International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 1– No.4, February 2012 – www.ijais.org

Analysis of Key Exchange Protocols using Session Keys

Pranav J Vyas
Smt. Chandaben Mohanbhai Patel Institute of Computer Applications,
Charusat, Changa. Anand, India

Bhushan H Trivedi
GLS Institute of Computer Technology, Ahmedabad-06

ABSTRACT
Security of information flowing through insecure network is becoming complicated with advent of internet and its usage. Encrypting information is one way of securing it from unauthorized access. This paper analyze techniques of exchanging key through which encryption is performed. We review the techniques on various parameters and find which technique is best suitable for use in mobile computers with limited processing power and battery capacity while efficiently working on wireless networks.

General Terms
Session Key, Key Exchange Protocols

Keywords
Session Key, Key Exchange Protocols, Mutual Authentication, User Anonymity, Forward Security.

1. INTRODUCTION
As the technological advancements are made by mankind usage of various types of computer network is increasing. While communicating over insecure networks such as Internet increases achievement of secrecy, anonymity, and authentication are becoming more and more important requirements in order to avoid unauthorized access to information. When a client wants to get a service from server whatever messages that are exchanged between them should be secret. In order to maintain this secrecy schemes are built with session key and are protected so that their following messages can maintain this secrecy.

In our paper we do review of popular session key exchange algorithm. We find out how they work. We compare these algorithms on various criteria to check their strength and weaknesses. In the end we conclude which algorithm is most suitable for key exchange with mobile computers and give reason behind its selection. We also highlight future challenges in applying algorithm with mobile computers and suggest remedial technique.

The paper is divided into five sections. Section 1 is of introduction of the paper and topic. Section 2 describes protocols and their working. Section 3 consists of justification of criteria that we choose. In section 4 we compare the protocols in detail and also present a tabular summary of main points. Section 5 concludes paper giving summery of work done and outcome that we deduced at the end of reviewing protocols.

2. PROTOCOL DESCRIPTION
Here we provide brief details on working of various protocols. Following is table of symbols that we have used.

Table1. Description of symbols used in describing protocols.

Symbol	Description
A	Alice's name
B	Bob's name
*E*A	Encryption with a key Trent shares with Alice
*E*B	Encryption with a key Trent shares with Bob
I	Index number
K	A random session key
L	Lifetime
*T*A,*T*B, *T*S	A timestamp
*R*A,*R*B	A nonce, chosen by Alice and Bob respectively

2.1 Needhlam-Shroeder protocol
This protocol, invented by Roger Needham and Michael Schroeder [1], also uses symmetric cryptography and Trent. Working of protocol is described in following steps:

(1). Alice sends a message to Trent consisting of her name, Bob's name, and a random number. *A*,*B*,*R*A
(2). Trent generates a random session key. He encrypts a message consisting of a random session key and Alice's name with the secret key he shares with Bob. Then he encrypts Alice's random value, Bob's name, the key, and the encrypted message with the secret key he shares with Alice. Finally, he sends her the encrypted message: *E*A(*R*A,*B*,*K*,*E*B(*K*,*A*))
(3). Alice decrypts the message and extracts *K*. She confirms that RA is the same value that she sent Trent in step (1). Then she sends Bob the message that Trent encrypted in his key. *E*B(*K*,*A*)
(4). Bob decrypts the message and extracts *K*. He then generates another random value, *R*B. He encrypts the message with *K* and sends it to Alice. *E*K(*R*B)
(5). Alice decrypts the message with *K*. She generates *R*B - 1 and encrypts it with *K*. Then she sends the message back to Bob. *E*K(*R*B - 1)
(6). Bob decrypts the message with *K* and verifies that it is *R*B - 1.

2.2 Diffie-Hellman protocol
The following protocol is given in [2]. It is due to Burrows. Working of protocol is described in following steps:

(1) Alice generates a session key *K*. Alice sends message to Trent consisting of her name and message encrypted with *E*A which consist of timestamp *T*A, Name of B and session key *K*. *A, E*A (*T*A, *B, K)*

(2) Trent checks *T*A if it finds it valid it forwards *K* to *B*. *B* checks the *T*A in message and compare it with any other message it received from A. *E*B(*T*B, *A, K)*

2.3 Oatway-Ress protocol

This protocol is designed for use on insecure networks. It allows individuals communicating over such a network to prove their identity to each other while also preventing eavesdropping or replay attacks and allowing for the detection of modification. This protocol uses symmetric cryptography [3]. Working of protocol is described in following steps:

(1) Alice generates a message consisting of an index number, her name, Bob's name, and a random number, all encrypted in the key she shares with Trent. She sends this message to Bob along with the index number, her name, and his name: *I,A,B,EA(RA,I,A,B)*
(2) Bob generates a message consisting of a new random number, the index number, Alice's name, and Bob's name, all encrypted in the key he shares with Trent. He sends it to Trent, along with Alice's encrypted message, the index number, her name, and his name: *I,A,B,E*A(*R*A,I,A,B),*E*B(*R*B,I,A,B)
(3) Trent generates a random session key. Then he creates two messages. One is Alice's random number and the session key, encrypted in the key he shares with Alice. The other is Bob's random number and the session key, encrypted in the key he shares with Bob. He sends these two messages, along with the index number, to Bob: *I,E*A(*R*A,*K*),*E*B(*R*B,K)
(4) Bob sends Alice the message encrypted in her key, along with the index number: *I,E*A(*R*A,K)
(5) Alice decrypts the message to recover her key and random number. She then confirms that both have not changed in the protocol.

2.4 Yahalom

In this protocol, both Alice and Bob share a secret key with Trent [2]. Working of protocol is described in following steps:

(1) Alice concatenates her name and a random number, and sends it to Bob. *A,R*A
(2)Bob concatenates Alice's name, Alice's random number, his own random number, and encrypts it with the key he shares with Trent. He sends this to Trent, along with his name. *B,E*B(*A,R*A,RB)
(3)Trent generates two messages. The first consists of Bob's name, a random session key, Alice's random number, and Bob's random number, all encrypted with the key he shares with Alice. The second consists of Alice's name and the random session key, encrypted with the key he shares with Bob. He sends both messages to Alice. *E*A(*B,K,R*A,*R*B),*E*B(*A,K).*
(4)Alice decrypts the first message, extracts *K,* and confirms that *R*A has the same value as it did in step
(1). Alice sends Bob two messages. The first is the message received from Trent, encrypted with Bob's key. The second is *R*B, encrypted with the session key. *E*B(*A,K*),*E*K(*R*B)
(5)Bob decrypts the message encrypted with his key, extracts *K,* and confirms that *R*B has the same value as it did in step (2).

3. CRITERIA

There are many criteria on which key exchange protocols can be compared. In this paper we have compared protocols on following criteria: (1) Vulnerability to attacks (2) Variants available (3) Usage of nonce (4) Mutual Authentication (5) User Anonymity. Following is the reason of choosing particular criteria.

3.1. Vulnerability

Vulnerability is one of the criteria on which session key exchange protocols are evaluated. This criterion helps in determining loop holes which can be exploited by attackers to gain control over communication session by obtaining session key. Usually vulnerability of a protocol is found out by using it in various situations and analyzing its response. Examples of vulnerabilities are reply attacks in Needham-Schroeder [4] [5], impersonation in Otway-Reese [6] [7] [8] [9] [10] etc. Analysis of vulnerability helps protocol designers in determining exact behavior that causes protocol to fail to provide secure communication between two parties. This analysis also gives insights on how to prevent these exploits in future.

3.2. Variants

Once vulnerability is found in protocol and analyzed, protocol designers address this vulnerability by introducing new technique or feature or extra bit of information that prevents attacker from applying known exploits on this protocol and making protocol secure in turn. Variants of Needhlam-Schroeder described by [11] are well known. Yahalom protocol is also a variant described by Paulson [12]. These variants are not proof from vulnerability and needs to be checked as original protocols to discover vulnerability if there are any.

3.3. Usage of nonce

A nonce is an identification of party involved in communication. Nonce can be a number or any random string. When Alice send a nonce encrypting it with public key of Trent and Trent replies with same nonce encrypted inside Alice's private key it confirms to Alice that Trent is actually trusted party and there is no impersonator involved. This is important in establishing identity when two un-trusted parties are about to communicate and want to make sure that they are communicating with each other and there is no impersonator involved. Protocols such as Needhlam-Schroeder, Otway-Reese and Yahalom make use of this technique for mutually authenticating parties involved in communication.

3.4. Mutual Authentication

Mutual authentication is very important property when communicating on insecure network channels. Network such as Internet where impersonation is easily achieved mutual authentication technique is used to make sure both hosts identify each other before communication can take place. There are many techniques of mutually authenticating such as EAP-IKEv213, pseudonym identity [14], trusted third parties and nonce. Protocols such as Needham-Schroeder and Yahalom provides mutual authentication.

3.5. User Anonymity

User anonymity refers to be able to communicate without revealing one's identity. Revealing identity on insecure networks such as Internet can be risky as it can be tracked on Internet. Sharing a session key without revealing host's identity is an important property of session key exchange protocols. According to [16] anonymity is said to be achieved when an adversary who are not in possession of secret key cannot learn the identity of signer of signature of secret key. Shoup[Sho99] defines anonymity in the context of the simulation framework for key exchange security, as opposed to the indistinguishability framework of, for example, Canetti-Krawczyk [18], which has now become more commonplace for analyzing key agreement protocols.

There are two forms of anonymity according to [GOL11]. There are protocols such as [20], [21], [22], [23] which aim to provide identity hiding where identity of one party remains hidden. However identity becomes available by the end of protocol to peer. There are protocols that have been suggested by [24], [25] that overcome this problem.

There are also protocols such as [17], [26], [27], [28], [29], [30] aim to give anonymity where even peer of party does not learn its long term identity. This property is very important for practical applications such as TOR [31].

4. POTOCOL COMPARISION

In this section we compare protocols on the above mentioned criteria. We check if the protocol meets criteria selected or not. We also discuss the reasons behind protocol not being able to meet criteria. We also try to provide solution where it can be improved to meet criteria.

4.1. Needham-Schroeder

This protocol was invented by Roger Needham and Michael Schroeder [1]; it uses symmetric cryptography and trusted third party Trent.

Since this protocol uses trusted third party Trent to exchange keys unless the third party isn't compromised it is very difficult for Mallory to get session key. So this protocol is secure to be used in unsecured network environments. This protocol is vulnerable to reply attacks but variants are proposed that patches this loophole with use of timestamps. Nonce is used in this protocol for mutual authentication and to determine freshness of message.

This protocol provides mutual authentication with usage of random values which involved parties' exchange. In this protocol Alice sends random value *RA* to Trent to which Trent replies by encrypting *RA* inside *EA* which is secret key of Alice which only Trent knows. If the correct *RA* is received by Alice that determines and authenticates ID of Trent to Alice. Bob and Alice can also mutually authenticate each other. Bob sends random value *RB* to Alice encrypting it with *K***.** If Alice replies with correct *RB-1* value Bob can determine that Alice is genuine party. Alice can also authenticate bob when she gets reply of her earlier message *EB (K, A)*.

User anonymity is not supported in this protocol as both Alice and Bob knows identity of each other.

In this protocol's implementation if the encryption algorithm is strong and keys are random then there is little possibility that key will be compromised. If the Mallory gets access to old *K* then he can launch successful attack as demonstrated by [5].If we assume that Mallory somehow gets access to *K*. He can now message to Bob pretending to be Alice by sending message *EB (K, A)***.** Bob will be led to believe that Alice has initiated new conversation and will send packet *K (RB)* to which Mallory intercepts message and reply with *K (RB-1)*. Now, Mallory can successfully impersonate as Alice.

According to [5] problem can be solved with use of timestamps at proper places while sending packet. Timestamps can be included in *E*A (*R*A, *B, K, T, EB* (*K, A, T*)) where *T* is time on Trent. It can be verified if the message is reply or not by *TI-T < TA+TB***.** However, this solution is not suitable for all type of connection e.g. terminals which doesn't maintain local clocks.

There are several variants available of this protocol which addresses problem of freshness of session keys as demonstrated by [32][5][33]. Original implementation of protocol does not use timestamps but subsequent variants uses timestamps to address key freshness issue [32][5].

According to [5] if the timestamps are used in step (2) and (3) of original algorithm then problem of reply attacks can be solved and freshness of key can be maintained. Trent replies back to Alice with message *E*A (*R*A, *B, K, TA, EB* (*K, A*)). After this when Bob receives message *E*B (*K, A, TB*). Now both Alice and Bob can check that their messages are not reply by comparing timestamps *T*.

4.2. Diffie-Hellman Protocol

Diffie-Hellman method of key exchange is one of the first methods of key exchange. It was proposed by Whitfield Diffie and Martin Hellman in 1976. It is an anonymous key agreement protocol and does not use Trent as trusted third party to exchange key. It belongs to public key cryptography family of protocols. In this method Alice and Bob uses a mathematical formula consisting of prime number and base numbers to derive a common key which is used as session key for that communication session.

Diffie-Hellman protocol does not provide mutual authentication as there is no way to guarantee identity of either party during protocol execution. However, there are variations of the protocol that are implemented in various systems which provide mutual authentication [34] [35][36].

DIffie-Hellman protocol provides user anonymity due to its design. Diffie-Hellman protocol is designed with a mathematical model which makes use of discreet logarithm problem. User does not need to provide any identification when discussing secret key agreement.

In Diffie-Hellman algorithm security of session key is dependent on how well Alice and Bob choose G and g. The attacker Mallory has to computer g^{ab}. Computing g^{ab} is very difficult provided there is no efficient algorithm to solve discreet logarithm problem.

Diffie-Hellman protocol is vulnerable man in the middle attacks. Station-To-Station protocol is variation of Diffie-Hellman protocol which prevents man in the middle attack[37]. STS protocol let Alice and Bob sign their message for each other. Here, mutual authentication is possible when Bob sends Alice his identification *E*k(*S*B(*x,y*)) where *S*B(*x,y*) carries authentication code of Bob. Alice can also send her identification to Bob in subsequent message *E*k(*S*A(*x,y*)) where *S*A(*x,y*) carries Alice's Identification.

4.3. Ottway - Rees Protocol

This protocol allows individuals communicating over insecure networks to exchange secure session key. Alice and Bob can achieve mutual authentication in this protocol with help of trusted third party Trent. Random numbers are used as nonce which is checked when a message is received by either Alice or Bob to verify if they are the same or not. If they are same then both trust identity of the other. When Trent replies to Bob with messages *I,E*A(*R*A,*K*), *E*B(*R*B,K) both Alice and Bob can view only message which is encrypted with their respective private keys if they find the same random number *RA* and *RB* respectively they can confirm identity of each other based on their trust relationship with Trent**.**

User anonymity is not present in this protocol as mutual authentication requires each party to identify itself before key is exchanged.

According to [6] when Alice wants to establish connection with Bob, Mallory can exploit this situation by impersonating as Bob. Mallory can intercept initial message *E*A (*R*A, *K*) from Alice to Bob and sends this message back to Alice. Alice will treat this message as from Bob with session key information. [7] Shows one more attack which shows responsibility of server. In this paper Boyed and Mao shows that Trent should not only check values of I, A, B in message but also should compare values of in other message and should check them to see if they are matched with values sent in plain text. At the end of the attack Alice will think that she has shared secret key *K* with Bob but its Mallory who is impersonating as Bob with whom Alice shared secret key *K*. Boyed and Mao also points out that if Cipher Block Chaining Algorithm is used to encrypt two former messages then attacker can user One Time Pad algorithm to find out secret key from Bob by impersonating as Alice.

In their paper [9] have found out two attacks one of which is against original version and other is against modified version. The first attack against original version works under assumption that trusted host Trent is not able to tell if the message sent is different or same. In this attack Mallory intercepts the message that Bob sends to Trent. Mallory then sends the same message to Trent impersonating as Bob. According to assumption Trent is not able to differentiate between two messages now Bob and Alice both will have different secret keys Alice will have key *K* and Bob will have key *K*`.

This protocol has a modified version proposed by [38] which counters attacks by [7]. Authors of paper [9] propose attack on this modified protocol. This attack works under assumption that the trusted third party Trent allows multiple parallel connections between Alice and Bob. In this attack Mallory intercepts message that Bob is sending to Trent. Mallory then sends this message twice so that Trent will think that Bob and Alice want to have established two parallel sessions and will give two sets of messages to Bob. These two sets of messages will be again intercepted by Mallory and will be used in construction of message where Alice and Bob will get two different keys.

4.4. Yahalom

This protocol was developed by Burrows, Abadi, and Needham. This protocol also uses trusted third party Trent to exchange key. Here, Alice can authenticate identity of Bob when she gets reply from Trent which is encrypted by her own secret key and when opened, reply contains her secret no *RA***.** Bob can also authenticate when he receives message *E*B (*A, K*), *EK* (*R*B). Bob can also authenticate identity of Alice since Bob's random number is encrypted by secret key *EK* which is again encrypted with Bobs secret key *EB* which in turn contain name of Alice *A***.** Only Trent and Bob know about this secret key.

This protocol does not support user anonymity as during process of mutual authentication both parties ascertain each other's identity.

According to [12] session key in this protocol is secure since *RB* is kept secret. Other then Bob himself *RB* is only known by Trent and Alice. Only Alice could have formed *EK* (*R*B**)**. This way Alice can associate *K* with fresh random nonce. Bob also learns that A has been active recently. According to Paulson, assumes that *K* is secret and so he goes on to assume that *RB* is also secret is faulty. In step 2 of protocol *RB* is sent to Trent as plain text. Paulson goes on to show that the original version of protocol can be attacked if Mallory can get an old *RB*. If Mallory is successful in his attempt he will easily be able to decrypt *E*B (*A, K*). Then, He will be able to easily extract secret key *K* from message and impersonate as Alice from step 4.

This protocol is different because here Bob first contacts trusted third party Trent. Trent then sends one and only one message to Alice [39].

5. CONCLUSION

In this paper we review key exchange protocols which do not make use of timestamps during communication. We learned that it is possible to maintain freshness key even without using timestamps. We also reviewed protocols for criteria such as mutual authentication, user anonymity and security of session key.

During our review we found out that mutual authentication and user anonymity are two mutually exclusive conditions. It is not possible to achieve both of them in same implementation of a protocol. Mutual authentication is an important on insecure networks such as internet when exchanging information. But then mutual authentication also carries extra information that is used to identify hosts. So, mutual authentication carries some overhead.

User anonymity also has its pits and falls. User anonymity is useful on secure private network where each and every host trust each other. For example, a private network of cell phone service provider. Protocols providing user anonymity do not carry extra information so they are little faster as compared to protocols providing mutual authentication. On one hand where user anonymity protects privacy of users by hiding his/her identity from their peers and on other it carries no guarantee that host is genuine.

Selecting a protocol category that fits best for a mobile computer to exchange key is a trade off. The selection is dependent on the network environment in which the mobile computer will be operated. If the computer is to be operated on in largely insecure public network such as Wi-FI hot spots then it is necessary to have mutual authentication feature in protocol. However if the computer is to be used for communication over internet via mobile service operator it is suggested to make use of protocol that provides user anonymity as the communication between the mobile service provider is between two trusted parties thus removing overhead and making communication faster.

Based on analysis of existing protocols discussed in this paper, we can derive that no one protocol has features required to be operated on mobile computers. We require a protocol that makes use of less power in terms of processing and battery and can combine strength of all protocol combine.

Table2. Protocol Comparison

Criteria/ Protocol	Vulnerability	Variants	Nonce	Mutual Authentication	User Anonymity
Needham-Schroeder	Reply Attack	Yes	Yes	Yes	No
Diffie Hellman	Keys are vulnerable to a small subgroup attack	Yes	No	No	Yes
Otway Rees	Impersonation	Yes	Yes	Yes	No
Yahalom	Reply Attack	Yes	Yes	Yes	No

6. REFERENCES

[1] R.M. Needham and M.D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," Communications of the ACM, v. 21, n. 12, Dec 1978, pp. 993-999.

[2]Michael Burrows, Martin Abadi, and Roger Needham. A Logic of Authentication. Technical Report 39, Digital Systems Research Center, February 1989.

[3]D. Otway and O. Rees, "Efficient and Timely Mutual Authentication," Operating Systems Review, v. 21, n. 1, 1987, pp. 8-10.

[4] Gavin Lowe. An attack on the Needham-Schroeder public key authentication protocol. Information Processing Letters, 56(3):131--136, November 1995.

[5]D.E. Denning and G.M. Sacco, "Timestamps in Key Distribution Protocols," Communications of the ACM, v. 24, n. 8, Aug 1981, pp. 533-536.

[6]Ding Yi-Qiang. The formal analysis of cryptographic protocols. Ph.D. Dissertation, Institute of Software, The Chinese Academy of Science. June 1999.

[7] Colin Boyed, Wenbo Mao. On a Limitation of BAN Logic. Adances in Cryptoology-Eurocrypt'93, Lecture Notes in Computer Science 765, Tor Helleseth(Ed.), pp. 240-247, May 1993.

[8]Wenbo Mao. An Augmentation of BAN-Like Logics, The Eighth IEEE Computer Security Foundations Workshop (CSFW '95), March 13-15, 1995, Kenmare, County Kerry, Ireland 1995

[9]Guilin Wang, Sihan Quing, Two new attacks against Otway Rees Protocol IFIP/SEC2000, Information Security.

16th World Computer Congress 2000, August 21-25, Beijing, China. Beijing: International Academic Publishers, 2000. 137-139

[10]John Clark and Jeremy Jacob. A survey of authentication protocol literature : Version 1.0., November 1997.

[11]Gavin Lowe. A family of attacks upon authentication protocols. Technical Report 1997/5., Department of Mathematics and Computer Science, University of Leicester, 1997.

[12]Lawrence C. Paulson. Relations between secrets: Two formal analyses of the yahalom protocol. Computer Security, 2001.

[13] The Extensible Authentication Protocol-Internet Key Exchange Protocol version 2 (EAP-IKEv2) Method.

[14]Yixin Jiang, Chuang Lin, Xuemin (Sherman) Shen, Mutual Authentication and Key Exchange Protocols for Roaming Services in Wireless Mobile Networks, IEEE transactions on wireless communication vol.5 No. 9.

[15]Markus Jakobsson and David Pointcheval Mutual Authentication for Low-Power Mobile Devices Syverson (Ed.): FC 2001, LNCS 2339, pp. 178–195, 2002.(Springer-Verlag Berlin Heidelberg) 2002

[16]Jesse walker, Jiangtao Li, Key Exchange with Anonymous Authentication using DAA-SIGMA Protocol In IACR eprint archive, 2010

[17]Victor Shoup, On Formal Model for Secure key exchange, IBM Research Report RZ3120, 1999

[18]Ran Canetti, Hugo Krawczyk, Analysi fo Key-Exchange Protocols and their use for building secure channels, Advances in Cryptology- EUROCRYPT'01, Vol2045 LNCS, PP. 453-474. Springer 2001

[19]Ian Goldberg, Douglas Stebila, and Berkant Ustaoglu, Anonymity and one-way authentication in key exchange protocols Technical Report CACR 2010-11, University of Waterloo Centre for Applied Crypto- graphic Research, 2011

[20]William Aiello, Steven M. Bellovin, Matt Blaze, Ran Canetti, John Ioannidis, Angelos D. Keromytis, and Omer Reingold. Just Fast Keying: Key agreement in a hostile et. ACM Transactions on Information and System Security, 7(2):1 May 2004.

[21] Zhaohui Cheng, Liqun Chen, Richard Comley, and Qiang Tang. Identity-based key agreement with unilateral identity privacy using pairings. Proc. Information Security Practice and Experience (IS- PEC) 2006, LNCS, volume 3903, pp. 202,213. Springer, 2006.

[22]Hung-Yu Chien. ID-based key agreement with anonymity for ad hoc networks. In Tei-Wei Huo, Edwin Sha, Minyi Guo, Laurence Yang, and Zili Shao, editors, Proc. Embedded and Ubiquitous Computing (EUC) 2007, LNCS, volume 4808, pp. 333,345. Springer, 2007.

[23]Hugo Krawczyk. SIGMA: The `SIGn-and-MAc' approach to authenticated Diffie-Hellman and its use in the IKE protocols. Advances in Cryptology CRYPTO 2003, LNCS, volume 2729, pp. 400,425. Springer, 2003.

[24]Ran Canetti and Hugo Krawczyk. Security analysis of IKE's signature based key-exchange protocol. In Moti Yung, editor, Advances in Cryptology Proc. CRYPTO 2002, LNCS,protocol. volume 2442, pp. 27-52. Springer, 2002.

[25]Alfred J. Menezes and Berkant Ustaoglu. Comparing the pre- and post-specified peer models for key agreement. International Journal of Applied Cryptography, 1(3):236-250, 2009.

[26] Ian Goldberg. On the security of the Tor authentication protocol. In George Danezis and Philippe Golle, editors, Privacy Enhancing Technologies (PET) 2006, LNCS, 4258, pp. 316-331. Springer, 2006.

[27]Lasse verlier and Paul Syverson. Improving efficiency and simplicity of tor circuit establishmentand hidden services. In Privacy Enhancing Technologies, LNCS, volume 4776, pp. 134{152. Springer, 2007.

[28]Aniket Kate, Greg M. Zaverucha, and Ian Goldberg. Pairing-based onion routing with improved forward secrecy. ACM Transactions on Information and System Security, 13(4):29, 2010.

[29]Sk. Md. Mizanur Rahman, Atsuo Inomata, Takeshi Okamoto, Masahiro Mambo, and Eiji Okamoto. Anonymous secure communication in wireless mobile ad-hoc networks. In Frank Stajano, Hyoung Joong Kim, Jong-Suk Chae, and Seong-Dong Kim, editors, Proc. International Converence on Ubiquitous Convergence Technology (ICUCT) 2006, LNCS, volume 4412, pp. 140,149. Springer, 2007.

[30]Sherman S. M. Chow and Kim-Kwang Raymond Choo. Strongly-secure identity-based key agreement and anonymous extension. In Juan Garay, Arjen Lenstra, Masahiro Mambo, and Ren e Peralta, editors, Proc. 10th International Conference on Informa- tion Security Conference (ISC) 2007, LNCS, volume 4779, pp. 203,220. Springer, 2007.

[31]Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In Proc. 13th USENIX Security Symposium. The USENIX Association, 2004.

[32]D.E. Denning, Cryptography and Data Security (Addison-Wesley, 1982).

[33]R.M. Needham and M.D. Schroeder, "Authentication Revisited, "Operating Systems Review, v. 21, n. 1, 1987, p. 7.

[34]Chiu-Hsiung Liao, Hon-Chan Chen, Ching-Te Wang, An Exquisite Mutual Authentication Scheme with Key Agreement Using Smart Card, Informatica Vol. 33 (2009) 125–132

[35]Arpa network protocol notes URL: tools.ietf.org/html/rfc46

[36]Blake-Wilson, S.; Menezes, A. , Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol, Public Key Cryptography, Lecture Notes in Computer Science, vol. 1560, Springer, pp. 154–170, 1999

[37]W. Diffie, P.C. van Oorschot, and M.J. Wiener, "Authentication and Authenticated Key Exchanges," Designs, Codes and Cryptography, v. 2, 1992, 107-125.

[38]Ross Anderson and Roger Needham. Robustness principles for public key protocols. In Proceedings of Crypto '95, 1995

[39]Bruce Schneier, Applied Cryptography 2nd Edition, (Willy Publishing Company)