Energy Efficient Recognition Protocol for Ad Hoc Networks

Taresh Singh[1] and S Qamar[2]

[1]Department of Information Technology, College of Engineering Roorkee

Taresh.singh@gmail.com

[2]Professor and Head, department of Information Technology, SRM University, NCR Campus Modinagar (U. p.), India

jsqamar@gmail.com

Abstract

The recognition is a new security principle closely related to authentication. Low-power ad hoc networks with no pre-deployment information require the less authoritative security in recognition. We have studied previously proposed low-power protocols according to the environment and security model presented. We have implemented the New Message Recognition Protocol (NMRP) and Zero Common Knowledge (ZCK) protocol in C and matlab. From our comparison between NMRP and ZCK, we observed that NMRP satisfied the properties of low power environment.

Keywords

Ad hoc Networks, Security, Authentication, Low-Power, Recognition.

1. Introduction

Low-power environments[1], such as sensor networks, present a difficulty in performing traditional security protocols. Sensor motes, as the devices involved in a ad hoc network are called, are intended to be small and cheap. The continued desire to make these motes smaller offsets the technological advancements of increasing computational power in a smaller area. Thus, the extremely low computational power of such devices has severe performance decreases for asymmetric cryptography and exponential mathematics. Moreover, the dynamic network topology and self-organizing properties of this environment prevent any sort of pre-deployment information like shared secrets or network addresses of trusted third parties. New security classification [1] called entity recognition whereby after the first communications with an entity, future recognitions assure that one is communicating with the same entitywithout the guarantee that participants are actively involved. This is closely related to the commonly studied term entity authentication. Using these weaker security requirements allows one to draw security conclusions in low-power environments where, the stronger requirements are unnecessary and often even impossible to achieve.

Some protocols that satisfy the low power environment include the Resurrecting Duckling [3], TESLA [4], μTESLA [5], Guy Fawkes [6], Remote User Authentication [7], Zero Common-Knowledge (ZCK) [8] and New Message Recognition Protocol (NMRP) [2].

Section 1 introduces the low power environment. Target environment model is givenin Section 2. We outline two key disclosure techniques important to recognition protocols, and some

DOI : 10.5121/ijcses.2012.3202

consequences of the target environment in Section 3.Previous work is presented in Section 4.NMRP is given in section 5. Section 6describes our implementation of NMRP and ZCK. Results and comparisons are provided in Section 7and conclusions and a summary of key results are given in Section 8.

2. Adversarial Model

It is assume that an adversary has full control over the connection between Alice and Bob [1]. Thus, Eve controls all the messages sent. Thatmeans Eve can do the following:

1) Read all messages sent by Alice or by Bob.
2) Modify messages, delay messages, or replay messages.
3) Insert messages generated by herself to Alice or Bob or both.

Recognition requires one exception. However at the beginning of the communication between Alice and Bob a trustworthy relay channel needs to be available, similar to that required for the Diffie-Hellman protocol. The initial contact between Alice and Bob then determines the future relationship. Without such a trustworthy initial relay channel, the whole notion of entity recognition does not make sense since Alice and Bob do not actually know who they are communicating with. This modification can be defined as an *adversarial model.*

Thus, in the adversarial model it is assumed that an initial phase where the adversary can read messages but relays them faithfully. Based on the assumptions we do not consider denial of service attacks here. The adversary could just delay all messages. The adversary aims to forge a message, that is, to make Bob accept a message believing that it originated from Alice. Considered protocols are expected to be sound and robust. Hence, if Alice and Bob behave as intended, Bob will accept messages that Alice sent but he will not accept messages that Bob did not send or that were manipulated. Furthermore, all protocols are expected to be recoverable, so if Bob refuses to accept a message, soundness is regained for future messages.

In this model it is assumed that the adversary is not able to compromise Alice or Bob in a sense that he gains knowledge of secret key material. In particular, if the adversary compromises Alice then all relationships of Alice to further entities are compromised. This is not an unusual drawback but a usual property of today’s deployed systems.

3. Basis of Securityin Low Power Environment

Since the computational requirement of asymmetric cryptography causes performance decreases in low-power environments, many low-power protocols use a delayed key disclosure technique to achieve similar security (see Table I). This section briefly examines two common methods for key disclosure used in low-power protocols.

3.1 Time Delayed Key Disclosure

For an authentication or recognition scheme to use any notion of time, the two entities involved must have loosely synchronized clocks.

Table 1: Key Disclosure of Recognition Protocol

Protocol	Key Disclosure		Others
	Time Delay	Interactive	
Resurrecting Duckling			X
TESLA	X		
μTESLA	X		
Guy Fawkes	X		
Remote User Authentication		X	
ZCK		X	
NMRP		X	

The Network Time Protocol (NTP) [10] provides scalable clock synchronization over the wired network of the Internet. The network assumptions that NTP makes are not true in sensor networks [11]. The main issues are the low energy, multi-hop, self-organizing, and dynamic topology properties of the environment. Maintaining the synchronization of clocks requires a secure authentication scheme [12]. Consequently, an authentication (or recognition) scheme on clock synchronization requires another authentication scheme to validate the time.

Moreover, time delayed disclosure requires an upper bound on the message delivery delay to be certain that the message is delivered before the key is disclosed. In multi-hop wireless networks, this message delay could be significant.

3.2 Interactive Key Disclosure

By disclosing the key after an interactive exchange of messages, an algorithm does not need to worry about the issues involved with clock synchronization. However, there are still a couple trade-offs that must be addressed.

Note that this method of key disclosure requires at least three messages between entities *A* and *B*. In the first message, *A* sends the authenticated message to *B*. The second message is an acknowledgement by *B* for receiving the message from *A*. Message three discloses the key to *B*.

The three message exchange could be a problem in an environment with high packet loss. In such a situation, time delayed disclosure may be a better solution.A more important issue is, however, that in order to prevent an adversary *E* from causing *A* to disclose the key too early, the second message must have the data origin authenticated as coming from *B*.

4. Recognition Protocols

In the following, we present some of the protocols in more detail. For this purposes we assume entity *A* is approached by entity *B* to be recognized.

4.1 The Resurrecting Duckling

The Resurrecting Duckling [3], [13] is described as a security policy for low-power environments. We point out that the method of key exchange during the imprinting phase is open to an attack by a passive observer *E*. If *E* observes this key-sharing phase between *A* and *B*, *E* can impersonate *B* to *A* at any point in the future. The protocol attempts to address this by recommending this imprinting phase take place over a secure channel like direct contact. In an ad hoc network, it is unlikely that this requirement could be satisfied.

4.2 TESLA

TESLA[4], [14],a broadcast authentication protocol, sends messages with a MAC keyed according to time intervals. The receiver can verify the message when the key is sent in a future time interval based on a key-disclosure delay. Clock synchronization is negotiated using a digital signature algorithm, like RSA or DSA (Digital Signature Algorithm).

The time delayed key disclosure of TESLA relies on loose, but bounded clock synchronization between the two involved parties. Clock synchronization requires authenticated synchronization messages as described in Section 3.2.1. TESLA attempts to deal with this issue by suggesting the use of digital signatures to authenticate the time response. The computational, bandwidth and memory requirements do not make it a viable solution for a sensor network environment.

4.3 µTESLA

Recognizing the limitations of TESLA in a low-power environment, it wasmodified to address the issues above and named the result μTESLA [5]. Like TESLA, it uses time intervals for disclosing keys,however, clock synchronization is performed in negotiation with a base station. Now, instead of authenticating the clock synchronization messages with a digital signature, μTESLA assumes a master pre-shared secret between the base station and authenticating nodes. This pre-deployment information might not be possible in some deployment scenarios as in our target environment.

4.4 Guy Fawkes

Guy Fawkes protocol [6] uses code words to publish messages and future code words in a hash so that the codeword can be revealed later to prove that you are communicating with the same party. In the original scheme the commitment would be published in a newspaper such that the commitment would be stored in a public directory with a time-stamp and could be verified at any time. However, in an *ad hoc* network, in most cases there is no such central directory that provides time-stamps, so an explicit acknowledgement of the receipt is necessary for the security of the protocol. However, this requires the acknowledgement data to be authenticated as coming from the receiver. A second variant of the Guy Fawkes protocol was presented that fixes this issue. Here, basically both parties publish messages and future code words in a hash that are revealed later on.

The Guy Fawkes scheme requires negligible computations. However, the Guy Fawkes protocol also requires quite some bytes to be exchanged and it is more complex. To clarify, if a pair Alice and Bob only wants to authenticate a single message m_0, they need to perform two iterations of the Guy Fawkes protocol since the key for the authenticated message is opened in the next iteration.

4.5 Remote User Authentication

The Remote User Authentication Protocol [7] uses amessage authentication code (MAC) and requires that users compute a lot ofMAC values. The MAC values are sent over the authenticated channel. Thisis a concern in our setting since the authenticated channels usually have lowbandwidth. Moreover, the amount of computations and communication assumedin this protocol may not be desirable in a pervasive network of devices with lowcomputational power.

4.6. Zero Common-Knowledge

The Zero Common-Knowledge (ZCK) protocol [8]uses the values of a hash chain as keys for a MAC. This protocol was implemented in [1] as a proof-of-concept. The observations from this implementation ensuredthat this protocol is suitable for devices with low computational power, low codespace, low communication bandwidth and low energy resources. It also raised acouple of areas of concern, mainly denial-of-service and memory complexity.

If Alice and Bob want to communicate then they randomly choose a_0 and b_0, respectively. Then, they respectively form hash chains $a_i = h\ (a_{i-1})$ and $b_i = h\ (b_{i-1})$,i = 1… n. Note that for each pair of users wishing to communicate, theremust be a separate pair of hash chains. This means that if a device wants tocommunicate with m users, it has to deal with m different hash chains of lengthn. This is of concern when dealing with small devices in a ad hoc networkwith memory constraints [1].

4.7. New Message Recognition Protocol (NMRP) without use of Hash Chain

NMRP [2] is a new design for message recognition protocols suitable for ad hoc networks and itdoes not make use of hash chains.Hash chaining techniques have been used in recent designs of message recognition protocols. In this approach, the small devices are required to save values of a hash chain in their memories for every single user they want to communicate with. Since they do not use this technique, they no longer require the small devices to save values of a hash chain in their memories. This relaxes the memory requirements. Moreover, the passwords are chosen at random in each session. Hence, they are independent of one another and are being refreshed in each session. This can be done for any arbitrary number of times, so we do not need to fix the total number of times the protocol can be executed which implies a desired flexibility in this regard. As the passwords are independent of one another, we do not need to consider assumptions that depend on the number of sessions the protocol is executed.

5. New Message Recognition Protocol without use of Hash Chain

In this section, we describe the details of NMRP [2]. The internal states of Alice and Bob, initialization phase and execution of the protocol are three phases of this protocol.

5.1 Internal States of Alice and Bob

- x_0,x_1 and y_0,y_1 are the passwords for this session and the next session, respectively.
- $X_0 = H(x_0)$, $X_1 = H(x_1)$ and $Y_0 = H\ (y_0)$,$Y_1 = H\ (y_1)$ are the committing hash values of the passwords.
- $M_0 = H(x_0, X_1) = H(x_0, H(x_1))$, $N_0 = H\ (y_0, Y_1) = H\ (y_0, H\ (y_1))$ are the binding hash value of the passwords.
- y^*_{-1}, Y^*_0, N^*_0 and x^*_{-1}, X^*_0, N^*_0 are Bob's and Alice’s most recent password, committing hash value, and binding hash value accepted by Alice.

5.2 Initialization of Alice and Bob

- Choose random x_0,x_1 and y_0, y_1.
- Compute $X_0 = H((x_0), X_1) = H(x_1)$,$M_0 = H(x_0,X_1)$
and $Y_0 = H\ ((y_0), Y_1) = H(y_1)$, $N_0 = H(y_0,Y_1)$.
- Send X_0, M_0 and Y_0, N_0 from each other over the authenticated channel.
- Receive Y_0, N_0 and X_0, M_0 from each other over the authenticated channel.
- Let y^*_{-1} =NULL, $Y^*_0 =Y_0$, $N^*_0 =N_0$ andx^*_{-1} =NULL, $X^*_0 =X_0$, $M^*_0 =M_0$.

5.3 Execution

At Alice	At Bob
Alice wants to send a message m to Bob. Alice's execution can be described as follows: ▪ Choose a random x_2. ▪ Compute $X_2 = H(x_2)$,	▪ After receiving m^-, h^-, choose a random y_2. ▪ Compute $Y_2 = H\ (y_2)$, $N_1 = H\ (y_1, Y_2)$. ▪ Send y_0, Y_1, N_1 to Alice and

$M_1 = H(x_1, X_2)$, and h = H [m, x_0]. ▪ Send m, h to Bob and wait to receive $y^{-}_{0}, Y^{-}_{1}, N^{-}_{1}$ from Bob. Resend m, h if Bob did not respond. ▪ If $H(y^{-}_{0}) = Y^{*}_{0}$ and $H(y^{-}_{0}, Y^{-}_{1}) = N^{*}_{0}$, then send x_0, X_1, M_1 to Bob and update internal state: $y^{*}_{-1} = y^{-}_{0}$, $Y^{*}_{0} = Y^{-}_{1}$, $N^{*}_{0} = N^{-}_{1}$, $x_0 = x_1$, $x_1 = x_2$, $X_0 = X_1$, $X_1 = X_2$, $M_0 = M_1$. ▪ Otherwise, initiate resynchronization with Bob.	wait to receive $x^{-}_{0}, X^{-}_{1}, M^{-}_{1}$. Resend y_0, Y_1, N_1 to Alice, if Alice did not respond. ▪ If $H(x^{-}_{0}) = X^{*}_{0}$, ▪ $H(x^{-}_{0}, X^{-}_{1}) = M^{*}_{0,}$ and $h^{-}=H[m^{-}, x^{-}]$, then update internal state $x^{*}_{-1} = x^{-}_{0}$, $X^{*}_{0} = X^{-}_{1}$, $M^{*}_{0} = M^{-}_{1}$, $y_0 = y_1$, $y_1 = y_2$, $Y_0 = Y_1$, $Y_1 = Y_2$, $N_0 = N_1$, Output (Alice, m'). ▪ Otherwise, initiate resynchronization with Alice.

6. Implementation

We implemented NMRP (New Message Recognition Protocol)[2]and ZCK[1] protocol in C++ programming language and MATLAB. For the purpose of comparing NMRP and ZCK protocols, it is assumed that it should not effect the relative measurements of protocols (MD5 being the common computation). For a one-way MD5 hash function we assume that each hash element takes two bytes of memory space and one unit time for its calculation.In our implementation we developed a *hash ()* function that is used in both protocol.

In our implementation of NMRP we defined class *Node* with its two objects Alice and Bob. The member variables of node are *curpwd* for current session password, *nextpwd* for next session password, *Comhash1, Comhash2* for committing hash value of the passwords, *Bindhash* for binding hash value etc. the member functions of the class Node are *Hash()*, *Bindinghash()* and *Masghashing()* etc.

We implemented ZCK by declaring a class Node with two objects *Alice* and *Bob*. Member variables of *Node* are *Rand* for random number, *Hashchain[]* for holding hash chain values, *Rechash* for holding received hash value, *Msgdigest* for holding digest of the message etc.. Member functions of *Node* are *Hash ()*, *Msghashing ()*, *Hashchaingen ()* etc.

7. Results

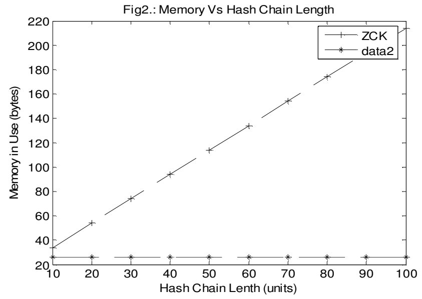
In this section we present results based on our implementation of the NMRPwithout use of Hash chain and ZCK with use of hash chain for ad hoc networks. Fig 1and Fig 2 show an overview of our results. The properties of such environments are determined by Axioms 1 to 4. Hence we relate our results with these axioms.

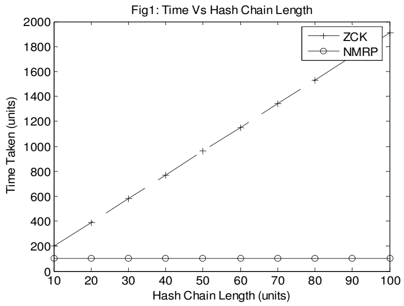
7.1 Axiom 1 (low computational power)

The observations from our implementation ensure that NMRP protocol is more suitable for devices with low computational power. As the length of hash chain increases the time taken by ZCK protocol increases. In New Message Recognition Protocol, the running time remains constant.

7.2 Axiom 2 (low code space)

In NMRP the memory usage remains constant even if the node in communication with several different nodes. In ZCK nodes in communication with several different nodes, it has to manage several hash chain for each node separately becauseit uses different hash chain for each different node. Obviously, a node must keep space for multiple hash chains since a separate one is needed for each communication partner.The observations from our implementation ensure that NMRP protocol is more suitable for devices with low memory space.





7.3 Axiom 3 (low communication bandwidth)

In NMRP recognition phase has three messages transmissions similar to ZCK. Thus the total messages transmission overhead for each message would be equal to ZCK.

7.4 Axiom 4 (low energy resources)

We validated that NMRP implementation fits to Axiom 1,2 and 3. Since the energy consumption of a protocol is composed of the computational effort and the data transmission, one can infer that NMRP requires less energy overhead than ZCK protocol.

7.5 Remarks and Observations

7.5.1Security

The NMRP satisfies the above presented adversarial model. In particular, the scheme is secure if there is a reliable relay channel available for sending data during the initialization phase. During the first recognition Alice cannot be sure that it is communicating with the correct entity Bob.However, the more information that Alice receives (that it expected Bob to have) the more certain Alice can be that it is correct.

7.5.2 Pair wise Memory Complexity

In the case of ZCK, separate shared keys are needed for transmitting and receiving. The shared key is in the form of hash chains which have a significant memory requirement. This implies a limit on the number of partners that one node negotiates with. In practice this may be a limiting factor. In NMRP there is no limitation on number of partners that one node to communicate with.

8. Conclusions

The security principle called recognition is closely related to authentication. This new principle is appropriate for low-power environments where identification is not possible. It is observed that few proposed protocols satisfy the requirements of this limited resource environment.

NMRP fits the requirements best out of the protocols we analyzed. By presenting our implementation we showed that the environment can be satisfied but we also identified areas for concern including pair wise memory complexity. Future work in recognition protocols to satisfy low-power protocols will improve on our presented techniques.

References

[1] Jonathan Hammell, Andr_eWeimerskirch, JoaoGirao, and Dirk Westho, (2005) "Recognition in a low-power environment". In ICDCSW '05: Proceedings of the SecondInternational Workshop on Wireless Ad Hoc Networking (WWAN) ICDCSW'05),pages 933{938, Washington, DC, USA, IEEE Computer Society.

[2] AtefehMashatan and Douglas R. Stinson. (2008) A new message recognition protocol forad hoc pervasive networks. Technical Report 2008-15, Centre for Applied Cryptographic Research (CACR), University of Waterloo, Canada.

[3] F. Stajano and R. Anderson, (1999)"The resurrecting duckling: Security issues for ad-hoc wireless networks," in *Security Protocols, 7th International Workshop Proceedings, Lecture Notes in Computer Science*.

[4] A. Perrig, R. Canetti, J. D. Tygar, and D. Song,(2002) "The TESLA broadcast authentication protocol," *Cryptobytes*, vol. 5, no. 2, pp. 2–13, RSA Laboratories, Summer/Fall 2002.

[5] A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar,(2001) "SPINS: Security protocols for sensor networks," in *Mobile Computing and Networking*, pp. 189–199.

[6] R. Anderson, F. Bergadano, B. Crispo, J. Lee, C. Manifavas, and R. Needham,(1998) "A new family of authentication protocols," *ACM OperatingSystems Review*, vol. 32.

[7] C. Mitchell, (2003) "Remote user authentication using public information," in *Cryptography and Coding, 9th IMA International Conference on Cryptography and Coding Proceedings*.

[8] A. Weimerskirch and D. Westhoff,(2003) "Zero common-knowledge authentication for pervasive networks," in *Proceedings of Selected Areas of Cryptography 2003 (SAC 2003).*

[9] A. Menezes, P. vanOorschot, and S. Vanstone, (1997) *Handbook of Applied Cryptography*, CRC Press.

[10] D. L. Mills,(1994) "Internet time synchronization: The network time protocol," in *Global States and Time in Distributed Systems*, Z. Yang and T. A. Marsland, Eds. IEEE Computer Society Press.

[11] J. Elson and K. R¨omer, (2002) "Wireless sensor networks: A new regime for time synchronization," in *Proceedings of the First Workshop on Hot Topics in Networks*, Princeton, New Jersey, Oct.

[12] L. Gong, (1993) "Variations on the themes of message freshness and replay," in *Proceedings of the IEEE Computer Security Foundations WorkshopVI*, Franconia, New Hampshire, June, pp. 131–136.

[13] F. Stajano, (2000) "The resurrecting duckling — what next?" in *Security Protocols, 8th International Workshop Proceedings, Lecture Notes in Computer Science*.

[14] A. Perrig, R. Canetti, D. Song, and J. D. Tygar,(2001) "Efficient and secure source authentication for multicast," in *Network and Distributed System Security Symposium Conference Proceedings*.