Signature-Based Authentication in Future Internet of Things (IoT) Applications

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 25, 2022

Internet of Things (IoT)

- A “thing” in the IoT can be a person, animal or physical/virtual object with a unique identifier (IP address or device ID) that has the ability to transfer data (sensing information from surrounding area) via the Internet.
  - *Physical object:* Smartphone, camera, sensor, vehicle, drone, etc.
- The “Things” in IoT usually referes to IoT devices. IoT devices can perform remote sensing, actuating (making an action), and monitoring capabilities.
- A thing can be smart and thus, the thing can make a decision without human’s help (intervention).
  Majority of things are expected to be smart in the future.
- The objective of IoT is to integrate computer-based systems and the physical world for economic benefit and to improve accuracy and efficiency while reducing human involvement.
- An estimated 50 billion objects will be a part of IoT by 2020.

Internet of Things (IoT)

Table: IoT units installed based by category (millions of units)

Category	2016	2017	2018	2020
Consumer	3,963.00	5,244.30	7,036.30	12,863.00
Business: cross-industry	1,102.10	1,501	2,132.60	4,381.40
Business: vertical-specific	1,316.60	1,635.40	2,027.70	3,171
Grand total	6,381.80	8,380.60	11,196.60	20,415.40

Table: IoT endpoint spending by category (millions of dollars)

Category	2016	2017	2018	2020
Consumer	532,515	725,696	985,384	1,494,466
Business: cross-industry	212,069	280,059	372,989	567,659
Business: vertical-specific	634,921	683,817	736,543	863,662
Grand total	1379,505	1,689,572	2,094,881	2,925,787

Ref. Information Matters. The Business of Data and the Internet of Things (IoT). http://informationmatters.net/internet-of-things-statistics/. Accessed on August 2018.

Signature-Based Authentication in Future Internet of Things (IoT) Applications

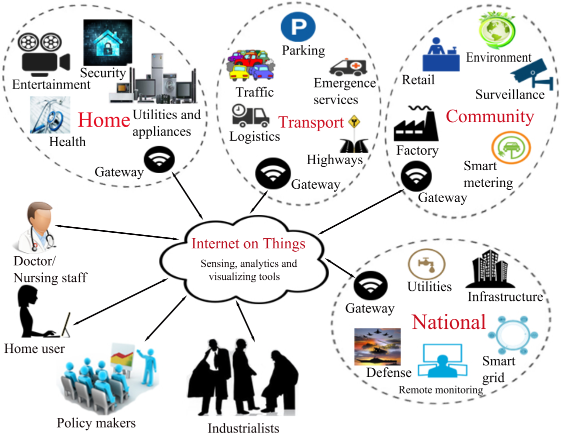
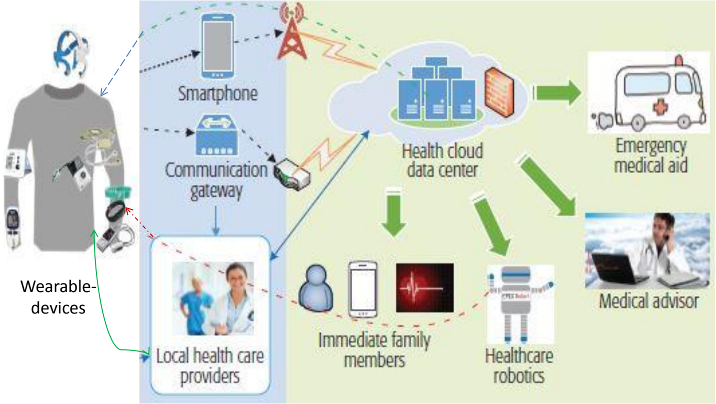


Figure: IoT authentication model

IoT authentication model

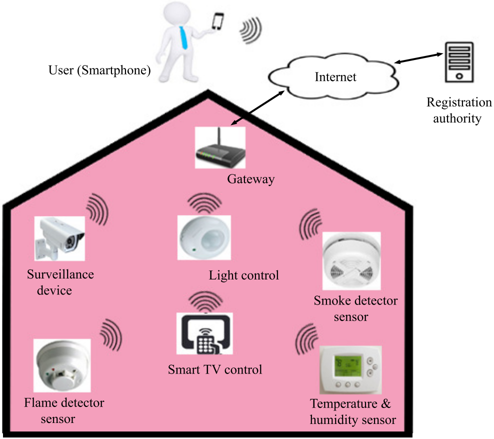
- IoT authentication model considers four different scenarios, i.e., Home, Transport, Community and National.
- All these scenarios have smart devices, such as sensors and actuators, which facilitate the day to day life of people.
- In the given scenarios, all smart devices are connected to the Internet through the gateway nodes (*GWN*s).
- Different types of users (for example, smart home user and doctor) can access the data of relevant IoT devices through the *GWN*.
- Mutual authentication between a user and a device through the *GWN* provides access to device data to the user.

IoT Applications: Healthcare



Jangirala Srinivas, Ashok Kumar Das, Neeraj Kumar, and Joel J. P. C. Rodrigues. “Cloud Centric Authentication for Wearable Healthcare Monitoring System,” in **IEEE Transactions on Dependable and Secure Computing**, Vol. 17, No. 5, pp. 942-956, September/October 2020, DOI: 10.1109/TDSC.2018.2828306.

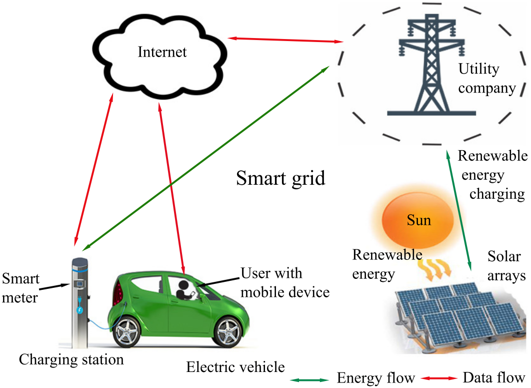
IoT Applications: Smart Home



Smart Home

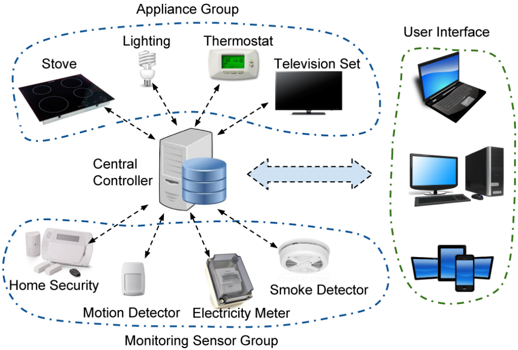
Mohammad Wazid, Ashok Kumar Das, Vanga Odelu, Neeraj Kumar, and Willy Susilo. “Secure Remote User Authenticated Key Establishment Protocol for Smart Home Environment,” in **IEEE Transactions on Dependable and Secure Computing**, Vol. 17, No. 2, pp. 391-406, 2020, DOI: 10.1109/TDSC.2017.2764083.

IoT Applications: Renewable Energy-Based Smart Grid



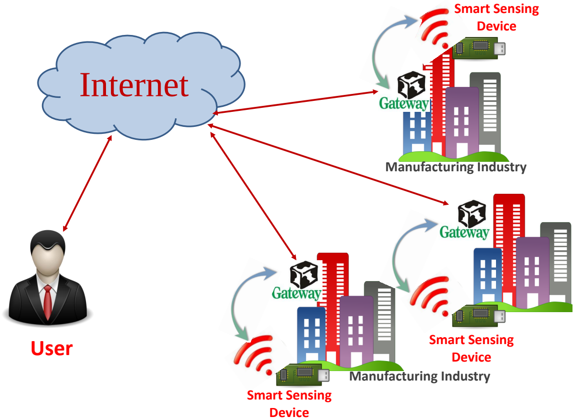
Mohammad Wazid, Ashok Kumar Das, Neeraj Kumar, and Joel J. P. C. Rodrigues. "Secure Three-factor User Authentication Scheme for Renewable Energy Based Smart Grid Environment," in **IEEE Transactions on Industrial Informatics**, Vol. 13, No. 6, pp. 3144-3153, 2017, DOI: 10.1109/TII.2017.2732999.

IoT Applications: Smart Home



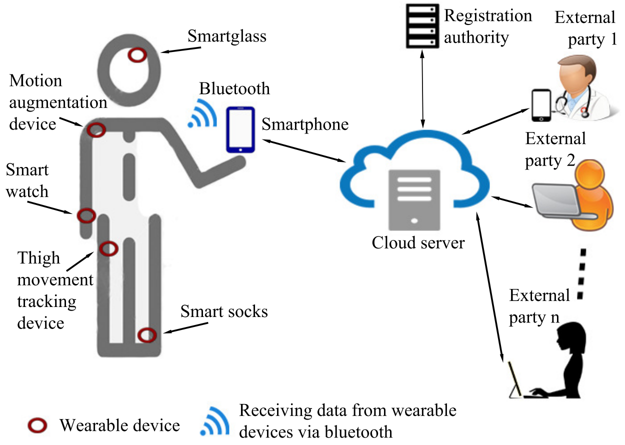
Mohammad Wazid, Ashok Kumar Das, Vanga Odelu, Neeraj Kumar, Mauro Conti, and Minho Jo. "Design of Secure User Authenticated Key Management Protocol for Generic IoT Network," in *IEEE Internet of Things Journal*, Vol. 5, No. 1, pp. 269-282, 2018.

IoT Applications: Industrial IoT



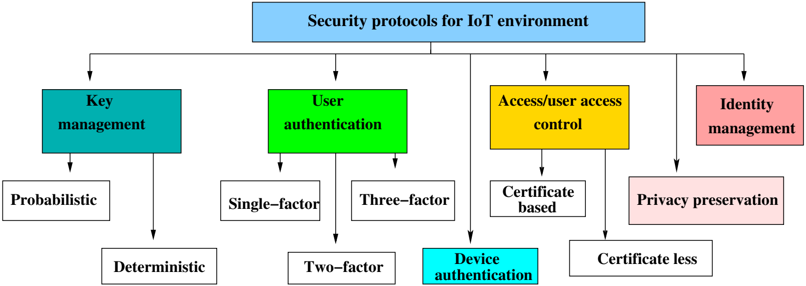
Jangirala Srinivas, Ashok Kumar Das, Mohammad Wazid, and Neeraj Kumar. "Anonymous Lightweight Chaotic Map-Based Authenticated Key Agreement Protocol for Industrial Internet of Things," in *IEEE Transactions on Dependable and Secure Computing*, ol. 17, No. 6, pp. 1133-1146, 2020, DOI: 10.1109/TDSC.2018.2857811.

IoT Applications: Healthcare



Ashok Kumar Das, Mohammad Wazid, Neeraj Kumar, Muhammad Khurram Khan, Kim-Kwang Raymond Choo, and YoungHo Park. “Design of Secure and Lightweight Authentication Protocol for Wearable Devices Environment,” in **IEEE Journal of Biomedical and Health Informatics (Formerly, IEEE Transactions on Information Technology in Biomedicine)**, Vol. 22, No. 4, pp. 1310-1322, 2018.

Taxonomy of security protocols in IoT



Ashok Kumar Das, Sherali Zeadally, and Debiao He. "Taxonomy and Analysis of Security Protocols for Internet of Things," in *Future Generation Computer Systems (Elsevier)*, Vol. 89, pp. 110-125, 2018, DOI: 10.1016/j.future.2018.06.027.

Security requirements in IoT environment

- **Authentication:** It involves authentication of sensing devices, users and gateway nodes before allowing access to a restricted resource, or revealing crucial information.
- **Integrity:** The message or the entity under consideration must not be changed to ensure integrity.
- **Confidentiality:** Confidentiality or privacy of the wireless communication channel protects from the unauthorized disclosure of information.
- **Availability:** The relevant network services should be made available to authorized users even under denial-of-service attacks on the system.
- **Non-repudiation:** It aims to prevent a mischievous entity from hiding his/her actions.
- **Authorization:** It confirms that only the legitimate IoT sensing (smart) devices can supply information to network services.
- **Freshness:** It confirms that the information is fresh and the old messages cannot be replayed by any adversary.
- Apart from the above security requirements, the following two important security properties should also be satisfied:
  - **Forward secrecy:** If an IoT sensing node quits the network, any future messages after its exit must be prohibited.
  - **Backward secrecy:** If a new IoT sensing node is added in the network, it must not read any previously transmitted message.

Security attacks in IoT environment

- **Replay attack:** A replay attack is one in which an adversary, $\mathcal{A}$ attempts to mislead another authorized entity by reusing the information during the transmission.
- **Man-in-the-middle attack:** Under such an attack, $\mathcal{A}$ intercepts the transmitted messages and tries to change/delete/modify the contents of the messages delivered to the recipients.
- **Stolen-verifier attack:** This attack can occur if the *GWN* in the IoT network stores any verifier/password table for user/device verification. In such an attack, $\mathcal{A}$ can steal a user's credentials such as identity or password from the table.
- **Stolen/lost smart card attack:** If $\mathcal{A}$ has a lost/stolen smart card, he/she can extract all the credentials stored into its memory by using techniques such as power analysis attacks. Using the extracted information, $\mathcal{A}$ can then derive the secret credentials.
- **Password guessing attack:** In a password-based scheme, $\mathcal{A}$ may attempt to guess the password of a legal registered user either online or offline mode with the help of the eavesdropped messages and also stored credentials in the system or a user's smart card (mobile device).

Security attacks in IoT environment (Cont...)

- **Password change attack:** Under this attack, $\mathcal{A}$ may try to change the password of an authorized registered user.
- **Denial-of-Service attack:** A Denial-of-Service (DoS) attack is any event that prevents a system's or network' capability to perform its expected function.
- **Privileged-insider attack:** In this kind of attack, a trusted user within the organization (also known as an insider) can act as a privileged-insider attacker.
- **Impersonation attack:** In an impersonation attack, an attacker may attempt to falsify a fake message to defraud other recipient entities in a network on behalf of a sending entity.
- **Resilience against smart device physical capture attack:** In IoT environment, except the *GWN* the IoT sensing devices are not physically protected. Hence, there is a possibility of physical capturing of the sensing devices by an attacker $\mathcal{A}$. $\mathcal{A}$ can then use the extracted information stored in those captured sensing devices to compromise communication between other non-compromised sensing devices.

Signature-Based Authentication in Future Internet of Things (IoT) Applications

- Sravani Challa, Mohammad Wazid, Ashok Kumar Das, Neeraj Kumar, Alavalapati Goutham Reddy, Eun-Jun Yoon, and Kee-Young Yoo, “Secure Signature-Based Authenticated Key Establishment Scheme for Future IoT Applications,” in **IEEE Access**, Vol. 5, pp. 3028-3043, 2017. **[This article is one of the top 50 most frequently downloaded documents for Popular Articles (May-June 2017)]**

Threat model

- We follow the widely-accepted Dolev-Yao threat (DY) model [1].
- Under the DY model, communication between two entities is performed over a public channel.
- An adversary can then have an opportunity to eavesdrop, modify or delete the content of the messages being transmitted.
- An adversary can physically capture one or more sensing devices in IoT, and can extract all the sensitive information stored in the captured devices using the power analysis attacks [2], [3].

[1] D. Dolev and A. Yao, "On the security of public key protocols," ***IEEE Transactions on Information Theory***, vol. 29, no. 2, pp. 198–208, 1983.
[2] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," ***IEEE Transactions on Computers***, vol. 51, no. 5, pp. 541–552, 2002.
[3] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in 19th Annual IACR Crypto Conference (Advances in Cryptology) - ***CRYPTO'99***, LNCS, vol. 1666, Santa Barbara, California, USA, 1999, pp. 388–397.

Contributions in this work

- An authentication model for IoT is presented and the security challenges involved and its requirements are discussed.
- A secure signature-based authentication and key agreement scheme has been proposed to address these issues.
- A formal security analysis using the widely-used Burrows-Abadi-Needham logic (BAN logic) and an informal security analysis have been presented to prove that the scheme is secure.
- Simulation using the broadly-accepted Automated Validation of Internet Security Protocols and Applications (AVISPA) tool for the formal verification of the scheme's security has also been provided.
- Using NS2 simulator, the scheme's impact on network performance parameters has been measured for practical demonstration of the scheme.
- Finally, it has been shown that the scheme is also efficient in terms of communication and computation costs.

Notations

Symbol	Description
GWN	Gateway node
SD_j	j^{th} sensing device
ID_j	SD_j's identity
U_i	i^{th} user
SC_i	U_i's smart card
ID_i	U_i's identity
PW_i	U_i's password
BIO_i	U_i's personal biometrics template
σ_i	Biometric secret key
τ_i	Biometric public reproduction parameter
t	Error tolerance threshold used by fuzzy extractor
$Gen(\cdot)$	Probabilistic generation procedure used by fuzzy extractor
$Rep(\cdot)$	Deterministic reproduction procedure used by fuzzy extractor
$h(\cdot)$	Collision-resistant one-way cryptographic hash function

Notations (Continued...)

Symbol	Description
p	A large prime number
Z_p	$Z_p = \{0, 1, \cdots, p-1\}$, a prime finite field
E_p	An elliptic curve over prime field Z_p
$P = ((P)_x, (P)_y)$	An elliptic curve point in elliptic curve E_p, $(P)_x$ and $(P)_y$ are x and y coordinates of P, respectively
$k.P$	Elliptic curve point multiplication; $k \in Z_p^*$ being a scalar and $P \in E_p$
d	private key of involved entities
Q	$Q = d.P$, public key of involved entities
T_i, T_s	Current system timestamps
ΔT	Maximum transmission delay
sk_{ij}	Session key between U_i and SD_j
$\oplus$, $\|\|$	Bitwise XOR and concatenation operations, respectively

Elliptic Curve Cryptography (ECC)

Elliptic curves over modulo a prime $GF(p)$

Let $p > 3$ be a prime. The elliptic curve $y^2 = x^3 + ax + b$ over Z_p is the set $E_p(a, b)$ of solutions $(x, y) \in E_p(a, b)$ to the congruence

$$y^2 = x^3 + ax + b \pmod p,$$

where $a, b \in Z_p$ are constants such that $4a^3 + 27b^2 \neq 0 \pmod p$, together with a special point $\mathcal{O}$ called the point at infinity (or zero point).

Properties of Elliptic Curves

- An elliptic curve $E_p(a, b)$ over Z_p (p prime, $p > 3$) will have roughly p points on it.
- More precisely, a well-known theorem due to Hasse asserts that the number of points on $E_p(a, b)$, which is denoted by $\#E$, satisfies the following inequality:

$$p + 1 - 2\sqrt{p} \leq \#E \leq p + 1 + 2\sqrt{p}.$$

- In addition, $E_p(a, b)$ forms an abelian or commutative group under addition modulo p operation.

Elliptic Curve Cryptography (ECC)

- **Point addition on elliptic curve over finite field $GF(p)$**
  If $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be two points on elliptic curve $y^2 = x^3 + ax + b \pmod p$, $R = (x_R, y_R) = P + Q$ is computed as follows:

$$x_R = (\lambda^2 - x_P - x_Q) \pmod p,$$
$$y_R = (\lambda(x_P - x_R) - y_P) \pmod p,$$
$$\text{where } \lambda = \begin{cases} \frac{y_Q - y_P}{x_Q - x_P} \pmod p, \text{if } P \neq -Q \\ \frac{3{x_P}^2 + a}{2y_P} \pmod p, \text{if } P = Q. \end{cases}$$

- **Scalar/point multiplication on elliptic curve over finite field $GF(p)$**
  If $P = (x_P, y_P)$ be a point on elliptic curve $y^2 = x^3 + ax + b \pmod p$, then $5P$ is computed as $5P = P + P + P + P + P$.

Elliptic Curve Cryptography (ECC)

Definition (Elliptic Curve Discrete Logarithm Problem (ECDLP))

Let $E_p(a, b)$ be an elliptic curve modulo a prime p. Given two points $P \in E_p(a, b)$ and $Q = kP \in E_p(a, b)$, for some positive integer k, where $Q = kP$ represent the point P on elliptic curve $E_p(a, b)$ be added to itself k times. Then the elliptic curve discrete logarithm problem (ECDLP) is to determine k given P and Q.

Definition (Elliptic Curve Decisional Diffie-Hellman Problem (ECDDHP))

Let $E_p(a, b)$ be an elliptic curve and $G \in E_p(a, b)$ be a base point. The elliptic curve decisional Diffie-Hellman problem (ECDDHP) is defined as follows. Given a quadruple $(G, u.G, v.G, w.G)$, decides whether $w = u.v \pmod{p}$.

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 $\sigma_i \in \{0, 1\}^l$ 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: Hamming distance $d(B_i, B_i') \leq et$ is met.

One of the estimations on error tolerance threshold values provided by Cheon *et al.* is as follows: If the Hamming distance between the original biometric template B_i and current biometric template B_i' is h_T and the number of bits in input biometric is n_b, we then have $et = \frac{h_T}{n_b}$.

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. **[This article is one of the top 50 most frequently downloaded documents for Popular Articles (June - November 2015)]**

System Setup Phase

- *GWN* chooses a non-singular elliptic curve E_p over $GF(p)$ and a base point P of order n as large as the prime p.
- *GWN* also picks its private key d_{GWN} and computes the corresponding public key $Q_{GWN} = d_{GWN}.P$.
- *GWN* then chooses a collision-resistant one-way cryptographic hash function $h(\cdot)$, fuzzy extractor functions $Gen(\cdot)$ and $Rep(\cdot)$.
- The system parameters $\{E_p(a, b), p, P, h(\cdot), Q_{GWN}, Gen(\cdot), Rep(\cdot), t\}$ are made public, whereas d_{GWN} is kept secret by *GWN*.

Sensing Device Registration Phase

All the sensing devices in IoT are registered offline by the *GWN* as follows.

- For each device SD_j, the *GWN* chooses a unique identity ID_j and a unique private key d_j, and calculates the corresponding public key $Q_j = d_j.P$. It further computes $RID_j = h(ID_j \parallel d_j)$.
- The *GWN* pre-loads $\{ID_j, d_j, RID_j\}$ in the memory of SD_j. Furthermore, the *GWN* stores $\{ID_j, RID_j, Q_j\}$ in its database, and then makes Q_j as public.

User Registration Phase

User (U_i)	Gateway node (GWN)
Select identity ID_i, private key d_i.	
Compute public key $Q_i = d_i.P$	
$RID_i = h(d_i \parallel ID_i)$.	
$\xrightarrow{\langle RID_i \rangle}$	
(Secure channel)	Compute
	$R_i = h(RID_i \parallel d_{GWN})$.
	$\xleftarrow{\langle \text{Smart Card}\{R_i\} \rangle}$
Select password PW_i.	(Secure channel)
Imprint personal biometric Bio_i.	
Compute $Gen(Bio_i) = (\sigma_i, \tau_i)$,	
$RPW_i = h(PW_i \parallel d_i \parallel ID_i \parallel \sigma_i)$,	
$R_i^* = R_i \oplus h(ID_i \parallel PW_i \parallel \sigma_i)$,	
$d_i^* = d_i \oplus h(ID_i \parallel \sigma_i)$.	
Insert $\{d_i^*$, RPW_i, τ_i, t, $h(\cdot)$,	
$Gen(\cdot)$ and $Rep(\cdot)\}$ into smart card.	
Replace R_i with R_i^* in smart card.	

Login Phase

User (U_i) $\{RPW_i, d_i^*, R_i^*, Gen(\cdot), Rep(\cdot), \tau_i, h(\cdot), t\}$	Gateway Node (GWN) $\{ID_j, RID_j, Q_j, d_{GWN}\}$
Enter ID_i' and PW_i'. Imprint Bio_i'. Compute $\sigma_i' = Rep(Bio_i', \tau_i)$, $d_i' = d_i^* \oplus h(ID_i' \parallel \sigma_i')$, $RPW_i' = h(PW_i' \parallel ID_i' \parallel d_i' \parallel \sigma_i')$. Check if $RPW_i' = RPW_i$? Choose random $a \in Z_p^*$. Generate timestamp T_i. Compute $A_i = a.P$, $N_i = a.Q_{GWN} = ((N_i)_x, (N_i)_y)$, $RID_i' = h(d_i' \parallel ID_i')$, $R_i' = R_i^* \oplus h(ID_i' \parallel PW_i \parallel \sigma_i')$, $DID_i' = RID_i' \oplus (N_i)_y$, $DID_j' = ID_j \oplus (N_i)_y$, $V_i = h(ID_j \parallel T_i \parallel N_i \parallel R_i')$, $r_i = (N_i)_x$, $s_i = a^{-1}(V_i + r_i d_i')$. $\xrightarrow{\langle DID_i', DID_j', A_i, T_i, r_i, s_i \rangle}$ (via public channel)	

Authentication and Key Agreement Phase

Gateway Node (GWN)	Sensing Device (SD_j)
$\{ID_j, RID_j, Q_j, d_{GWN}\}$	$\{ID_j, d_j, RID_j\}$

Check if $T_i' - T_i \leq \Delta T$?
Compute $N_{GWN} = d_{GWN}.A_i = ((N_{GWN})_x, (N_{GWN})_y)$,
$RID_i^* = DID_i' \oplus (N_{GWN})_y$, $ID_j^* = DID_j' \oplus (N_{GWN})_y$.
Check if $ID_j^* = ID_j$? If so, compute $R_i = h(RID_i^* \parallel d_{GWN})$,
$V_i^* = h(ID_j^* \parallel T_i \parallel N_{GWN} \parallel R_i)$.
Verify U_i's signature by computing $w_{GWN} = s_i^{-1} \pmod p$,
$u_{GWN} = V_i^* w_{GWN} \pmod p$, $t_{GWN} = r_i w_{GWN} \pmod p$,
$N_i^* = (u_{GWN}.P + t_{GWN}.Q_i)d_{GWN} = ((N_i^*)_x, (N_i^*)_y)$.
Check if $(r_i^* = (N_i^*)_x) = ((N_i)_x = r_i)$?
Choose random $c \in Z_p^*$. Generate timestamp T_{GWN}.
Compute $C_{GWN} = c.P = ((C_{GWN})_x, (C_{GWN})_y)$,
$V_{GWN} = h(R_i \parallel T_i) \oplus h(A_i \parallel RID_j \parallel T_{GWN} \parallel T_i)$,
$r_{GWN} = (C_{GWN})_x$, $s_{GWN} = c^{-1}(h(R_i \parallel T_i) +$
$r_{GWN} d_{GWN}) \pmod p$.
$\xrightarrow{\langle V_{GWN}, T_{GWN}, T_i, A_i, C_{GWN}, s_{GWN} \rangle}$
(via public channel)

Authentication and Key Agreement Phase

- Note that $N_i^* = (u_{GWN}.P + t_{GWN}.Q_i)d_{GWN}$
$= ((N_i^*)_x, (N_i^*)_y)$.
Now,

$$(u_{GWN}.P + t_{GWN}.Q_i)d_{GWN}$$
$$\begin{aligned} &= (((V_i^* P)/s_i) + (((r_i d_i).P)/s_i))d_{GWN} \\ &= (1/s_i)(V_i^* + r_i d_i)d_{GWN}.P \\ &= (1/s_i)(as_i)d_{GWN}.P \\ &= a.Q_{GWN} \\ &= N_i \\ &= ((N_i)_x, (N_i)_y). \end{aligned}$$

- Hence, $r_i^* = (N_i^*)_x = (N_i)_x = r_i$.

User (U_i)	Sensing Device (SD_j)
	Check if $T'_{GWN} - T_{GWN} \leq \Delta T$?
	Compute $h(R_i \parallel T_i) = V_{GWN} \oplus$
	$h(A_i \vert\vert RID_j \vert\vert T_{GWN} \vert\vert T_i)$.
	Verify GWN's signature by computing
	$w_{SD_j} = s_{GWN}^{-1} \pmod p$,
	$u_{SD_j} = h(R_i \parallel T_i) w_{SD_j} \pmod p$,
	$r_{GWN} = (C_{GWN})_x$, $t_{SD_j} = r_{GWN} w_{SD_j} \pmod p$,
Check if $T'_j - T_j \leq \Delta T$?	$C^*_{GWN} = u_{SD_j}.P + t_{SD_j}.Q_{GWN}$
Compute $k'_{ij} = a.B_{SD_j} = a.(b.P)$,	$= ((C^*_{GWN})_x, (C^*_{GWN})_y)$.
$sk'_{ij} = h(ID_j \parallel h(R^*_i \parallel T_i) \parallel k'_{ij} \parallel T_i \parallel T_j)$,	Check if $(r^*_{GWN} = (C^*_{GWN})_x) =$
Verify SD_j's signature by computing	$((C_{GWN})_x = r_{GWN})$?
$w_i = s_{SD_j}^{-1} \pmod p$,	Generate random $b \in Z_p^*$, timestamp T_j.
$u_i = h(sk'_{ij}) w_i \pmod p$, $r_{SD_j} = (B_{SD_j})_x$,	Compute $k_{ij} = b.A_i = b.(a.P)$,
$t_i = r_{SD_j} w_i \pmod p$,	$sk_{ij} = h(ID_j \parallel h(R_i \parallel T_i) \parallel k_{ij} \parallel T_i \parallel T_j)$,
$B^*_{SD_j} = u_i.P + t_i.Q_j = ((B^*_{SD_j})_x, (B^*_{SD_j})_y)$.	$B_{SD_j} = b.P = ((B_{SD_j})_x, (B_{SD_j})_y)$,
Check if $(r^*_{SD_j} = (B^*_{SD_j})_x) = ((B_{SD_j})_x = r_{SD_j})$?	$r_{SD_j} = (B_{SD_j})_x$,
	$s_{SD_j} = b^{-1}(h(sk_{ij}) + r_{SD_j} d_j) \pmod p$.
	$\xleftarrow{\langle B_{SD_j}, s_{SD_j}, T_j \rangle}$
	(public channel)
Store the session key sk'_{ij} shared with SD_j.	Store the session key sk_{ij} shared with U_i.

Authentication and Key Agreement Phase

- $C^*_{GWN} = u_{SD_j}.P + t_{SD_j}.Q_{GWN}$
  $= h(R_i \parallel T_i)w_{SD_j}.P + r_{GWN}w_{SD_j}(d_{GWN}.P)$
  $= w_{SD_j}(h(R_i \parallel T_i) + r_{GWN}d_{GWN}).P$
  $= (1/s_{GWN})(c.s_{GWN}).P$
  $= c.P$
  $= C_{GWN}$
  $= ((C_{GWN})_x, (C_{GWN})_y)$.
  Hence, $r^*_{GWN} = (C^*_{GWN})_x = (C_{GWN})_x = r_{GWN}$.
- $B^*_{SD_j} = u_i.P + t_i.Q_j$
  $= (h(sk'_{ij})w_i).P + (r_{SD_j}w_id_j).P$
  $= w_i(h(sk'_{ij}) + r_{SD_j}d_j).P$
  $= (1/s_{SD_j})(b.s_{SD_j}).P$
  $= b.P$
  $= ((B_{SD_j})_x, (B_{SD_j})_y)$.
  Hence, $r^*_{SD_j} = (B^*_{SD_j})_x = (B_{SD_j})_x = r_{SD_j}$.

Password and Biometric Update Phase

User (U_i)	Smart card (SC_i)
Enter $ID_i, PW_i^{old}, Bio_i^{old}$.	
$\xrightarrow{\{ID_i, PW_i^{old}, Bio_i^{old}\}}$	
	Compute $\sigma_i^{old} = Rep(Bio_i^{old}, \tau_i)$, $d_i' = d_i^* \oplus h(ID_i \parallel \sigma_i^{old})$, $R_i' = R_i^* \oplus h(ID_i \vert\vert PW_i^{old} \vert\vert \sigma_i^{old})$, $RPW_i^{old} = h(PW_i^{old} \parallel d_i' \parallel ID_i \parallel \sigma_i^{old})$.
Enter PW_i^{new}, Bio_i^{new}.	If $RPW_i^{old} = RPW_i$ does not hold, terminate.
	$\xleftarrow{\{\text{Permit user to change password/biometric}\}}$
$\xrightarrow{\{PW_i^{new}, Bio_i^{new}\}}$	
	Compute $Gen(Bio_i^{new}) = (\sigma_i^{new}, \tau_i^{new})$, $RPW_i^{new} = h(PW_i^{new} \parallel d_i' \parallel ID_i \parallel \sigma_i^{new})$, $(d_i^*)^{new} = d_i' \oplus h(ID_i \parallel \sigma_i^{new})$, $(R_i^*)^{new} = R_i' \oplus h(ID_i \vert\vert PW_i^{new} \vert\vert \sigma_i^{new})$. Replace the old values RPW_i, d_i^*, R_i^* and τ_i with new ones RPW_i^{new}, $(d_i^*)^{new}$, $(R_i^*)^{new}$, and τ_i^{new}, respectively.

Smart Card Revocation Phase

User (U_i)	Gateway node (GWN)
Select d_i^{new}. Enter current ID_i. Compute $Q_i^{new} = d_i^{new}.P$ $RID_i^{new} = h(d_i^{new} \parallel ID_i)$. $\xrightarrow{\langle RID_i^{new} \rangle}$	Compute $R_i^{new} = h(RID_i^{new} \parallel d_{GWN})$. $\xleftarrow{\langle \text{Smart Card}\{R_i^{new}\} \rangle}$
Use current PW_i and Bio_i. Enter PW_i and imprint Bio_i. Compute $Gen(Bio_i) = (\sigma_i, \tau_i)$, $RPW_i^{new} = h(PW_i \parallel d_i^{new} \parallel ID_i \parallel \sigma_i)$, $Q_i^{new} = d_i^{new}.P$, $(R_i^*)^{new} = R_i^{new} \oplus h(ID_i \parallel PW_i \parallel \sigma_i)$, $(d_i^*)^{new} = d_i^{new} \oplus h(ID_i \parallel \sigma_i)$. Insert $\{(d_i^*)^{new}$, RPW_i^{new}, τ_i, t, $h(\cdot)$, $(R_i^*)^{new}$ $Gen(\cdot)$ and $Rep(\cdot)\}$ into smart card. Make Q_i^{new} public.	

Dynamic Sensing Device Addition Phase

Suppose a new sensing device SD_j^{new} is to be deployed in the network. The GWN then performs the following steps offline:

- The GWN chooses a unique identity ID_j^{new} and a unique private key d_j^{new}, and calculates the corresponding public key $Q_j^{new} = d_j^{new}.P$. It further computes $RID_j^{new} = h(ID_j^{new} \parallel d_j^{new})$.
- The GWN pre-loads RID_j^{new} in the memory of SD_j^{new}. In addition, the GWN stores $\{ID_j^{new}, RID_j^{new}, Q_j^{new}\}$ in its database, and also makes Q_j^{new} public.

After the deployment of SD_j^{new}, the GWN informs the users in the network so that they can access SD_j^{new} using the login and authentication & key agreement phases, respectively.

Security Analysis

- **BAN logic Proof:**

Theorem

The proposed scheme provides secure mutual authentication between a legal user U_i and a sensing device SD_j.

- **Dicussion on Other Attacks:**
  - Privileged-insider Attack
  - User Impersonation Attack
  - Offline Password Guessing Attack
  - Stolen Smart Card Attack
  - Denial-of-Service Attack
  - Replay Attack
  - Man-in-the-Middle Attack
  - Resilience against Sensing Device Attack
  - Anonymity and Untraceability

Formal Security using Random Oracle Model

Theorem

If $\mathcal{A}$ be an adversary running in polynomial time t against our authenticated key-agreement (AKE) protocol, $\mathcal{P}$ in the random oracle, the advantage of $\mathcal{A}$ in breaking the security of the session key sk_{ij} is given by

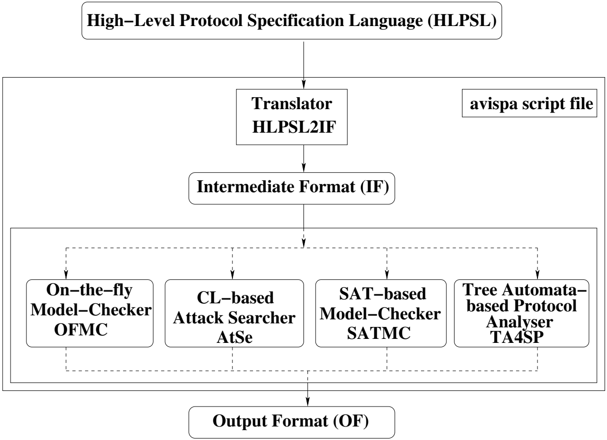
$$Adv_{\mathcal{P}}^{AKE} \leq \frac{q_h^2}{|Hash|} + \frac{q_{send}}{2^{l-1}.|\mathcal{D}|} + 2Adv^{ECDDHP}(t),$$

where q_h, q_{send}, $|Hash|$, $|\mathcal{D}|$, l and $Adv^{ECDDHP}(t)$ are the number of HASH queries, the number of Send queries, the range space of hash function $h(\cdot)$, the size of the distributed password dictionary $\mathcal{D}$, the number of bits in biometric key σ_i, and the advantage of $\mathcal{A}$ in breaking the elliptic curve decisional Diffie-Hellman problem (ECDDHP), respectively.

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.

Architecture of 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.

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*.

Formal Security Verification using AVISPA tool

Analysis of simulation results using OFMC and CL-AtSe backends

```
% 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: 0.15s
  visitedNodes: 49 nodes
  depth: 6 plies
```

```
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   : 3 states
  Reachable  : 0 states
  Translation: 0.03 seconds
  Computation: 0.01 seconds
```

Performance Comparison

- The performance of the proposed scheme is compared with other related authentication schemes [4], [5], [6] previously proposed for IoT applications.

[4] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase authentication protocol for wireless sensor networks in distributed IoT applications," in **IEEE Wireless Communications and Networking Conference (WCNC)**, Istanbul, Turkey, 2014, pp. 2728–2733.
[5] P. Porambage, A. Braeken, C. Schmitt, A. Gurtov, M. Ylianttila, and B. Stiller, "Group Key Establishment for Enabling Secure Multicast Communication in Wireless Sensor Networks Deployed for IoT Applications," **IEEE Access**, vol. 3, pp. 1503–1511, 2015.
[6] M. Turkanovi, B. Brumen, and M. Holbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," **Ad Hoc Networks**, vol. 20, pp. 96–112, 2014.

Comparison of communication overhead of our scheme with related IoT schemes

Protocol	No. of messages	No. of bits
Our	3	2528
Porambage *et al.* [4]	4	1344
Porambage *et al.* [5]		
-Protocol-1	4	3360
-Protocol-2	2	1136
Turkanovic *et al.* [6]	4	2720

Comparison of computation overheads of our scheme with related IoT schemes

Protocol	User side	GWN/Base station side	Sensing device/ Sensor side	Total overhead
Our	$5T_{ecm} + 5T_h$ $\approx 0.0871s$	$5T_{ecm} + 4T_h$ $\approx 0.08678s$	$4T_{ecm} + 3T_h$ $\approx 0.06936s$	$14T_{ecm} + 12T_h$ $\approx 0.24324s$
[4]	$3T_h + 2T_{ecm}$ $+T_{eca}$ $\approx 0.0396s$	$-$	$3T_h + 2T_{ecm}$ $+T_{eca}$ $\approx 0.0396s$	$6T_h + 4T_{ecm}$ $+2T_{eca}$ $\approx 0.0792s$
$-$ Protocol-1 [5]	$4T_{ecm} + 8T_h$ $+T_{eca}$ $\approx 0.0754s$	$-$	$11T_{ecm} + 10T_h$ $+3T_{eca}$ $\approx 0.2045s$	$15T_{ecm} + 18T_h$ $+4T_{eca}$ $\approx 0.2799s$
$-$ Protocol-2 [5]	$3T_{ecm} + 7T_h$ $+T_{eca}$ $\approx 0.0579s$	$-$ $-$	$5T_{ecm} + 7T_h$ $+2T_{eca}$ $\approx 0.0965s$	$8T_{ecm} + 14T_h$ $+3T_{eca}$ $\approx 0.1544s$
[6]	$7T_h$ $\approx 0.00224s$	$5T_h$ $\approx 0.0016s$	$7T_h$ $\approx 0.00224s$	$19T_h$ $\approx 0.00608s$

Note: T_h: time for one-way hash function $h(\cdot)$; T_{ecm}: time for ECC point multiplication; T_{eca}: time for ECC point addition.

Comparison of functionality features of the proposed scheme with related IoT schemes

Feature	[4]	[5]	[6]	Our
User anonymity property	×	×	✓	✓
Insider attack	×	✓	×	✓
Off-line password guessing attack	—	—	×	✓
Stolen smart card attack	—	—	×	✓
Denial-of-service attack	×	✓	✓	✓
Known session key attack	✓	×	✓	✓
User impersonation attack	×	✓	×	✓
Man-in-the middle attack	×	✓	✓	✓
Replay attack	×	×	✓	✓
Mutual authentication	✓	✓	✓	✓
Session key agreement	✓	✓	✓	✓
Forward secrecy	✓	✓	✓	✓
Stolen/lost device revocation	—	—	×	✓
Untraceability property	✓	×	×	✓
Resilience against sensing device capture attack	×	✓	✓	✓
GWN independent password update phase	—	—	✓	✓
Support biometric update phase	—	×	×	✓
Provide formal security analysis using random oracle model	×	×	×	✓
Provide security analysis using BAN logic	×	×	×	✓
Provide formal security verification using AVISPA tool	×	×	×	✓

Practical Perspective: NS2 Simulation Study

Simulation Environment

- **Scenario 1.** This scenario has three users (U_is): one is static and other two are moving with the speeds of 2 *mps* (meters per second) and 15 *mps*, respectively.
- **Scenario 2.** This scenario has five users (U_is): two are static and other three are moving with the speeds of 2 *mps*, 15 *mps* and 15 *mps*, respectively.
- **Scenario 3.** This scenario has eight users (U_is): four are static and other four are moving with the speeds of 2 *mps*, 2 *mps*, 10 *mps* and 15 *mps*, respectively.

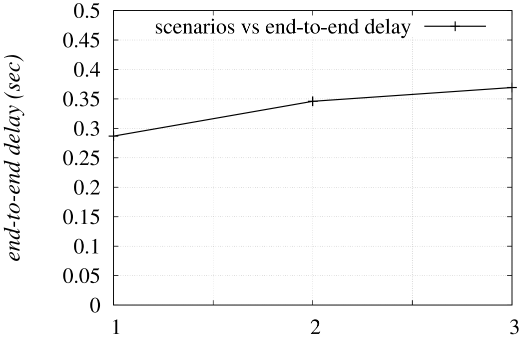
Practical Perspective: NS2 Simulation Study

Simulation Parameters

Parameter	Description
Platform	Ubuntu 14.04 LTS
Network scenarios	1, 2 and 3
Number of users (U_i)	$3, 5, 8$ for scenarios $1, 2, 3$
Number of gateway nodes (GWN)	1 for all scenarios
Number of smart devices (SD_j)	50 for all scenarios
Mobility	2 *mps*, 10 *mps*, 15 *mps*
Simulation time	1800 seconds (30 minutes)
Communicatio range of SD_j	50 meters
Communication range of GWN	200 meters
MAC protocol	IEEE 802.15.4
Routing protocol	Ad Hoc On-Demand Distance Vector (AODV)

Simulation Results

- **Impact on End-to-end Delay:**
  End-to-end delay (EED) is computed as the average time taken by the data packets (messages) to arrive at the destination from the source. EED can be formulated as $\sum_{i=1}^{n_{pkt}} (T_{rec_i} - T_{send_i})/n_{pkt}$, where T_{rec_i} and T_{send_i} are the receiving and sending time of a packet i, respectively, and n_{pkt} the total number of packets.



Simulation Results

- **Impact on Throughput:**
  Throughput is measured as the number of bits transmitted per unit time. The throughput can be calculated as $\frac{n_r \times |pkt|}{T_d}$, where T_d is the total time (in seconds), $|pkt|$ the size of a packet, and n_r the total number of received packets.