Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 8, August 2013, pg.115 – 119

RESEARCH ARTICLE

Parallel Encryption Technique Combined With Secure Single Sign-On Mechanism for Distributed Computer Networks

R. Suganya[1], A.K. Sathiya Bama[2]

[1]Research Scholar, Department Of Computer Science, Vivekanandha College, Tiruchengode, Tamil Nadu, India
[2]Assistant Professor, Department Of Computer Application, Vivekanandha College, Tiruchengode, Tamil Nadu, India

[1] suganya2121@gmail.com; [2] ask_jnf@yahoo.com

Abstract— These security-enhanced communication tools in a wide-area Globus test bed that we are constructing, called GUSTO (Guidance Utilizing Stable Timing Oscillator). This deployment will allow large-scale application experiments and hence provide feedback on how our security mechanisms work in practical situations. It seems certain that encryption performance will be a bottleneck in many situations. Hence, we will experiment with various performance enhancement techniques, including specialized protocols, parallel encryption algorithms combined with secure single sign mechanism, and use of dedicated encryption processors. Another interesting direction for further work will be to investigate the feasibility of using the Meta computing Directory Service to determine when secure communication mechanisms must be employed, for example because communication occurs over insecure network connections. Clearly one issue that will be important to address in this context is the authenticity of resource database entries.

Key Terms: - Communication; Dos; Key Distribution; Mobile Devices; User Identification

I. INTRODUCTION

The exercise of high-performance networks to pair geographically distributed supercomputers, database systems, generalized scientific instruments, etc., is enabling novel applications in areas such as collaborative engineering, computer-enhanced instrumentation, and ultra-large-scale scientific simulation. However, widespread use of such applications depends crucially on the availability of appropriate security mechanisms. Owners of resources require authentication mechanisms to protect themselves against malicious users. Users of resources may also demand authentication of resources, in order to protect themselves against spoofing by malicious resource providers. Users will often need to ensure that the integrity and confidentiality of data communicated between resources are not compromised, particularly when communication occurs over public networks. Other forms of attack can also be of concern, such as denial of service attacks against applications that use supercomputers to control remote devices.

The task of meeting these security requirements is complicated by the distinctive program structures, computing environments, and performance requirements encountered in high-performance systems. Traditional

distributed systems often have a client-server structure, with limited mutual trust between client and server. In contrast, parallel programs may comprise hundreds or thousands of tightly coupled, fully trusting processes. Distributed systems employ remote procedure call (RPC) or TCP/IP as their primary communication mechanism. In contrast, the applications that we consider here may communicate by using two-sided message passing, streaming protocols, multicast, and/or single-sided get/put operations, as well as RPC; furthermore, they are typically programmed by using message-passing libraries such as the standard Message Passing Interface (MPI [13]) or with specialized parallel languages. Programs must run on parallel computers, which typically provide specialized mechanisms for process creation, communication, and so forth, and which may even run specialized operating systems. At the same time, programs often must achieve a substantial fraction of peak computer and network performance.

II. RELATED WORK

A user identification and key distribution scheme to maintain user anonymity in distributed computer networks. Later, Wu and Hsu [8] pointed out that Lee-Chang scheme is insecure against both impersonation attack and identity disclosure attack. Meanwhile, Yang et al. [9] identified a weakness in Wu-Hsu scheme and proposed an improvement. In 2006, however, Mangipudi and Katti [10] pointed out that Yang et all's scheme suffers from DoS (Deniable of Service) attack and presented a new scheme. In 2009, Hsu and Chuang [11] showed that both Yang et al. and Mangipudi-Katti schemes were insecure under identity disclosure attack, and proposed an RSA-based user identification scheme to overcome the drawbacks. On the other hand, it is usually not practical by asking one user to maintain different pairs of identity and passwords for different service providers, since this could increase the workload of both users and service providers as well as the communication overhead of networks.

To tackle this problem, single sign-on (SSO) mechanism [12] has been introduced so that after obtaining a credential from a trusted authority, each legal user can use this single credential to authenticate itself and then access multiple service providers. Intuitively, an SSO scheme should meet at least two basic security requirements, i.e., soundness and credential privacy. Soundness means that an unregistered user without a credential should not be able to access the services offered by service providers. Credential privacy guarantees that colluded dishonest service providers should not be able to fully recover a user's credential and then impersonate the user to log in other service providers. Formal security definitions of SSO schemes were given in [13]. Chang and Lee made a careful study of SSO mechanism. Firstly, they argued that Hsu-Chuang user identification scheme, actually an SSO scheme, has two weaknesses: (a) An outsider can forge a valid credential by mounting a credential forging attack since Hsu-Chang scheme employed naive RSA signature without any hash function to issue a credential for any random identity selected by a user; and (b) Hsu-Chuang scheme requires clock synchronization since timestamp is used in their scheme. Then, Chang and Lee presented an interesting RSA based SSO scheme, which is highly efficient in computation and communication (So it is suitable for mobile devices), and does not rely on clock synchronization by using nonce instead of timestamp. Finally, they presented well-organized security analysis to show that their SSO scheme supports secure mutual authentication, session key agreement, and user anonymity. In [13], Han et al. proposed a generic SSO construction which relies on broadcast encryption plus zero knowledge (ZK) proof showing that the prove knows the corresponding private key of a given public key. So, implicitly each user is assumed to have been issued a public key in a public key infrastructure (PKI). In the setting of RSA cryptosystem, such a ZK proof is very inefficient due to the complexity of interactive communications between the prover (a user) and the verifier (a service provider). Therefore, compared with Han et al.‟s generic scheme, Chang-Lee scheme has several attracting features: less underlying primitives without using broadcast encryption, high efficiency without resort to ZK proof, and no requirement of PKI for users.

III. PARALLEL ENCRYPTION TECHNIQUE

A Parallel Programs use process creation mechanisms to initiate computation on other computers. In Nexus, process creation involves a call to a create process" function, which invokes machine-specific mechanism to create the new process and instantiates a start point referencing an endpoint in the newly created context. Subsequent communication with that context occurs over the new communication link. The same interface is used to create multiple contexts (for example, when initiating computation on a parallel computer), except that the call returns a vector of start points, one per new process. Typically, process creation involves interaction with some remote service, whether this be an rash daemon, a scheduler on a supercomputer, or some other specialized server. Authentication of the requester and/or the remote server may be required, and an initial security context must be established for subsequent communication between requester and newly created

process. As noted previously, we need to deal with a wide variety of process creation and authentication mechanisms, and must address scalability issues that arise when creating large number of processes.

We seek to address the requirements outlined in the preceding section by constructing a secure communications infrastructure based on a portable communications library called Nexus. We chose to work with Nexus for two reasons. First, it supports many of the tools that are commonly used for application development in parallel and distributed systems, such as the Message Passing Interface (MPI) [13], High Performance Fortran (HPF) [17], and CAVE common a specialized library for collaborative environment applications). Second, its architecture has been designed to support the coexistence and concurrent use of different process creation and communication methods. The latter feature simplifies the integration and management of Different security methods. Some of the parallel tools that have been constructed with Nexus mechanisms. Each of these libraries or languages use Nexus facilities to create processes and to exchange data between processes; Nexus handles automatically the various low-level issues relating to the process creation and communication methods to be used in different situations.

A. Encryption and Decryption Phase

Encryption and Decryption between user and provider is ensured using PET algorithm which is more secure than DES and there are currently no known non-brute-force attacks against PET. Data which is send from each provider to user is encrypted and send to the user, then the user decrypts it and the original data is retrieved. All these encryption and decryption are done using the more Parallel encryption Technique. To run in different machines, programming is based on IP address of the systems. Using the multithreading features of Java, all the providers can be run in parallel. The overall checking of authentication of user and provider are explained in fig.1.

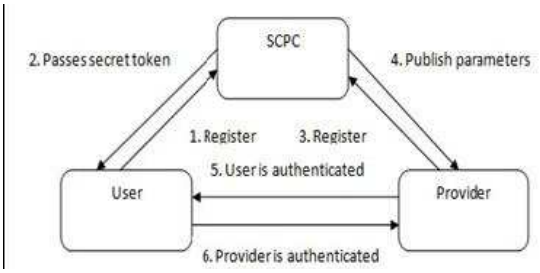

Fig.1 Example Parallel Encryption Architecture

For example, Google Accounts allows a user to sign on to different services provided by Google using the same username/password pair. Another famous example is RSA Secure ID [14], which a two factor authentication solution based on a OTP token and classical username/password credentials, allowing a user to sign on to several Secure ID enabled services using the same token. However, a recent attack to EMC facilities exposed the overall fragility of this heuristic system. Even though their security was unaffected by current attacks, both solutions still require the user to repeatedly perform the sign on procedure. In most of current transparent single sign-on architectures, the user receives some kind of "authentication ticket" after he successfully signs on to the identity provider. When the user desires to sign on, he sends this ticket to the intended service provider or application, which then verifies it's validity by direct communication with the identity provider. This approach has several drawbacks, such as complex management and the requirement of secure online communication between applications and identity providers, which increases network traffic and processing loads.

B. One Time Passwords

Passwords are very often used to user's authentication in computer systems. Regardless the popularity of passwords, they can hardly be used for authentication to Grid services since they do not provide SSO. However, passwords are still usually used in Grids to initial login to the infrastructure, i.e., to access private keys in files or My Proxy repositories, obtain a Kerberos tickets, etc.

One interesting implementation of passwords is one-time passwords (OTPs) that can be used even from untrusted location since their potential sniffing does not cause any harm. As suggested by the name, an OTP can be used only once and is not accepted for authentication after using. From the user point of view an OTP works in the same way as usual passwords do, except to be always different. If an OTP-based system is to be to be deployed within an infrastructure which already uses passwords, changes required to be applied are minimal. Only the components to validate the passwords have to be substituted, which is usually an acceptable change for

deployment. The other change, which is more difficult, is to provide users with tools to manage their OTP and train them in proper usage of the tools.

OTP requires users to manage their lists of generated passwords which are sequentially used for authentication; fortunately there exist hardware devices or applications (so called tokens or soft-tokens) which facilitate managements the OTPs for the user. OTPs can be managed in several ways by these tokens differing in generating, storing and usage of the password. Application tokens are usually meant for mobile devices such as PDA or mobile phones. Both types of the tokens provide two-factor authentication, where user has to prove that they owns something (i.e., the token) and also knows something (PIN or password for the token). The following chapters describe methods available for managing OTPs.

C. Regenerated Sequence of Passwords

In this model OTPs are generated before their first use, so the user gets a list of passwords. Passwords are usually printed on paper or maintained by a soft-token that generates the requested password on demand. A user needs to know which password from the list has to be used for authentication to the server, there for a server has to send the index of the required password at the beginning of the authentication process. This system is pretty easy but the user has to update the list of passwords whenever they have used last one from the list. This principle is used by two systems known as S\Key [18] and OPIE [19], both of them are based on the Lamport's schema [20]. The schema uses an one-way function which eases implementation of password maintained on the server side without requiring the server to store the whole list of OTPs for each user

D. Challenge-Response Password Generating

OTP passwords based on this mechanism are generated by user's PIN code and authentication server's challenge. Typical representative of this category is the CryptoCard RB-1 token2. The user enters their PIN code to the payment card-sized token using an embedded keyboard, resulting in an OTP being then displayed on the small display. The OTP is generated by the MD5 hash function, based on the challenge entered by the user and secret key stored on the token as parameters. The secret key is shared with the authentication server. The token is programmable and various levels of security can be set (length of the PIN code, number of invalid PIN enters) there for the token can be used indefinitely. As is common for smart cards, the token is disabled if an invalid PIN code is entered several times subsequently.

E. Our Solution

As describe in the preceding section, our motivation is to allow for single sign-on (SSO) of a user. Initial authentication occurs in a web browser. Authentication is required for access to the initial web portal and other data services. We use MyProxy as an authentication service for all other services involved. The services authenticate a user by passing username and password to a MyProxy server. MyProxy indicates successful authentication by responding with a positive answer and/or the X.509 credential for that username/password.

At this point, SSO could be achieved by passing the user's secret password from the web portal to the JWS application. However this is undesirable from a security perspective since it risks exposure of the long-lived password. Our solution is to create a session password. This short-lived password can be passed to the JWS application and utilized to contact both the original portal and any other service which uses MyProxy as an authentication service. The lifetime of the session password is set to the expected duration of the user's actual session. Upon expiration of the session password, it can no longer be utilized to authenticate the user. The short lifetime mitigates risk of password theft.

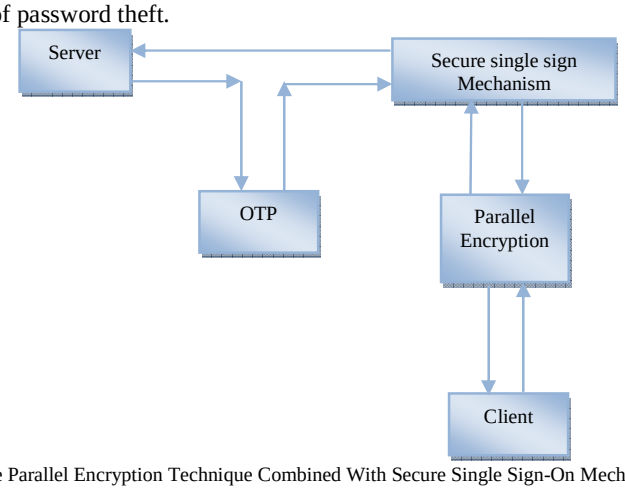

Fig. 2 Example Parallel Encryption Technique Combined With Secure Single Sign-On Mechanism Architecture

IV. CONCLUSION

This paper proposes parallel encryption technique combined with secure single sign-on mechanism based on one-way hash functions and random nonce's to solve the weaknesses described above and to decrease the overhead of the system. Encryption and Decryption of data sent between user and provider can improve security of communication. Encryption and Decryption process can be done using a more secure algorithm, ie, parallel Encryption. Parallel technique is strong enough to be certified for use by the US govt. for top secret information. Parallel is federal information processing standard and there are currently no known non-brute-force attacks against parallel. Thus parallel is given priority than other standards when security is taken into consideration. By using this sso scheme, users need only one password for secure access to all applications and systems and would lock out the hackers entering into the system. But there are some vulnerability problems and there should be a good password, one that is very hard to crack. This paper proposes further research into more efficient enhancements for security of single sign on for distributed computer networks. For third-party sites, credential generation and synced, cloud-based storage can be provided. Auto login, Smart cards, Biometrics are other methods to enhance security for single sign on mechanism for distributed computer networks.

REFERENCES

[1] I. Chlamtac, M. Conti, and J.-N. Liu, "Mobile Ad hoc Networking: Imperatives and Challenges," Ad Hoc Networks, vol. 1, no. 1, pp. 13– 64, July 2003.

[2] R. Rajaraman, "Topology Control and Routing in Ad hoc Networks: ASurvey," SIGACT News, vol. 33, pp. 60–73, June 2002.

[3] S. Biswas and R. Morris, "ExOR: Opportunistic Multi-Hop Routing for Wireless Networks," in Proc. ACM Conference of the Special Interest Group on Data Communication (SIGCOMM), Philadelphia, PA, USA, August 2005, pp. 133–144.

[4] P. Larsson, "Selection Diversity Forwarding in a Multihop Packet Radio Network With Fading Channel and Capture," ACM Mobile Computingand Communications Review, vol. 5, no. 4, pp. 47–54, October 2001.

[5] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing," in Proc. ACM Conference of the Special Interest Group on Data Communication (SIGCOMM), Kyoto, Japan, August 2007, pp. 169–180.

[6] C. Fragouli, J.-Y. L. Boudec, and J. Widmer, "Network Coding: an Instant Primer," SIGCOMM Computer Communication Review, vol. 36, pp. 63–68, January 2006.

[7] I. Leontiadis and C. Mascolo, "GeOpps: Geographical Opportunistic Routing for Vehicular Networks," in Proc. IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks(WoWMoM), Helsinki, Finland, June 2007, pp. 1–6.

[8] S. Yang, F. Zhong, C. K. Yeo, B. S. Lee, and J. Boleng, "Position Based Opportunistic Routing for Robust Data Delivery in MANETs," in Proc. 2009 IEEE Conference on Global Telecommunications (GLOBECOM), Honolulu, Hawaii, USA, December 2009, pp. 1325–1330.

[9] S. Murthy, "Routing in Packet-Switched Networks Using Path-Finding Algorithms," Ph.D. dissertation, University of California - Santa Cruz, 1156 High Street, Santa Cruz, CA 95064, United States, 1996.

[10] J. Behrens and J. J. Garcia-Luna-Aceves, "Distributed, Scalable Routing based on Link-State Vectors," in Proc. ACM SIGCOMM, 1994, pp. 136 147.

[11] S. Murthy and J. J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," Mobile Networks and Applications, vol. 1, no. 2, pp. 183–197, October 1996.

[12] Z. Wang, C. Li, and Y. Chen, "PSR: Proactive Source Routing in Mobile Ad Hoc Networks," in Proc. 2011 IEEE Conference on Global Telecommunications (GLOBECOM), Houston, TX USA, December 2011.

[13] Z. Wang, Y. Chen, and C. Li, "A New Loop-Free Proactive Source Routing Scheme for Opportunistic Data Forwarding in Wireless Networks," IEEE Communications Letters, to appear.

[14] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, July 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3561.txt

[15] M. K. Marina and S. R. Das, "Routing Performance in the Presence of Unidirectional Links in Multihop Wireless Networks," in The Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02),Lausanne, Switzerland, June 2002, pp. 12–23.

