International Journal of Advance Research in Science and Engineering
Volume No.07, Issue No.03, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

Key Distribution in a group using digital envelope based sponge function

I.Ravikumar1,K.V.Kiran2

1Asst. Proff, Dept. of CSE,WISTM Engineering college

2Assoc.Proff, Dept. of CSE,WISTM Engineering college

ABSTRACT

T*he main goal of this paper is to communicate secrete message (Key) among the group members those who are registered in group in a server / client environment using sponge function as tool. The communication is dynamically changed. It is implemented in a python programming. Most of modern cryptography primitives have no provability secure constructions. Their security is defined on the basis of well known in the given time cryptanalytic attacks. Moreover, the asymptotic nature of cryptographic definitions does not let us say anything about how hard it is to break a given cryptographic primitive for keys a certain fixed length. Sponge construction equipped with one ideal permutation and appropriate security parameters are suitable for building provably secure cryptographic primitives.*

Keywords: Sponge function, RSA algorithm, Group communication, encryption and decryption.

I.INTRODUCTION

Now a day's information security places a vital role, when two persons are communicating each other by sending messages then there is a chance of accessing the same message by unauthorized persons. So by providing the security services the unauthorized persons are denied access to these messages. To do this we have encryption algorithms. There are two types of encryption algorithms, symmetric encryption and asymmetric encryption (or) public-key encryption.

The symmetric encryption will have single key where as the public-key encryption involves two keys, public and private keys. In the public-key encryption algorithm for sending the messages the sender will create the cipher text by encrypting the message by one key, the receiver will get the plain text by decrypting the cipher text with other key. If the sender is to send the message to group of members then the sender must send the messages individually one after another.

To overcome this we go for the Secure Key distribution in a group using sponge function. A group is established and a master key is generated using the sponge function and text messages are transmitted between users of group with help of Master key and also between sub groups. In our approach, group key will be updated when a user joins or leaves the group.

This is to maintain confidentiality. Confidentiality can be achieved through changing the key material, known as rekeying every time a new member joins the group or existing member leaves from the group. The new group

International Journal of Advance Research in Science and Engineering
Volume No.07, Issue No.03, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

key is computed guaranteeing forward and backward secrecy. Whenever there is a membership change; group key must be changed to prevent a new user from reading past communication, called backward access control and a departed user from reading future communications, called forward access control.

Once a communication group is formed, the members in a group can send /receive messages from other members of same/different groups. To send a message a user needs to login, compose a message and send to user/users. The message is encrypted using RSA algorithm and is transmitted to the receiver user/users. The entire process of communication is done by establishing Client-Server Environment using socket programming.

In our project the members can be organized into several groups and these groups are connected to other groups. The main objective of our proposed scheme is to establish the Client-Server Environment. When a new member is added or deleted from the group the server generates the new group keys and send them to the members.

The server also generates the individual public and private keys for the members and these are constant. The message can be sent by the members within the organization either to a single user or to any group of users in the organization by encrypting the plain text by their public keys. The messages will be received successfully and they can view the messages by decrypting the cipher text with their private key.

If the message is sent to any particular user then he can view the message by decrypting with his group private key. If the message is sent by a member to a group then the members within the group will view the message by decrypting it with their corresponding group private key.

Establishing Client-Server Environment through Socket Programming:

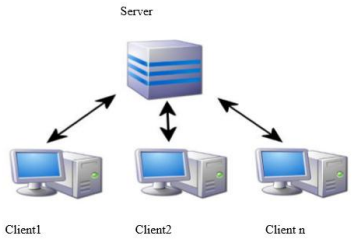

Fig 1.1: client - server environment

The socket associates the server program with a specific hardware port on the machine where it runs so any client program anywhere in the network with a socket associated with that same port can communicate with the server program. Client programs send requests to the server program, and the server program responds to the requests. In order to make a socket connection, you need to know a couple of pieces of information. First you need a host to connect to. Here, we are going to be running the client(s) and the server on the same machine.

Steps for socket programming in java:

At Server side:

i. Creation of a Server Socket. ii. Accept an input connection from the Clients. iii. Reading from the socket. iv. Writing to a socket.

At Client side:

i. Initialize InetAddress of the Server.

ii. Creation of a Socket.

iii. Writing to a Socket.

International Journal of Advance Research in Science and Engineering
Volume No.07, Issue No.03, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

iv. Reading from the Socket.

v. Closing of a connection.

When a user registers, the client password is stored in the form of a hash value in the database. So, when a user login the hash value is calculated for the password and is compared with the password that is stored in the database and then the user is authenticated. The group keys are computed by applying the sponge function to the stored hash values.

II.SPONGE CONSTRUCTION

In the context of cryptography, sponge functions provide a particular way to generalize hash functions to more general functions whose output length is arbitrary. A sponge function instantiates the sponge construction, which is a simple iterated construction building a variablelength input variable-output function based on a fixed length permutation(or transformation).

The sponge construction is a simple iterated construction for building a function F with variable-length input and arbitrary output length based on fixed length transformation or permutation f operating on a fixed number b of bits. Here b is the width.

The sponge construction operates on a state of b = r + c bits. The value of r is called the bit rate and c is called the capacity.

First, all the bits of the state are initialized to zero. The input message is padded and cut into blocks of r bits. The sponge construction proceeds in two phases: The absorbing phase followed by the squeezing phase.

i. In the absorbing phase, the r-bit input message blocks are XORed into the first r bits of the state, interleaved with the applications of the function f. When all the message blocks are processed, the sponge construction switches to the squeezing phase. ii. In squeezing phase, the first r bits of the state are returned as output blocks, interleaved with the applications of the function f. The number of output blocks is chosen at the will by the user.

The last c bits of the state are never directly affected by the input blocks and are never output during the squeezing phase.

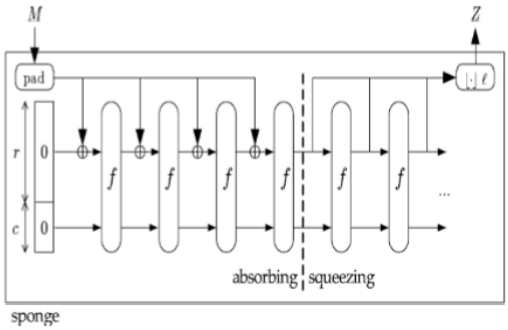

Fig.1.2: Sponge Architecture

International Journal of Advance Research in Science and Engineering
Volume No.07, Issue No.03, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

RSA Public-key Encryption Algorithm:

RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described it in 1978. RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages.

Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated the following way:

i. Choose two distinct prime numbers p and q.

ii. For security purposes, the integer's 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. iii.

iii. Compute n = pq.

iv. n is used as the modulus for both the public and private keys

v. Compute φ (n) = (p – 1) (q – 1), where φ **is Euler's totient** function.

vi. Choose an integer e such that 1 < e < φ (n) and greatest common divisor of (e, φ (n)) = 1; i.e., e and φ (n) are coprime. e is released as the public key exponent.

vii. Determine d as:

a. $d \equiv e^{-1} \pmod{\varphi(n)}$

b. i.e., d is the multiplicative inverse of e mod φ (n).

a. This is more clearly stated as solve for d given (de) mod φ(n) = 1

b. This is often computed using the extended Euclidean algorithm.

c. 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.

Encryption:

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

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.

Existing System:

The members in the organization are organized to form into groups and sub groups. The logical key hierarchy is an efficient approach that supports dynamic group membership.

International Journal of Advance Research in Science and Engineering
Volume No.07, Issue No.03, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

Fig.2.1: Logical key hierarchy

Each member at leaf node have access to the keys of their private key ,all public keys and the private key of the nearest group private key. If any member wishes to send the message to other users who are under a particular group, he/she should encrypt the message with the available group private key and sends to the higher level key generation centers where the message is decrypted and group public key and as the message is not intended for the group, it is again encrypted and is sent till the message to the respective desired group. So, for any message that is to be sent to a particular group involves a number of encryptions and decryptions.

Proposed System:

By using the Secure Key distribution in a group using sponge function, if a member wishes to send a message to any particular group, he encrypts the message with the respective group public key to which he wishes to send the message, and broadcasts the encrypted message. The message is decrypted by the respective group private key and viewed by the respective group members.

Sponge as a reference of security claims:

One could exhaustively list all the properties that a hash function should resist to and assign them resistance levels. Alternatively, claiming the security of a concrete function with regard to a model means comparing the success probability of an attack on the concrete function against that on the model. This allows compact security claims, which address all the possible properties at once, including future requirements not foreseen in an exhaustive list. In fixed digest-length hash functions, the required resistance against attacks is expressed relative to the digest length. Until recently one has always found it reasonable to expect a hash function to be as strong as a random oracle with respect to the classical attacks. An iterated function uses a finite memory to store its state and processes the input, block per block. At any point in time, the state of the iterated function summarizes the input blocks received so far. Because it contains a finite number of bits, collisions can happen in this state. Random oracles, on the other hand, do not have collisions in their "state" as such a concept does not exist. This is the main reason for which random oracles cannot be used directly to express security claims of functions with variable-length output: they would simply never exhibit any effects of the finite memory any concrete iterated function has Random sponges functions, on the other hand, provide an alternative to the random oracle model for expressing security claims. A random sponge is an instance of the sponge construction with f chosen randomly from the set of transformations (or of permutations) over b bits. A random sponge can serve as a reference model for expressing compact security claims for iterated hash functions and stream ciphers. When using a random sponge as a security model, one considers the success of a particular attack. Such a success probability depends not only on the nature of the attack considered but also on the chosen parameters of the random sponge, i.e., its capacity, bit-rate and whether it calls a random permutation or a random transformation.

International Journal of Advance Research in Science and Engineering
Volume No.07, Issue No.03, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

The flat sponge claim is a simplification in the sense that we consider only the worst-case success probability, determined by the RO differentiability bound, which depends solely on the capacity of the random sponge. Hence, it flattens the claimed success probabilities of all attacks using a single parameter: the claimed capacity c claim.

III.SPONGE AS A DESIGN TOOL

As said, our initial goal was to define a reference for security properties of hash function designs. Despite our original intention we realized that the sponge construction could also lead to practical hash function designs. An important aspect is that the design can be based on a permutation, as opposed to a compression function or a block cipher. Designing a suitable permutation is easier than designing a suitable compression function or block cipher. This is rather good news in itself: all the symmetric cryptographic primitives can be based on a fixed-length permutation. A permutation has a single input and therefore treats all the input bits on an equal footing. This is a welcome simplification compared to modes making use of a block cipher or a tweak able block cipher. Generic attacks are attacks that do not exploit the properties of the concrete primitive but only the properties of the construction. The in differentiability framework provides us with a way to upper bound the success probability of generic attacks, and we used it to show that sponge functions are actually resistant to such attacks below a complexity of 2c/2. In fact, these results show that any attack against a sponge function implies that the permutation it uses can be distinguished from a typical randomlychosen permutation. This naturally leads to the following design strategy, which we called the hermetic sponge strategy: adopting the sponge construction and building an underlying permutation f that should not have any structural distinguishers. In this approach, one designs a permutation f on b = r + c bits and uses it in the sponge construction to build the sponge function F. In addition, one makes a flat sponge claim on F with a claimed capacity equal to the capacity used in the sponge construction, namely c claim = c. In other words, the claim states that the best attacks on F must be generic attacks. Hence, c claim = c means that any attack on F with expected complexity below 2c/2 implies a structural distinguisher on f, and the design of the permutation must therefore avoid such distinguishers. In the hermetic sponge strategy, the capacity determines the claimed level of security, and one can trade claimed security for speed by increasing the capacity c and decreasing the bit-rate r accordingly, or vice-versa.

Sponge as a versatile cryptographic primitive: With its arbitrarily long input and output sizes, the sponge construction allows building various primitives such as a hash function, a stream cipher or a MAC. In some applications the input is short (e.g., a key and a nonce) while the output is long (e.g., a key stream). In Cryptographic sponge functions where the input is long (e.g., a message to hash) and the output is short (e.g., a digest or a MAC).Another set of usage modes takes advantage of the duplex construction, a construction that is closely related to the sponge construction and whose security can be shown to be equivalent. The duplex construction allows the alternation of input and output blocks at the same rate as the sponge construction, like a full-duplex communication. This allows one to implement an efficient reseed able pseudo random bit sequence generation and an authenticated encryption scheme requiring only one call to f per input block.

International Journal of Advance Research in Science and Engineering
Volume No.07, Issue No.03, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

IV.IMPORTANCE OF OUR SYSTEM

Involves less number of encryptions and decryptions, as the users have access to the group private keys.

Scope of the system:

This Document plays a vital role in the software development life cycle (SDLC). As it describes the complete requirements of the system, it is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

Objective of the System:

More than one person in the selected group will receive the messages securely within the organization. Confidentiality is maintained for the messages being transmitted by rekeying. Secure Group Communication provides :

i. Confidentiality

ii. Authenticity

iii. Integrity of messages delivered between group members.

Applications:

The Secure Group Communication can be used in organizations, colleges, for sending messages and also in places where we have to send messages with authentication and security. This provides us security and also a very secure way of sending and receiving messages.

Security Issues: Security and confidentiality are the top most concerns of the client. Quality issues refer to how reliable, available and robust should the system be, while developing the proposed system the developer must be able to guarantee the reliability transactions so that they will be processed completely and accurately. The ability of system to detect failures and recovery from those failures refers to the availability of system.

V.CONCLUSION

For the messages delivered between the members of the group, the confidentiality, authenticity, integrity. As there is a very important need for Secure Group Communications by many networking applications also on the internet, many network applications such as teleconferencing, information services, distributed interactive simulation, collaborative work, and group meetings. Information Services is a system most commonly used on the internet these days where information on any subject is updated by the server to the peers registered to it .Distributed Interactive Simulation (DIS) is an open standard for conducting real time platform level war across multiple host computers and is used worldwide, especially by military organizations but also by other agencies such as those involved in space exploration and medicine. Future Work: This paper is done for limited number of users in a group. In future this can be implemented with more number of users and groups.

REFERENCES

[1] "NETWORK SECURITY ESSENTIALS" BY William Stallings

[2] Cryptographic Sponge Functions by Michael Peeters.

[3] Generation of random keys for cryptographic systems by M. BOROWSKI, M. LESNIEWICZ.

[4] Security of Keyed Sponge Construction by Guido Bertoni , Joan Daemen.

International Journal of Advance Research in Science and Engineering
Volume No.07, Issue No.03, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

[5] Security Analysis of Extended Sponge Functions by Thomas Peyrin.

[6] Implementation Aspects of Sponge based Authenticated Encryption for Pervasive devices By Elif Bilge Kavun.

[7] The Mathematics of RSA Public Key Crypto Systems by Burt Kaliski.

[6] SiddharthBanga, SakshamMongia, Vaibhav Tiwari, Mrs. SunitaDhotre, ―Regression and Augmentation Analytics on Earth's Surface Temperature‖ , IJCST, 2017.

[7] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey, J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg, ―Top 10 algorithms in data mining‖ , Springer-Verlag London, 2007

[8] Arthur Chol, NazgolTavabi, Adnan Darwiche, ―Structured Features in Naive Bayes Classification‖ , Association for the Advancement of Artificial Intelligence, 2016.

[9] Harry Zhang, ―The Optimality of Naive Bayes‖ , American Association for Artificial Intelligence, 2004.

[10] Toon Calders, SiccoVerwer, ―Three naive Bayes approaches for discrimination-free classification‖ , Data Min Knowl Disk, 2010.