Improved Robust User Authentication Scheme for Wireless Sensor Networks

Binod Vaidya[1], Min Chen[2] and Joel J. P. C. Rodrigues[3]
[1]Instituto de Telecomunicaçoes,
Covilha, PORTUGAL
bnvaidya@co.it.pt
[2]WISENET Lab, Hebei Polytechnic University,
Tangshan, CHINA
minchen@ieee.org
[3]Instituto de Telecomunicaçoes, and University of Beira Interior
Covilha, PORTUGAL
joeljr@ieee.org

Abstract- **Wireless sensor networks (WSNs) have been widely used in a wide variety of applications. Deploying WSNs in unattended environments can cause various security threats. In these scenarios, provisioning user authentication is a critical issue. In this paper, we propose improved robust user authentication scheme for WSNs, which is a variation of strong-password based solution. We have analyzed security properties of the proposed scheme and compared with the previous schemes in terms of overhead cost. We have also conducted formal verification of the proposed scheme.**

I. INTRODUCTION

Wireless sensor networks (WSNs) commonly consist of a large number of miniaturized sensor/actuator devices with low processing power, limited storage and energy that communicate over an ad hoc wireless network. Because of the ubiquitous nature, ease of deployment, and wide range of potential applications, WSNs have been widely used in a wide variety of applications, for instance, emergency medical care, structural health monitoring, environmental control, vehicular tracking, habitat monitoring, military operations or surveillance.

The sensors are usually cheap, small devices with battery and memory constraints and little computation power. When the number of sensors in the network is large or the deployment area is inaccessible, replacing the nodes is very costly or impossible.

Practically, most queries in WSN applications are issued at the base stations or at the backend of the application system. However, in many applications, the real-time data may no longer be accessed at the gateway node only. It can be accessed from any sensor login node in an ad hoc manner.

Providing user authentication while accessing real-time data is critical. Hence, in order to prevent unauthorized users from gaining the information, we need to use robust user authentication scheme. However, to date, only few schemes [2-5,10] have been proposed those are well suited for WSNs.

In this paper, we propose an improved robust dynamic user authentication schemes in WSN, which is a variation of strong-password based solution proposed by Wong *et al.* [3] and modified version of the robust scheme [10].

The rest of this paper is organized as follows. In Section II, we will briefly review the related works regarding the user authentication, while in Section III, we present the cryptanalysis some existing schemes. Next, in Section IV, we discuss the proposed user authentication protocol. Then, in Section V, we will analyze our proposed schemes in terms of security features and provide a comparative study with existing schemes in terms of cost overhead. And in Section VI, we have conducted formal verification of the proposed scheme. Finally, we will conclude our paper in Section VII.

II. RELATED WORKS

User authentication is an important topic for communication security and there are many schemes existed for the purpose in the literatures. In 1981, Lamport [1] proposed a user authentication scheme for communication in insecure channel. Later, several user authentication schemes [2-10] have been proposed to prevent from unauthorized users from gaining access to the system.

In 2006, Wong *et al.* [3] proposed a lightweight strong-password based dynamic user authentication protocol for WSNs. Wong et al. scheme uses basically one-way hash function and exclusive-OR operation to provide the dynamic user authentication in WSN. It consists of three phases: Registration, Login, and Authentication. We briefly describe the operation of this protocol below.

In Registration phase, a user submits his/her identification *userID* and password *PW* to the GW-node. The GW-node computes some values *A* and *B*. The GW-node replies to the user for successful registration and stores these values along with *userID* and PW. The GW-node distributes *userID* along with A and timestamp (*TS*) to those sensor nodes, which are able to provide a login interface to users.

In Login phase, a user submits *userID* and *PW* to a login node. The login node checks the validity of *userID*. If true then the login node retrieves the *A* and computes not only *B* but also authenticators C_2 and C_1. The login node then sends *userID* along with C_2, C_1 and current timestamp (*T*) to the GW-node for final authentication process.

978-1-4244-5875-2/09/$26.00 ©2009 IEEE

In Authentication phase, the GW-node checks the validity of the user and the timestamp. If both are valid, then the GW-node retrieves corresponding *A* and *B* and computes C_2 and C_1. On validating the authenticators, an accept message is sent to the login node which is forwarded to the user.

Tseng *et al.* [5] proposed an improved user authentication scheme that is modification of Wong *et al.*'s scheme [3] such that it not only fixes the weaknesses but also enhances the security of Wong *et al.*'s scheme. Tseng *et al.*'s scheme is divided into four phases: registration, login, authentication, and password-changing phases.

However, these schemes still have security flaws and cannot fully prevent from various malicious attacks. In [10], we proposed the robust dynamic user authentication scheme for WSNs. Nonetheless, it also does not provide complete mutual authentication. Thus, in this paper, we will propose improved robust user authentication scheme in WSN that can provide better security features than above-mentioned schemes.

III. CRYPTANALYSIS

Although Wong *et al.* [3] proposed a dynamic user authentication scheme that allows legitimate users to query at any of the sensor nodes and imposes very light computational load, there still remains several security weaknesses in their scheme. Tseng *et al.* [5] showed some of the security weaknesses in Wong *et al.*'s scheme. In this section, we show some of the security weaknesses for two schemes – namely Wong *et al.*'s scheme, and Tseng *et al.*'s scheme.

A. Wong et al.'s scheme

Wong *et al*'s scheme cannot resist forgery attacks when there is node capture attacks. Adversary captures LN to obtain *UID, A, TS* and eavesdrops *UID, PW.* Then it computes $B_e = H(A \,||\, H\,(PW));\ C_{1e} = H(T'\oplus B_e);\ C_{2e} = B_e \oplus A$. It sends message ($UID,\ C_{1e}, C_{2e},\ T'$) to GW. As long as $(T - T') < \Delta T$ then it is passed.

It cannot also prevent replay attacks of *Acc_login* that are possible in two ways.

Firstly, while transmitting *Acc_login* message from GW to LN, the malicious intermediate node can intercept it before forwarding it. In next session when this adversary receives message to GW from legitimate LN, it just drops that message and the captured *Acc_login* is replayed to LN as pretending legal GW.

Secondly, while transmitting *Acc_login* from LN to UD, adversary node can eavesdrop it. Next time the login message from UD can be blocked by adversary node. The captured *Acc_login* message is replayed to UD as pretending legal LN. As UD does not check the correctness, they will be counterfeited.

B. Tseng et al.'s scheme

Tseng *et al*.'s scheme cannot resist replay attacks of *Acc_login* that are possible in two ways.

In first case, on transmitting *Acc_login* from GW to LN, the malicious intermediate node can intercept it before forwarding it. Next time when this malicious node receives message to GW from legitimate LN, it just drops that message and the captured.

Acc_login is replayed to LN as pretending legal GW. LN does not check the correctness, so it will also send *Acc_login* to UD.

In second case, while transmitting *Acc_login* from LN to UD, adversary node eavesdrops it. Next time the login message from UD can be blocked by adversary node. The captured *Acc_login* message is replayed to UD as pretending legal LN. As UD does not check the correctness, they will be counterfeited.

It cannot prevent man-in-the-middle attacks. *UID, A, t* is intercepted or eavesdropped by an adversary. It then intercepts *UID, C, T, t*. After computing $C^* = H\,(A\oplus T^*)$, it will forward $UID,\ C^*,\ T^*,\ t$ to GW.

IV. PROPOSED AUTHENTICATION PROTOCOL

In this section, we propose an improved user authentication scheme to overcome weaknesses of the robust scheme [10].

TABLE I
NOTATIONS USED IN PROPOSED SCHEME

Symbols	Descriptions
UD	User's Device such PDA, PC
GW	Registration Sensor Gateway
LN	Sensor Login node
H()	One-way hash function
$\oplus$	Exclusive-OR (XOR) operation
||	Concentration
Succ_Reg	Successful Registration message
Acc_login	Accept login message
Succ_Change	Successful Changes message
x	Secret key known to the GW
UID	User's identity
PW	Password chosen by user
TS	Timestamp for particular user
t, T , T_0	Current time recorded by one of the nodes
ΔT	Allowed time interval for transmission delay

Table I shows the notations used in the proposed scheme. In the proposed scheme, it is assumed that as one-hop communication between UD and LN occurs, it is less likely to have malicious action. So we have only considered mutual authentication between GW and LN. The proposed scheme is composed of four phases: registration phase, login phase, authentication phase, and password change phase.

In Registration phase, the UD randomly chooses a password *PW* and calculates $vpw = H(PW)$. Afterwards, the UD submits its identity *UID* and *vpw* to the GW in a secure way. The GW computes $X = H(UID \,||\, x)$ and stores (*UID, vpw, X, TS*). Then the GW replies to the user for successful registration with *X*, (*Succ_Reg(X))*. Upon receiving this message, UD will store *X* for future use. The GW distributes *(UID, X, TS)* to those sensor nodes, which are able to provide a login interface to users.

RP1 -	UD	:	Compute $vpw = H(PW)$
RP2 -	UD → GW	:	UID, vpw
RP3 -	GW	:	Compute $X = H(UID \| x)$ Store UID, vpw, X, TS
RP4 -	GW → UD	:	*Succ_Reg (X)*
RP5	UD		Store X
RP5 -	GW → LNs	:	UID, X, TS
RP6 -	LN	:	Store UID, X, TS

In Login phase, a user submits (UID, A, t) to a login node. Upon receiving the login request at time T_0 , the login node checks its lookup table to see if UID is a valid user and checks $T_0 - t \geq \Delta T$. The login request is rejected if it is not. Otherwise, the login node retrieves the corresponding A and computes $C_K = (X \oplus A \oplus T_0)$. It then sends ($UID, C_K , T_0, t$) to the GW.

LP1 -	UD	:	Compute $A = H(vpw \| t)$
LP2 -	UD → LN	:	UID, A, t
LP3 -	LN	:	Check UID Check $T_0 - t \geq \Delta T$ Compute $C_K = (X \oplus A \oplus T_0)$
LP4 -	LN → GW	:	UID, C_K , T_0, t

In Authentication phase, the GW checks whether or not UID, t is a valid user and t. The login request is rejected if it is not. Otherwise, the GW verifies if $T_1 - T_0 \geq \Delta T$; $T_0 - t \geq \Delta T$. If the condition is satisfied, then the login request is considered as a replay message and thus is rejected. On the other hand, the GW retrieves the corresponding vpw and A and computes $A' = H(vpw \| t)$ and $C_K' = (X \oplus A' \oplus T_0)$. A reject message is sent to the login node if $C_K \neq C_K'$. Otherwise, computes $V_M = H(X \| A' \| T_1)$ and sends accept message (*Acc_login*, V_M, T_1) to the LN.

The LN computes V'_M and after verification of $V_M = V'_M$, it computes $Y_K = H(V'_M \| T_2)$. The LN sends (*Acc_login*, Y_K, T_1, T_2) to the UD. Upon receiving the message at time T_3, the UD checks if $T_1 - T_0 \geq \Delta T$; $T_0 - t \geq \Delta T$. If the conditions are true, then the login request is rejected. Otherwise, the login node retrieves the corresponding A, performs $V''_M = H(X \| A \| T_1)$ and $Y'_K = H(V''_M \| T_2)$, and checks if $Y_K = Y'_K$. If it is true, then the UD starts obtaining data if the condition holds. Otherwise, accept login message is rejected.

Figure 1 shows communication flows for first three phases (registration, login and authentication) of the proposed scheme Whereas Figure 2 shows the communication flow for password changing phase of the proposed scheme.

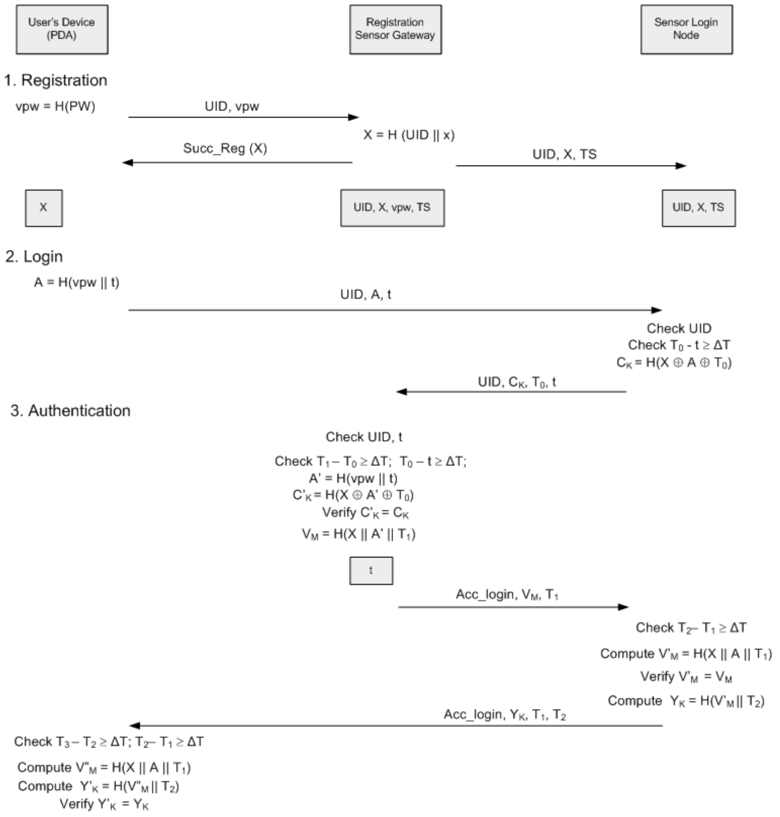

Figure 1. Communication flows for proposed scheme.

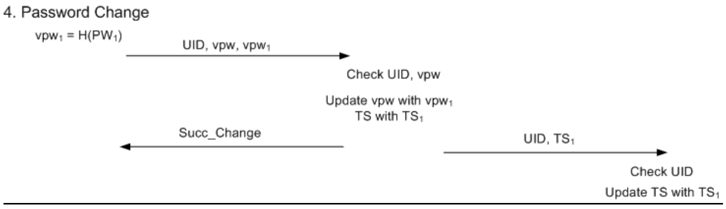

Figure 2. Password changing phase for proposed scheme.

AP1 -	GW	:	Check *UID, t* Check $T_1 - T_0 \geq \Delta T$; $T_0 - t \geq \Delta T$ Compute $A' = H(vpw \| t)$ Compute $C_K' = (X \oplus A' \oplus T_0)$ *Verify* $C_K = C_K'$ Compute $V_M = H(X \| A' \| T_1)$ Store *t*
AP2 -	GW → LN	:	*Acc_login*, V_M, T_1
AP3 -	LN	:	Check $T_2 - T_1 \geq \Delta T$ Compute $V'_M = H(X \| A \| T_1)$ *Verify* $V_M = V'_M$ Compute $Y_K = H(V'_M \| T_2)$
AP4 -	LN → UD	:	*Acc_login*, Y_K , T_1 , T_2
AP5	UD	:	Check $T_1 - T_0 \geq \Delta T$; $T_0 - t \geq \Delta T$ Compute $V''_M = H(X \| A \| T_1)$ Compute $Y'_K = H(V''_M \| T_2)$ *Verify* $Y_K = Y'_K$

In the Password-changing phase, UD changes his password *PW* to PW_1. Then it computes $vpw_1 = H(PW_1)$ and sends the triple (*UID, vpw,* vpw_1) to the GW in the secure channel. The GW checks *UID* and *vpw*. If both of them are true, GW updates its database. Then GW sends success change *Succ_Change* to the UD. At the same time, the GW distributes updated information to all the LNs. Upon receiving updates, LNs check *UID* and update their databases.

PP1 -	UD	:	Computes $vpw_1 = H(PW_1)$
PP2 -	UD → GW	:	*UID, vpw,* vpw_1
PP3 -	GW	:	Checks UID, vpw Updates *vpw, TS* with vpw_1, TS_1 respectively
PP4 -	GW → UD	:	*Succ_Change*
PP5 -	GW → LNs	:	*UID,* TS_1
PP6 -	LN	:	Checks *UID* Updates *TS* with TS_1

V. ANALYSIS

In this section, we present the security analysis of the proposed scheme and the comparison of the cost overhead with some existing schemes.

A. *Security Analysis*

The proposed scheme has several advantages over the existing schemes. It has following security features.

The proposed scheme can provide protection against the replay attacks of login message as well as accept login message (*Acc_login*). In case of login message, as GW-node checks user ID and timestamp, the adversary node cannot replay it, whereas in case of *Acc_login* message, as a login node checks the authenticator, the adversary node cannot replay it.

The proposed scheme can protect against the forgery attacks. In both schemes, the adversary node could not be able to compute C_k as it has no knowledge of the value of *A*. And if the adversary tries forgery attack using login message, it still cannot have access because of the timestamp used.

The proposed scheme can protect against the MITM attacks. In both schemes, the adversary node could not be able to compute C_k as it has no knowledge of the value of *X*.

The proposed scheme provide full mutual authentication. It can provide mutual authentication between login node and gateway node as well as mutual authentication between gateway and the user. In first case, a gateway node verifies the authenticator containing C_k supplied by the login node while a login node verifies the authenticator containing *X* furnished by a gateway node. While in the second case, the UD gives *vpw* and the GW gives *X* (securely exchange) during registration phase. Therefore, UD and GW can use *X* and *vpw* respectively to provide mutual authentication between the GW and UD.

B. *Overhead Cost Comparisons*

Table II summarizes the comparisons of the Wong *et al.*'s scheme, Tseng *et al.*'s scheme, the robust scheme and proposed scheme in terms of cost overheads.

TABLE II
OVERHEAD COST COMPARISON

Protocols	Overhead Cost			
	Registration	Login	Authentication	Total
Wong *et al.*'s scheme [2]	$3T_H+1C_{MH}$	$3T_H+2T_{XOR}+1C_{MH}$	$1T_H+2T_{XOR}+1C_{MH}$	$7T_H+4T_{XOR}+3C_{MH}$
Tseng *et al.*'s scheme [4]	$1T_H+1C_{MH}$	$2T_H+2T_{XOR}+1C_{MH}$	$2T_H+2T_{XOR}+1C_{MH}$	$5T_H+4T_{XOR}+3C_{MH}$
Robust scheme [10]	$2T_H+1C_{MH}$	$2T_H+2T_{XOR}+1C_{MH}$	$4T_H+2T_{XOR}+1C_{MH}$	$8T_H+4T_{XOR}+3C_{MH}$
Proposed scheme	$2T_H+1C_{MH}$	$2T_H+2T_{XOR}+1C_{MH}$	$7T_H+2T_{XOR}+1C_{MH}$	$11T_H+4T_{XOR}+3C_{MH}$

It can be seen that the overhead cost of the proposed scheme is slightly higher than robust scheme and Wong et al.'s scheme. Even though the proposed scheme has little higher overhead cost than all three schemes, it provides full mutual authentication. That means it can provide mutual authentication between the gateway and login sensor node as well as between the gateway and the user device.

VI. FORMAL VERIFICATION

In this section, we discuss the formal verification of the proposed scheme using Nonmonotonic Cryptographic Protocols (NCP)[11,12], which is also known as Rubin Logic. Rubin Logic describes the logic of authentication and the beliefs of various entities involved in a protocol as a consequence of communication, in which there is no idealization step in specifying protocols. The specification of protocols, which is close to the actual implementation, is simply the starting point of the analysis.

There are two sets in non-monotonic logic, which are applicable of analyzing a protocol. One is global to the protocol and other is local to each entity. Global Set is public to each principal in a protocol specification. It contains Principal set, Rule set, Secret set, and Observers set. Local sets are private to each principal in a protocol specification. For each principal, it mainly has following sets: Possession set, Belief set, and Behavior list. The knowledge and belief sets for each principal are modified via actions and inference rules.

Global sets:

Principal Set: *P = {U, SN, GW}*. U is the initiator of protocol.

Rule Set: Inference rules defined in Appendix 1 B.

Secret Set: {PW, x}

Observers Set:

Observers(PW): {U}

Observers(x): {GW}

Local Sets: The Local set consists of U, SN, and GW.

Principle U

```
POSS(U) = {PW, h(UID || x), {UID}}
BEL(U) ={#PW, # h(UID || x)}
BL(U) =
    Send (SN, {UID, h(vpw || t), t})
    Update({UID, h(vpw || t), t})
    Receive(SN, {AccLogin, $Y_K$, $T_1$, $T_2$})
    Check-freshness($T_3 – T_2 \geq \Delta T$})
    Check-freshness($T_2 – T_1 \geq \Delta T$)
    Hash(h(.); Concat(h(UID || x), h(vpw || t), $T_1$)) -> $V''_M$
    Hash(h(.); Concat($V''_M$ , $T_2$)) -> $Y'_K$
    Check($Y'_K$, $Y_K$)
```

Principle SN

```
POSS(SN) = {h(UID || x),TS, {UID}}
BEL(SN) = {# h(UID || x), #TS}
BL(SN) =
    Receive (U, {UID, H(vpw || t), t})
    Check(UID, stored UID)
    Check-freshness($T_0 – t \geq \Delta T$)
    Hash(h(.); XOR(h(UID || x), h(vpw || t), $T_0$)) -> $C_K$
    Send(GW, {UID, $C_K$, $T_0$, t})
    Update({UID, $C_K$, $T_0$, t})
    Receive(GW, {AccLogin, $V_M$, $T_1$})
    Check-freshness($T_2 – T_1 \geq \Delta T$)
    Hash(h(.); Concat(h(UID || x), h(vpw || t), $T_1$)) -> $V'_M$
    Check($V'_M$, $V_M$)
    Hash(h(.); Concat($V'_M$, $T_2$)) -> $Y_K$
    Send (U, {AccLogin, $Y_K$, $T_1$, $T_2$})
    Update({AccLogin, $Y_K$, $T_1$, $T_2$})
```

Principle GW

```
POSS(GW) = {x, vpw, TS, {UID}}
BEL(GW) = {#x, #vpw, #TS}
BL(GW) =
    Receive(SN, {UID, $C_K$, $T_0$, t})
    Check(UID, stored UID)
    Check-freshness($T_1 – T_0 \geq \Delta T$)
    Check-freshness($T_0 – t \geq \Delta T$)
    Hash(h(.); Concat(vpw, t)) -> A'
    Hash(h(.); XOR(h(UID || x), A', $T_0$)) -> $C'_K$
    Check($C'_K$, $C_K$)
    Hash(h(.); Concat(h(UID || x), A', $T_1$)) -> $V_M$
    Send (SN, {AccLogin, $V_M$, $T_1$})
    Update({AccLogin, $V_M$, $T_1$})
```

We assumed the registration phase occurs in secure manner. And during registration phase, principles U, SN, and GW share some parameters, which are kept secure as possible. Thus we have analyzed only login and authentication phases in the proposed scheme.

It can be seen that the actions are executed in sequential order starting from the U since it is the initiator of the protocol. With every *Update* action, there will be change in the action sequences from one principle to another principle.

The freshness can be verified with *Check-freshness* action. Thus the submessage freshness rule can be applied. Similarly the *Check* action verifies the authenticity of authentication tag. For instance while verifying *Check(C'_K, C_K)* action, the GW accepts the login message, only if the obtained value C_K from the LN is equal to the computed value C'_K by the GW; otherwise it will reject the login message and send failure message.

It can be conferred that the mutual authentications are achieved between the gateway node and the user device and between the gateway node and the login sensor node.

VII. CONCLUSION

In this paper, we have proposed an improved version of robust dynamic user authentication scheme for wireless sensor networks. We have conducted the cryptanalysis of the existing schemes, which shows that they cannot thwart various malicious attacks. We have provided the security analysis of our proposed scheme and also shown the overhead cost comparison with the existing schemes. Even though our scheme has slightly higher overhead cost, it can provide full mutual authentication and thwart various attacks. We have also conducted formal verification of the proposed scheme, which shows that full mutual authentication can be achieved.

ACKNOWLEDGMENT

Part of this work has been supported by the *Instituto de Telecomunicações,* Next Generation Networks and Applications Group, Portugal, and by the Euro-NF Network of Excellence from Seventh Framework Programme of EU.

REFERENCES

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

[2] Z. Benenson, N. Gedicke, and O. Raivio "Realizing Robust User Authentication in Sensor Networks." *In Proc. of Workshop on Real-World Wireless Sensor Networks (REALWSN 2005),* Sweden, June 2005.

[3] K.H.M. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks." *In Proc. of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, Jun. 2006; 1: 318–327.

[4] C. Jiang, B. Li, and H. Xu, "An Efficient Scheme for User Authentication in Wireless Sensor Networks." *In Proc. of 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07),* 2007.

[5] C.C. Lee, L.H. Li, and M.S. Hwang. "A remote user authentication scheme using hash functions". *ACM SIGOPS Operating Systems Review*, Oct. 2002; 36(4): 23–29.

[6] J.J. Shen, C.W. Lin, and M.S. Hwang. "A modified remote user authentication scheme using smart cards". *IEEE Transactions on Consumer Electronics*, May 2003; 49(2): 414–416.

[7] M.L. Das, A. Saxena, and V.P. Gulati. "A Dynamic ID-based Remote User Authentication Scheme". *IEEE Transactions on Consumer Electronics*, May 2004; 50(2): 629–631.

[8] E.J. Yoon, E.K. Ryu, and K.Y. Yoo. "An improvement of Hwang Lee Tang's simple remote user authentication scheme". *Elsevier Computers & Security*, Feb. 2005; 24(1):50–56..

[9] Y.Y. Wang, J.Y. Liu, F.X. Xiao, and J. Dan, "A more efficient and secure dynamic ID-based remote user authentication scheme", *Elsevier Computer Communications*, 2009.

[10] B Vaidya, J.S. Silva, J.J. Rodrigues, "Robust Dynamic User Authentication Scheme for Wireless Sensor Networks", *In Proc. of the 5th ACM Symposium on QoS and Security for wireless and mobile networks (Q2SWinet 2009)*, Tenerife, Spain, Oct. 2009, pp 88-91.

[11] A.D. Rubin, and P. Honeyman, "Nonmonotonic cryptographic protocol." *In Proc. of the Computer Security Foundation Workshop* VII, pp. 100-116, June 1994.

[12] M.L. Das, and V.L. Narasimhan, "Towards a Formal Verification of an Authentication Protocol using Non-monotonic Logic." *In Proc. of 5th International Conference on Information Technology: New Generation*, 2008.

Appendix 1: Rubin Logics Basics

A. Actions

1. *Concat*$(X_1, X_2,\dots, X_n)$
   Condition: $X_1, X_2,\dots,X_n \in POSS(P_i)$.
   Result: $POSS(P_i) := POSS(P_i) \cup \{X_1, X_2,\dots, X_n\}$.
   Description: This action is used when a principal constructs a message, X, out of submessages $X_1, X_2,\dots, X_n$.

2. *Hash*$(h(\cdot); X)$
   Condition: $h(\cdot), X \in POSS(P_i)$
   Result: $POSS(P_i) := POSS(P_i) \cup \{h(X)\}$
   Description: This action is used to hash data

3. *XOR*$(X_1, X_2,\dots, X_n)$
   Condition: $X_1, X_2,\dots, X_n \in POSS(P_i)$
   Result: $POSS(P_i) := POSS(P_i) \in \{X_1, X_2,\dots, X_n\}$
   Description: This action is used to XORing data.

4. *Check*(X, Y)
   Condition: $X, Y \in POSS(P_i)$
   Result: Valid if $X = Y$, else Invalid

5. *Send*(P_j, X)*:* It means that P_i sends X to P_j.

6. *Receive*(P_j, X)*:* It means that P_i receives X from P_j. In this case, X will be marked as coming from P_j and added to $POSS(P_i)$.

7. *Update*(X)*:* The Update function is used to maintain the observers of X.

B. Inference Rules

1. Nonce verification rule:

$$\frac{\#(X) \in BEL(P),\ from\ Q \in POSS(P)}{BEL(P) := BEL(P) \cup \{Q\ believes\ \#(X)\}}$$

2. Message meaning rule:

$$\frac{\{X\}_k\ from\ Q \in POSS(P), \{P,Q\} \subseteq Observers(k)}{BEL(P) := \mathrm{BEL}(P) \cup \{X \in POSS(P)\}}$$

3. Origin rule:

$$\frac{X \in POSS(P), \{X\ contains\ x_1\}, Q \in Observers(x_1)}{x_1\ from\ Q \in POSS(P)}$$

4. Submessage origin rule:

$$\frac{X \in POSS(P), X\ contains\ \{x_1, x_2\}\ from\ Q}{x_2\ from\ Q \in POSS(P)}$$

5. Submessage freshness rule:

$$\frac{\#(x) \in BEL(P), \{X\ contains\ x_1, X\ contains\ x_2\} \subseteq POSS(P)}{BEL(P) := BEL(P) \cup \#(x_1)}$$