New Enhanced Authentication Protocol for Internet of Things

Mourade Azrour
*IDMS Team, Department of Computer Science, Faculty of Sciences and Techniques, Moulay Ismail University, Errachidia 52000, Morocco.*

Jamal Mabrouki
*Laboratory of Spectroscopy, Molecular Modeling, Materials, Nanomaterial, Water and Environment, CERNE2D, Faculty of Science, Mohammed V University in Rabat, Rabat 10000, Morocco.*

Azedine Guezzaz
*Department of Computer Science and Mathematics, High School of Technology, Cadi Ayyad University, Marrakesh 40000, Morocco.*

Yousef Farhaoui
*IDMS Team, Department of Computer Science, Faculty of Sciences and Techniques, Moulay Ismail University, Errachidia 52000, Morocco.*

Follow this and additional works at: [https://dc.tsinghuajournals.com/big-data-mining-and-analytics](https://dc.tsinghuajournals.com/big-data-mining-and-analytics)

Part of the Computer Engineering Commons, Computer Sciences Commons, and the Data Science Commons

**Recommended Citation**


This Research Article is brought to you for free and open access by Tsinghua University Press: Journals Publishing. It has been accepted for inclusion in Big Data Mining and Analytics by an authorized editor of Tsinghua University Press: Journals Publishing.
New Enhanced Authentication Protocol for Internet of Things

Mourade Azrour*, Jamal Mabrouki, Azedine Guezzaz, and Yousef Farhaoui

Abstract: Internet of Things (IoT) refers to a new extended network that enables to any object to be linked to the Internet in order to exchange data and to be controlled remotely. Nowadays, due to its multiple advantages, the IoT is useful in many areas like environment, water monitoring, industry, public security, medicine, and so on. For covering all spaces and operating correctly, the IoT benefits from advantages of other recent technologies, like radio frequency identification, wireless sensor networks, big data, and mobile network. However, despite of the integration of various things in one network and the exchange of data among heterogeneous sources, the security of user's data is a central question. For this reason, the authentication of interconnected objects is received as an interested importance. In 2012, Ye et al. suggested a new authentication and key exchanging protocol for Internet of things devices. However, we have proved that their protocol cannot resist to various attacks. In this paper, we propose an enhanced authentication protocol for IoT. Furthermore, we present the comparative results between our proposed scheme and other related ones.

Key words: authentication; Internet of Things (IoT); sensor; security; authorization

1 Introduction

In recent digital world, the Internet of Things (IoT) is a great network technology that interconnects between innumerable devices. The IoT lets people living and working smarter by offering smart objects. Due to its importance, IoT is applied in many fields, as well as for water monitoring[1–5], healthcare[6–9], smart environment[10–13], smart home[14–19], and others. In fact, there is no standard architecture of IoT. However, according to Fig. 1, the architecture illustrated consists of three main layers, including perception layer, networking layer, and application layer. The first layer

Fig. 1 Architecture of IoTs.
The second layer has a role to transfer captured values and link between the various elements of the network. Many network solutions can be adopted in this layer, including Bluetooth, Wi-Fi, GSM, 4G/5G, Zigbee, Wimax, etc. For the application layer, it is responsible to process and store received data for the operators to develop the applications, for example, smart environment.

Currently, the number of linked things to the internet is growing very rapidly. According to the Gartner results, the total of linked objects by 2020 will be more than 20 billion[20]. As a result, our daily life is touched by IoT applications either in personal or in a full social order. So, IoT covers several fields, it makes our usual actions simple and alters the relationship between people and different objects[21].

Regardless of the great importance of IoT in our life, this technology has to resolve certain issues, for example, to link between hybrid resources and systems, the invention of new protocol is required. Furthermore, the security of private data must be satisfied and the security is considered as the most important issue, then it must receive great interest. The type of security that is required for IoT comprehends data integrity, the privacy protection, access control, service availability, and so on[22]. Therefore, it surpasses the standard security that enables to protect transferred data. Furthermore, the two mechanisms used in IoT for preventing unauthorized users and devices to access to the sensed and captured values are the authentication and access control.

For assuring this two security services, many researches have proposed various authentication protocols for IoT. Therefore, in recent years, Ye et al.[23] suggested a new authentication and key exchanging protocol for IoTs devices. So, they demonstrated that the protocol can resist against some attacks. However, Azzour et al.[24] demonstrated that the protocol in Ref. [23] is vulnerable to some attacks and it has many security problems. Accordingly, in this study, we propose an enhanced authentication protocol for IoT. Additionally, we expose the result of the comparative study between our proposed protocol and other correlated ones.

The rest of this paper is structured as follows. Section 2 is reserved for presenting the literature review. In Section 3, we detail our proposed IoT authentication protocol. The security analysis and performance comparison are given in Sections 4 and 5, respectively. Then the paper is concluded in Section 6.

2 Literature Review

Due to the importance of the security in IoT and the necessity of assuring the confidentiality of the user’s data, the execution authentication protocol between connected devices is obligatory. Furthermore, in order to strengthen the authentication process, many specialists have presented different authentication protocol models based on various encryption techniques and algorithms.

In 2014, Jan et al.[25] presented an efficient authentication protocol that allows server and IoT objects to authenticate each other mutually. The protocol is based on shared key method to verify the identity of server and IoT devices, and also to share the secret key between two entities. In the later year, in order to secure the communication between IoT devices and cloud servers, Kalra and Soo[26] introduced a robust mutual authentication protocol that is founded on primitive of Elliptic Curve Cryptography (ECC)[27] and that uses HTTP cookies. After simulating their protocol under Automated Validation of Internet Security Protocols and Applications (AVISPA) tools, they confirmed that their scheme can be used to provide the mutual authentication and it can resist against several attacks.

In 2017, Wu et al.[28] presented a new authentication protocol for IoT-based Wireless Sensor Networks (WSNs) which aims to resolve the recommended directives for IoT, that are introduced by Fantacci et al.[29] and Nguyen et al.[30]. For establishing the communication the user firstly sends a message to gateway, then the last one transfers it to the sensor. At the end, the user, gateway, and sensor authenticate each other mutually. Nevertheless, Bayat et al.[31] proved that the protocol in Ref. [28] is not very secured as it cannot deal with some security attacks. Consequently, for enhancing the protocol in Ref. [28], Bayat et al.[31] proposed a provable secure authentication protocol. They justified that the enhanced protocol is secured after the simulation under ProVerif analyzer.

In recent times, Li et al.[32] proposed a three-factor anonymous authentication protocol based on a fuzzy commitment protocol and error correction code to handle the user’s biometric data. The protocol is applied for authenticate devices-based WSNs in IoT environments. They demonstrated that their scheme might guarantee
computational efficiency and achieve more security and functional features. However, in 2019, Tai et al.\cite{33} showed that the protocol in Ref. [32] is not secured against various attacks.

The quantum cryptography technic is also used by Sharma and Kalra\cite{34} and Karla and Sood\cite{26} for proposing two-factor authentication scheme to identify users and cloud servers. Hereafter, they demonstrated that their protocol is secured against possible attacks.

On the other hand, Dhillon and Kalra\cite{35} suggested an authentication protocol for authenticating the identity both users and devices if they want have an authentication protocol for IoTs. Our proposed protocol can resist against several attacks.

3 Our Proposed Protocol

In this section, we detailed a new enhanced authentication protocol for IoTs. Our proposed protocol consists of four phases, which are new sensor adding phase, user registration phase, login and authentication phase, and password changing phase. The used notations are illustrated in Table 1.

3.1 New sensor addition phase

In order to add new sensor node $Sn_i$ in an existing sensor network, the gateway generates random and particular ID$_{Sn}$ and $K_{GSn_i}$ as identifier and key of the new sensor, respectively. Then, the gateway loads this information into the node memory before deployment. Hence, it stores ID$_{Sn}$, $K_{GSn_i}$, and its database for future usage.

Note here that $K_{GSn_i}$ is the secret key shared between the gateway and the sensor.

3.2 User registration phase

As depicted in Fig. 2, the user can register by performing registration phase through a secure channel as follows:

R1: The user $U$ chooses his/her identity ID$_{u}$ and the corresponding password PW$_{u}$. Then, he/she selects randomly two numbers $r_1$ and $r_2$. After that, it computes

$$HID = h(ID_u || r_1) \text{ and } HPW = h(ID_u || PW_u || r_2)$$

are sent to the gateway through a secure channel.

R2: The gateway GW selects a random number $r_3$ and computes $V = h(x_{Gw} || r_3)P \oplus HPW$. Then, the gateway stores HID and $r_3$ in its database and sends V to user $U$.

R3: The user stores this information $\{V, r_1, r_2, HID\}$ in the smartcard.

3.3 Login and authentication phase

In this phase, the communication is established among user, gateway, and sensor node through a public channel. The five steps of this phase are illustrated in Fig. 3 and are detailed as follows:

Auth1: $U \rightarrow GW: \{HID’, V_{U1}, T_1, a, ID_{Sn}\}$. After inputting the ID$_{u}$ and PW$_{u}$, user $U$ chooses a random integer $a$, computes HID’ = $h(ID_{u} || r_1)$, then checks if HID’ $\neq$ HID or not. If it is ok, $U$ computes $HPW’ = h(ID_{u} || PW_{u} || r_2)$ and $V_{U1} = h(V \oplus HPW’ || a)$. Finally, it sends this message $\{HID’, V_{U1}, T_1, a, ID_{Sn}\}$ to the gateway.

Auth2: $GW \rightarrow Sn_i: \{V_{Sn1}, \text{HID}, T_2, b\}$. When user’s message is received, the gateway verifies the timestamp $T_2 - T_1 \leq \Delta T$ and checks if $V_{U1} \overset{?}{=} h(h(x_{Gw} || r_3)P || a)$. If it is ok, the gateway chooses $b$ as random integer and computes

$$V_{Sn1} = h(HID’ || ID_{Sn} || T_2 || b || K_{GSn_i})P.$$  

Lastly, the gateway send this information $\{V_{Sn1}, \text{HID}, T_2, b\}$ to the sensor node.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Notation & Explanation \\
\hline
$U$ & User \\
$Sn_i$ & Sensor \\
GW & Gateway \\
ID$_{u}$/ID$_{Sn}$ & Identify of user/sensor \\
PW$_{u}$ & Gateway private key \\
$x_{Gw}$ & User’s password \\
$E(a, b)$ & Elliptic curve equation with order $n$ \\
P & Point on $E(a, b)$ \\
$h(\cdot)$ & Secret key shared between GW and Sn\_i \\
$h(\cdot)$ & One way hash function \\
$\oplus$ & String concatenation operator \\
$\oplus$ & XOR operator \\
$T_i$ & Timestamp ($i = 1, 2, \ldots, 5$) \\
\hline
\end{tabular}
\caption{Symbolizations and their meanings.}
\end{table}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig2.png}
\caption{Registration phase.}
\end{figure}
In order to change spontaneously his/her password through a public channel, the user has to execute the login and authentication phase with his/her ID_u and old password PW_u. Once getting the successful authentication and sharing the session key, the user chooses his new password PW_u* as illustrated in Fig. 4, the details are as follows:

**Change1:** U → GW: \{M_1\}. The user inputs ID_u and PW_u, then checks HID = h(ID_u;r_1). In the case it is ok, it chooses freely his/her new PW_u*, selects two random integers r_1^* and r_2^*, then calculates the values of HID^* = h(ID_u;r_1^*) and HPW^* = h(ID_u||PW_u^*||r_2^*).

Next, it encrypts the message with the session key M_1 = E_{SK}(HPW||HPW^*||HID||HID^*||V). Finally, it sends message M_1 to the gateway.

**Change2:** GW → U: \{M_2\}. Upon receiving the user’s message, the gateway decrypts it, M'_1 = D_{SK}(HPW||HPW^*||HID^*||HID^*||V). Next, it checks if V = h(x_{GW}||r_3)P ⊕ HPW. If it is ok, the gateway randomly selects an integer r_1^* and replaces HID and r_3 by HID^* and r_2^*. Afterward, it computes V^* = h(x_{GW}||r_1^*)P ⊕ HPW^* and encrypts V using the session key, M_2 = E_{SK}(V^*). This last one is sent back to user.

**Change3:** Once the gateway response is arrived, the user descripts M'_2 = D_{SK}(V^*). Then, it replaces V, r_1, r_2, and HID with the new values V^*, r_1^*, r_2^*, and HID^*, respectively.

### 4 Security Analysis

#### 4.1 Informal analysis

- **Mutual authentication**

  In the proposed scheme, the gateway authenticates the user and the sensor device. For the user, the gateway compares the value of V_u1 received from the user with the computed h(x_{GW}||r_3)P ||a in Auth2. For the sensor, in Auth4, the gateway checks the correctness

#### 3.4 Password changing phase

In order to change spontaneously his/her password through a public channel, the user has to execute the login and authentication phase with his/her ID_u and old password PW_u. Once getting the successful authentication and sharing the session key, the user chooses his new password PW_u* as illustrated in Fig. 4, the details are as follows:

**Change1:** U → GW: \{M_1\}. The user inputs ID_u and PW_u, then checks HID = h(ID_u;r_1). In the case it is ok, it chooses freely his/her new PW_u*, selects two random integers r_1^* and r_2^*, then calculates the values of HID^* = h(ID_u;r_1^*) and HPW^* = h(ID_u||PW_u^*||r_2^*).

Next, it encrypts the message with the session key M_1 = E_{SK}(HPW||HPW^*||HID||HID^*||V). Finally, it sends message M_1 to the gateway.

**Change2:** GW → U: \{M_2\}. Upon receiving the user’s message, the gateway decrypts it, M'_1 = D_{SK}(HPW||HPW^*||HID^*||HID^*||V). Next, it checks if V = h(x_{GW}||r_3)P ⊕ HPW. If it is ok, the gateway randomly selects an integer r_1^* and replaces HID and r_3 by HID^* and r_2^*. Afterward, it computes V^* = h(x_{GW}||r_1^*)P ⊕ HPW^* and encrypts V using the session key, M_2 = E_{SK}(V^*). This last one is sent back to user.

**Change3:** Once the gateway response is arrived, the user descripts M'_2 = D_{SK}(V^*). Then, it replaces V, r_1, r_2, and HID with the new values V^*, r_1^*, r_2^*, and HID^*, respectively.

**4 Security Analysis**

**4.1 Informal analysis**

- **Mutual authentication**

  In the proposed scheme, the gateway authenticates the user and the sensor device. For the user, the gateway compares the value of V_u1 received from the user with the computed h(x_{GW}||r_3)P ||a in Auth2. For the sensor, in Auth4, the gateway checks the correctness

![Fig. 4 Password changing phase.](image)
of \( V_{Sn2} \equiv h(HID \parallel ID_{Sn} \parallel T_3 \parallel c \parallel K_{GSn})P \). On the other hand, the user can authenticate the server in Auth5, by checking the validity of \( V_{U2} \equiv h(V \oplus HPW\parallel d\parallel ID_{Sn}) \). Finally, in Auth3, the sensor verifies the correctness of \( V_{Sn1} \equiv h(HID\parallel ID_{Sn} \parallel T_2 \parallel b \parallel K_{GSn})P \). If it is ok, the server is authenticated correctly. Accordingly, our proposed protocol offers mutual authentication.

- **Session key secrecy**

  Session key secrecy means that at the end of authentication phase and key exchange, anybody cannot know the session key excluding the user and gateway. In our proposed protocol, the session key is calculated in this way,

  \[
  SK = h(HID \parallel d \parallel a \parallel h(x_{Gw} \parallel r_3)P) \quad \text{or} \quad SK = h(HID \parallel d \parallel a \parallel V \oplus HPW'),\]

  where \( HPW' = h(ID_a \parallel PW_a \parallel r_2) \). Since, \( PW_a, r_2 \), and \( ID_a \) are secret, the session key cannot be computed by anyone except the user and the gateway. For that reason, our proposed protocol provides session key secrecy.

- **Password guessing attack**

  Suppose that an attacker eavesdrops the communication between user \( U \), gateway \( Gw \), and sensor \( Sn \), then gets \( V_{U1} \), where \( V_{U1} = h(V \oplus HPW\parallel a) \) and \( HPW' = h(ID_a \parallel PW_a \parallel r_2) \). For verifying the password, the attacker must know \( ID_a, r_2 \), and the value of \( V \). Since, those values are not transmitted directly; the pirate cannot verify the correctness of guessed password. Therefore, our scheme can resist against password guessing attack.

- **Insider attack**

  Insider attack refers to a security risk in which a legitimate user or device executes a malicious code or tries to access to other account. In our proposed protocol, all session parameters are limited to a specific session. So, they are recomputed in every new session. Therefore, the insider (user or device) is not capable to execute the insider attack, due to it has no new parameters of other sessions. Therefore, our proposed protocol is secured against insider attack.

- **Replay attack**

  In our proposed scheme, in the case an adversary attempts to alter user’s password, it needs to discover a session key and validate user ID. Even if the adversary has it or a valid user has not closed an old session, the adversary accesses to the user’s application, it cannot modify the password without knowing the old one. For that reason, our proposed protocol is secured against replay attack.

- **Denning-sacco attack**

  Denning-sacco attack denotes the ability to have a long-term private key, like password, gateway private key, or the session key, through old session key. In our proposed scheme, the session key is calculated in this way,

  \[
  SK = h(HID \parallel d \parallel a \parallel h(x_{Gw} \parallel r_3)P),
  \]

  where \( a \) and \( d \) are two random numbers generated for each session. Hence, it is impossible to get user password, because it is not used for generation session keys. Furthermore, it is hard to compute the private key of gateway from \( h(x_{Gw} \parallel r_3)P \), because the pirate has to face Elliptic Curve Cryptography Diffie-Hellman (ECCDH) and to get the secret random number \( r_3 \). Finally, we can say that the proposed scheme is resistant against denning-sacco attack.

- **Stolen verifier**

  In our proposed authentication protocol, any secret information, including user password or gateway secret key, is saved in the database. Therefore, the attacker is not capable to get correct user password, even if he gets an unauthorized access to the database. Besides, if he has accessed to the sensor, he cannot compute the session key \( K_{GSn} \), as it depends on a random number. As a result, our proposed scheme is secure against stolen verifier attack.

- **Denial of Service (DoS) attack**

  For verifying the newly received message, we have used the time stamps. In addition, random values are generated in each step and in each session, and since the repetitive messages are not allowed, the pirate cannot execute the DoS attack messages. Consequently, our proposed protocol can deal with DoS attack.

4.2 **Formal security analysis under scyther tool**

In this section, we firstly explain the utility of scyther tool\cite{36}, which is used for formal security analysis of our protocol. Then, we present the obtained results under this tool. Scyther is a tool that is developed and designed for formal analysis security protocol under the perfect cryptography assumption. It can identify the security requirements and vulnerabilities of a given protocol. The algorithms developed in scyther tool can provide features, such as

- Leading achievements, which have enabled new models for protocol analysis, including multi-protocol analysis.
The powerful creation of a finite description of an unlimited number of model traces, also known as a full feature.

Our proposed scheme is then specified according to Security Protocol Description Language (SPDL). This specification describes the roles of user, gateway, and sensor. Each role includes sequences of events (send, receive, announcements, and claim events). Figure 5 illustrates the verification result obtained from our protocol under scyther tool. The result shows that our protocol satisfies all security requirements and no attack is found.

5 Performance Analysis

In this section, our proposed protocol is compared with other related ones. In this comparison, we have done our analysis according to the two points of views, the security and performance.

As illustrated in Table 2 and security analysis, our protocol is secured against stolen verifier attack, denning-sacco attack, password guessing attack, replay attack, DoS attack, and insider attack. Furthermore, it can provide mutual authentication and session key secrecy. On the other hand, Karla and Sooud[26] scheme is vulnerable to password guessing, DoS and insider attacks do not provide mutual authentication. The protocol in Ref. [37] cannot resist against replay and DoS attacks. The protocol in Ref. [38] is vulnerable against stolen verifier and insider attacks. Moreover, it is not able to provide session key secrecy. Finally, the scheme in Ref. [39] cannot resist against password guessing attack.

The computation costs of our proposed protocol are compared with other related protocol. In this evaluation, very lightweight functions, such as string concatenation operation and XOR operation, are not neglected, because nearby calculation cost is slight. The symbolizations used are as follows:

- $T_h$: Computational charge of one-way hash operation;
- $T_{pm}$: Computational charge of elliptic curve point multiplication;
- $T_{inv}$: Computational charge of modular inversion;
- $T_{E/D}$: Computational charge of encryption and decryption algorithm.

In our protocol’s authentication phase, the user computes $5T_h$, the gateway calculates $6T_h + 4T_{pm}$, and the sensor computes $2T_h + 2T_{pm}$. Therefore, the total computation cost of our protocol is $13T_h + 6T_{pm}$. According to Table 3, we can notice that modular inversion operation, symmetric encryption, and decryption algorithms are not used in our protocol, they

![Fig. 5 Scyther test results.](image)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Stolen verifier</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Denning-sacco</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Password guessing</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Replay</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DoS</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Insider</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mutual authentication</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Session key secrecy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Note: ✓: Yes  ×: No

<table>
<thead>
<tr>
<th>Item</th>
<th>Computation cost in Ref. [40]</th>
<th>Computation cost in Ref. [41]</th>
<th>Computation cost in Ref. [42]</th>
<th>Computation cost in ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>$2T_h + 2T_{inv}$</td>
<td>$2T_h$</td>
<td>$5T_h$</td>
<td></td>
</tr>
<tr>
<td>Gateway</td>
<td>$4T_h + 4T_{inv}$</td>
<td>$2T_h + T_{E} + T_{D}$</td>
<td>$6T_h + 4T_{pm}$</td>
<td></td>
</tr>
<tr>
<td>Sensor</td>
<td>$3T_h + 2T_{inv}$</td>
<td>$T_h$</td>
<td>$2T_h + 2T_{pm}$</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>$8T_h + 8T_{inv}$</td>
<td>$28T_h$</td>
<td>$5T_h + T_{E} + T_{D}$</td>
<td>$13T_h + 6T_{pm}$</td>
</tr>
</tbody>
</table>
are replaced by the ECC, which is very fast and offers same security. We can remark also that with our protocol, user computes only $5T_h$ and the sensor calculates just $2T_h + 2T_{pm}$, consequently that is faster than the protocols in Refs. [40] and [42]. For that reason, we can say that our protocol is appropriate for IoTs applications.

6 Conclusion

In this paper, we recalled the vulnerabilities that we have discovered in protocol of Ref. [40]. Then, we proposed a new enhanced authentication protocol for IoTs applications. Afterward, we analyzed our protocol informally and we justified that it can resist against various known attacks, including stolen verifier attack, denning-sacco attack, password guessing attack, replay attack, DoS attack, and insider attack. On the other hand, we have used scyther tool for analyzing the protocol formally, the obtained results confirm that our scheme can satisfy security requirements. Finally, we compared the performance and computation charges of the protocol with other related ones.

References


Mourade Azrour received the PhD degree from Faculty of Sciences and Technologies, Moulay Ismail University, Errachidia, Morocco in 2019, and the MS degree in computer and distributed systems from Faculty of Sciences, Ibn Zohr University, Agadir, Morocco in 2014. He currently works as a computer science professor at the Department of Computer Science, Faculty of Sciences and Technologies, Moulay Ismail University. His research interests include authentication protocol, computer security, Internet of Things, and smart systems. He is a scientific committee member of numerous international conferences. He is also a reviewer of various scientific journals, such as International Journal of Cloud Computing and International Journal of Cyber-Security and Digital Forensics (IJCSDF).

Yousef Farhaoui received the PhD degree in computer security from Ibn Zohr University of Science, Morocco in 2012. He is now a professor at Faculty of Sciences and Techniques, Moulay Ismail University. His research interests include e-learning, computer security, big data analytics, and business intelligence. He is a member of various international associations. He has authored 4 books and many book chapters with reputed publishers, such as Springer and IGI. He is served as a reviewer for IEEE, IET, Springer, Inderscience, and Elsevier journals. He is also the guest editor of many journals with Wiley, Springer, Inderscience, etc. He has been the general chair, session chair, and panelist in several conferences.
Azidine Guezzaz received the MS degree in the field of computer science and distributed systems from Department of Mathematics and Computer Science, Faculty of Science, University Ibn Zohr, Agadir, Morocco in 2013. He received the PhD degree from Faculty of Science, University Ibn Zohr, Agadir, Morocco in 2018. He was a professor at the Technology High School and BTS in the period 2014–2018. He then joined Cadi Ayyad University in 2018 as an assistant professor. His main field of research interests are intrusion detection and prevention, computer and network security, and cryptography.

Jamal Mabrouki received the PhD degree in water science and technology from Faculty of Sciences, Mohamed V University in Rabat, Morocco in 2020. He is an engineer in environment and climate. He is working on the project of migration and water and has the role of water governance in migration policy in Africa with the cooperation between MedYWat and World Bank. He is currently a researcher for the environment and climate program at ECOMED in Morocco, where he started the coordinator of the project “Adaptation of Citizens to Climate Change”.