International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume: 3 | Issue: 2 | Jan-Feb 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 - 6470

An Identity-Based Mutual Authentication with Key Agreement

B. V. S. Manikya Rao[1], Y. Triveni[2]

[1]Final M.Tech Student, [2]Associate Professor
[1,2]Department of CSE, Sarada Institute of Science,
[1,2]Technology and Management (SISTAM), Srikakulam, Andhra Pradesh, India

ABSTRACT
Now days mobile networks are rapid development by performing the e-commerce transaction such as online shopping, internet banking and e- payment. So that to provide secure communication, authentication and key agreement is important issue in the mobile networks. Hence, schemes for authentication and key agreement have been studied widely. So that to provide efficient and more secure techniques is necessary. In this paper we are proposed random prime order key agreement protocol proposed for authentication and key agreement. Another technique is used to provide security of transferred data using key xor data transpose technique. By using this technique, we provide more security and more efficiency for transferring data.

1. INTRODUCTION

During secure communication, authentication should be performed to protect users and a secret session key should be established for confidentiality. As the development of cryptography, schemes for authentication and key agreement develop accordingly. Early schemes are based on passwords. The first password authentication scheme to authenticate a remote user over an insecure channel was proposed by Lamport. Introducing public key cryptography into cryptography, Diffie and Hellman proposed the first key agreement scheme. Many authentication and key agreement schemes based on traditional public key cryptography were constructed. Despite the vulnerability and lackness of authentication, Diffie and Hellman's key agreement scheme is the foundation for other schemes and most of key agreement schemes use Diffie and Hellman's technique. Since the introduction of identity based cryptography by Shamir , many identity-base cryptosystems were presented in application. It is not until Boneh and Franklin proposed an identity-base encryption scheme with bilinear pairings on elliptic curves that identity-base cryptography develops rapidly.

Cryptography is the study of "mathematical" systems involving two kinds of security problems: privacy and authentication. A privacy system prevents the extraction information by unauthorized parties from messages transmitted over a public channel, thus assuring the sender of a message that it is being read only in conventional cryptographic by the intended recipient. An authentication system prevents the unauthorized injection of messages into a public channel, assuring the receiver of a message of the legitimacy of its sender A channel is considered public if its security is inadequate for the needs of its users. A channel such as a telephone line may therefore be considered private by some users and public by others. Any channel may be threatened with eavesdropping or injection or both, depending on its use. In telephone communication, the threat of injection is paramount, since the called party cannot determine which phone is calling. Eavesdropping, which requires the use of a wiretap, is technically more difficult and legally hazardous. In radio, by comparison, the situation is reversed. Eavesdropping is passive and involves no legal hazard, while injection exposes the illegitimate transmitter to discovery and prosecution. Having divided our problems into those of privacy and authentication we will sometimes further subdivide authentication into message authentication, which is the problem defined above, and user authentication, in which the only task of the system is to verify that an individual is who he claims to be.

2. Existing System

During secure communication, authentication should be performed to protect users and a secret session key should be established for confidentiality. As the development of cryptography, schemes for authentication and key agreement develop accordingly. Early schemes are based on passwords. The first password authentication scheme to authenticate a remote user over an insecure channel was proposed by Lamport. Introducing public key cryptography into cryptography, Diffie and Hellman proposed the first key agreement scheme. Many authentication and key agreement schemes based on traditional public key cryptography were constructed. Since Diffie and Hellman's scheme lacks authentication and it is vulnerable to Man-in-the middle attack, then authentication with key agreement is necessary and attractive in practical implementation. Despite the vulnerability and laciness' of authentication, Diffie and Hellman's key agreement scheme is the foundation for other schemes and most of key agreement schemes use Diffie and Hellman's technique. Since the introduction of identity based cryptography by Shamir, many identity-base cryptosystems were presented in application. It is not until Boneh and Franklin proposed an identity-base encryption scheme with bilinear pairings on elliptic curves that identity-base cryptography develops rapidly. Various identity-based authentication and key agreement schemes are constructed and made into application. Some authentication schemes can be found in . However, these schemes do not provide mutual authentication and key exchange between the client and the server, which is required in mobile client-server environment.

3. PROPOSED SYSTEM

The proposed system contains mainly two concepts for the authentication, key agreement and security of transferring data. By implementing those concepts we can perform mutual authentication of users and key agreement in both users.

The process of mutual authentication and key agreement as follows.

Random prime order key agreement protocol:

1. The sender will generate private key randomly.
2. Calculate public key using this formula pub=g private % P.
3. After calculating public key the sender randomly choose SRand SV values.
4. Calculate Sa value by performing following steps.
   i. C=(message.hashCode)%200000;
   ii. C=C%200000;
   iii. C3=C-(privatekey*SR);
   iv. Sk=Gcd(C3)
   v. Int k1=Inverse(sk,p-1);
   vi. V=c3*k1;
   vii. Sa=V%P;
5. Calculate SA by performing following steps.
   SA = for(int i=1;i<=Sa;i++)
   temp=(temp*SR)%P;
6. Send public key,SRand SAto Receiver.
7. The receiver also perform the step 1 to 5.
8. After that we can calculate RB value using following steps.
   RB = for(int i=1;i<=VB;i++)
   temp=(temp*SR)%P;
9. The Receiver will send RBvalue to Sender.
10. The sender will receive the RB value and calculate SA1and acknowledgment.
11. After calculating the sender will send to receiver .
12. The receiver will retrieve the both values and perform the authentication status.
13. After that the sender will generate key by using following equation.
    Key=**for**(**int** i=1;i<=RV;i++)
    temp=(temp*RB)%P;
14. The receiver will generate key by using following equation.

Key=**for**(**int** i=1;i<=CA1;i++)
temp=(temp*RB)%P;
After generating shared key the sender will perform the encryption process as follows

Key for data transpose technique:

1. The transferring message can be converted into 32 X 32 matrix format.
2. After generating matrix format we transpose into rows and columns.
3. After transpose matrix that data can be converted into Ascii values.
4. The transpose data AND key can be xor again convert into binary format.
5. After that binary data can be converted into ascii format and that data can be sended to receiver.
6. The receiver will retrieve that cipher data and perform the reverse process.
7. After completion of reverse process we can get original message.

4. Rquirement Analysis

Software Requirements:

Functional requirements: It includes a set of use cases that describe all the interactions the users will have with the software.

Non-functional requirements: requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

Software Requirements for Present Project:

1. Operating System: Any Operating System
2. Run-Time: OS Compatible JVM

Hardware Requirements

1. VDU: Monitor/ LCD TFT / Projector
2. Input Devices: Keyboard and Mouse
3. RAM: 512 MB
4. Processor: P4 or above
5. Storage: 10 to 100 MB of HDD space.

5. SOFTWARE DEVELOPMENT LIFE CYCLE

The **Systems Development Life Cycle (SDLC)**, or *Software Development Life Cycle* in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems.

In software engineering the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system the software development process.

Software Model or Architecture Analysis:

THE GENERAL MODEL

Software life cycle models describe phases of the software cycle and the order in which those phases are executed. There are tons of models, and many companies adopt their own, but all have very similar patterns. The general, basic model is shown below:

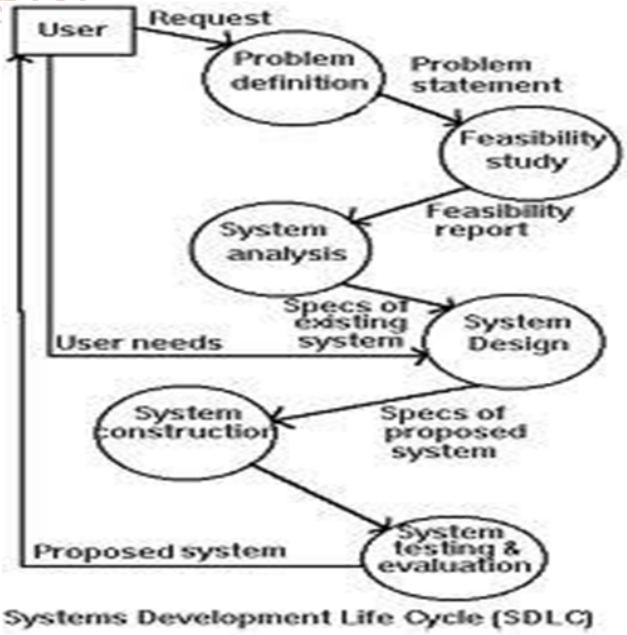


Figure 1 System Development Life Cycle

Each phase produces deliverables required by the next phase in the life cycle. Requirements are translated into design. Code is produced during implementation that is driven by the design. Testing verifies the deliverable of the implementation phase against requirements

SPIRAL MODEL:

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models

The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.

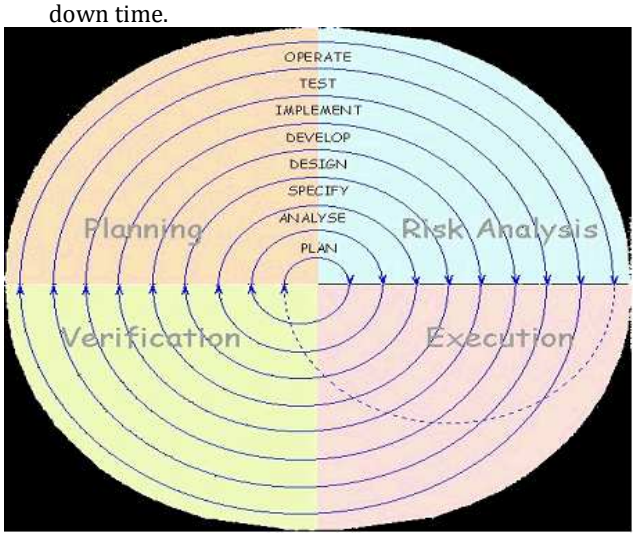
A second prototype is evolved by a fourfold procedure:

1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
2. Defining the requirements of the second prototype.
3. Planning an designing the second prototype.
4. Constructing and testing the second prototype.

At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product. The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.

The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.

- The final system is constructed, based on the refined prototype. The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.



SPIRAL LIFE CYCLE MODEL:

The spiral model is similar to the incremental model, with more emphases placed on risk analysis. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation. A software project repeatedly passes through these phases in iterations (called Spirals in this model). The baseline spiral,starting in the planning phase, requirements are gathered and risk is assessed. Each subsequent spirals builds on the baseline spiral. Requirements are gathered during the planning phase. In the risk analysis phase, a process is undertaken to identify risk and alternate solutions. A prototype is produced at the end of therisk analysis phase. Software is produced in the engineering phase, along with testing atthe end of the phase. The evaluation phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral. In the spiral model, the angular component represents progress, and the radius of the spiral represents cost.

6. TESTING

Software testing can also be stated as the process of validating and verifying that a software program/application/product:

1. meets the business and technical requirements that guided its design and development;
2. Works as expected; and
3. Can be implemented with the same characteristics.

TESTING METHODS

Software testing methods are traditionally divided into white- and black-box testing.

White box testing

API testing (application programming interface) - testing of the application using public
and private APIs

- Code coverage - creating tests to satisfy some criteria of code coverage (e.g., the test designer cancreate tests to cause all statements in the program to be executed at least once)
- Fault injection methods - improving the coverage of a test by introducing faults to test code paths Mutation testing methods
- Static testing - White box testing includes all static testing

TEST COVERAGE

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

Two common forms of code coverage are:

- Function coverage, which reports on functions executed
- Statement coverage, which reports on the number of lines executed to complete the test

BLACK BOX TESTING

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing. Specification-based testing: Specification-based testing aims to test the functionality of

software according to the applicable requirements. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case. Specification-based testing is necessary, but it is insufficient to guard against certain risks.

7. CONCLUSION

This paper proposes random prime order key protocol and key xor data transpose technique for mobile client server environment. Compared with known our scheme is more efficient and good properties against for various types of attacks. This paper also provides more security of transferring data. So that by implementing those techniques we can improve efficiency given project and also provide more security for transferring data.

8. REFERENCES

[1] L. Lamport. "Password authentication with insecure communication", Communications of the ACM 24 (11) 770-772, (1981).

[2] W. Diffie, M. Hellman. "New directions in cryptography". IEEE Trans Inf Theory.IT- 22(6):644-654, (1976).

[3] A. Shamir. "Identity-based cryptosystems and signature protocols", in: Proceedings of the Advances in Cryptology-Crypto84, Santa Barbara, USA, pp. 47-53, (1984).

[4] D. Boneh, M. Franklin. "Identity-based encryption from the Weil pairing", SIAM J. Comput. 32(3) 586-615,(2003) .

[5] M. L. Das, A. Saxena, V.P. Gulati, D.B. Phatak. "A novel remote user authentication scheme using bilinear pairings", Comput. Security 25 (3) 184-189, (2006).

[6] Y. M. Tseng, T. Y. Wu, J. D. Wu. "A pairing-based user authentication scheme for wireless clients with smart cards", Informatica 19 (2) 285-302, (2008).

[7] T. Goriparthi, M.L. Das, A. Saxena. "An improved bilinear pairing based remote user authentication scheme", Comput. Standard Interf. 31 (1) 181-185, (2009).

[8] J. Yang, C. Chang. "An ID-based remote mutual authentication with key agreement protocol for mobile devices on elliptic curve cryptosystem", Computers and Security 138-143,(2009).

[9] E. Yoon, K. Yoo. "Robust ID-based remote mutual authentication with key agreement protocol for mobile devices on ECC", in: 2009 International Conference on Computational Science and Engineering, Vancouver, Canada, pp. 633-640, (2009).

[10] T. Wu, Y. Tseng. "An efficient client authentication and key agreement protocol for mobile client-server environment", Computer Networks. 54, 1520-1530,(2010).

[11] T. H. Chen, Y. C. Chen, and W. K. Shih. "An advanced ecc id-based remote mutual authentication scheme for mobile devices", 2010 Symposia and Workshops on Ubiquitous,Autonomic and Trusted Computing, pp. 116- 120, (2010).

[12] D. He, J. Chen, J. Hu. "An ID-based client authentication with key agreement protocol for mobile clientCserver environment on ECC with provable security". InformationFusion 13 223-230, (2012).