User Authentication in Wireless Sensor Networks

Dr. Ashok Kumar Das

IEEE Senior Member
Associate Professor
Center for Security, Theory and Algorithmic Research
(Department of Computer Science and Engineering)
International Institute of Information Technology, Hyderabad
(Formerly **Indian Institute of Information Technology, Hyderabad**)

E-mail: *ashok.das@iiit.ac.in*
Homepage: `http://www.iiit.ac.in/people/faculty/ashokkdas`
Personal Homepage: `https://sites.google.com/view/iitkgpakdas/`

September 11, 2022

Temporal credential-based three-factor user authentication for distributed wireless sensor networks

This work is published in the paper:

> **Ashok Kumar Das. "A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks," in *Peer-to-Peer Networking and Applications (Springer)*, Vol. 9, No. 1, pp. 223-244, 2016, DOI: 10.1007/s12083-014- 0324-9. (2021 SCI Impact Factor: 3.488)**

[This article is one of the top five most popular downloaded articles during December 2014 to January 2015 of the Peer-to-Peer Networking and Applications]

Necessity for user authentication

- Most queries in wireless sensor network (WSN) applications are issued at the point of the base station or gateway node of the network.
- However, for critical applications of WSNs (e.g., battle field, healthcare application) there is a great need to access the real time data inside the WSN from the nodes, because the real-time data may no longer be accessed through the base station only.
- The real-time data can be given access directly to the external users (parties) those who are authorized to access data as and when they demand.
- The user authentication plays a vital role for this purpose.

User authentication in DWSNs

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Three factors used in the designed scheme

- Smart card
- Password
- Personal biometrics (for example, fingerprints, faces, irises, hand geometry and palm-prints, etc.)

Biometrics and Fuzzy Extractor

- Uses the user's personal biometrics along with traditional password to design user authentication protocols in WSNs.
- The biometric verification allows one to confirm or establish an individual's identity.
- There are major advantages of using biometric keys (for example, fingerprints, faces, irises, hand geometry and palm-prints, etc.):
  - Biometric keys can not be lost or forgotten.
  - Biometric keys are very difficult to copy or share.
  - Biometric keys are extremely hard to forge or distribute.
  - Biometric keys can not be guessed easily.
  - Someone's biometrics is not easy to break than others.

Biometrics and Fuzzy Extractor

- The output of a conventional hash function $h(\cdot)$ is sensitive and it may also return completely different outputs even if there is a little variation in inputs.
- The biometric information is prone to various noises during data acquisition, and the reproduction of actual biometric is hard in common practice.
- To avoid such problem, a fuzzy extractor method is preferred, which can extract a uniformly random string and a public information from the biometric template with a given error tolerance t.

Biometrics and Fuzzy Extractor

Definition

The fuzzy extractor is a tuple $(\mathcal{M}, l, t)$, which is composed of the following two algorithms, called *Gen* and *Rep*:

- **Gen:** It is a probabilistic algorithm, which takes a biometric information $B_i \in \mathcal{M}$ as input, and then outputs a secret key data $\sigma_i \in \{0, 1\}^l$ and a public reproduction parameter τ_i, where $Gen(B_i) = \{\sigma_i, \tau_i\}$.
- **Rep:** This is a deterministic algorithm, which takes a noisy biometric information $B_i' \in \mathcal{M}$ and a public parameter τ_i and t related to B_i, and then it reproduces (recovers) the biometric key data σ_i. In other words, we have $Rep(B_i', \tau_i) = \sigma_i$ provided that the condition $d(B_i, B_i') \leq t$ is met.

Biometrics and Fuzzy Extractor

- The probability to guess the biometric key data $\sigma \in \{0,1\}^l$ by an attacker is approximately $\frac{1}{2^l}$, where $l = m - 2\log(\frac{1}{\epsilon}) + O(1)$, where ϵ is the statistical distance between two given probability distributions, and m is the min-entropy given as follows.
  the min-entropy $H_\infty(A)$ of a random variable A is $-log(max_a Pr[A = a])$.
  Vanga Odelu, Ashok Kumar Das, and Adrijit Goswami. "A Secure Biometrics-Based Multi-Server Authentication Protocol using Smart Cards," in *IEEE Transactions on Information Forensics and Security*, Vol. 10, No. 9, pp. 1953 - 1966, 2015, DOI: 10.1109/ TIFS.2015.2439964. (2021 SCI Impact Factor: 7.231) [This article is one of the top 50 most frequently downloaded documents for Popular Articles (June - November 2015)]

User authentication in DWSNs

Threat Model In the following we consider the three types of models:

- **Honest-but-Curious adversary model**: This model [HCAM] is a passive adversarial model where the adversary $\mathcal{A}$ will behave like a legitimate entity and follow the specified protocol. However, $\mathcal{A}$ can read all the transmitting information between the corrupted entities in the network.
- **Dolev-Yao (DY) threat model**: This model is known as the DY model [DYM]. In the DY model, an adversary $\mathcal{A}$ has the potential ability to eavesdrop, intercept, modify and delete messages that are being communicated among various agents through a wireless network.
- **Canetti and Krawczyk's model**: This model is also known as the "CK-adversary model" [CKM]. Keeping all the fundamental assumptions used in the DY model, the CK-adversary model empowers $\mathcal{A}$ to compromise secret keys, secret credentials, and session states through the session hijacking attacks. Thus, leakage of the short term secrets from the *UE* node's memory can lead to disclosure of session key and other secrets.

User authentication in DWSNs

Threat Model

- [HCAM]: B. Narwal and A. K. Mohapatra, “A survey on security and authentication in wireless body area networks,” *Journal of Systems Architecture*, vol. 113, p. 101883, 2021. https://www.sciencedirect.com/science/article/pii/S1383762120301600
- [DYM]: D. Dolev and A. Yao, “On the security of public key protocols,” *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198208, 1983. https://ieeexplore.ieee.org/document/1056650
- [CKM]: R. Canetti and H. Krawczyk, “Universally Composable Notions of Key Exchange and Secure Channels,” in *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT’02)*, Amsterdam, The Netherlands, 2002, pp. 337–351. https://link.springer.com/chapter/10.1007/3-540-46035-7_22

User authentication in DWSNs

Threat Model

- Due to the hostile environments in the deployment field, nodes can be physically captured by an attacker.
- Sensor nodes as well as cluster heads can be compromised or captured by an attacker. Usually, nodes are not equipped with tamper-resistant hardware due to cost constraints and hence we assume that once a node is captured by an attacker, all the stored sensitive data as well as cryptographic information are revealed to the attacker.
- In any case, the *GWN* will not be compromised by an attacker.
- Finally, we make use of the famous Dolev-Yao threat model in which two communicating parties (nodes) communicate over an insecure channel. We adopt the similar threat model for WSNs where the channel is insecure and the end-points (users, sensor nodes) cannot in general be trustworthy.

User authentication in DWSNs

Table: Notations used

Symbol	Description
GWN	WSN gateway node (base station)
U_i	i^{th} user
SC_i	Smart card of U_i
ID_i	Identity of user U_i
PW_i	Password of user U_i
B_i	Biometric information of U_i
K	1024-bit secret number known to U_i only
$h(\cdot)$	Secure collision-free one-way hash function
X_s	1024-bit secret master key of GWN
SN_j	j^{th} sensor node in WSN
ID_{SN_j}	Identity of SN_j
TE_i	Expiration time of U_i's temporal credential
TS_X	Current timestamp of an entity X
$Gen(\cdot)$	Fuzzy generator function
$Rep(\cdot)$	Fuzzy reproduction function
t	Error tolerance threshold used in fuzzy extractor
ΔT	Maximum transmission delay
$A \oplus B$	Bitwise XORed of data A with data B
$A\|\|B$	Data A concatenates with data B

Pre-Deployment Phase

Before deployment of nodes in the network, the *GWN* does the following steps.

- Step PD1. For each deployed sensor node SN_j, the GWN selects a unique identifier ID_{SN_j}.
- Step PD2. The GWN generates randomly a large 1024-bit number K_{GWN-S}, which is considered as the GWN's private key only known to the GWN. After that for each deployed sensor node SN_j, the GWN computes $TC_j = h(K_{GWN-S} || ID_{SN_j})$, which is the temporal credential for SN_j.
- Step PD3. Finally, each deployed sensor node SN_j is pre-loaded with the information TC_j as its temporal credential prior to its deployment in the target field.

User authentication in DWSNs

Pre-Deployment Phase

ID_{SN_j}	$TC_j = h(K_{GWN-S} \mid\mid ID_{SN_j})$

Figure: Pre-loaded information into SN_j's memory.

User authentication in DWSNs

Registration Phase

- Before accessing data from a particular sensor node in the sensor network, the user U_i needs to register with the GWN of the network.
- U_i first selects a unique identity ID_i and chooses a password PW_i.
- U_i generates randomly a large 1024-bit secret number K. U_i computes the masked password $RPW_i = h(ID_i||K||PW_i)$ and sends the registration request message $\langle ID_i, RPW_i \rangle$ to the GWN via a secure channel.
- The remaining steps are summarized in the following table.

User authentication in DWSNs

User (U_i)/Smart Card (SC_i)	GWN
Inputs ID_i, PW_i, B_i. Generates a random secret number K. Computes $RPW_i = h(ID_i \| \| K \| \| PW_i)$. $\xrightarrow{\langle ID_i, RPW_i \rangle}$ (via a secure channel)	
	Generates private key K_{GWN-U}. Computes $TC_i = h(K_{GWN-U} \| \| ID_i \| \| TE_i)$, $PTC_i = TC_i \oplus RPW_i$. Generates secret information X_s and computes $r_i = h(ID_i \| \| X_s)$. Selects temporary identity TID_i of U_i and initializes it. Stores the tuple (TID_i, ID_i, TE_i) in its verification table. $\xleftarrow{\langle Smart\ Card(h(\cdot), TID_i, TE_i, PTC_i, r_i) \rangle}$ (via a secure channel)
Computes $Gen(B_i) = (\sigma_i, \tau_i)$, $e_i = h(ID_i \| \| \sigma_i) \oplus K$, $f_i = h(ID_i \| \| RPW_i \| \| \sigma_i)$, $r_i^* = r_i \oplus h(ID_i \| \| K)$. Replaces r_i with r_i^* in smart card. Stores e_i, f_i, $Gen(\cdot)$, $Rep(\cdot)$, t and τ_i in smart card.	

Registration Phase

$h(\cdot), TID_i, TE_i, PTC_i, r_i^*, f_i, e_i, Gen(\cdot),$ $\text{Rep}(\cdot), t, \tau_i$.

Figure: Information stored into SC_i's memory.

Login Phase

User (U_i)/Smart Card (SC_i)	GWN
Inserts smart card and inputs ID_i, PW_i, B_i. Computes $\sigma_i^* = Rep(B_i, \tau_i)$, $K^* = e_i \oplus h(ID_i \| \| \sigma_i^*)$, $RPW_i^* = h(ID_i \| \| K^* \| \| PW_i)$ and $f_i^* = h(ID_i \| \| RPW_i^* \| \| \sigma_i^*)$. Checks if $f_i^* = f_i$? If so, generates a current timestamp TS_1, temporary key K_i, and computes $TC_i = PTC_i \oplus RPW_i^*$, $M_1 = r_i^* \oplus h(ID_i \| \| K^*) = h(ID_i \| \| X_s)$, $PKS_i = K_i \oplus h(TC_i \| \| M_1 \| \| TS_1)$, $C_i = h(ID_i \| \| K_i \| \| TC_i \| \| M_1 \| \| TID_i \| \| TS_1)$. $\xrightarrow{\langle TID_i, C_i, PKS_i, TS_1 \rangle}$ (via a public channel)	

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Authentication and Key Agreement Phase

User (U_i)/Smart Card (SC_i)	GWN	Sensor node (SN_j)
	Checks the timeliness of TS_1 by the condition $\vert T^*_{GWN} - TS_1\vert < \Delta T$, where T^*_{GWN} is the current timestamp of the GWN. If it is valid, computes $M_2 = h(ID_i \vert\vert X_s)$, $TC_i = h(K_{GWN-U} \vert\vert ID_i \vert\vert TE_i)$, $K_i = PKS_i \oplus h(TC_i \vert\vert M_2 \vert\vert TS_1)$. $C^*_i = h(ID_i \vert\vert K_i \vert\vert TC_i \vert\vert M_2 \vert\vert TS_1)$. Checks if $C^*_i = C_i$? If so, computes $TC_j = h(K_{GWN-S} \vert\vert ID_{SN_j})$, $C_{GWN} = h(TID_i \vert\vert TC_j \vert\vert TS_2)$, $PKS_{GWN} = (K_i \oplus M_2)$ $\oplus h(TC_j \vert\vert TS_2)$. $\xrightarrow{\langle TS_2, TID_i, C_{GWN}, PKS_{GWN}\rangle}$ (via a public channel)	

Authentication and Key Agreement Phase (Cont...)

U_i/SC_i	GWN	Sensor node (SN_j)
		Checks if $\|T_j^* - TS_2\| < \Delta T$? If it is valid, computes $C_{GWN}^* = h(TID_i \|\| TC_j \|\| TS_2)$. Checks if $C_{GWN}^* = C_{GWN}$? If it holds, computes $M_3 = PKS_{GWN} \oplus h(TC_j \|\| TS_2)$, $C_j = h(K_j \|\| TID_i \|\| ID_{SN_j} \|\| TS_3)$, $PKS_j = K_j \oplus h(M_3 \|\| TS_3)$. $\langle ID_{SN_j}, TS_3, C_j, PKS_j \rangle$ $\xleftarrow{\hspace{4cm}}$ (via a public channel)
	Computes $K_j = PKS_j \oplus h((K_i \oplus M_2) \|\| TS_3)$, $C_j^* = h(K_j \|\| TID_i \|\| ID_{SN_j} \|\| TS_3)$. Verifies if $C_j^* = C_j$? If it is valid, generates TID_i^{new} and computes $D_{GWN} = TID_i^{new} \oplus h((K_i \oplus M_2) \|\| TS_3 \|\| TS_4)$. Updates TID_i with TID_i^{new}, and computes $E_{GWN} = h(ID_i \|\| ID_{SN_j} \|\| TC_i \|\| D_{GWN}$ $\|\| K_j \|\| TS_3 \|\| TS_4)$. $\langle ID_{SN_j}, TS_3, TS_4, PKS_j, D_{GWN}, E_{GWN} \rangle$ $\xleftarrow{\hspace{7cm}}$ (via a public channel)	

Authentication and Key Agreement Phase (Cont...)

User (U_i)/Smart Card (SC_i)	GWN	Sensor node (SN_j)
Checks the timeliness of TS_4. If it is valid, computes $TID_i^{new} = D_{GWN} \oplus h((K_i \oplus M_1)$ $\|\|TS_3\|\|TS_4)$, $K_j = PKS_j \oplus h((K_i$ $\oplus M_1)\|\|TS_3)$, $E^*_{GWN} = h(ID_i\|\|$ $ID_{SN_j}\|\|TC_i\|\|D_{GWN}\|\|K_j\|\|TS_3$ $\|\|TS_4)$. Checks if $E^*_{GWN} = E_{GWN}$? If it passes, computes session key $SK_{ij} = h((K_i \oplus M_1) \oplus K_j)$. Replaces TID_i with TID_i^{new}.	Replaces TID_i with TID_i^{new}.	Computes session key $SK^*_{ij} = h(M_3 \oplus K_j)$ $= h((K_i \oplus M_1) \oplus K_j)$.

Password and biometric update phase

User (U_i)	Smart Card (SC_i)
Inserts SC_i, and inputs ID_i, PW_i^{old} and also imprints B_i^{old}. $\xrightarrow{\langle ID_i, PW_i^{old}, B_i^{old}\rangle}$	Computes $\sigma_i^{old} = Rep(B_i^{old}, \tau_i)$,
	$K^* = e_i \oplus h(ID_i \| \sigma_i^{old})$, $RPW_i^{old} = h(ID_i \| K^* \| PW_i^{old})$, $f_i^{old} = h(ID_i \| RPW_i^{old} \| \sigma_i^{old})$. Checks if $f_i^{old} = f_i$? $\xleftarrow{\text{Request for new password \& biometrics}}$
Inputs PW_i^{new}, B_i^{new} $\xrightarrow{\langle PW_i^{new}, B_i^{new}\rangle}$	Computes $x = PTC_i \oplus RPW_i^{old}$
	$= TC_i \oplus RPW_i \oplus RPW_i^{old} = TC_i$, $RPW_i^{new} = h(ID_i \| K^* \| PW_i^{new})$, $PTC_i^{new} = x \oplus RPW_i^{new}$, $Gen(B_i^{new}) = (\sigma_i^{new}, \tau_i^{new})$, $e_i^{new} = h(ID_i \| \sigma_i^{new}) \oplus K^*$, $f_i^{new} = h(ID_i \| RPW_i^{new} \| \sigma_i^{new})$. Replaces PTC_i, f_i, e_i, and τ_i with PTC_i^{new}, f_i^{new}, e_i^{new}, and τ_i^{new}, respectively.

Dynamic node addition phase

Suppose a new sensor node SN_j^{new} is to be deployed in the existing sensor network. For this purpose, the following steps are executed by the GWN in offline prior to its deployment in the target field:

- Step DA1. The GWN first assigns a unique random identity $ID_{SN_j}^{new}$ for SN_j^{new}.
- Step DA2. The GWN then computes the temporal credential for SN_j^{new} as $TC_i^{new} = h(K_{GWN-S}||ID_{SN_j}^{new})$.
- Step DA3. Finally, the GWN loads $ID_{SN_j}^{new}$ and TC_i^{new} in the memory of SN_j^{new} prior to its deployment.

After deployment of the new sensor node SN_j^{new} in the target field, the GWN needs to inform the user U_i so that he/she can access the real-time data from it later.

Security Analysis

It is shown that the proposed scheme has the ability to tolerate the following attacks:

- Privileged insider attack
- Online password and biometric key guessing attack
- Offline password and biometric key guessing attack
- Replay attack
- Man-in-the-middle attack
- Stolen-verifier attack
- Forgery (impersonation) attacks
  - U_i forgery attack
  - GWN forgery attack
  - SN_j forgery attack

Security Analysis

It is also shown that the proposed scheme has the ability to tolerate the following other attacks:

- Many logged-in users with the same login-id attack
- Identity guessing attack
- Tracing attack
- Password and biometric change attack
- User anonymity and unlinkability
- Three-factor security

Formal Security Analysis Using Random Oracle Model

More precisely, we have the following theorem:

Theorem

Let $\mathcal{A}$ be an adversary running in polynomial time t against our protocol $\mathcal{P}$ in random oracle, $\mathcal{D}$ be a uniformly distributed password dictionary and l be the number of bits in the biometrics key σ_i. Then, the probability of deriving the identity ID_i, the password PW_i, the biometric key data σ_i of a legal user U_i, and the secret information X_s of the GWN, even if the user U_i's smart card SC_i is lost/stolen, in the proposed protocol $\mathcal{P}$ by $\mathcal{A}$ is estimated as

$$Adv_{\mathcal{P}}^{ake} \leq \frac{q_h^2}{|Hash|} + \frac{q_{send}}{2^{l-1}.|\mathcal{D}|},$$

where q_h, q_{send}, $|Hash|$ and $|\mathcal{D}|$ denote the number of hash queries, the number of Send queries, the range space of the hash function and the size of $\mathcal{D}$, respectively.

User Authentication in DWSNs

Performance comparison

- SF_1 : whether resilient against privileged insider attack;
- SF_2 : whether resilient against stolen-verifier attack;
- SF_3 : whether protects password guessing attack;
- SF_4 : whether resilient against stolen smart card attack;
- SF_5 : whether prevents forgery attack;
- SF_6 : whether resists replay attack;
- SF_7 : whether resilient against user identity guessing attack;
- SF_8 : whether resilient against tracing attack;
- SF_9 : whether provides mutual authentication between U_i and GWN;
- SF_{10} : whether provides mutual authentication between GWN and SN_j;
- SF_{11} : whether provides user anonymity;
- SF_{12} : whether provides user untraceability property;

Performance comparison

- SF_{13} : whether supports key agreement between U_i and SN_j;
- SF_{14} : whether supports correct password update;
- SF_{15} : whether supports correct biometric update;
- SF_{16} : whether provides non-repudiation;
- SF_{17} : whether resilient against node capture attack;
- SF_{18} : whether provides three-factor security;
- SF_{19} : whether provides formal security analysis and verification;
- SF_{20} : whether supports dynamic sensor node addition after initial deployment.

User Authentication in DWSNs

The proposed scheme is compared with the following recent related existing schemes:

- [1]. Das, M.L.: Two-Factor User Authentication in Wireless Sensor Networks. **IEEE Transactions on Wireless Communications** 8(3), 1086–1090 (2009)
- [2]. Yoo, S.G., Park, K.Y., Kim, J.: A Security-Performance-Balanced User Authentication Scheme for Wireless Sensor Networks. **International Journal of Distributed Sensor Networks** 2012 (2012). Article ID 382810, 11 pages, 2012. doi:10.1155/2012/382810
- [3]. Sun, D.Z., Li, J.X., Feng, Z.Y., Cao, Z.F., Xu, G.Q.: On the secu- rity and improvement of a two-factor user authentication scheme in wireless sensor networks. **Personal and Ubiquitous Computing** 17(5), 895–905 (2013)
- [4]. Xue, K., Ma, C., Hong, P., Ding, R.: A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. **Journal of Network and Computer Applications** 36(1), 316–323 (2013)
- [5]. Jiang, Q., Ma, J., Lu, X., Tian, Y.: An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. **Peer-to-Peer Networking and Applications** pp. 8(6), 1070–1081 (2015)

User Authentication in DWSNs

Table: Features comparison between the proposed scheme and other schemes

Security features	[1]	[2]	[3]	[4]	[5]	Proposed scheme
SF_1	No	Yes	Yes	No	No	Yes
SF_2	Yes	Yes	Yes	Yes	Yes	Yes
SF_3	Yes	Yes	Yes	Yes	Yes	Yes
SF_4	No	No	Yes	No	Yes	Yes
SF_5	Yes	Yes	Yes	Yes	Yes	Yes
SF_6	Yes	Yes	Yes	Yes	Yes	Yes
SF_7	No	No	No	No	Yes	Yes
SF_8	No	No	No	No	Yes	Yes
SF_9	No	Yes	No	Yes	Yes	Yes
SF_{10}	No	Yes	Yes	Yes	Yes	Yes
SF_{11}	No	No	No	No	Yes	Yes
SF_{12}	No	No	No	No	Yes	Yes

User Authentication in DWSNs

Table: Features comparison between the proposed scheme and other schemes (Continued...)

Security features	[1]	[2]	[3]	[4]	[5]	Proposed scheme
SF_{13}	No	Yes	Yes	Yes	Yes	Yes
SF_{14}	No	Yes	No	No	No	Yes
SF_{15}	No	No	No	No	No	Yes
SF_{16}	No	No	No	No	No	Yes
SF_{17}	No	Yes	Yes	Yes	Yes	Yes
SF_{18}	No	No	No	No	No	Yes
SF_{19}	No	No	No	No	No	Yes
SF_{20}	No	No	No	No	No	Yes

User Authentication in DWSNs

Table: Computational overhead comparison between our scheme and other schemes

Phase	Entity	[1]	[2]	[3]	[4]	[5]	Proposed
User reg	U_i	$-$	t_h	$-$	$2t_h$	t_h	$4t_h + t_{fe}$
	GWN	$3t_h$	$3t_h$	$2t_h$	$4t_h$	t_h	$2t_h$
Login +	U_i	$4t_h$	$5t_h$	$2t_h$	$10t_h$	$7t_h$	$t_{fe} + 9t_h$
Authen	GWN	$4t_h$	$8t_h$	$5t_h$	$13t_h$	$10t_h$	$11t_h$
	SN_j	t_h	$2t_h$	$2t_h$	$6t_h$	$5t_h$	$5t_h$
Total cost		$12t_h$	$19t_h$	$11t_h$	$35t_h$	$24t_h$	$31t_h + 2t_{fe}$
Rough (in seconds)		0.0038	0.0061	0.0035	0.0112	0.00768	0.04412

User Authentication in DWSNs

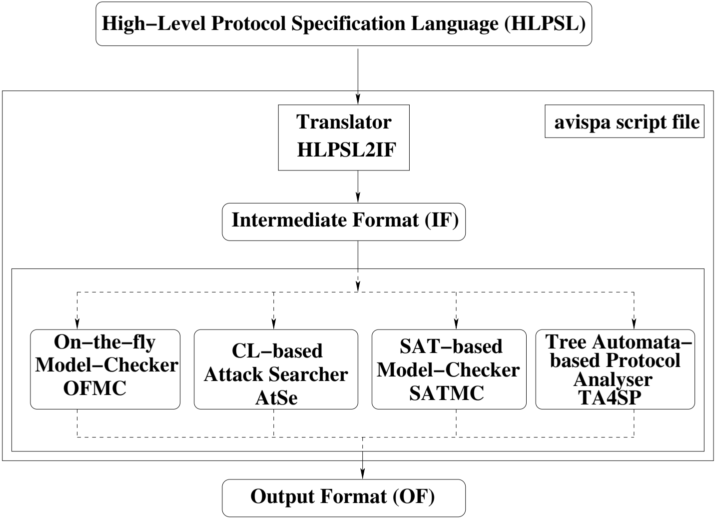
Table: Communication overhead comparison between our scheme and other schemes

Scheme	Communication overhead
M. L. Das [1]	2 messages (704 bits)
Yoo et al. [2]	6 messages (1824 bits)
Sun et al. [3]	5 messages (1296 bits)
Xue et al. [4]	4 messages (2256 bits)
Jiang et al. [5]	4 messages (1920 bits)
Proposed scheme	4 messages (1952 bits)

Formal security verification using AVISPA tool

- AVISPA (Automated Validation of Internet Security Protocols and Applications), is a push-button tool for the automated validation of Internet security-sensitive protocols and applications.
- Consists of four backends:
  - On-the-fly Model-Checker (OFMC) is responsible for performing several symbolic techniques to explore the state space in a demand-driven way.
  - Constraint-Logic-based Attack Searcher (CL-AtSe) provides a translation from any security protocol specification written as transition relation in intermediate format into a set of constraints which are effectively used to find whether there are attacks on protocols.
  - SAT-based Model-Checker (SATMC) builds a propositional formula and then the formula is fed to a state-of-the-art SAT solver to verify whether there is an attack or not.
  - Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP) approximates the intruder knowledge by using regular tree languages.

Formal security verification using AVISPA tool



Formal security verification using AVISPA tool

- Protocols described using the high level language, HLPSL is a role-oriented language.
- Each principal is implemented in transitional roles in which the transitions of a principal takes place during the protocol run as specified. The protocol session is a parallel composition of these transitional roles.
- The intruder is modeled using the Dolev Yao model (according to our threat model) with the possibility for the intruder to assume a legitimate role in a protocol run.
- The role system defines the number of sessions, the number of principals and the roles.

- *agent:* Values of type *agent* represent principal names. The intruder is always assumed to have the special identifier *i*.
- *public_key:* These values represent agents' public keys in a public-key cryptosystem. For example, given a public (respectively private) key *pk*, its inverse private (respectively public) key is obtained by inv(*pk*).
- *symmetric_key:* Variables of this type represent keys for a symmetric-key cryptosystem.
- *text:* In HLPSL, *text* values are often used as nonces. These values can be used for messages. If *Na* is of type *text (fresh)*, then *Na'* will be a fresh value which the intruder cannot guess.
- *nat:* The *nat* type represents the natural numbers in non-message contexts.
- *const:* This type represents constants.
- *hash_func:* The base type *hash_func* represents cryptographic hash functions. The base type function also represents functions on the space of messages. It is assumed that the intruder cannot invert hash functions (in essence, that they are one-way).

Role specification in HLPSL language

- The type declaration *channel* (*dy*) declares that the channel is for the Dolev-Yao threat model (as described in our threat model). In this case, the intruder (*i*) will have the ability to intercept, analyze, and/or modify messages transmitted over the insecure channel.
- witness(A,B,id,E) declares for a (weak) authentication property of *A* by *B* on *E*, declares that agent *A* is witness for the information *E*; this goal will be identified by the constant *id* in the goal section.
- request(B,A,id,E) means for a strong authentication property of *A* by *B* on *E*, declares that agent *B* requests a check of the value *E*; this goal will be identified by the constant *id* in the goal section.
- A message is sent with the *Snd*( ) operation.
- A message is received by the *Rcv*( ) operation.
- The intruder is always denoted by *i*.

Role specification in HLPSL language

- In this implementation, we have three basic roles:
  - *alice* for representing the user U_i
  - *server* for representing the GWN
  - *bob* for representing a sensor node SN_j
- Apart from these, we must have two mandatory roles:
  - *session*: In the session segment, all the basic roles including the roles for U_i, the GWN and SN_j are instanced with concrete arguments.
  - *environment*: The top-level role, which is called the environment, defines in the specification of HLPSL. It contains the global constants and a composition of one or more sessions, where the intruder may play some roles as legitimate users. In HLPSL, the intruder also participates in the execution of protocol as a concrete session.

```
role bob (Sj,BS, U : agent, MKsj : symmetric_key, H : hash_func,
F : hash_func, IDsj, PWi, Bi, S : text, Snd, Rcv: channel(dy))
played_by Sj
def=
 local State  : nat,
     IDi, RNui, RNbs  : text
     const  alice_server, server_bob, bob_server, subs1,
          subs2 : protocol_id
  init  State := 0
 transition
  1. State   = 0 /\ Rcv(BS.Sj.IDsj.IDi.{xor(H(IDi.PWi.F(Bi)),RNui')
                  .H(xor(H(IDi.PWi.F(Bi)),RNui').IDsj.RNui'
                  .RNbs').RNui'.RNbs'}_MKsj)  =|>
   State' := 1 /\ Snd(BS.U.IDi.IDsj.{RNui'}_H(IDi.IDsj.RNui'.
 xor(H(IDi.PWi.F(Bi)),RNui'))) %% Send an acknowledgement to the BS
            /\ secret({PWi,Bi},subs1,U)
            /\ secret(S, subs2, BS)
            /\ request(BS, Sj, server_bob, RNbs)
            /\ request(U, Sj, alice_bob, RNui)
end role
```

Figure: Role specification in HLPSL for the sensor SN_j.

```
role session(U,BS,Sj: agent,
    MKsj : symmetric_key,
          % H is hash function
          H     : hash_func,
          F     : hash_func,
          PWi, Bi, S : text,
          IDi, IDsj, RNui, RNbs :text)
def=

  local  US, UR, SS, SR, VS, VR: channel (dy)

  composition
         alice(U, BS, Sj, MKsj, H, F, IDi, PWi, Bi, S, US, UR)
      /\ server(BS, Sj, U, MKsj, H, F, PWi, Bi, S, SS, SR)
      /\  bob(Sj, BS, U, MKsj, H, F, IDsj, PWi, Bi, S, VS, VR)
end role
```

Figure: Role specification in HLPSL for the session.

Role specification for the goal and environment

```
role environment()
def=

  const u, bs, sj  : agent,
        mksj : symmetric_key,
        h     : hash_func,
        f     : hash_func,
        pwi, bi, s, idi, idsj, rnui, rnbs : text,
       alice_server, server_bob,  bob_server,
       alice_bob, subs1, subs2 : protocol_id

  intruder_knowledge = {u, bs, sj, h, f, idi, idsj}

  composition
session(u, bs, sj, mksj, h, f, pwi, bi, s,
                idi, idsj, rnui, rnbs) /\
session(u, bs, sj, mksj, h, f, pwi, bi, s,
                idi, idsj, rnui, rnbs) /\
        session(u, bs, sj, mksj, h, f, pwi, bi, s,
                idi, idsj, rnui, rnbs)

end role

goal
  secrecy_of subs1
  secrecy_of subs2
  authentication_on alice_bob
  authentication_on alice_server
  authentication_on server_bob
  authentication_on bob_server
end goal

environment()
```

Role specification in HLPSL language

- secret ({PWi,Bi, K}, subs1, Ui) declaration tells that *PWi*, *Bi*, *K* are kept to the user U_i only, which is characterized by the protocol id subs1.
- witness (SNj, GWN, bob_server_ts3, TS3') tells that SN_j has freshly generated the value TS_3 for the GWN.
- request(GWN, SNj, server_bob_ts2, TS2') is meant for SN_j's acceptance of the value TS_2, which was generated for SN_j by the GWN.
- secrecy_of subs1: It represents that *PWi*, *Bi*, *K* are kept secret to the user U_i only.
- Similarly for others: subs2, subs3, sub4, subs5
- authentication_on alice_server_ts1: U_i (the smart card) generates a timestamp TS_1. When the GWN receives TS_1 from the message from U_i, the *GWN* authenticates U_i based on TS_1.

Result of the analysis using OFMC backend

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\auth.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.07s
  visitedNodes: 8 nodes
  depth: 3 plies
```

Result of the analysis using CL-AtSe backend

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  C:\progra~1\SPAN\testsuite\results\auth.if

GOAL
  As Specified

BACKEND
  CL–AtSe

STATISTICS

  Analysed    : 63 states
  Reachable   : 15 states
  Translation: 0.09 seconds
  Computation: 0.00 seconds
```

Summary

- We have proposed a user authentication and key agreement scheme using biometric, password and smart card of a legal user for large-scale distributed wireless sensor networks.
- The proposed scheme allows the user to authenticate at both the *GWN* and the sensor nodes inside WSN.
- After successful authentication, both the user and the sensor node from which user wants to access real-time data in the target field, will be able to establish a secret session key between them. Later using this session key, the user can contact the sensor node directly for real-time data inside WSN.
- The proposed scheme supports password and biometric change phase by the user at any time locally without contacting the *GWN*.
- The proposed scheme provides better security features and higher security level than other schemes, which are demonstrated through the formal and informal security analysis.
- Overall, considering better security features and higher security level, and efficiency that our scheme provides, we conclude that our scheme is more appropriate for practical applications such as healthcare and battlefield applications of WSNs as compared to other existing approaches.

Important References

- C.-C. Chang and H.-D. Le, "A Provably Secure, Efficient and Flexible Authentication Scheme for Ad hoc Wireless Sensor Networks," **IEEE Transactions on Wireless Communications**, 2015, DOI: 10.1109/TWC.2015.2473165.
- Ashok Kumar Das, Santanu Chatterjee, and Jamuna Kanta Sing. "A New Biometric-Based Remote User Authentication Scheme in Hierarchical Wireless Body Area Sensor Networks," in **Ad Hoc & Sensor Wireless Networks (Old City Publishing)**, Vol. 28, No. 3-4, pp. 221-256, 2015. (2012 SCI Impact Factor: 0.41)
- Ashok Kumar Das. "A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor," in **International Journal of Communication Systems (Wiley)**, 2015, In Press, DOI: 10.1002/dac.2933. (2013 SCI Impact Factor: 1.106)

Important References (Continued...)

- Ashok Kumar Das. "A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks," in **Peer-to-Peer Networking and Applications (Springer)**, 2015, In Press, DOI: 10.1007/s12083-014-0324-9. (2014 SCI Impact Factor: 0.632) [This article is one of the top five most popular downloaded articles during December 2014 to January 2015 of the Peer-to-Peer Networking and Applications.]
- J. Yuan, C. Jiang, and Z. Jiang. A Biometric-Based User Authentication for Wireless Sensor Networks. **Wuhan University Journal of Natural Sciences**, 15(3):272-276, 2010.
- Ashok Kumar Das, Pranay Sharma, Santanu Chatterjee, and Jamuna Kanta Sing. "A dynamic password-based user authentication scheme for hierarchical wireless sensor networks," in **Journal of Network and Computer Applications (Elsevier)**, Vol. 35, No. 5, pp. 1646 - 1656, 2012, doi:10.1016/j.jnca.2012.03.011. [This article is one of the top 25 most downloaded articles during April 2012 to December 2012 of the Journal of Network and Computer Applications.] (2014 SCI Impact Factor: 2.229)

Important References (Continued...)

- Das, M.L.: Two-Factor User Authentication in Wireless Sensor Networks. **IEEE Transactions on Wireless Communications** 8(3), 1086–1090 (2009)
- Yoo, S.G., Park, K.Y., Kim, J.: A Security-Performance-Balanced User Authentication Scheme for Wireless Sensor Networks. **International Journal of Distributed Sensor Networks** 2012 (2012). Article ID 382810, 11 pages, 2012. doi:10.1155/2012/382810
- Sun, D.Z., Li, J.X., Feng, Z.Y., Cao, Z.F., Xu, G.Q.: On the security and improvement of a two-factor user authentication scheme in wireless sensor networks. **Personal and Ubiquitous Computing** 17(5), 895–905 (2013)
- Xue, K., Ma, C., Hong, P., Ding, R.: A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. **Journal of Network and Computer Applications** 36(1), 316–323 (2013)

Important References (Continued...)

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

- Ashok Kumar Das. “A Secure and Efficient User Anonymity-Preserving Three-Factor Authentication Protocol for Large-Scale Distributed Wireless Sensor Networks,” in **Wireless Personal Communications (Springer)**, Vol. 82, No. 3, pp. 1377 - 1404, 2015, DOI: 10.1007/ s11277-015-2288-3. (2013 SCI Impact Factor: 0.979)
- Ashok Kumar Das. “An efficient and novel three-factor user authentication scheme for large-scale heterogeneous wireless sensor networks,” in **International Journal of Communication Networks and Distributed Systems (Inderscience)**, Vol. 15, No. 1, pp. 22-60, 2015.
- Q. Jiang, J. Ma, X. Lu, Y. Tian: An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. **Peer-to-Peer Networking and Applications** pp. 8(6), 1070–1081 (2015)

Important References (Continued...)

- P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain. “Lightweight and secure session-key establishment scheme in smart home environments,” **IEEE Sensors Journal**, Vol. 16, No. 1, pp. 254-264, 2016.
- J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami. “Internet of Things (IoT): A vision, architectural elements, and future directions,” **Future Generation Computer Systems**, Vol. 29, No. 7, pp. 1645-1660, 2013.

Thank you