPYbHZaryNxyTxL5E9McvjZ5Y+ze07CeeoT56K+35
VQcY7oW+zw62QbW29wIDAQAB
-----END PUBLIC KEY-----Client B || Client A || 1551202309
Message sent to A (original): 8a86e2713e3e4498a6dd0eae0cc2794a|c5c8fee1cbc4412dbc26e39ab3dbdfc1
}0004S000c0k.0004ok0000~00~000000%0000V,zS0c000Ry0"00m5K00_0#0000000000vh000\0}000Z3s0700Z00W00B-K_4#0%
Message received from A (encrypted): a0v000d0/000000|0^f0t00RoY0$>000030b_x{H0$000E00_0000000X:}000K000000sn000>00J0S0Q
I0000QJ0000c00000{0F09B0R00e0P00:
Message received from A (decrypted): c5c8fee1cbc4412dbc26e39ab3dbdfc1
```

3. Communication between Client A and Client B

```
Enter q to quit
A: Hi 1
Message sent to B (original): Hi 1
Message sent to B (encrypted): WV00I070080gD<00洩0000+0h000N0)0(0/010C0%0L0Sw0m00:0Z000)C000.000000000s0m00I0!~00000>0"00J00Z00U0
000000g080qiTT00 P00
Message received from B (encrypted): 0000010!0000R00A)000+10?0AL00~C|-000r0
000 00c.000$000su000W000x00d090000
0~0000000A)000
000T

,I0P:`t00r00pz0100nq
^[[?62;cMessage received from B (decrypted): 7df006318f704092ae45d9af5707c36c||e2bbb88df8b649219e1f6ff2647fab7
A: Hi 2
Message sent to B (original): Hi 2
Message sent to B (encrypted): 000000000^000}Ac='%00lj0~000i00B;000V0
C0C000Z000@_P0E0_00000l0x-000{#60-W0000z000d\02R00?00k8
Message received from B (encrypted): 0000010!0000R00A)000+10?0AL00~C|-000r0
000 00c.000$000su000W000x00d090000
0~0000000A)000
000T

,I0P:`t00r00pz0100nq
Message received from B (decrypted): 7df006318f704092ae45d9af5707c36c||e2bbb88df8b649219e1f6ff2647fab7
A: q
Message sent to B (original): q
0q000%A^Y0010L0000]00000a800000;yP)0g000&r0'002ba00j00%000000(a8k00% s000b0u0000cD0s00000amb00;0R#000
```

```
Message received from A (encrypted): WV00I070080gD<00洩0000+0h000N0)0(0/010C0%0L0Sw0m00:0Z000)C000.000000000s0m00I0!~00000>0"00J0
0Z00U000000g080qiTT00 P00
Message received from A (decrypted): Hi 1
B:Got-it 1
Message sent to A (original): Got-it 1
Message sent to A (encrypted): *B1.H./t000E[(0[#-0000M3T0Z00
0a0x# 0000000}00<xM+0000c3009000000R00]000k0{F000E0tk0*000]0 0HQ0000Yj0K~000Hy0~00:0}0
Message received from A (encrypted): 000000000^000}Ac='%00lj0~000i00B;000V0
C0C000Z000@_P0E0_00000l0x-000{#60-W0000z000d\02R00?00k8
Message received from A (decrypted): Hi 2
B:Got-it 2
Message sent to A (original): Got-it 2
Message sent to A (encrypted): A&S00~,000( 0000000
0000_00+'s0000010000%004[00
000
U-000Y00y0A`000>0)00A0#000000000
0q000%A^Y0010L0000]00000a800000;yP)0g000&r0'002ba00j00%000000(a8k00% s000b0u0000cD0s00000amb00;0R#000
Message received from A (decrypted): q
```

