

An Efficient Authenticated Asymmetric Key Exchange Scheme

Indraneel Chakraborty and Sukumar Nandi
 Indian Institute of Technology, Guwahati
 North Guwahati, Guwahati
 Assam, India. Pin-781031
 Email: sukumar@iitg.ernet.in
 Phone: +91-0361-790321 to 328 Extn 2025
 Fax: +91-0361-790762, 630355.

Abstract—In this paper, an efficient authenticated asymmetric key exchange scheme has been designed based on the features of the Threshold Cryptography [11]. The method provides authentication and key establishment (like RSA, $N = pq$) over an insecure channel using shares of two prime numbers and is secure against even off-line dictionary attack. In the proposed scheme, N , p and q are all secret and each of the two parties knows one of the shares. This provides more security as compared to usual symmetric authenticated key exchange scheme.

Keywords—Product, Prime, Secure, Symmetric, Threshold, Cryptography

I. INTRODUCTION

AUTHENTICATED key exchange schemes are very important for strong authentication over an insecure channel. Most of the schemes proposed [1-8] are based on a secret password. These schemes generate systematic session key, which is vulnerable to all types of attacks, in particular dictionary attack, which depends on the size of the password.

In this paper, utilizing generation and transfer of secret shares we have proposed an efficient authenticated asymmetric key exchange scheme. The shares are generated as per Shamir scheme [10], and exchanged to generate $N = f(p, q)$ analogous to the RSA scheme. Two parties will exchange shares of the two prime numbers p and q , and decide the value of N , a large number. In this method, they will know only one prime number p (or q) and a large number N , which is a function of p and q for a session. Though all partial information transactions for the session key generation take place through insecure channel, the final key remains secret. It is also shown

that the scheme is well guarded against various attacks.

The organization of the paper is as follows: the next section presents the Mathematical background. In Section 3 the notations and the key exchange scheme are provided. The scheme is analyzed against attacks in Section 4. Section 5 describes the uses of the proposed scheme.

II. MATHEMATICAL FOUNDATIONS

THE shares of a key have been generated as in the Shamir Scheme [10] of dividing each key into numbers that on interpolation give the secret key.

Definition 1: Given a secret σ , choose a random polynomial $f(x)$ of degree t , such that $f(0) \bmod k = \sigma$. Then each share is $\sigma_i = f(i) \bmod k$. Where k is a large prime number.

On generating n shares of a polynomial $f(x)$, $(t + 1)$ of which combine to give $\sigma = f(0)$. So it can be written as

$$(\sigma_1, \sigma_2, \dots, \sigma_n) \xleftarrow{(t,n)} \sigma.$$

The generation of shares has been made on the basis of the following theorem. The proof of the theorem is trivial.

Theorem 1: For $n \geq t + 1$, σ can be determined, if $(\sigma_1, \sigma_2, \dots, \sigma_n) \xleftarrow{(t,n)} \sigma$ i.e. for a polynomial $f(x)$ of degree t , $f(x)$ can be determined if there are $n \geq t + 1$ distinct shares are available.

As a minimum of $(t + 1)$ shares is required to reveal the number used as a key, this value $(t + 1)$ is known as the *threshold* of the system. The cryptosystems with such a method of key storage and generation are called *ThresholdCryptosystems* [11]. The above theorem

leads to the following corollary.

Corollary 1: It is not computationally very expensive to arrive at $(t + 1)$ shares of a secret σ when one is allowed the choice of a polynomial $g(x)$ of degree t . By the choice of a polynomial, it is meant that the coefficient of the polynomial are chosen to get $g(0) = \sigma$.

This is based on the famous *birthday problem* that “it is much easier to find two people in a room with the same birthday that two people with a particular birth-date”.

In RSA scheme, there is a pair of prime numbers (p, q) and another large number N that is a function of p and q . One pair (p, N) is used for encryption and the other pair (q, N) is used for decryption or vice-versa. If $N = pq \text{ mod } k$, where k is a *third* prime number then the following theorem holds.

Theorem 2: Given $N = pq \text{ mod } k$ where p, q, k are three prime numbers, p cannot be determined if q, k and N are known and vice versa for q .

Proof: Let $p\alpha = \beta k$ and $q' = (q + \alpha)$ So

$$\begin{aligned} N &= p(q + \alpha) \text{ mod } k \\ &= (pq \text{ mod } k + p\alpha \text{ mod } k) \text{ mod } k \\ &= (pq \text{ mod } k + 0) \text{ mod } k \\ &= pq \text{ mod } k \end{aligned}$$

Hence it is very difficult to differentiate between q and q' . The above theorem could as well be shown to work for p .

The proposed asymmetric key exchange scheme is designed using the theorems and corollary of this section. In the next section the scheme is presented.

III. THE PROPOSED KEY EXCHANGE SCHEME

THE proposed scheme operates in two phases: the first phase establishes a secret session key (i.e. N) through a series of exchanges of generated shares and local computation of N ; and in the second phase, both parties confirm each other's knowledge of the keys.

A. Key Establishment Phase

In the first phase, when the key is generated both parties A and B exchange t shares of their prime numbers, where each prime numbers corresponds to a polynomial of degree t (as mentioned in definition 1). Then they exchange t shares of the resulting product $N = pq$ by

multiplying shares of p and q and adding that to the corresponding shares of a polynomial of degree which has a value of zero. Also they exchange $(2t + 1)$ th share of their prime numbers with adjustable factors β 's. Based on all these information each party locally computes N as a function of two prime numbers. The algorithm is proposed below with the following notations.

- p = prime number known to party A
- q = prime number known to party B
- k = a big prime number known to both parties (Universally known)
- $N = pq \text{ mod } k$ (N is the number to be generated). The generated N is a polynomial of degree $2t$
- $p_i = f_p(i) \text{ mod } k$, where $f_p(\cdot)$ is a random polynomial of degree t and $f_p(0) \text{ mod } k = p$
- $q_i = f_q(i) \text{ mod } k$ where $f_q(\cdot)$ is also a random polynomial of degree t and $f_q(0) \text{ mod } k = q$.
- b_i = shares of a random polynomial $f_z(\cdot)$ of degree $2t$ such that $f_z(0) \text{ mod } k = 0$.
- β_1, β_2 are two random numbers such that $b_{2t+1} = (p_{2t+1}\beta_2 + q_{2t+1}\beta_1 + \beta_1\beta_2) \text{ mod } k$.

β_1 is known to party A , β_2 is known to party B .

Proposed Algorithm

- Step 1: Party A finds $2t + 1$ shares of p . Gives $p_1, p_2, p_3, \dots, p_t$ to party B .
- Step 2: Party B finds $2t + 1$ shares of q . Gives $q_{t+1}, q_{t+2}, q_{t+3}, \dots, q_{2t}$ to party A .
- Step 3: Each of the parties multiplies $p_i q_i$ and adds b_i to it. A does it for $i = 1$ to t and B does it for $i = t + 1$ to $2t$.
- Step 4: Party A adds β_1 to p_{2t+1} and gives it to B with $p_i q_i + b_i$ where $i = 1$ to t .
- Step 5: Party B adds β_2 to q_{2t+1} and gives it to A with $p_i q_i + b_i$ where $i = t + 1$ to $2t$.
- Step 6: Each party now locally computes a polynomial $f(\cdot)$ which has values $p_i q_i + b_i$ for $i = 1$ to $2t$ and $(p_{2t+1} + \beta_1)(q_{2t+1} + \beta_2)$ for $i = 2t + 1$. The generated product $N' = f(0) \text{ mod } k$.
- Step 7: N' is now checked by both the parties to verify that it could be generated by their shares (p or q). If so, they have found a pair β_1 and β_2 such that $(p_{2t+1}\beta_2 + q_{2t+1}\beta_1 + \beta_1\beta_2) = b_{2t+1}$ where b_i interpolate to $f_z(\cdot)$ such that $f_z(0) \text{ mod } k = 0$.

- Step 8: If N' is not passed by both the parties then $b_1, b_2, b_3, \dots, b_{2t+1}$ shares are re-chosen by fine-tuning by the difference between N' and the expected N .

It is to be noted that $f_z(\cdot)$ is not known to any party as no party knows b_{2t+1} and more than half of $b_1, b_2, b_3, \dots, b_{2t}$ thus providing security. Once a set of $b_1, b_2, b_3, \dots, b_{2t}$ and b_1, b_2 are found they could be used with any values of p and q provided $p_{2t} + 1, q_{2t} + 1$ be the same.

This is required as $b_{2t+1} = (p_{2t+1}\beta_2 + q_{2t+1}\beta_1 + \beta_1\beta_2) \text{ mod } k$ — so no constituents should change for b_{2t+1} . But $p_1, p_2, p_3, \dots, p_{2t}$ can be changed to produce a new p and similarly for q . Step 8 of the algorithm could also be done by a trusted party who provides $b_1, b_2, b_3, \dots, b_{2t}, \beta_1, \beta_2, p_{2t+1}$ and q_{2t+1} to the respective parties. If so the algorithm is finished in $O(t^2)$ time.

B. Authentication phase

In the second phase, both parties A and B confirm each other's knowledge before proceeding to use N as the session key. As in the RSA scheme, encryption can be done using p ; decryption can be done using q and vice-versa. For confirmation, each communicates to other by RSA like encryption and decryption with randomly chosen number (say R_A or R_B) and time stamp (say T_A or T_B). The steps for authentication are as follows.

Party A	Party B
Step 1: $E_p(R_A, T_A) \rightarrow$	$D_q(E_p(R_A, T_A)) = R_A, T_A$
Step 2: $R_{A-1}, T_B, R_B = D_p(E_q(R_{A-1}, T_B, R_B))$	$E_q(R_{A-1}, T_B, R_B)$
\leftarrow	
Step 3: $E_p(R_{B-1}, T_A) \rightarrow$	$D_q(E_p(R_{B-1}, T_A)) = R_{B-1}, T_A$

Where E_p and E_q mean encryption using key p and q , D_p and D_q mean decryption using key p and q respectively. R means a random number generated by party whose name is the subscript of R . T stands for time-stamp to protect from replay attack. The arrows show the direction in which data is sent.

The above is a standard method and any other standard and secure method for authentication could as well be used.

IV. ANALYSIS OF THE PROPOSED SCHEME

THE proposed methodology is a key exchange scheme supported by an authentication provision. The scheme generates a session key for securing a subsequent authenticated session between two parties, and does not provide any extra information regarding the individual steps. It also fulfils the much desired feature of a strongly secure key exchange scheme, that is to have minimum persistent recorded data, or in other words, the scheme does not generate any persistent data, which have to be distributed and securely stored. The scheme is secured against factorization attacks, as both p and q are unknown to any third party.

A detailed analysis of the scheme against various possible attacks follows in the subsections.

A. Dictionary Attack

All password-based schemes proposed in literature are vulnerable to *dictionary attack* if any opportunity is given. So, the primary job of the designer is to remove such opportunities of attack. Dictionary attacks can be *online* as well as *off-line*. In case of the proposed scheme, p, q and N are unknown to any third party so the online dictionary attack is not applicable to this scheme.

However, offline dictionary attack is a more complex and it needs to be handled with care. One can make this attack by posing as a legitimate party to gather information, or by one who monitors the messages between two parties during a legitimate valid exchange. Even a little information leakage during an exchange can be exploited.

To minimize the information leakage this subsection describes the necessary precautions in selecting the shares of numbers.

Selection criteria for shares of p and q

Given

$$p_i = f_p(0) + b = f_p(i)$$

$$q_j = f_q(0) + a = f_q(j)$$

Now

$$(f_p(0) + b)(f_q(0) + a) \text{ mod } k = (f_p(0)f_q(0) + af_p(0) + bf_q(0) + ab) \text{ mod } k \\ = pq \text{ mod } k$$

if $(af_p(0) + bf_q(0) + ab)$ is a multiple of k .

As k is known beforehand, so it will be less complex to find p or q and N . To avoid such a situation a proper selection of shares of p and q is necessary. Shares of p , $f_p(i) = f_p(0) + b$ should be generated in such a way that $f_p(0)$ and b should have a factor c , where c is small number relatively prime to k . Similar precautions must be taken for the generation of shares of q by the other party.

B. Stolen Session Key Attack

In this type of attack, a stolen session key (p, N) (or (q, N)) is used to find the other prime number q (or p). Subsequently, one tries to find polynomials that were used for the session key generation. As per Theorem 2, the proposed scheme is guarded against this attack. Also as the system is fast and easy, the session key can be changed after small intervals to avoid stolen key attacks.

C. Verification Stage Attack

The verification stage is where both parties prove to each other the knowledge of keys. As p , q , and N is cryptographically large, the second stage is presumed to be immune to any brute-force attack.

V. APPLICATIONS OF THE PROPOSED SCHEME

THE proposed scheme is broadly useful for any applications where prolonged key storage is risky or impractical, and where the communication channel may be insecure. Some of its common applications are: user-to-user applications, disk-less workstations, bootstrapping new system installation, cellular phones or other key-pad systems etc.

From the economic point of view as well as to encounter the stolen key problems, these authentication schemes are always preferable than the smart cards.

VI. CONCLUSIONS

IN this paper, a new and efficient asymmetric key exchange scheme with authentication has been designed based on exchange of shares of numbers. The scheme is more robust as compared to usual symmetric authenticated key exchange schemes, which are vulnerable to on-line and off-line dictionary attacks and stolen key attacks.

REFERENCES

- [1] B Schneier, "Applied Cryptography", Second Edition, John Wiley & Sons, 1996.
- [2] D P Jablon, "Strong Password-Only Authenticated Key Exchange", ACM SIGCOMM, Computer Comm. Review, pp 5-26, 1997.
- [3] S M Bellovin et al, "Encrypted Key Exchange: Password-based Protocols Secure Against Dictionary Attack", Proc. of the IEEE Symp. on Research in Security & Privacy, Oakland, May 1992.
- [4] W Diffie et al., "Authentication and Authenticated Key Exchanges", Design Codes and Cryptography, 2, 107-125, 1992.
- [5] W Diffie and M E Hellman, "Privacy and Authentication: An Introduction to Cryptography", Proc. of the IEEE, vol 67, no 3, pp. 397-427, Mar 1992.
- [6] B Jaspán, "Dual-Workfactor Encrypted Key Exchange: Efficiently Preventing Password Chaining and Dictionary Attacks", Proceedings of the Sixth Annual USENIX Security Conference, pp 43-50, July 1996.
- [7] C Ellison, "Establishing Identity Without Certification Authorities", Proc. of the Sixth Annual USENIX Security Symposium, San Jose, pp 67-76, July 1996.
- [8] M Steiner, et al., "Refinement and Extension of Encrypted Key Exchange", Operating Systems Review, vol 29, no 3, pp 22-30, July 1995.
- [9] R Rivest, A Shamir and L Adleman, "On Digital Signatures and Public-Key Cryptosystems", MIT laboratory for Computer Science, Technical Report, MIT/LCS/TR-212, Jan 1979.
- [10] A Shamir, "How to Share a Secret", Communications of the ACM, Vol. No. 22, pp. 612-613, 1979.
- [11] Y Desmedt and Y Frankel, "Threshold Cryptosystems", Proc. CRYPTO 89, pp. 307-315, Springer-verlag, 1990, LNCS 435.