Authentication of Scalable Mobile Presence Cloud Using RSA Mechanism

#Matla Sravan[1], M.Tech C*SE E mail:* matlasravan@gmail.com
#Ashish Ladda [2], *Asst. prof., Department of CSE E mail:* matlasravan@gmail.com
#Mohd. Fasi Ahmed Parvez[3], *Assoc. prof. and HOD, Department of CSE E mail:* parvez40509@gmail.com
#Balaji Institute of Engineering & Sciences, Warangal, Telangana State, India.

Abstract:
A mobile presence service is an essential component of a social network application because it maintains each mobile user's presence information. We use efficient and scalable server architecture, called Presence Cloud, which enables mobile presence services to support large-scale social network applications. When a mobile user joins a network, Presence Cloud searches for the presence of his/her friends and notifies them of his/her arrival. The presence server authentication problem is a security problem in distributed presence services. In centralized presence architectures, there is presence server authentication problem, since users only connect to an authenticated presence server. In Presence Cloud, however, requires a system that assumes no trust between presence servers, it means that a malicious presence server is possible. To address this authentication problem, we used RSA for authentication using encryption and decryption.

1. Introduction

Social networking services on the Internet are growing and increasing numbers of people are using these new ways to communicate and share information. Many users are communicating with both friends from outside the service as well as with people they have only been in contact with through a social networking service. At the same time mobile phones are becoming more powerful and increasingly offer high speed Internet connectivity. Because of this people expect these social networking services to be available on their mobile device, as well as on their personal computer. Given the capabilities of today's mobile devices, it is possible to extend the existing phonebook with capabilities to support a variety of social networking services in addition to the existing communication options. By integrating the contacts gained from the social networking service into the mobile phonebook the user can reach these contacts easily.

2. Proposed System

We analyze the performance of Presence Cloud in terms of the search cost and search satisfaction level. Our current Presence Cloud does not address the communication security problem, and the presence server authentication problem, we discuss the possible solutions as follows. The distributed presence service may make the mobile presence service more prone to communication security problems, such as malicious user attacks and the user privacy. Several approaches are possible for addressing the communication security issues. For example, the Skype protocol offers private key mechanisms for end-to-end encryption. In Presence Cloud, the TCP connection between a presence server and users, or a presence server could be established over SSL to prohibit user impersonation and man-in-the-middle attacks. This end-to-end encryption approach is also used in XMPP/SIMPLE protocol. The presence server authentication problem is another security problem in distributed presence services. In centralized presence architectures, it is no presence server authentication problem, since users only connect to an authenticated presence server. In Presence Cloud, however, requires a system that assumes no trust between presence servers, it means that a

malicious presence server is possible in Presence Cloud. To address this authentication problem, a simple approach is to apply a centralized authentication server. Every presence server needs to register an authentication server; Presence Cloud could certificate the presence server every time when the presence server joins to Presence Cloud. An alternative solution is PGP web of trust model , which is a decentralized approach. In this model, a presence server wishing to join the system would create a certifying authority and ask any existing presence server to validate the new presence server's certificate. However, such a certificate is only valid to another presence server if the relying party recognizes the verifier as a trusted introducer in the system. These two mechanisms both can address the directory authentication problem principally. In additional, the user satisfaction of mobile presence service is another search issue. Several studies have investigated the issues of user satisfaction in several domains, including VOIP WWW search engine. To the best of our knowledge, there is no study of exploring the user satisfaction issues, such as search response time, search precise, etc, about mobile presence services. Given the growth of social network applications and mobile device computing capacity, it is an interesting research direction to explore the user satisfaction both on mobile presence services or mobile devices.

3. Algorithm

RSA is an Internet encryption and authentication system that uses an algorithm developed in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman. The RSA algorithm is the most commonly used encryption and authentication algorithm and is included as part of the Web browsers from Microsoft and Netscape. It's also part of Lotus Notes, Intuit's Quicken, and many other products. The encryption system is owned by RSA Security. The company licenses the algorithm technologies and also sells development kits. The technologies are part of existing or proposed Web, Internet, and computing standards.

The algorithm used for obtaining the public and private keys. The algorithm involves multiplying two large prime numbers (a prime number is a number divisible only by that number and 1) and through additional operations deriving a set of two numbers that constitutes the public key and another set that is the private key. Once the keys have been developed, the original prime numbers are no longer important and can be discarded. Both the public and the private keys are needed for encryption decryption but only the owner of a private key ever needs to know it. Using the RSA system, the private key never needs to be sent across the Internet. The private key is used to decrypt text that has been encrypted with the public key. Thus, if I send you a message, I can find out your public key (but not your private key) from a central administrator and encrypt a message to you using your public key. When you receive it, you decrypt it with your private key. In addition to encrypting messages (which ensures privacy), you can authenticate yourself to me (so I know that it is really you who sent the message) by using your private key to encrypt a digital certificate. When I receive it, I can use your public key to decrypt it. The RSA algorithm involves three steps: key generation, encryption and decryption.

4. Key generation

RSA involves a public key and a private key. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. The keys for the RSA algorithm are generated the following way:

1. Choose two distinct prime numbers p and q. For security purposes, the integers p and q should be chosen at random, and should be of similar bit-length. Prime integers can be efficiently found using a primality test.

2. Compute n = pq. n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.
3. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1) = n - (p + q - 1)$, where φ is Euler's totient function. Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; i.e., e and φ(n) are coprime. e is released as the public key exponent. e having a short bit-length and small Hamming weight results in more efficient. However, much smaller values of e (such as 3) have been shown to be less secure in some settings.[5]
4. Determine d as $d \equiv e^{-1} \pmod{\varphi(n)}$; i.e., d is the multiplicative inverse of e (modulo φ(n)). This is more clearly stated as: solve for d given $d \cdot e \equiv 1 \pmod{\varphi(n)}$. This is often computed using the extended Euclidean algorithm. Using the pseudo code in the Modular integers section, inputs a and n correspond to e and φ(n), respectively. d is kept as the private key exponent.

The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the modulus n and the private (or decryption) exponent d, which must be kept secret. p, q, and φ(n) must also be kept secret because they can be used to calculate d. An alternative, used by PKCS#1, is to choose d matching $de \equiv 1 \pmod{\lambda}$ with $\lambda = \text{lcm}(p - 1, q - 1)$, where lcm is the least common multiple. Using λ instead of φ(n) allows more choices for d. λ can also be defined using the Carmichael function, λ(n).

3.1 Encryption:

Alice transmits her public key (n, e) to Bob and keeps the private key d secret. Bob then wishes to send message M to Alice. He first turns M into an integer m, such that $0 \leq m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to

$c = m^e \pmod{n}$

This can be done quickly using the method of exponentiation by squaring. Bob then transmits c to Alice.

3.2 Decryption:

Alice can recover m from c by using her private key exponent d via computing

$m = c^d \pmod{n}$

Given m, she can recover the original message M by reversing the padding scheme.

4 Conclusion

In this paper we used a scalable server architecture called PresenceCloud that supports mobile presence services in large scale social network services. The authentication problem is the security problem in distributed presence services and it could be solved through RSA for generating the key used for Encryption and Decryption approach.

5 References

1. R. B. Jennings, E. M. Nahum, D. P. Olshefski, D. Saha, Z.-Y. Shae, and C. Waters, "A study of internet instant messaging and chat protocols," *IEEE Network*, 2006.
2.http://en.wikipedia.org/wiki/RSA(cryptosystem)
3. D. Abdus Subhahan, P. Lakshmi Samyuktha " Scalable Mobile Presence Cloud with Authentication through PGP", ijarcsse, ISSN: 2277 128X, 2014
4. Chi-Jen Wu, Jan-Ming Ho, *Member, IEEE*, Ming-Syan Chen, *Fellow, IEEE, "*A Scalable Server Architecture for Mobile Presence Services in Social Network Applications", IEEE TRANSACTIONS ON MOBILE COMPUTING YEAR 2013
5. R. B. Jennings, E. M. Nahum, D. P. Olshefski, D. Saha, Z.-Y. Shae, and C. Waters, "A study of internet instant messaging and chat protocols," *IEEE Network*, 2006.
6.Gobalindex, ttp://www.skype.com/intl/en-us/support/user-guides/p2pexplained/.

7. Z. Xiao, L. Guo, and J. Tracey, "Understanding instant messaging traffic characteristics," *Proc. of IEEE ICDCS*, 2007.
8. C. Chi, R. Hao, D. Wang, and Z.-Z. Cao, "Ims presence server: Traffic analysis and performance modelling," *Proc. of IEEE ICNP*, 2008.

Matla Sravan

M.Tech in Computer Science Engineering from JNTU Hyderabad. I completed by MCA from Bhavathiyar niversity, Coimbattore in 2010. My research areas includes Programming Languages, Data Base Management Systems, Mobile Applications, Data Mining

ASHISH LADDA

Currently working as Asst Professor in CSE Department, Balaji Institute of Engineering Science Interested Area is Networking, Secure Computing etc.

Mohammed Fasi Ahmed Parvez

Currently working as Associate professor and Head of Department Balaji Institute of Engineering Sciences Narsampet, with 12+ years of Experience. Completed M.Tech from JNTU Hyderabad in2010. His research areas includes programming languages, Database Management System and Data Warehouse &Data Mining.