American Journal of Engineering Research (AJER)
e-ISSN : 2320-0847 p-ISSN : 2320-0936
Volume-02, Issue-09, pp-154-162
www.ajer.org

Research Paper Open Access

IBC Secured Key Partition for A Peer-To-Peer Network

Suneeta peddireddy, Lokesh A, Prapulla C
Suneeta Peddireddy is with Asst.Professor,RajivGandhi Institute of Technology,Bengaluru-560032.
Lokesh A is with Asst.Professor, M S Engineering College, Bengaluru-560064
Prapulla C is with M.Tech III SEM CS&E, M S Engineering College, Bengaluru-560064.

Abstract: - For identity verification and authentication purposes we introduced an Identity based cryptography (IBC) into peer-to-peer (P2P) . However problems on secure private key issuing could not be addressed by current IBC-based solution. In this paper we present a IBC infrastructure setup phase, a peer registration solution using Shamir's (k, n) secret sharing scheme, and a secure key issuing scheme, which adopts key generate center (KGC) and key privacy authorities (KPAs) to issue private keys to peers securely in order to enable the IBC systems to be more acceptable and applicable in real-world P2P networks[1]. We use Byzantine fault tolerance protocol to develop a scheme to authenticate KPAs and to maintain a security of KPAs. This has theoretical analysis and experimental results that shows performance is effective and is able to support large scale network.

Keywords: - *Peer-to-Peer, Secured key issuing, Secrete key sharing, fault tolerance*

BACKGROUND

P2P networks are extremely vulnerable to large spectrum of attacks , with its self-organized and self-maintenance nature. This is mainly due to the lack of certification service responsible for identity verification and for authentication purposes.

We can solve some of the problems by verifying the authenticated nodes identities and by issusing public keys to the nodes for certification using traditional certificate-based public key infrastructure.PKI based on security protocol is difficult to deployed as many nodes that store certificates to each node may become invalid quickly as node churn is highly frequent in P2P network. Practically it is difficult to implement as each node requires more space to store public key certificates. If overlay nodes have a common shared key for secure communication then secured P2P overlay communication is efficient. In dynamic P2P overlay network achieving such an efficiency is difficult as a new key must be generated every time a node membership changes occurs in order to preserve secrecy.
Suneeta peddireddy is with Asst.Professor, Rajiv

I. INTRODUCTION

In Securing Key issuing trust level that is to be placed on third party is important .In regular transmission of data that the users supplies information must be blinded which is also called as blinding. The third party provides a partial private key which is called as blinded. That key is passed on many other third party. Once the users gets the key the users can unblind the keys and retrieve the information. Here the secured key is divided into several parts and any three of the key part can be used by the user to retrieve the information

II. SYSTEM DESIGN

We have four section namely system setup,peer registration, secure keying and system maintenance. In system setup phase we describe how KGC and KPA work in the beginning of the system. In Peer Registration phase and secure keying phase we describe how a peer joins the system . In system maintenance phase the maintenance of KPAs takes place

The requirements of the system are :

- Secure peer registration:

It must be able to a methodology to mitigate attacks such as man-in-middle attack, collusion attacks during peer registration phase.

- Robust system maintenance :

The system must provide a online method to add new substitution of KPAs remove,
identify malicious KPAs.

- Secure key issuing :

The key that is being issued must be secure that will secure the keys without secure channel and defend attacks of all types.

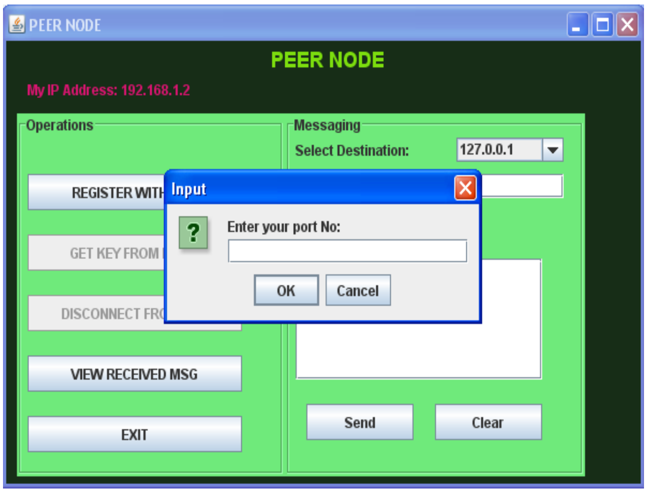
The software requirements analysis (SRA) step of a software development process yields specifications that are used in software engineering. If the software is "semi automated" or user centered, software design may involve user experience design yielding a story board to help determine those specifications. If the software is completely automated (meaning no user or user interface), a software design may be as simple as a flow chart or text describing a planned sequence of events. There are also semi-standard methods like Unified Modeling Language and Fundamental modeling concepts. In either case some documentation of the plan is usually the product of the design.
A software design may be platform-independent or platform-specific, depending on the availability of the technology called for by the design

Design Considerations
There are many aspects to consider in the design of a piece of software. The importance of each should reflect the goals the software is trying to achieve. Some of these aspects are:

- **Compatibility** –The software is able to operate with other products that are designed for interoperability with another product. Since our software is developed on java it is compatible with other products.
- **Packaging -** In java many packages are available. Thus we have used various packages which made coding simpler.
- **Reliability -** The software is able to perform a required function under stated conditions for a specified period of time. Hence the software reliable.
- **Reusability** -The modular components designed capture the essence of the functionality expected out of them and hence the modules are reusable.
- **Robustness -** The software is able to tolerate unpredictable or invalid input.
- **Usability -** The software has user friendly interface which makes its usability easy.

III. RESULT

4.1 Creating Peer Node



PEER NODE

My IP Address: 192.168.1.2

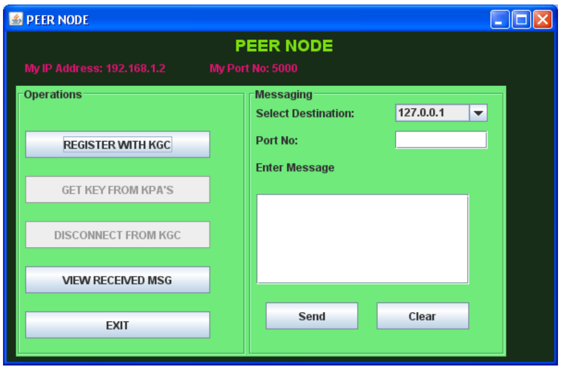
Select Destination: 127.0.0.1

Input

Enter your port No:

4.1: Creating Peer Node

4.2 Register The Peer Node



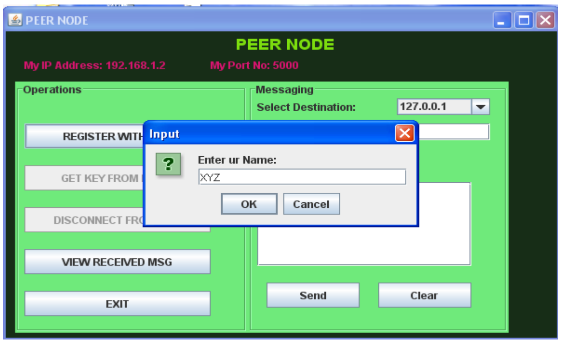
My IP Address: 192.168.1.2 My Port No: 5000

Select Destination: 127.0.0.1

Port No:

Enter Message

4.2.1: Register The Peer Node

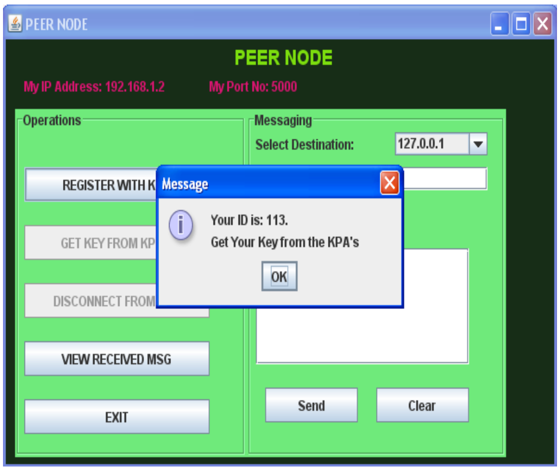


My IP Address: 192.168.1.2 My Port No: 5000

Select Destination: 127.0.0.1

Enter ur Name: XYZ

4.2.2: Giving name to peer node



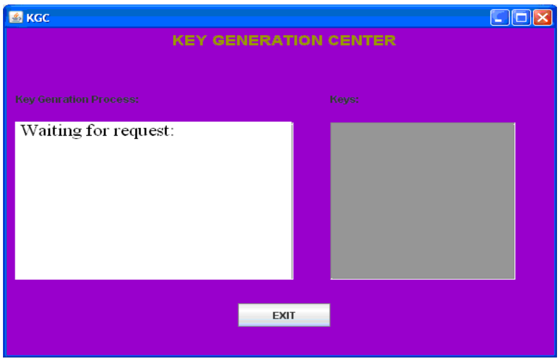
My IP Address: 192.168.1.2 My Port No: 5000

Select Destination: 127.0.0.1

Your ID is: 113.
Get Your Key from the KPA's

4.2.3: ID is received

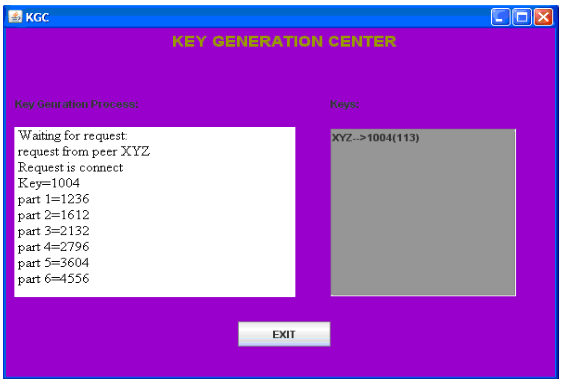
4.3 Key Generation Centre



Waiting for request:

4.3.1: Key Generation Center waiting for request

4.4 Key Generation Process

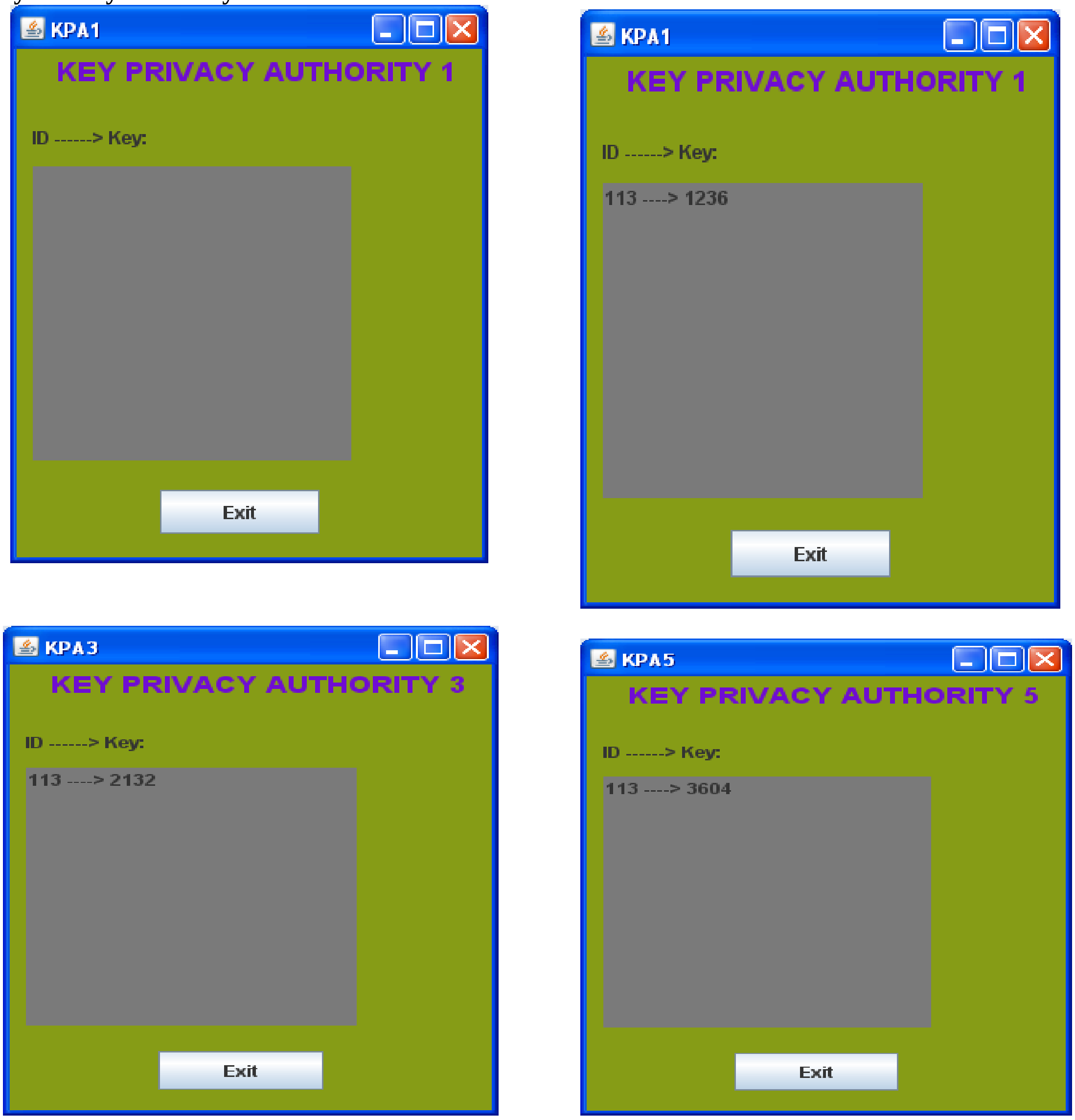


```
Waiting for request:
request from peer XYZ
Request is connect
Key=1004
part 1=1236
part 2=1612
part 3=2132
part 4=2796
part 5=3604
part 6=4556
```

```
XYZ-->1004(113)
```

4.4.1: Key Generation Process

4.5 Key Privacy Authority



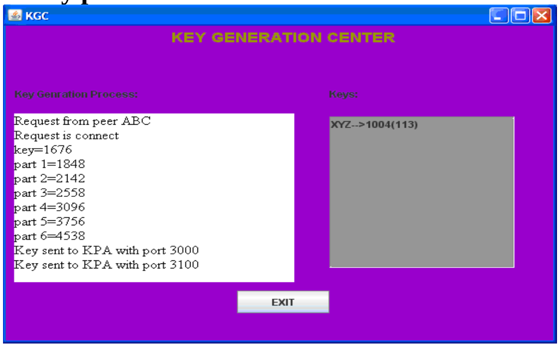
```
113 ----> 1236
```

```
113 ----> 2132
```

```
113 ----> 3604
```

4.5.1 : Key Privacy Authority

4.6 Key generation centre with key parts

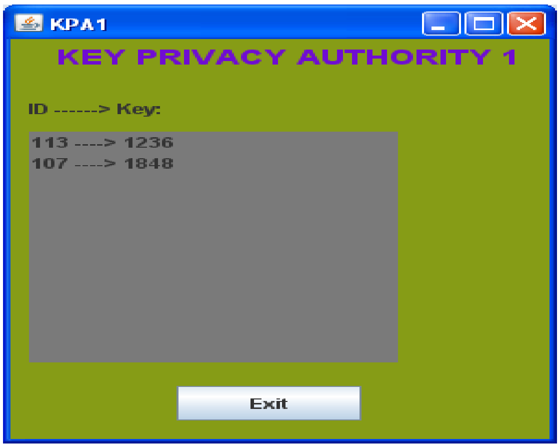


```
Request from peer ABC
Request is connect
key=1676
part 1=1848
part 2=2142
part 3=2558
part 4=3096
part 5=3756
part 6=4538
Key sent to KPA with port 3000
Key sent to KPA with port 3100
```

```
XYZ-->1004(113)
```

4.6.1: Key generation centre with key parts

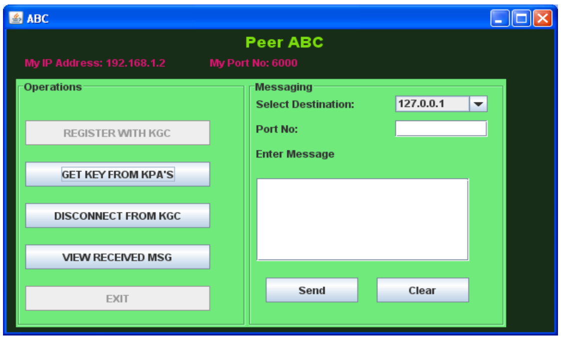
4.7 KPA with a key part



```
ID ------> Key:
113 ----> 1236
107 ----> 1848
```

4.7: KPA with a key part

4.8 Get Key from KPAs



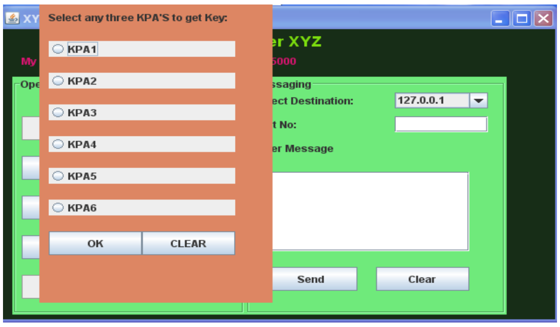
My IP Address: 192.168.1.2 My Port No: 6000

Select Destination: 127.0.0.1

Port No:

4.8.1: Get Key from KPAs

4.9 Retrieving the key



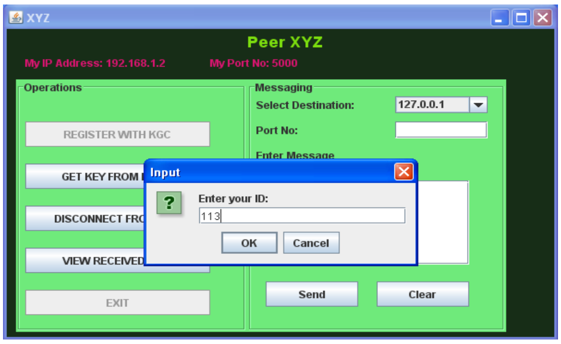
Select any three KPA'S to get Key:

○ KPA1
○ KPA2
○ KPA3
○ KPA4
○ KPA5
○ KPA6

127.0.0.1

4.9.1: Retrieving the key

4.10 Entering ID of the peer node



Peer XYZ

My IP Address: 192.168.1.2 My Port No: 5000

Select Destination: 127.0.0.1

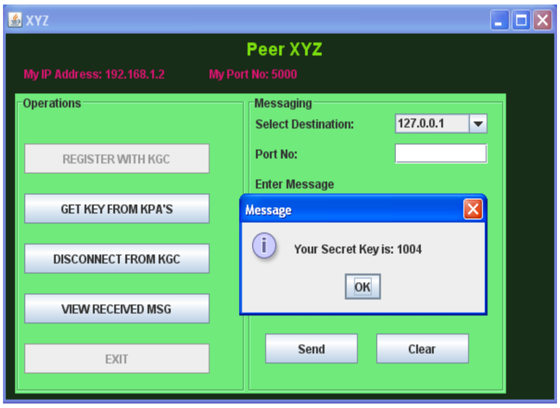
Port No:

Enter Message

Enter your ID:
113

4.10.1: Entering ID of the peer node

4.11 Using the key parts



Peer XYZ

My IP Address: 192.168.1.2 My Port No: 5000

Select Destination: 127.0.0.1

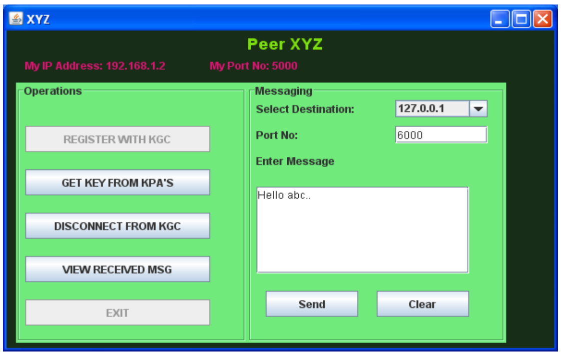
Port No:

Enter Message

Your Secret Key is: 1004

4.11.1: Using the key parts

4.12 Sending Message to Receiver Peer



Peer XYZ

My IP Address: 192.168.1.2 My Port No: 5000

Select Destination: 127.0.0.1

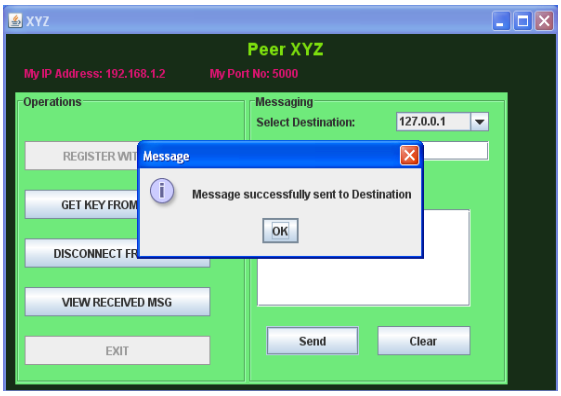
Port No: 6000

Enter Message

Hello abc..

4.12.1: Sending Message to Receiver Peer

4.13Confirmation



Peer XYZ

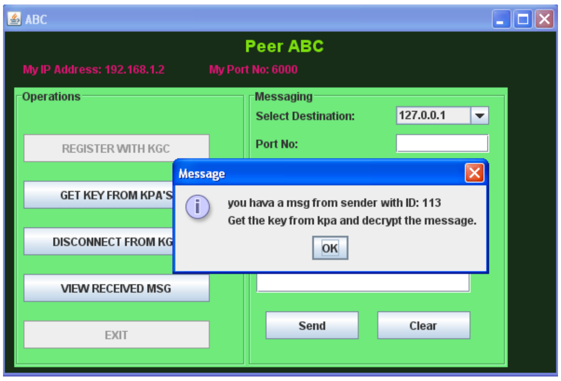
My IP Address: 192.168.1.2 My Port No: 5000

Select Destination: 127.0.0.1

Message successfully sent to Destination

4.13.1: Confirmation

4.14 Receiving a Message



My IP Address: 192.168.1.2 My Port No: 6000

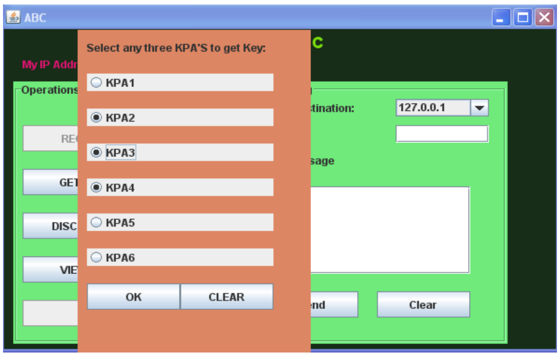
Select Destination: 127.0.0.1

Port No:

you hava a msg from sender with ID: 113
Get the key from kpa and decrypt the message.

4.14.1: Receiving a Message

4.15 Selection of KPAs

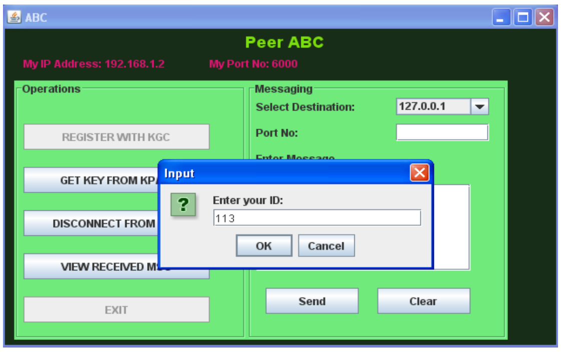


Select any three KPA'S to get Key:

○ KPA1
● KPA2
● KPA3
● KPA4
○ KPA5
○ KPA6

4.15.1: Selection of KPAs

4.16 Entering the peer's ID



My IP Address: 192.168.1.2 My Port No: 6000

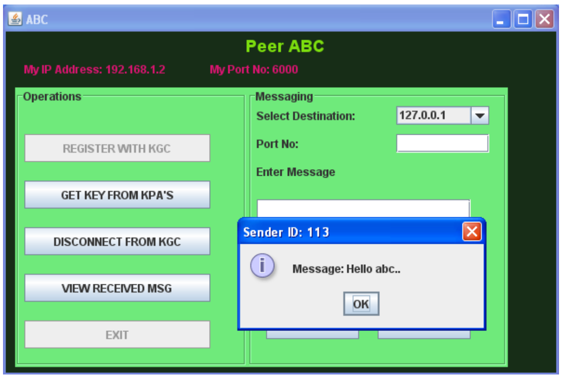
Select Destination: 127.0.0.1

Port No:

Enter your ID: 113

4.16.1: Entering the peer's ID

4.17 Decryption



My IP Address: 192.168.1.2 My Port No: 6000

Select Destination: 127.0.0.1

Port No:

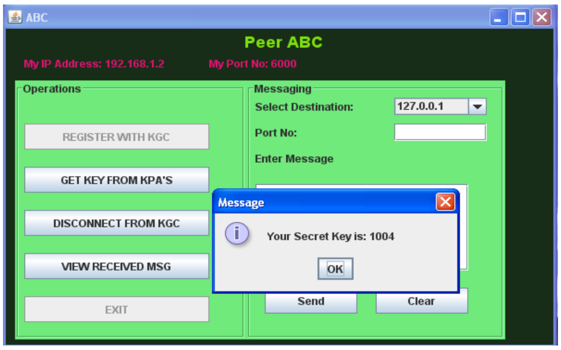
Enter Message

Sender ID: 113

Message: Hello abc..

4.17.1: Decryption

4.18 Key display



ABC

Peer ABC

My IP Address: 192.168.1.2 My Port No: 6000

Select Destination: 127.0.0.1

Port No:

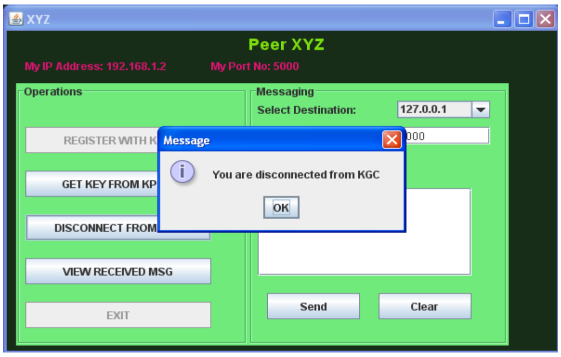
Enter Message

Message

Your Secret Key is: 1004

4.18 Key display

4.19 Disconnected from KGC



XYZ

Peer XYZ

My IP Address: 192.168.1.2 My Port No: 5000

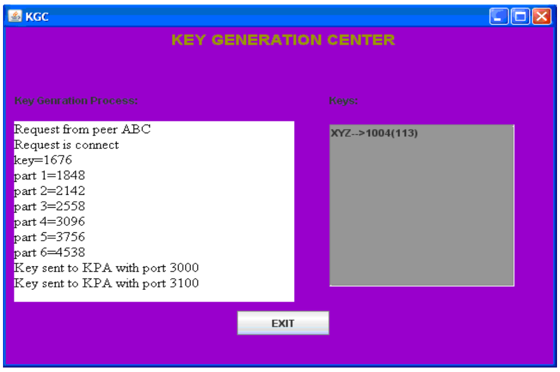
Select Destination: 127.0.0.1

Message

You are disconnected from KGC

6.19.1: Disconnected from KGC

4.20 KGC after disconnection



KGC

KEY GENERATION CENTER

Key Genration Process:

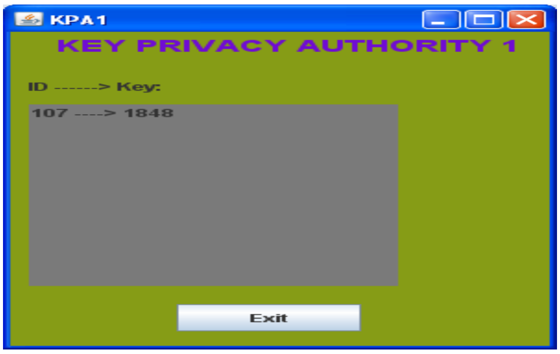
```
Request from peer ABC
Request is connect
key=1676
part 1=1848
part 2=2142
part 3=2558
part 4=3096
part 5=3756
part 6=4538
Key sent to KPA with port 3000
Key sent to KPA with port 3100
```

Keys:

XYZ-->1004(113)

4.20.1: KGC after disconnection

4.21 KPA after disconnection



KPA1

KEY PRIVACY AUTHORITY 1

ID -------> Key:

107 -----> 1848

4.21.1: KPA after disconnection

IV. CONCLUSION & ENHANCEMENT

We have developed a secure key issuing scheme for P2P networks using IBC, SKIP. SKIP provides a peer registration service using Shamir's (k,n) secret sharing scheme. We develop a secure key issuing protocol, which adopts KGC and KPAs to issue private keys to peers securely.

Our future work includes developing a scheme to authenticate KPAs, remove malicious ones and finding out alternate ones to join in the system using the BFT protocol.

REFERENCES

[1] Cong Tang, Ruichuan Chen, Zhuhua Cai, Anming Xie, Jianbin Hu∗ , Liyong Tang, Zhong Chen, "SKIP: A *S*ecure *K*ey *I*ssuing Scheme for *P*eer-to-Peer Networks", in Institute of Software, Peking University, China,2010,

[2] E. Sit and R. Morris, "Security considerations for peer-to-peer distributed hash tables," in *IPTPS*, 2002, pp. 261–269.

[3] Shamir, "Identity-based cryptosystems and signature schemes," in *CRYPTO*, 1984, pp. 47–53.

[4] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *CRYPTO*, 2001, pp. 213–229.

[5] Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo, "Secure key issuing in id-based cryptography," in *ACSW Frontiers*, 2004, pp. 69–74.

[6] R. Gangishetti, M. C. Gorantla, M. L. Das, A. Saxena, and V. P. Gulati, "An efficient secure key issuing protocol in idbased cryptosystems," in *ITCC (1)*, 2005, pp. 674–678.

[7] Saxena, "Threshold ski protocol for id-based cryptosystems," in *IAS*, 2007, pp. 65–70.

[8] Z.-L. Lu, G.-H.; Zhang, "Wheel of trust: A secure framework for overlay-based services," *ICC*, pp. 1148–1153, 2007.

[9] Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *SIGCOMM*, 2001, pp. 149–160.

[9] E. K. Lua, "Securing peer-to-peer overlay networks from Sybil attack," in *ISCIT'07*, Sydney, Australia, 2007.

[9] S. Ryu, K. R. B. Butler, P. Traynor, and P. D. McDaniel, "Leveraging identity-based cryptography for node id assignment in structured p2p systems," in *AINA Workshops (1)*, 2007, pp. 519–524.

[10] L. M. Aiello, M. Milanesio, G. Ruffo, and R. Schifanella, "Tempering kademlia with a robust identity based system," in *Peer-to-Peer Computing*, 2008, pp. 30–39.

[11] Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.