Token-Based Lightweight Authentication to Secure IoT Networks

Maissa Dammak, Omar Rafik Merad Boudia, Mohamed-Ayoub Messous, Sidi-Mohammed Senouci, Christophe Gransart

Token-Based Lightweight Authentication to Secure IoT Networks

Maissa Dammak[1], Omar Rafik Merad Boudia[2], Mohamed Ayoub Messous[1], Sidi Mohammed Senouci[1], Christophe Gransart[3]
[1]DRIVE EA1859, Univ. Bourgogne Franche Comté, France
[2]STIC Laboratory, University of Tlemcen, Algeria
[3]IFSTTAR, COSYS/LEOST Villeneuve d'Ascq, France
{maissa.dammak ; ayoub.messous ; sidi-mohammed.senouci}@u-bourgogne.fr; om_meradboudia@mail.univ-tlemcen.dz; christophe.gransart@ifsttar.fr

***Abstract*—The rapid growth of Internet of Things (IoT) technology offers huge opportunities and also brings many new challenges related to the authentication in IoT devices. Using passwords or pre-defined keys have drawbacks that limit their use for different IoT applications like smart hotel and smart office. In fact, they didn't provide temporary access to data in such reservation systems. Thus, authenticating users basing on password mechanism is not feasible. In this paper, we propose a new *Token-Based Lightweight User Authentication* (*TBLUA*) for IoT devices, which is based on token technique in order to enhance the robustness of authentication. Security analysis shows the security strength of the proposed scheme such as token security, *Perfect Forward Secrecy (PFS)*, etc. In addition, the presented performance analysis shows that it is a strong competitor among existing ones for user authentication in IoT environments.**

Keywords—*Lightweight Authentication, Token, Security, Perfect Forward Secrecy, De-Synchronization Attack.*

I. Introduction

The proliferation of the internet of thing (IoT) have expended into many aspects of our life style and social interactions, the next frontier is digital industry environment (e.g. smart city, smart hotel, smart office) [1]. Although, the future of smart industry environment is promising, many technical challenges must be addressed to achieve convenience and security [2]. Specifically, user lightweight authentication for reservation system has been a critical issue due to the communication between the user and smart devices which is limited in time. The reason is that the users may want to reserve a list of smart devices to establish communications for a period of time. For this purpose, it is important to authenticate the legitimacy of a user for a predefined time interval. In this context, tokens have been introduced as an efficient solution to create a strong binding between the users that requested the reservation and the smart device. At the same time, token-based authentication reduces the risk of stolen authentication factors as tokens are protected against misuse, and it does not require much more user effort than password-based mechanism [3].

The user-to-device authentication is fundamental, however, most of IoT devices are resource-constrained devices and they need to transmit sensed data periodically. Hence, it is necessary for smart things to adopt a lightweight authentication protocol to reduce their energy consumption when a device aims to authenticate and transmit data to its targeted peer. Likewise, IoT devices communicate over insecure communication channels and an illegal user (attacker) can break the security and also gain access over the smart device [4][5]. Furthermore, by compromising one secret key, an attacker may deduce any previous session key which represents a serious threat. Thus, Perfect Forward Secrecy is a basic security property for session key-based authentication [6].

To the best of our knowledge, most of the authentication schemes have several security limitations especially PFS, which is the basic and important security property for authentication in IoT environments. Besides, all most previously proposed schemes are based on two or three-factor authentication [5][8][9][12] which limit their use in reservation system.

In this paper, we propose a lightweight authentication protocol based on token technique to reach the design goals. This protocol:

- Generates an additional security layer of authentication by adopting the token technique which offers access to a specific resource for a predefined period of time.
- Reduces the computation overhead and save energy for authenticating devices during the authentication session, by using only lightweight computation operations such as XOR and hash function.
- Is designed to withstand the most popular security attacks and ensures the known security property especially PFS.
- Proves, by simulating its performance with existing schemes, that it is more efficient and lightweight solution.

The rest of the paper is structured as follows. The network model and the threat model are presented in section II. The proposed scheme *TBLUA* for user authentication in IoT environments is presented in Section III. Informal security analysis and the performance comparison with the existing relevant schemes are given in Section IV. Finally, Section VII concludes this paper.

II. System Model

In this model, we have followed two models which are discussed below:

A. Network and threat models

In this section we present the network model which is depicted in Fig.1. Our model consists of the end user U_i who needs to register herself/himself at the trusted Reservation Server (RS) in order to communicate with Smart Devices SD_j. RS is responsible for generating reservation tokens for U_i and distribute them for the Registration Authority (RA). The latter is responsible for registering all smart devices and gateway (GW) securely. Moreover, we assume that all the heterogeneous devices are synchronized with their clocks and agree a maximum transmission delay (ΔT) to protect our scheme against replay attacks [10]. We have used the Dolev-Yao threat model [9], in which two communicating parties (U_i, SD_j) interact over insecure channel and they are not considered as trustworthy. An adversary, let's call it *A*, can

eavesdrop the exchanged messages, and thus modify or delete the messages during transmission. Furthermore, SD_j are not tamper-resistant and thus, they can be physically compromised by A. Also, the user's smart phone SP can be lost/stolen by A. Therefore, A can extract sensitive information stored in those nodes using the well-known power analysis attacks [13]. Nevertheless, we assume that the GW in the proposed scheme is a trusted node and is not compromised under any circumstances; otherwise, the whole network is compromised [5]. Furthermore, RA and RS are also fully trusted and cannot be compromised by an adversary.

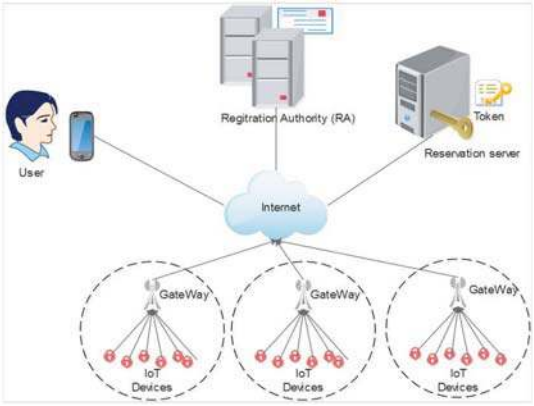

FIG.1. PROPOSED NETWORK MODEL

III. *TBLUA* SCHEME DESCRIPTION

In this section, we describe the proposed authentication and key negotiation protocol to make secure data transmission after a successful reservation. The proposed authentication protocol includes the following phases: (i) Offline smart device and GW registration, (ii) User reservation, (iii) Token distribution between GW and smart devices and (iv) Login and Authentication. All these phases are detailed in the following subsections. Notations are presented in Table I.

TABLE I. SYMBOLS AND THEIR DESCRIPTIONS

Symbols	Descriptions
RS	Reservation Server
RA	Registration Authority
U_i	User
GW	Gateway node
SD_j	Smart device node
PW_i	Password of U_i
ID_i	Identity of U_i
SP	User's Smart Phone
ID_{SDj}	Identity of SD_j
K	Secret key of GW
K_{UG}	Shared key between U_i and GW
K_{SG}	Shared key between SD_j and GW
TID_i	Temporary identity generated by GW for U_i
R_1	Random nonce created by U_i
R_2	Random nonce created by GW
R_3	Random nonce created by SD_j
$EK(\cdot)/DK(\cdot)$	Symmetric encrypt/decrypt using key K
NS_j	Sequence number
T_i , t_i	Current timestamp
ΔT	Maximum transmission delay
$h(\cdot)$	Cryptographic one-way hash function
$\|\|,\oplus$	Concatenation operation, Bitwise XOR operation

A. *Offline smart device and GW registration phase*

The offline sensing node registration phase is executed by the RA. The RA selects a unique identity ID_{SDj} for each deployed smart device SD_j and also generates a unique random 160-bits secret shared key, K_{SG}, between the GW and SD_i, where $1 \le j \le n$ (n is the number of smart devices) and the initial sequence numbers $NS_j=NS_{j0}=0$. The RA stores $\{ID_{SDj}, NS_j, K_{SG}\}$ into the smart device SD_j memory, and $\{ID_{SDj}, NS_{j0}, K_{SG}\}$ into the GW memory.

The RA further randomly generates a unique GW's identity ID_{GW} and a unique random 1024-bit secret key K. RA defines a group of SD_j which is identified by $G_i=\{SD_j; 1<j< N$, N is the number of SD_j in $G_i\}$ and computes $S_j= h(ID_{SDj} || G_i || K)$ for each SD_j and updates the SD_j node information table entry with $< ID_{SDj}, S_j, NS_{j0}, K_{SG}, G_i >$ in the GW memory.

B. *User reservation phase*

To access the services from a particular smart device SD_j, a user U_i first needs to register with the RA securely. The following steps in Table II are required for this registration:

TABLE II. SUMMARY OF USER RESERVATION PHASE

User (Ui)/ Smart Phone (SP)	Reservation server	Registration authority
1-Choose ID_i and PW_i, Compute $MPW_i=h(ID_i \oplus PW_i)$ $<ID_i, MPW_i>$ ----------> (via secure channel)	**2-**Reserve a group of smart devices G_i Generate $Token_u=$ Ek (ID_i, ID_{GW}, G_i, Te), (Te is the expiration time of the Token). $< Token_u >$ ----------> (via secure channel)	**3-**Generates a unique random 128-bits number n Computes $K_{UG}=h(ID_i\|\|n)\oplus ID_{GW}$. Generates also a random number R_i Computes $Reg_i=h(ID_i\|\|R_i \|\|MPW_i\|\|K_{UG})$, $A_i=R_i\oplus MPW_i$, $TK_{Ui}=Token_u\oplus h(ID_i\oplus R_i\oplus MPW_i\oplus K_{UG})$, $D_i=R_i\oplus h(TID_i\|\|K_{GW})$. $<TID_i, Reg_i, A_i, TK_{Ui}, K_{UG}>$
4-Compute $K_{UG}^*=K_{UG}\oplus h(h(ID_i)\oplus h(PW_i))$ Replace $K_{UG}=K_{UG}^*$	<---------- (Forward to User through secure channel)	**5-**Stores $<TID_i, D_i>$ into the GW memory

C. *Token distribution between GW and smart device phase*

In this phase, the GW distributes periodically the token of a user Ui to a group of smart devices after a successful reservation phase. It is detailed on 4 steps as follows:

TABLE III. SUMMARY OF TOKEN DISTRIBUTION PHASE

Gateway node (GW)	Smart Device (SD_j)
1- Decrypt $(Token_u)_K = (ID_i, ID_{GW}, G_i, Te)$ Retrieve all smart devices ID_{SDj} of the group G_i. Generate a random number r_i and timestamps t_1 Compute $D_1=h(K_{SGj}\|\|r_i\|\|ID_{SDj}\|\|t_1)$, $D_2=r_i\oplus h(K_{SGj})$ $<D_1, D_2, t_1>$ ----------> (via public channel)	**2-**Checks if $\|t_1^*-t_1\|<\Delta T$? If so, computes $r_i^*=D_2\oplus h(K_{SGj})$, $D_1^*=h(K_{SGj}\|\|r_i^*\|\|ID_{SDj}\|\|t_1)$, Check if $D_1^*=D_1$? If so, Generates a random number s_j and timestamp t_2 Computes $D_3=h(ID_{SDj}\|\|K_{SG}\|\|r_i^*\|\|s_j\|\|t_2)$, $D_4=s_j\oplus h(K_{SG})$, $<D_3, D_4, t_2>$, Otherwise, this phase is corrupted
3-Check if $\|t_2^*-t_2\|<\Delta T$? If so, compute: $s_j^*=D_4\oplus h(K_{SGj})$, $D_3^*=h(ID_{SDj}\|\|K_{SGj}\|\|r_i\|\|s_j^*\|\|t2)$, Check $D_3^* = D_3$? Generate timestamp t_3, Compute $Tx=Te\oplus h(K_{SGj})$, $F=h(Token_u\|\|K\|\|ID_{SDj})$, factor F to identify SD_j with the corresponding token. Update $K_{SGnew}=h(ID_{SDj}\|\|K_{SGj})$. $<F, Tx, t_3>$ else, this phase is corrupted ---------->	<---------- (via public channel) **4-**Update $K_{SGnew}=h(ID_{SDj}\|\|K_{SGj})$, Store F and Tx in its memory

TABLE IV. SUMMARY OF LOGIN AND AUTHENTICATION PHASE

User (Ui)/ Smart Phone (SP)	Gateway node (GWN)	Smart device (SDi)
1-Enter ID_i and PW_i into SP Compute MPW_i*=h($ID_i \oplus PW_i$), R_i*=$A_i \oplus MPW_i$*, K_{UG}=K_{UG}*$\oplus$h(h(ID_i*)$\oplus$h(PW_i*)), Reg_i*=h($ID_i||R_i$*$||MPW_i$*$||K_{UG}$), Check if Reg_i*=Reg_i ? if so, choose ID_{SDj} Calculates $Token_u$*=$TK_{Ui} \oplus$h(ID_i*$||R_i$*$||MPW_i$*$||K_{UG}$), CID_i=$ID_i \oplus$h($TID_i||K_{UG}||R_i$*$||T_1$), R_0=h($K_{UG}||R_i$*)$\oplus R_1$ CID_{SDj}=$ID_{SDj} \oplus$h($Token_u$*$||K_{UG}||R_i$*$||T_1$), M_1=h($ID_i||R_1||Token_u$*$||K_{UG}||T_1$) <TID_i, CID_i, CID_{SDj}, M_1, R_0, T_1> ----------> (via open channel)	**2-**Check if $\|T1$* $- T1 \| < \Delta T$? Search in the table against TID_i and retrieve D_i , Compute R_i*=$D_i \oplus$h($TID_i||K$), ID_i*=$CID_i \oplus$h($TID_i||K_{UG}||R_i$*$||T_1$), R_1*=$R_0 \oplus$h(R_i*$||K_{UG}$), M_1*=h(ID_i*$||R_1$*$||Token_u||K_{UG}||T_1$). Check if $M1$*=M_1? Decrypt ($Token_u$)$_K$=(IDi, ID_{GW}, G_i, Te) Generate a current timestamp T_2 Check if $Te<T_2$? If so, $Token_u$ is not expired. Compute ID_{SDj}=$CID_{SDj} \oplus$h($Token_u||K_{UG}||R_i$*$||T_1$), S_j*=h(ID_{SDj}*$||G_i||K$) Check if S_j*=S_j ? if so, produce a random nonce R_2 Compute M_2=h(h($ID_i||ID_{SDj}$*$||R_1$*$||R_2$)$\|$h($Token_u||K||ID_{SDj}$)$\|K_{SGj}||R_2||NS_{j0}||T2$), M_3=h($ID_i||ID_{SDj}||R_1$*$||R_2$)$\oplus K_{SGj}$, M_4=$R_2 \oplus$ h(K_{SGj}), NS_{j0}=NS_{j0}+1 <M_2, M_3, M_4, NS_{j0}, T_2>, Otherwise, this phase is corrupted ----------> (via public channel)	**3-**Check if $\|T_2$*$-T_2\|<\Delta T$? And if $1 \le NS_{j0}$-$NS_j \le N$, where N is a threshold. If so compute R_2*=$M_4 \oplus$h(K_{SGj}), M_3=$M_3 \oplus K_{SGj}$, M_2*=h(M_3|h($Token_u$|K |ID_{SDj}) | $K_{SGj}||R_2$*$||NS_{j0}$-1$||T_2$). Check if M_2*=M_2 ? if so generate a random number R_3 and current timestamp T_3, Compute SK=h($M_5||R_2$*$||R_3||T_3$), M_6=h($SK||R_3||K_{SGj}||NS_{j0}||T_3$), M_7=$R_3 \oplus$h(R_2). Update the shared key K_{SGnewj}=h($K_{SGj}||ID_{SDj}$), K_{SGj}=K_{SGnewj}, Update the sequence number NS_j=NS_{j0} <M_6, M_7, T_3>, Otherwise, this phase is corrupted <---------- (via public channel)
	4-Check if $\|T_3$*$-T_3\|<\Delta T$? If so, Compute: R_3*=$M_7 \oplus$h(R_2), SK*=h(h($ID_i||ID_{SDj}||R_1$*$||R_2$)$||R_2||R_3$*$||T_3$), M_6*=h(SK*$||R_3$*$||K_{SGj}||NS_{j0}||T_3$) Check if M_6*=M_6 ? if so;	
5-Check if $\|T_4$*$-T_4\|<\Delta T$? If so, compute: R_2*=$M_8 \oplus$h($ID_i||R_1$), R_3*=$M_7 \oplus$h(R_2*), TID_i*=$M_{10} \oplus$h(R_2*$\oplus R_3$*), SK*=h(h($ID_i||ID_{SDj}||R_1||R_2$*)$||R_2$*$||R_3$*$||T_3$), M_9*=h($ID_i||SK$*$||R_3$*$||K_{UG}$). Check if M_9*= M_9 ? If so, update TID_i= TID_i*, Compute K_{UGnew}=h($ID_i||K_{UG}$), Update K_{UG}= K_{UGnew},	generate a timestamps T_4 and a new unique identity TID_i*$\neq TID_i$ Compute M_8=$R_2 \oplus$h($ID_i||R_1$), M_9=h($ID_i||SK$*$||R_3$*$||K_{UG}$), M_{10}=TID_i*$\oplus$h($R_2 \oplus R_3$*), Update its memory K_{UGnew}=h($K_{UG}||ID_i$) and K_{SGnewj}=h($K_{SGj}||ID_{SDj}$). <M_7, M_8, M_9, M_{10}, T_4>, Otherwise, this phase is corrupted <---------- (via public channel) Compute D_i*=$R_i \oplus$h(TID_i*$||K$) Replace <TID_i, D_i> = <TID_i*,D_i*>.	

D. Login and authentication phase

Once the registration process is completed, a user U_i is now ready to login in the system. This phase achieves the goal of authentication among the U_i, GW, and SD_j. Besides, at the end of the execution of this phase, a session key is established between U_i and SD_j. This phase is explored in Table IV.

IV. SECURITY ANALYSIS

Our scheme ensures many security properties and resists most popular attack. At first, TBLUA ensures the anonymity, thus any adversary A is unable to break the anonymity using the public messages. This is because the identities ID_i and ID_{SDj} are protected by h(·). In addition, A needs to know ID_i, $ID_{SDj,}$ the long secret key and $Token_u$ to compute CID_i, CID_{SDj}. Thus, our protocol can resist user impersonation attack. Furthermore, as A cannot retrieve ID_i , ID_{SDj} and the shared Key K_{SG} of SD_j, he/she cannot masquerade as a valid smart device and TBLUA resists the node impersonation attack. Besides, TBLUA is mainly designed to ensure the PFS service, let suppose A has obtained K_{UG} and K_{SG}, He/she cannot get the session key SK. This is due to that after each successful session, K_{UG} and K_{SG} will be updated by one-way hash function. More than that, as we propose to use distinct shared secret keys K_{SGj} , SD_i establishes a distinct session key with U_i, thus, although if A can capture a SD_i , all non-compromised devices still can communicate with the legitimate user U_i with higher secrecy. Thus TBLUA withstands SD_j node capture attacks. Moreover, our scheme can resist against token impersonation attack because without prior knowledge of K, an adversary A cannot create a Token. And as the $Token_u$ is protected with a symmetric cipher function using K, A cannot modify a valid token. Hence, TBLUA resists token modification attack.

V. PERFORMANCE ANALYSIS

In this section, we compare the communication and computation costs of the proposed scheme with three prior related works [5][8][12]. Since the distribution token phase is not used frequently and the cost is negligible (i.e. GW; $4T_h$ = 2ms and Smart device: $3T_h$ = 1.5 ms), we only concentrate on comparing login and authentication phase.

A. Functionality comparison

The functionality features of the existing schemes and the proposed scheme are compared in Table V. TBLUA can resist against various kinds of known attacks and fulfill the desirable security features such as PFS.

TABLE V. FUNCTIONALITY FEATURES COMPARISON

Properties	[8]	[12]		[5]	TBLUA
Mutual Authentication	+	+		+	+
Key agreement	+	+		+	+
Intractability	-	+		+	+
User anonymity	-	+		+	+
SD. anonymity	-	+		+	+
Offline PW guessing	-	+		+	+
User impersonation	-	+		+	+
GW impersonation	-	+		+	+
SD impersonation	-	+		+	+
Privileged-insider	-	+		+	+
PFS	+	-		-	+
Replay attack	-	+		+	+
Stolen verifier	-	+		+	+
De-synchronization	-	-		-	+
Node capture	-	+		+	+
Token impersonation	N/A	N/A		N/A	+
Token modification	N/A	N/A		N/A	+

Note: N/A: Not Applicable

B. Computation costs comparison

For the computation comparison, let the notations T_h=0.5ms be the time for one hashing operation and T_{Enc}=T_{Dec}=8.7ms be respectively the time for one encryption/decryption using symmetric cryptography operation and T_{ECC}=T_{FE}=63.075ms represent respectively the time for one elliptic curve cryptography one fuzzy extraction operation [4][5], we omit XOR operation due to its negligible computational cost. In Table VI, we provide computation cost separately for user, GW node and S of the login and authentication phase.

TABLE VI. COMPUTATION COSTS COMPARISON

Scheme	User	GW	Smart Device
[8]	$2T_{ECC}+7T_h$= 129,6ms	$9T_h$ =4,5ms	$2T_{ECC}+5T_h$= 128,6ms
[12]	$7T_h+T_{Dec}+T_{Enc}$= 20ms	$11T_h+2*T_{Dec}+2 * T_{Enc}$= 40.3ms	$4T_h+T_{Dec}+T_{Enc}$= 19.4ms
[5]	$T_{FE}+13T_h+T_{Dec}+T_{Enc}$= 87.0ms	$5T_h+2T_{Dec}+2T_{Enc}$= 37.3ms	$4T_h+T_{Dec}+T_{Enc}$= 19.4ms
TBLUA	$16T_h$= 8ms	$19T_h+T_{Dec}$= 18.2ms	$7T_h$= 3.5ms

C. Communication costs comparison

The communication costs of different existing schemes along with our proposed protocol are given in Table VII. It presents the comparison only for the phases that are executed frequently which are login and authentication phases. We assume that ID_i is of length 160 bits, the identity of smart device node is 32 bits, random nonce is of 128 bits, symmetric encryption/decryption block size is of 128 bits (i.e., if we apply AES-128 algorithm [11]), timestamp is of 32 bits, sequence number is of 64 bits, and hash digest is of 160 bits (i.e. if SHA-1 hashing algorithm is applied [14]). For elliptic curve cryptography (ECC) based schemes, we consider a security level of 160-bit.

TABLE VII. COMMUNICATION COSTS COMPARISON

Scheme	User	GW	Smart device	Total Cost
[8]	84 Byte	64Byte	136 Byte	284 Byte
[12]	80 Byte	120 Byte	40 Byte	240 Byte
[5]	92 Byte	168 Byte	64 Byte	324 Byte
TBLUA	84 Byte	156 Byte	44 Byte	284 Byte

Fig.2 shows that the simulation results confirm the efficiency of our proposed scheme. In fact, our scheme has the lowest computation cost compared to benchmarking schemes and achieves a desirable communication overhead.

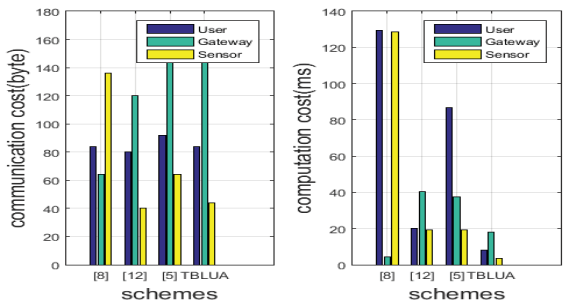

Fig. 2. Performance comparison

VI. CONCLUSION

In this paper, we have proposed a lightweight authentication protocol based on token technique which provides an authentication for a period of time and response to the needs of modern cities. In fact, the proposed protocol TBLUA is adopted in system reservation to ensure a mutual authentication between the communicating parties (User, GW, IoT device). Then, we demonstrated the trade-off between effectiveness and efficiency of the proposed scheme. From security perspective, it provides relatively more security features and high security level such as anonymity, Perfect Forward Secrecy, and resilience against the well-known attacks. Furthermore, performance analysis proved that TBLUA has a low computation and communication overhead compared to benchmarking schemes. In future works, further results will be conducted along with a formal verification using the AVISPA tool.

ACKNOWLEDMGELNT

This work is achieved as part of the European project ITEA PARFAIT [15], which is partially funded by FEDER (European Regional Development Fund), BPIFRANCE, and the BFC region (Bourgogne-Franche-Comté).

REFERENCES

[1] K. T. Nguyen, M. Laurent, N. Oualha, "Survey on secure communication protocols for the Internet of Things", *Ad Hoc Netw.*, vol. 32, pp. 17-31, Sep. 2015.

[2] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang, and L. Shu, "Authentication Protocols for Internet of Things: A Comprehensive Survey," Security and Communication Networks, vol. 2017.

[3] O. O. Bamasag and K. Youcef-Toumi. "Towards Continuous Authentication in Internet of Things Based on Secret Sharing Scheme", In Proceedings of the WESS'15: Workshop on Embedded Systems Security (WESS'15). ACM, New York, NY, USA.

[4] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. G. Reddy, E. J. Yoon, and K. Y. Yoo, "Secure Signature-Based Authenticated Key Establishment Scheme for Future IoT Applications," IEEE Access, vol. 5, pp. 3028– 3043, 2017.

[5] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti and M. Jo, "Design of Secure User Authenticated Key Management Protocol for Generic IoT Networks," in IEEE Internet of Things Journal, vol. 5, no. 1, pp. 269-282, Feb. 2018.

[6] L. Xiong, D.Peng, T.Peng, H.Liang, and Z.Liu, ―A lightweight anonymous authentication protocol with perfect forward secrecy for wireless sensor networks‖, Sensors, vol.17, no.11, pp.2681:1- 28,2017.

[7] M. Turkanovic, 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.

[8] 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, vol. 15, no. 1, pp. 357–366, 2016.

[9] P. Gope and T. Hwang, "A Realistic Lightweight Anonymous Authentication Protocol for Securing Real-Time Application Data Access in Wireless Sensor Networks," in IEEE Transactions on Industrial Electronics, vol. 63, no. 11, pp. 7124-7132, Nov. 2016.

[10] N. Khalil, M. R. Abid, D. Benhaddou, and M. Gerndt, "Wireless sensors networks for Internet of Things," in IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 2014, pp. 1–6.

[11] "Advanced Encryption Standard (AES)," FIPS PUB 197, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, November 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. Accessed on April 2018.

[12] Y. Lu, L. Li, H. Peng, et al. "An Energy Efficient Mutual Authentication and Key Agreement Scheme Preserving Anonymity for Wireless Sensor Networks," Sensors, vol. 16, no. 6, article no. 837, 2016.

[13] 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.

[14] "Secure Hash Standard," FIPS PUB 180-1, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, April 1995.

[15] http://www.itea3-parfait.com/.