IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 11, 2021, accepted June 23, 2021, date of publication July 5, 2021, date of current version July 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3094875

Questioning the Security of Three Recent Authentication and Key Agreement Protocols

AMIR MASOUD RAHMANI[1], MOKHTAR MOHAMMADI[2], SHIMA RASHIDI[3], JAN LANSKY[4], STANISLAVA MILDEOVA[4], MASOUMEH SAFKHANI[5], SARU KUMARI[6], AND MEHDI HOSSEINZADEH[7]

[1]Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, Taiwan
[2]Department of Information Technology, College of Engineering and Computer Science, Lebanese French University, Erbil 44001, Iraq
[3]Department of Computer Science, University of Human Development, Sulaymaniyah 13133, Iraq
[4]Department of Computer Science and Mathematics, Faculty of Economic Studies, University of Finance and Administration, 10100 Prague, Czech Republic
[5]Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran 16788-15811, Iran
[6]Department of Mathematics, Chaudhary Charan Singh University, Meerut 250004, India
[7]Pattern Recognition and Machine Learning Lab, Gachon University, Sujeonggu, Seongnam 13120, Republic of Korea

Corresponding author: Mehdi Hosseinzadeh (mehdi@gachon.ac.kr)

"The result was created in solving the standard project no. 7429/2020/02 System approach to selected information and communications technology trends" using institutional support for long-term conceptual development of research of the University of Finance and Administration.

ABSTRACT Providing the desired security for constrained devices in the edge of Internet of Things (IoT) systems is a challenging task. Given that those devices are in shortage of the area and energy, many lightweight and ultra-lightweight protocols have been proposed so far in the literature. On the other hand, while we see many new proposals in the literature to secure communications on IoT systems, security analysis of those schemes has not received enough attention. Hence, in this paper, we analyse the security of three recently protocols for constrained environments and show their security loopholes. The analysed schemes include two protocols which have been published by IEEE Access and a recently proposed protocol entitled Extremely Good Privacy (EGP). The designers of all those protocols claimed optimal security against active adversaries. However, in this paper, we propose an efficient secret disclosure attack against EGP that recovers the whole secret parameters of the protocol after eavesdropping/blocking several sessions of the protocol and doing some off-line computations. The probability of the adversary to recover whole $2l$ secret parameters of the tag after eavesdropping/blocking 68 sessions of the protocol is 0.99, targeting a 128-bit security level by $l = 128$. In addition, we show that an adversary can efficiently desynchronize the target tag from the reader/server in polynomial time. In the case of the other protocols, we also present efficient attacks that contradict the designers' security claims.

INDEX TERMS RFID, authentication, IoV-SMAP, EGP protocol, ultra-lightweight, secret disclosure attack, desynchronization attack.

I. INTRODUCTION

The Internet of Things (IoT) is an emerging technology which is going to affect all aspects of our life, where very soon we will be surrounded by many smart devices that can monitor and report every motion of us and even report related data through an interconnected network to many different sources where we may explicitly or implicitly allow them to do so. For example, your smartwatch can gather many sensitive data related to your body activities or a field camera can trace your motion step by step and so on. Permission to access your personal data can be provided during installing a new refrigerator or updating the SMART TV frame-ware. Many of those devices are using Radio Frequency IDentification (RFID) technology to connect to each other and identify the other party. RFID is a wireless identification method that uses radio frequency to identify or trace the objects to which the tags are attached. A typical RFID system includes tags, reader(s) and a back-end database, where the channel between the reader and the back-end database could be permanent and secure or wireless and insecure but the channel between the reader and tags is always wireless and insecure. Hence, any

The associate editor coordinating the review of this manuscript and approving it for publication was Sedat Akleylek.

transferred message from tags/sensors can be eavesdropped or intercepted by a malicious adversary.

The authentication mechanism is the first step that allows the protocol parties to trust each other. There are many authentication protocols today that allow authentication of multiple lightweight devices on the Internet of Things at the same time, such as [1], which are called grouping proof protocols. On the other hand, a large fraction of such sensors/tags could be passive, e.g. highly constrained microchip with an antenna that stores the unique sensor/tag identifier and other related information about an object that the sensor/tag has been attached to [2]. Hence, to provide the desired security for the sensitive data that may be transferred by such a sensor, we are not able to use the conventional solutions, e.g. provably secure cryptographic protocols. To provide acceptable security for such constrained devices, in the last decade, extensive efforts dedicated to designing secure ultra-lightweight/lightweight protocols, most of them targeting RFID applications. As pioneers, we can mention SASI [3] and LMAP [4] protocols that tried to provide acceptable security using a limited number of hardware efficient components, e.g. bit-wise operations, rotation and so on. Such protocols are known as ultra-lightweight protocols in the literature. However, almost all such protocols have been dramatically broken by later analysis, e.g. see [5]–[9] and [10]. In this direction, Tian *et al.* proposed a permutation-based protocol called RAPP [11], where it used a very efficient bit-based permutation that could be implemented efficiently in a hardware constrained environment such as RFID passive tags. However, it comes out soon the protocol has serious flaws, e.g. see [12]–[14]. Inspired by the RAPP designing paradigm and also to overcome its flaws, several similar protocols have been proposed later where R^2AP [15], RCIA [16], KMAP [17], SLAP [18] and UMAPSS [19] are just examples. In all those protocols in each session, only the reader contributes to the protocol's randomness. However, later Safkhani and Bagheri [20] showed that any such protocol will be vulnerable to tag impersonation and desynchronization attacks. In this vein, also keeping the Safkhani *et al.*'s attack in mind, Khalid *et al.* recently proposed an ultra-lightweight mutual authentication protocol called EGP (stands for Extremely Good Privacy). In this protocol, similar to other RAPP family of protocols, tags only use simple operations, e.g. bit-wise XOR and a very lightweight permutation. EGP is also supported by extensive formal/informal analysis and claimed optimal security against attacks, include desynchronization, traceability and secret disclosure.

While EGP aims to ensure the security of the communication between the protocol's parties with ultra-lightweight components, some other protocols employed more promising components. Among them, Son *et al.* recently used blockchain to propose an authentication protocol for cloud-based telecare medical information system (TMIS) [21]. The proposed protocol is based on bilinear pairings as the main source of the security. Another protocol is Yu *et al.*'s recent proposal which is a message authentication protocol for Internet of Vehicles (IoV) named IoV-SMAP [22]. It is a lightweight protocol and uses one-way hash function as the main source of the security. Both these protocols are also supported by the conducted security analysis by the designers.

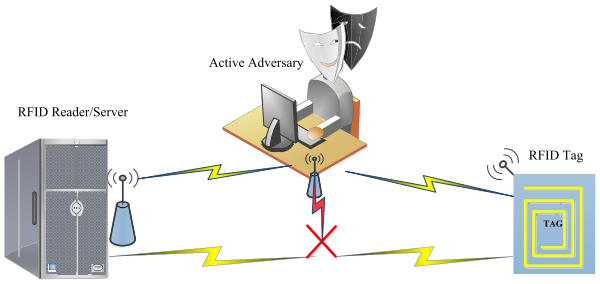


FIGURE 1. The used adversary model in this paper.

In this paper, we analyse the security of EGP, Son *et al.*'s protocol and IoV-SMAP. Through our analysis in this paper, we follow the "Dolev-Yao (DY)" adversary model [23], see Figure 1. In this model, the adversary $\mathcal{A}$ can control the communications between all the protocol's parties over a public channel and could interact passively or actively with them. However, $\mathcal{A}$ has no access to the internal values of the protocol's participants, e.g. the stored values such as keys and identifiers. In the case of insider attacks, we assume that the transferred values over secure channel is also accessible by the adversary, however, s/he has no access to the secret parameters. In addition, we also consider the recent adversarial model which has been proposed by Hosseinzadeh *et al.* [24], which investigates the impact of the compromising any node in the network on compromising whole the network.

A. OUR CONTRIBUTIONS

In this paper, based on the given adversarial model, as the first third-party analysis of the above mentioned protocols, we evaluate the security of three protocols and our contributions are as follows:

1) We show that the proposed protocol by Son *et al.* [21] suffers from secret disclosure attack and also insider attack.
2) We present an off-line password guessing by an insider attacker against the proposed protocol by Yu *et al.* [22], IoV-SMAP. We also show that the protocol could be compromised under the Hosseinzadeh *et al.* [24] adversarial model.
3) We show that it is possible to extend the Safkhani *et al.*'s attack on GUMAP family of protocols [20] to EGP also. Although the complexity of the attack is higher, compared to the attack on GUMAP, however, it is yet a linear function of memory size that is assigned to the tag's parameters on the reader/server's side. From this point of view, the attack complexity is similar to the attack on other members of the family. The proposed

attack is also applicable to the new design in this vein as well, i.e. [25].

4) We present a secret disclosure attack against EGP which can extract whole secrets, i.e., the tag static identifier and the secret key, in a semi-passive adversarial model. In this model, the adversary can only eavesdrop or block the transferred messages over the channel and cannot impersonate the tag/reader or modify the transferred messages.

It worth noting that the most important contribution of this paper is that the security analysis of cryptographic protocols contributes to the development of security protocol design science. Because it makes protocol designers familiar with the types of attacks that are applicable to the security protocols and so designers avoid repeating these errors that lead to such attacks in their design. Thus day by day the protocols that are designed for security, will be more secure, which will lead to their increasing use in many applications.

B. STATE OF THE ART

In recent years, to ensure that in many applications and for different environments messages are transmitted securely between different entities, the design of authentication and key exchange schemes has attracted a lot of attention. In addition, entity/device authentication and key exchange protocols are building blocks of access control in IoT security. Depending on the target application some constrains will be posed by the used component in designing the protocol. For example, a passive RFID tag which has no internal battery cannot support RSA as a cryptography primitive. Hence, depending on the target application, some designers have tried to design their protocol using only bit-wise operations and bit-wise permutation functions, e.g. [3], [11], [26], [27], and some other tried to use more sound components such as hash functions, e.g. [28], Physically Unclonable Functions (PUF), e.g. [29], [30], block ciphers, e.g. [31], authenticated encryption [32], to provide security for their protocol. However, it is not all those components could be considered as symmetric cryptography based protocols and either suffer from the lack of scalability or a variant of traceability attack. Hence, targeting less constrained environment, e.g. some medical services or vehicular networks, many researchers tried to provide desired security using public key based components which could be Elliptic Curve Cryptography (ECC) for instance, e.g. [33]–[37], bilinear pairings, e.g. [38], [39], or chaotic map such as [40]. Some other researchers tried to provide decentralized solutions by basing their protocol on the blockchain-based platforms such as [41]–[43]. Some recent works also have been conducted also that could be good choices for post quantum cryptography, because they are based on primitives that are secure against quantum computer such as lattice based cryptography, e.g. [44]–[46]. However, these solutions may not meet the constrained environment restrictions.

Although many interesting solutions have been proposed in literature, where we just mentioned a small fractions, yet we cannot trust those solutions if their security has not been analysed properly by independent researchers and their performance also meets the target applications requirements. Hence research is still going on in this active field of research, where designers are constantly trying to optimize their protocols for specific applications in terms of security and performance and on the other hand third parties try to show the drawbacks of the proposed protocols.

TABLE 1. Used notations in Son *et al.*'s protocol.

Symbol	Description
D_k	k^{th} doctor
P_i	i^{th} patient
CS	Cloud server
TA	Trusted authority
$SC/ID/PW$	Smart-card/Identity/Password of the patient
r_i, r_{CS}	Random numbers generated by P_i and CS respectively
SID_i	Secret identity of the patient
s_{TA}, α	Secret keys of TA
s_{CS}	Secret keys of CS
e	Bilinear map $e : G \times G \rightarrow G_T$
PK	Public key
$h(.)$	One-way hash function
$H(.)$	Map to point hash function
SK_{i-CS}	Session key between P_i and CS
$\|\|$	Concatenation
$\oplus$	XOR operation
$\mathcal{A}$	Adversary

C. PAPER ORGANIZATION

In the rest of this paper, we assign a section to each protocol and its security analysis. Hence, in Section II we describe our security analysis of Son *et al.*. Next, in Section III we describe and analyse the security of IoV-SMAP. In Section IV, we explain our security analysis of EGP. We discuss the results of the paper and some guidelines toward designing a secure protocol in Section V. Finally, we present the closing remarks in Section VI.

II. QUESTIONING THE SECURITY OF SON *et al.*'s PROTOCOL

Through this section, to describe and analyse the security of Son *et al.*'s protocol, we are using a list of notations that are represented in Table 1.

A. A BRIEF DESCRIPTION OF SON et al.'s PROTOCOL

Son *et al.* recently used blockchain to propose an authentication protocol for cloud-based telecare medical information system [21]. The proposed protocol is based on bilinear pairings as the main source of the security. Given cyclic groups G and G_T and a large prime q, a map $e : G \times G \rightarrow G_T$ is defined to be a bilinear pairings if it satisfies the below conditions:

- **Bilinearity:** For any $g, h \in G$ and any $a, b \in Zp$ we should have $e(g^a, h^b) = e(g, h)^{ab}$.

- **Non-degeneracy:** Assuming that g is the generator of G, $e(g, g) \neq 1$.
- **Efficiency:** For any $g, h \in G$, $e(g, h)$ can be calculated in polynomial time.

The protocol includes P_i as the i-th patient, D_K as the k-th doctor, CS as the cloud server and TA as the trusted authority. In the registration phase, P_i generates a random value $a_i \in Z_q^*$, calculates $HID_i = h(ID_i \| a_i)$ and sends it to TA over a secure channel. In response, TA stores HID_i in the secure memory, computes $SID_i = (HID_i \cdot s_{TA}) \cdot PK_{TA}$ and stores it in SC_i and sends it to P_i. Next, P_i generates $b_i \in Z_q^*$, computes $HPW_i = h(ID_i \| PW_i \| a_i)$, $A_i = h(ID_i \| PW_i) \oplus a_i$, $B_i = HPW_i \oplus b_i$, $C_i = SID_i \oplus b_i \cdot P$, and $Reg_i = h(a_i \| b_i \| HPW_i \| SID_i)$ and replaces SID_i with $(A_i; B_i; C_i; Reg_i)$ in SC_i. The cloud server also registers to TA, however, we omit its details because it has no impact on the proposed attack. The only point is that CS receives a list of registered HID_i and stores $CID_i = h(HID_i \| s_{CS})$ in its memory. Through the authentication phase between P_i and CS (see Figure 2), the patient inserts its SC_i into the card reader and inserts its ID_i and PW_i to do login. SC_i computes $a_i^* = A_i \oplus h(ID_i \| PW_i)$, $HID_i^* = h(ID_i \| a_i^*)$, $HPW_i^* = h(ID_i \| PW_i \| a_i^*)$, $b_i^* = HPW_i^* \oplus B_i$, and $SID_i^* = C_i \oplus b_i^* \cdot P$ and verifies whether $Reg_i \stackrel{?}{=} h(a_i^* \| b_i^* \| HPW_i^* \| SID_i^*)$ to accept the patient's login. Assuming that the login was successful, SC_i extracts the current timestamp T_1, generates a random value $r_i \in Z_q^*$ and computes $PK_i = (a_i \cdot r_i) \cdot P$, $X_i = (a_i \cdot r_i) \cdot PK_{CS}$, $D_i = HID_i \oplus h(X_i)$, $L_i^1 = h(X_i \| HID_i \| T_1 \| ID_{CS})$ and $PID_i = SID_i . L_i^1$. Next, P_i sends the tuple (PK_i, D_i, PID_i, T_1) to CS over a public channel.

CS verifies the received timestamp and given PK_i, computes $X_i = PK_i \cdot s_{CS}$, extracts $HID_i = h(X_i) \oplus D_i$ and verifies whether $h(HID_i \| s_{CS}) \stackrel{?}{=} CID_i$ matches any record in the database. Then, it computes $L_i^1 = h(X_i \| HID_i \| T_1 \| ID_{CS})$ and verifies whether $e(PID_i, P) \stackrel{?}{=} e((HID_i . L_i^1) \cdot PK_{TA}, PK_{TA})$ to authenticate P_i. Assuming that the authentication was successful, CS_i extracts the current timestamp T_2, generates a random value $r_{CS} \in Z_q^*$ and computes $R_{CS} = r_{CS} \cdot P$, $V_{CS} = r_{CS} \cdot PK_i$, $SK_{i-CS} = h(HID_i \| V_{CS} \| X_i)$, and $L_i^2 = h(V_{CS} \| SK_{i-CS} \| ID_{CS} \| T_2)$. Next, CS sends the tuple (R_{CS}, L_i^2, T_2) to P_i over a public channel.

P_i also verifies the received timestamp and given R_{CS}, computes $V_{CS}^* = (a_i \cdot r_i) . R_{CS}$, $SK_{i-CS}^* = h(HID_i \| V_{CS}^* \| X_i)$, and verifies whether $L_i^2 \stackrel{?}{=} h(V_{CS}^* \| SK_{i-CS}^* \| ID_{CS} \| T_2)$ to authenticate CS and accept the session key.

B. SECURITY FLAWS

In this section, the security analysis of Son *et al.*'s protocol is presented.

1) SECRET DISCLOSURE ATTACK BY *CS*

In a secure protocol, any secret value which is shared between the user (P_i in this case) and the trusted party (TA) should not be extractable by any service provider such as CS. However, in Son *et al.*'s protocol a malicious CS is able to extract SID_i which is a secret value for the P_i. To this end, once CS received (PK_i, D_i, PID_i, T_1) from the legitimate P_i, it computes $X_i = PK_i \cdot s_{CS}$, extracts $HID_i = h(X_i) \oplus D_i$ and computes $L_i^1 = h(X_i \| HID_i \| T_1 \| ID_{CS})$ and extracts $SID_i = (L_i^1)^{-1} \cdot PID_i$, where $(L_i^1)^{-1}$ is the multiplicative inverse of L_i^1. Given SID_i and HID_i, the malicious CS or an insider in CS can impersonate P_i to any other cloud server which has been authorized by the same TA to provide service for the target P_i.

2) INSIDER ATTACK

Through the authentication process between P_i and CS the temporal values are accessible by the insider adversary, which could be the CS operator. In Son *et al.*'s protocol, HID_i and L_i^1 are temporal values which are computed during the authentication phase in the CS side and can also be accessed by the insider, which has also access to ID_{CS} because it is known by any P_i so it is not secret. However, given HID_i, SID_i and ID_{CS} the adversary (insider) $\mathcal{A}$ can impersonate P_i at any time as follows:

1) To impersonate P_i, the insider adversary $\mathcal{A}$ with access to HID_i, SID_i and ID_{CS}, extracts the current timestamp T_1, generates a random value $r_i \in Z_q^*$ and computes $PK_i = (a_i \cdot r_i) \cdot P$, $X_i = (a_i \cdot r_i) \cdot PK_{CS}$, $D_i = HID_i \oplus h(X_i)$, $L_i^1 = h(X_i \| HID_i \| T_1 \| ID_{CS})$ and $PID_i = SID_i . L_i^1$. Next, $\mathcal{A}$ sends the tuple (PK_i, D_i, PID_i, T_1) to CS.
2) The legitimate CS verifies the received timestamp, computes $X_i = PK_i \cdot s_{CS}$, extracts $HID_i = h(X_i) \oplus D_i$ and finds related record to $h(HID_i \| s_{CS}) = CID_i$ in the database. Then, it computes $L_i^1 = h(X_i \| HID_i \| T_1 \| ID_{CS})$ and confirms that $e(PID_i, P) == e((HID_i . L_i^1) \cdot PK_{TA}, PK_{TA})$ and authenticates P_i. Then, CS_i extracts the current timestamp T_2, generates a random value $r_{CS} \in Z_q^*$ and computes $R_{CS} = r_{CS} \cdot P$, $V_{CS} = r_{CS} \cdot PK_i$, $SK_{i-CS} = h(HID_i \| V_{CS} \| X_i)$, and $L_i^2 = h(V_{CS} \| SK_{i-CS} \| ID_{CS} \| T_2)$. Next, CS sends the tuple (R_{CS}, L_i^2, T_2) to P_i over a public channel.

Following the above attack, $\mathcal{A}$ has been authenticated by CS as a legitimate P_i, while it has no access to the patient PW_i or even ID_i. It also has no access to any secret key of the cloud server through the attack. The success probability of the proposed attack is "1".

III. QUESTIONING THE SECURITY OF IoV-SMAP

Through this section, to describe and analyse the security of IoV-SMAP, we are using a list of notations that are represented in Table 2.

A. DESCRIPTION OF IoV-SMAP PROTOCOL

Yu *et al.* recently proposed a message authentication protocol for Internet of Vehicles (IoV) named IoV-SMAP [22]. It is a lightweight protocol which uses one-way hash function as

Patient P_i		**Cloud Server** CS
$(A_i; B_i; C_i; Reg_i)$		$\{CID_i = h(HID_i \| s_{CS})\}$
Inserts its SC_i in card reader, inserts its ID_i and PW_i to do login. SC_i computes $a_i^* = A_i \oplus h(ID_i \| PW_i)$, $HID_i^* = h(ID_i \| a_i^*)$ $HPW_i^* = h(ID_i \| PW_i \| a_i^*), b_i^* = HPW_i^* \oplus B_i$, $SID_i^* = C_i \oplus b_i^* \cdot P$, verifies whether $Reg_i \stackrel{?}{=} h(a_i^* \| b_i^* \| HPW_i^* \| SID_i^*)$ if successful, SC_i extracts T_1, generates $r_i \in Z_q^*$ and computes $PK_i = (a_i \cdot r_i) \cdot P$, $X_i = (a_i \cdot r_i) \cdot PK_{CS}$ $D_i = HID_i \oplus h(X_i)$, $L_i^1 = h(X_i \| HID_i \| T_1 \| ID_{CS})$, $PID_i = SID_i . L_i^1$	$\xrightarrow{PK_i, D_i, PID_i, T_1}$	
		Verifies T_1 and given PK_i, computes $X_i = PK_i \cdot s_{CS}$ extracts $HID_i = h(X_i) \oplus D_i$, verifies whether $h(HID_i \| s_{CS}) \stackrel{?}{=} CID_i$ computes $L_i^1 = h(X_i \| HID_i \| T_1 \| ID_{CS})$ verifies whether $e(PID_i, P) \stackrel{?}{=} e((HID_i . L_i^1) \cdot PK_{TA}, PK_{TA})$ If successful, CS_i extracts T_2, generates $r_{CS} \in Z_q^*$ computes $R_{CS} = r_{CS} \cdot P$, $V_{CS} = r_{CS} \cdot PK_i$, $SK_{i-CS} = h(HID_i \| V_{CS} \| X_i)$, $L_i^2 = h(V_{CS} \| SK_{i-CS} \| ID_{CS} \| T_2)$.
	$\xleftarrow{(R_{CS}, L_i^2, T_2)}$	
Verifies T_2 and given R_{CS}, computes $V_{CS}^* = (a_i \cdot r_i) . R_{CS}$, $SK_{i-CS}^* = h(HID_i \| V_{CS}^* \| X_i)$, verifies whether $L_i^2 \stackrel{?}{=} h(V_{CS}^* \| SK_{i-CS}^* \| ID_{CS} \| T_2)$		

FIGURE 2. Mutual authentication phase of Son *et al.*'s protocol [21].

the main source of the security. The protocol includes V_i as the *i*-th vehicle, *VS* as the vehicle server and *IS* as the infrastructure.

In the registration phase, V_i chooses its identity ID_{v_i} and password PW_{v_i}, generates a random value RN_i, calculates $RID_i = h(ID_{v_i} \| PW_{v_i})$, $RPW_i = h(PW_{v_i} \| RN_i)$ and sends them to *VS* over a secure channel. In response, *VS* computes $Q_i = K_{VS} \oplus h(RID_i \| RPW_i)$ and $W_i = h(RPW_i \| K_{VS})$ and stores them in SC_i and sends it to P_i. Next, V_i computes $E_i = RN_i \oplus h(PW_{v_i} \| RID_i)$ in SC_i. It should be noted K_{VS} is the master key of *VS*.

To register the infrastructure *IS*, it chooses its identity ID_{IS} and sends it to *VS* over a secure channel. In response, *VS* generates a random number N_{VS}, computes $C_i = h(ID_{IS} \| N_{VS}) \oplus K_{VS}$ and sends (C_i, N_{VS}) to *IS*.

Through the authentication phase between two vehicles V_i and V_j, the vehicle V_i inserts its ID_{v_i} and PW_{v_i} to do login. SC_i computes $RID_i = h(ID_{v_i} \| PW_{v_i})$, $RN_i = E_i \oplus h(PW_{v_i} \| RID_i)$, $RPW_i = h(PW_{v_i} \| RN_i)$, and $K_{VS} = Q_i \oplus h(RID_i \| RPW_i)$ and verifies whether $W_i \stackrel{?}{=} h(RPW_i \| K_{VS})$ to accept the login. Then, it generates a random number R_1, extracts the timestamp T_1, computes $M_1 = R_1 \oplus h(K_{VS} \| T_1)$, $M_2 = M_{request} \oplus h(R_1 \| K_{VS})$, $M_{ij} = h(M_{request} \| R_1 \| K_{VS} \| T_1)$ and sends (M_1, M_2, M_{ij}, T_1) to V_j.

After verification of the T_1, V_j also inserts its ID_{v_j} and PW_{v_j} to do login. SC_j computes $RN_j = E_j \oplus h(ID_{v_j} \| PW_{v_j})$, $RID_j = h(ID_{v_j} \| RN_j)$, $RPW_j = h(PW_{v_j} \| RN_j)$ and $K_{VS} = Q_j \oplus h(RID_j \| RPW_j)$ and verifies whether $W_j \stackrel{?}{=} h(RPW_j \| K_{VS})$ to accept the login. Next, it calculates $R_1 = M_1 \oplus h(K_{VS} \| T_1)$, $M_{request} = M_2 \oplus h(R_1 \| K_{VS})$ and verifies whether $M_{ij} \stackrel{?}{=} h(M_{request} \| R_1 \| K_{VS} \| T_1)$ to authenticate V_i. Then it generates a random number R_2 and extracts the current time stamp T_2 and computes the session key

TABLE 2. Used notations in IoV-SMAP.

Symbol	Description
V_i	i^{th} vehicle
VS	Vehicle server
IS	Infrastructure
ID/PW	Identity/password of the vehicle
ID_{IS}	Identity of infrastructure
RN_i	Random number generated by V_i
N_{VS}	Random number generated by VS
R_i, B_i	Random numbers generated by V_i and IS respectively
T_1, T_2	Timestamps
K_{VS}	Master key of VS
$h(.)$	One-way hash function
SK	Session key between P_i and CS
$\|$	Concatenation
$\oplus$	XOR operation
$\mathcal{A}$	Adversary

$SK = h(R_1\|R_2\|K_{VS})$, $M_3 = (M_{response}\|R_2) \oplus h(K_{VS}\|R_1\|T_2)$ and $M_{ji} = h(M_{response}\|SK\|T_2)$ and sends (M_3, M_{ji}, T_2) to V_i.

V_i verifies the received timestamp and computes $(M_{response}\|R_2) = M_3 \oplus h(K_{VS}\|R_1\|T_2)$, $SK = h(R_1\|R_2\|K_{VS})$ and verifies whether $M_{ji} \stackrel{?}{=} h(M_{response}\|SK\|T_2)$ to authenticate V_j. The authentication process between V_i and IS is almost the same.

B. SECURITY FLAWS

Here, the security analysis of IoV-SMAP is presented.

1) COMMON MASTER KEY

It is clear that the master key of VS, i.e. K_{VS} is known to any vehicle or IS. Hence, following the adversarial model which has been proposed by Hosseinzadeh *et al.* [24], compromising any vehicle compromises whole the network.

2) OFF-LINE PASSWORD GUESSING ATTACK BY AN INSIDER

In reality, the passwords and the usernames are selected form a limited dictionary. Hence, to make it impractical for the adversary to do off-line password guessing attack such values are masked by nonces. However, in any case any privilege adversary, even with access to the transferred messages (even in the registration phase) and also the memory of SC_i, should not be able to guess the user password using the dictionary attack. Let assume an insider adversary $\mathcal{A}$ knows the transferred values between V_i and VS, i.e. $RID_i = h(ID_{v_i}\|PW_{v_i})$, $RPW_i = h(PW_{v_i}\|RN_i)$ and also the content of the SC_i's memory, i.e. $Q_i = K_{VS} \oplus h(RID_i\|RPW_i)$, $W_i = h(RPW_i\|K_{VS})$ and $E_i = RN_i \oplus h(PW_{v_i}\|RID_i)$. Besides extracting the master key of VS, i.e. $K_{VS} = Q_i \oplus h(RID_i\|RPW_i)$, $\mathcal{A}$ can do off-line password guessing attack as follows:

1) $\mathcal{A}$ guesses a value for PW_{v_i}:
   a) $RN_i = E_i \oplus h(PW_{v_i}\|RID_i)$.
   b) If $RPW_i == h(PW_{v_i}\|RN_i)$, then returns PW_{v_i}; otherwise goes to Step 1.

TABLE 3. Used notations in EGP.

Symbol	Description
R	Reader
T	Tag
$\mathcal{A}$	Adversary
n_i	A random number
$\bar{b}$	The inverse of the binary value b
$(\mathcal{X})_i$	The i^{th} bit of the string $\mathcal{X}$
$P_x(m, n)$	The permutation of m based on n
$P_x^{-1}(m, n)$	Inverse of $P_x(m, n)$
ID^i	The tag's static identifier of the tag at i-th session
IDS^i	The pseudo identifier of the tag at the time i
K^i	The secret key of the tag at the time i
r	A security parameter which determines the buffer length in the reader side
b	The buffer size in the tag side
$\|$	Concatenation
$\oplus$	XOR operation
$Rot(.,.)$	Left rotation used in RAPP [11]
$Per(.,.)$	The permutation function used in RAPP [11]

Following the above attack, $\mathcal{A}$ is able to extract the password of V_i with the complexity of $2^{|D_{pw}|}$, where $|D_{pw}|$ denotes the entropy of password space in bits. Given PW_{v_i} and $RID_i = h(ID_{v_i}\|PW_{v_i})$, we can also use off-line guessing to determine ID_{v_i}.

IV. QUESTIONING THE SECURITY OF EGP

Through this section, we are using a list of notations that are represented in Table 3.

A. A BRIEF DESCRIPTION OF EGP

Khalid *et al.* recently proposed an ultra-lightweight mutual authentication protocol called EGP [26] which follows a framework already proposed by Tian *et al.*, to design an ultra-lightweight mutual authentication protocol called RAPP [11]. In EGP, tags only use two simple operations: bitwise XOR and a very lightweight reconstruction function $P_x(.)$, defined as follows:

Definition: Let $(\mathcal{X})_i$ denote the i^{th} bit of $\mathcal{X}$, $m = (m)_0\|(m)_1\|(m)_2\|\cdots\|(m)_{l-1}$ and $n = (n)_0\|(n)_1\|(n)_2\|\cdots\|(n)_{l-1}$, where $(m)_i, (n)_i \in \{0, 1\}$ and $\|$ denotes concatenation. Then the permutation of m based on n, denoted as $P_x(m, n)$, is as follows:

- $z = 0, t = 0$;
- for $i = \{0, \ldots, l-1\}$:
  - if $n_i = 1$ then $m^*_z = m_i$ and $z = z + 1$;
  - else-if $n_i = 0$, $m^*_{l-1-t} = m_i$ and $t = t + 1$;
- return $P_x(m, n) = m^* \oplus n$

In [20], a generalized attack has been proposed which can desynchronize the tag and the reader, if only a party of the protocol contributes to the protocol randomness and any of them keeps the history of the last successful

Vehicle V_i		**Vehicle** V_j
ID_{v_i}, PW_{v_i}, W_i		ID_{v_j}, PW_{v_j}, W_j
Inserts its ID_{v_i}, PW_{v_i} to do login. SC_i computes $RID_i = h(ID_{v_i} \| PW_{v_i})$, $RN_i = E_i \oplus h(PW_{v_i} \| RID_i)$, $RPW_i = h(PW_{v_i} \| RN_i)$, $K_{VS} = Q_i \oplus h(RID_i \| RPW_i)$, verifies whether $W_i \stackrel{?}{=} h(RPW_i \| K_{VS})$ to accept the login. If so, generates R_1, extracts T_1, computes $M_1 = R_1 \oplus h(K_{VS} \| T_1)$, $M_2 = M_{request} \oplus h(R_1 \| K_{VS})$, $M_{ij} = h(M_{request} \| R_1 \| K_{VS} \| T_1)$	$\xrightarrow{(M_1, M_2, M_{ij}, T_1)}$	After T_1 verification, inserts its ID_{v_j}, PW_{v_j} to do login. SC_j computes $RN_j = E_j \oplus h(ID_{v_j} \| PW_{v_j})$, $RID_j = h(ID_{v_j} \| RN_j)$, $RPW_j = h(PW_{v_j} \| RN_j)$, $K_{VS} = Q_j \oplus h(RID_j \| RPW_j)$ verifies whether $W_j \stackrel{?}{=} h(RPW_j \| K_{VS})$ if so, calculates $R_1 = M_1 \oplus h(K_{VS} \| T_1)$, $M_{request} = M_2 \oplus h(R_1 \| K_{VS})$ verifies whether $M_{ij} \stackrel{?}{=} h(M_{request} \| R_1 \| K_{VS} \| T_1)$ if so, generates R_2 , T_2, computes $SK = h(R_1 \| R_2 \| K_{VS})$, $M_3 = (M_{response} \| R_2) \oplus h(K_{VS} \| R_1 \| T_2)$ $M_{ji} = h(M_{response} \| SK \| T_2)$
	$\xleftarrow{(M_3, M_{ji}, T_2)}$	
V_i verifies T_2, computes $(M_{response} \| R_2) = M_3 \oplus h(K_{VS} \| R_1 \| T_2)$, $SK = h(R_1 \| R_2 \| K_{VS})$, verifies whether $M_{ji} \stackrel{?}{=} h(M_{response} \| SK \| T_2)$ to authenticate V_j		

FIGURE 3. Mutual authentication phase of IoV-SMAP protocol [22].

parameters. To overcome this attack, the EGP's designers assumed that the tag keeps the history of old shared parameters and the reader keeps a longer history, e.g. r previous used values, in a dynamic memory. Hence, in the i^{th} session of this protocol, as the shared parameters, the tag keeps $ID, IDS^i, IDS^{i-1}, K^i, K^{i-1}$ while the reader keeps ID,$(IDS^i, IDS^{i-1}, \ldots, IDS^{i-r+1})$, $(K^i, K^{i-1}, \ldots, K^{i-r+1})$, where ID is the tag's static identifier. The details of the EGP protocol, as depicted in Figure 4, are as follows:

1) The reader R, sends Hello to the target tag T.
2) T replies with its IDS.
3) R generates two random numbers n_1 and n_2 and computes $A\|B\|C$ and sends it to the tag, where $K^* = P_x(K \oplus n_2, n_1)$ and:

$$A = P_x(n_1, K) \tag{1}$$

$$B = P_x(n_2, K \oplus n_1) \tag{2}$$

$$C = P_x(K^* \oplus n_2 \oplus n_1, K) \tag{3}$$

4) T extracts n_1 from A as $n_1 = P_x^{-1}(A, K)$, extracts n_2 from B as $n_2 = P_x^{-1}(B, K \oplus n_1)$, and evaluates the received value for C to authenticate R. If the reader has been authenticated, the tag computes $D = P_x(K^* \oplus ID \oplus n_1, n_2)$ and sends it to R. In addition, T updates its new shared values as $K^i = K^*$, $IDS^i = P_x(IDS \oplus n_2, n_1 \oplus n_2)$, also keeps the history of the values as K^{i-1} and IDS^{i-1}.

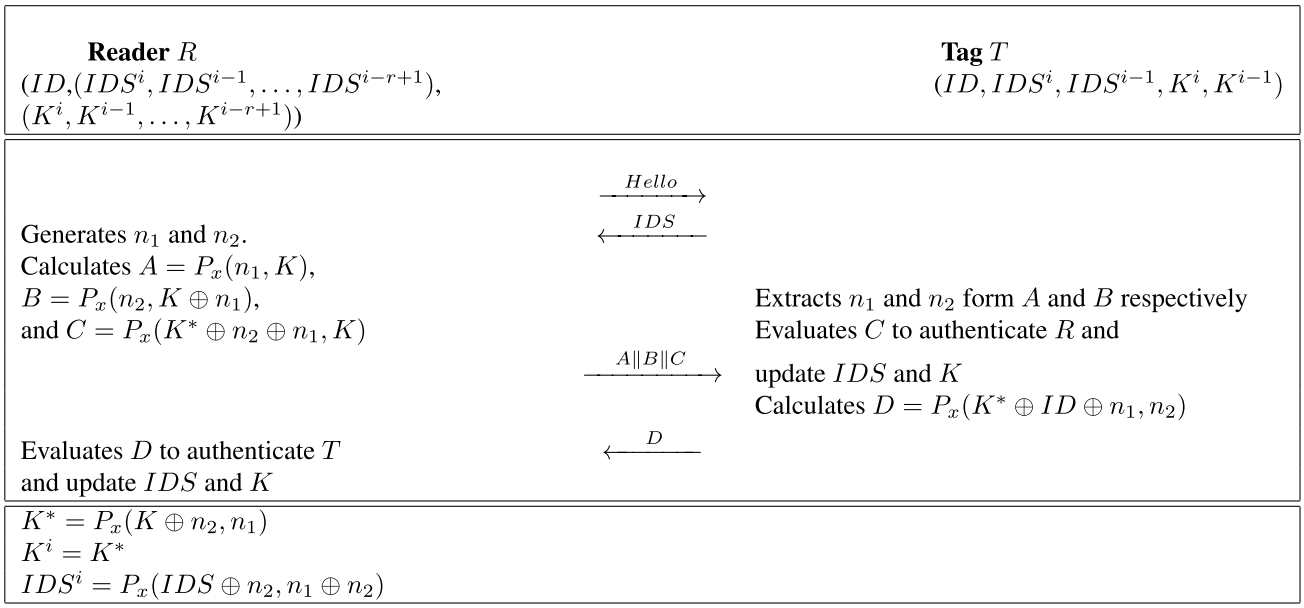


FIGURE 4. Mutual authentication phase of EGP protocol [26].

5) R evaluates the received value of D to authenticate T. If the tag has been authenticated, the reader adds $K^i = K^*$ and $IDS^i = P_x(IDS \oplus n_2, n_1 \oplus n_2)$ to its dynamic memory.

B. SECURITY ANALYSIS OF EGP

In this section, we present the first third party analysis of EGP, to the best of our knowledge. The presented attacks include desynchronization attack and secret disclosure attack.

1) DESYNCHRONIZATION ATTACK

In EGP, similar to many other protocols, e.g. RAPP [11], R^{2}AP [15], RCIA [16], KMAP [17], SLAP [18] and UMAPSS [19], in each session only the reader contributes to the protocol's randomness, by generating two random numbers denoted as n_1 and n_2. Hence, it is easy to impersonate the reader, by eavesdropping a session of the protocol between the reader and the tag and later broadcasting them to the target tag.

On the other hand, Safkhani and Bagheri [20] have shown that it is possible to do desynchronization attack on such protocol when either or both the protocol's parties keep the history of the current and the old shared values. However, to overcome this attack, in EGP the reader uses a dynamic memory and keeps the history of r shared values. Therefore, it is not possible to apply the Safkhani *et al.*'s method as it is to attack EGP. However, using a buffer overflow approach and assuming that the oldest location of the buffer is overwritten each time, it is possible to apply an extension of that attack on this protocol as follow, assuming that the reader keeps at most the history of r shared parameters:

1) Assuming that the reader's history of the shared parameters are $(ID, (IDS^r, IDS^{r-1}, IDS^{r-2}, \ldots, IDS^1)$, $(K^r, K^{r-1}, K^{r-2}, \ldots, K^1))$ and the tag's history of the shared parameters are $(ID, IDS^r, IDS^{r-1}, K^r, K^{r-1})$.
2) In session $r+1$:
   a) The reader R, sends Hello to the target tag T.
   b) T replies with its IDS^r.
   c) R generates two random numbers n_1^r and n_2^r and computes $A^r\|B^r\|C^r$ and sends it to the tag, where $A^r = P_x(n_1^r, K^r)$, $B^r = P_x(n_2^r, K^r \oplus n_1^r)$, $K^{r*} = P_x(K^r \oplus n_2^r, n_1^r)$ and $C^r = P_x(K^{r*} \oplus n_2^r \oplus n_1^r, K^r)$.
   d) The adversary $\mathcal{A}$ stores IDS^r and $A^r\|B^r\|C^r$.
   e) T extracts n_1^r from A^r, extracts n_2^r from B^r, evaluates the received value for C^r and authenticates R. The tag computes $D^r = P_x(K^{r*} \oplus ID \oplus n_1^r, n_2^r)$ and sends it to R. In addition, T updates its new shared values as $K^{r+1} = K^{r*}$, $IDS^{r+1} = P_x(IDS^r \oplus n_2^r, n_1^r \oplus n_2^r)$, also keeps the history of the old values as K^r and IDS^r.
   f) R evaluates the received value of D^r to authenticate T and adds K^{r+1} and IDS^{r+1} to its dynamic memory by overwriting them on K^1 and IDS^1 respectively.
3) In session $r+2$:
   a) R receives IDS^{r+1} from T, generates n_1^{r+1} and n_2^{r+1} and computes $A^{r+1}\|B^{r+1}\|C^{r+1}$ and sends it to the tag, where $A^{r+1} = P_x(n_1^{r+1}, K^{r+1})$, $B^{r+1} = P_x(n_2^{r+1}, K^{r+1} \oplus n_1^{r+1})$, $C^{r+1} = P_x(K^{r+1*} \oplus n_2^{r+1} \oplus n_1^{r+1}, K^{r+1})$, and $K^{r+1*} = P_x(K^r \oplus n_2^{r+1}, n_1^{r+1})$.
   b) The adversary $\mathcal{A}$ stores $IDS^{r+1}, A^{r+1}\|B^{r+1}\|C^{r+1}$ and prevents T from receiving $A^{r+1}\|B^{r+1}\|C^{r+1}$.
   c) In this point, the reader's records of the shared parameters are respectively

$(ID, (IDS^{r+1}, IDS^r, IDS^{r-1}, \ldots, IDS^2), (K^{r+1}, K^r, K^{r-1}, \ldots, K^2))$ and the tag's history of the shared parameters are $(ID, IDS^{r+1}, IDS^r, K^{r+1}, K^r)$.

4) For $c \in \{1, 2, \ldots, r+1\}$:
   a) Assume the legitimate reader R, sends Hello to the target tag T.
   b) T replies with its IDS^{r+c}.
   c) The adversary $\mathcal{A}$ blocks it and sends a random number to the reader.
   d) The legitimate reader R will not find a record and sends another Hello to the target tag T.
   e) T replies with its IDS^r.
   f) R generates two random numbers n_1^c and n_2^c and computes $A^c \| B^c \| C^c$ and sends it to the tag, where $A^c = P_x(n_1^c, K^r)$, $B^c = P_x(n_2^c, K^r \oplus n_1^c)$, $C^c = P_x(K^{r*} \oplus n_2^c \oplus n_1^c, K^r)$, and $K^{r*}_c = P_x(K^r \oplus n_2^c, n_1^c)$.
   g) T extracts n_1^c from A^c, extracts n_2^c from B^c, and evaluates the received value for C^c to authenticate R. The tag computes $D^c = P_x(K^{r*}_c \oplus ID \oplus n_1^c, n_2^c)$ and sends it to R. In addition, T updates its new shared values as $K^{r+c} = K^{r*}_c$, $IDS^{r+c} = P_x(IDS^{r-1} \oplus n_2^c, n_1^c \oplus n_2^c)$, also keeps the history of the values K^r and IDS^r as the last recently authenticated values. Hence, the tags parameters are $(ID, IDS^r, IDS^{r+c}, K^r, K^{r+c})$ in this step.
   h) R evaluates the received value of D^c to authenticate T. It also removes the least recently used parameters and includes new set of parameters to the buffer of the target tag. Hence, the reader's record will be $(ID, (IDS^{r+1+c}, \ldots, IDS^{2+c})$ $,(K^{r+1+c}, K^{r-2+c}, \ldots, K^{2+c}))$.

5) Now, the tags records are $(ID, IDS^r, IDS^{2r+1}, K^r, K^{2r+1})$ and the reader's record are $(ID, K^r, IDS^r, (IDS^{2r+1}, \ldots, IDS^{r+3}), (K^{2r+1}, \ldots, K^{r+3}))$.
   a) In session $r+4$:
      i) $\mathcal{A}$ impersonates R and sends *Hello* to T and the tag sends IDS^{2r+1}.
      ii) $\mathcal{A}$ sends another *Hello* to T and this time the tag responds with IDS^r.
      iii) In response, $\mathcal{A}$ sends the eavesdropped $A^r \| B^r \| C^r$ from Step 2d.
      iv) T authenticates $\mathcal{A}$, sends D^r to $\mathcal{A}$ and updates its records of shared values to $(ID, IDS^r, IDS^{r+1}, K^r, K^{r+1})$.
   b) In session $r+5$:
      i) $\mathcal{A}$, again, impersonates R and sends *Hello* to T and the tag sends IDS^{r+1}.
      ii) In response, $\mathcal{A}$ sends the eavesdropped $A^{r+1} \| B^{r+1} \| C^{r+1}$ from Step 3b.
      iii) T authenticates $\mathcal{A}$, sends D^{r+1} to $\mathcal{A}$ and updates its records of shared values to $(ID, IDS^{r+1}, IDS^{r+2}, K^{r+1}, K^{r+2})$.

At the end of this attack, the tag's records of IDS^{old} and K^{old} do not match any record in the reader side for this tag with the probability of $(1-2^{-2l})^r$, where l is the length of IDS and K in bits. Hence, the reader and the tag will be desynchronized after the above attack with a high probability. It should be noted in the above attack we considered overwriting the least recently used location of the buffer, for any other strategy of overwriting it is possible to adapt the attack. A possible solution could be using an unlimited buffer. However, in this case, the protocol's scalability will be under-question. It should be noted increasing the tag's buffer size, to keep more history of authenticated values, will not fix the problem, although can increase the attack complexity by $O(b)$, where b is the buffer size in the tag side. In general, if the tag uses a buffer of size b, i.e. keeps the history of b sessions, and the reader uses a buffer of size r, i.e. keeps the history of r sessions, it is possible to apply a desynchronization attack with the complexity of $O(r+b)$, which is a linear function of the used buffers sizes. In the previous protocols of RAPP family, e.g. KMAP, RAPP, R^{2}AP, RCIA, and UMAPSS $r, b \in \{1, 2\}$ while in EGP $b = 2$ and r could be a large value. A possible solution to fix this flaw is to force the tag to contribute to the protocol's randomness by generating fresh nonces and using them properly in the calculation of the tag and reader responses.

It worth noting Khalid *et al.* recently proposed another scheme following the EGP paradigm [25] on which the proposed desynchronization attack is applicable. However, for the sake of simplicity, we omit to repeat the attack against that scheme.

2) SECRET DISCLOSURE ATTACK

Inspired by a secret disclosure attack on RAPP [12], in this section we propose a secret disclosure attack against EGP. RAPP and EGP both have the same protocol structure and differ only in their defined permutation function and how the A, B, C, and D messages and K and IDS are defined. Therefore, the secret disclosure attack against RAPP [12] with modifications that make it suitable for EGP's permutation function and messages can also be applied to this protocol. However, the attack on RAPP will not be applicable as it is on EGP because the details are not the same. For example, in RAPP, A, B, C, D and E are transferred over the protocol and computed as follows [11]:

$$A = Per(K_2, K_1) \oplus n_1,$$
$$B = Per(K_1 \oplus K_2, Rot(n_1, n_1)) \oplus Per(n_1, K_1),$$
$$C = Per(n_1 \oplus K_1, n_1 \oplus K_3) \oplus ID,$$
$$D = Per(K_3, K_2) \oplus n_2$$
$$E = Per(K_3, Rot(n_2, n_2)) \oplus Per(n_1, K_3 \oplus K_2).$$

The structure of the used permutations is not the same also. Considering the differences, in this section we apply the modified secret disclosure attack on EGP.

In each session of EGP, R generates two random numbers n_1 and n_2 and computes $A \| B \| C$ and sends it to the tag, where

$A = P_x(n_1, K), B = P_x(n_2, K \oplus n_1), C = P_x(K^* \oplus n_2 \oplus n_1, K)$ and $K^* = P_x(K \oplus n_2, n_1)$. Hence, we can rewrite the message C as $C = P_x((P_x(K \oplus n_2, n_1)) \oplus n_2 \oplus n_1, K)$. In response, the tag sends $D = P_x(K^* \oplus ID \oplus n_1, n_2)$, which can be rewritten as $D = P_x((P_x(K \oplus n_2, n_1)) \oplus ID \oplus n_1, n_2)$. On the other hand, given the definition of the used permutation $P_x(m, n)$ in the structure of EGP, for m and n distributed randomly over $\{0, 1\}^l$, it is clear that $(P_x(m, n))_0 = \overline{m_0}$ with the probability of $\frac{1}{2}$, i.e. if $n_0 = 1$ then $(P_x(m, n))_0 = m_0 \oplus 1 = \overline{m_0}$, where given $x \in \{0, 1\}$ its inverse is denoted by $\bar{x}$; otherwise $(P_x(m, n))_{l-1} = m_0 \oplus n_{l-1}$. Based on these properties, we propose an attack to disclose secret parameters of EGP. The proposed attack includes two phases: learning phase and disclosure phase. In the learning phase of the attack, the adversary semi-actively (the adversary just eavesdrops or blocks the transferred messages and do not capable to modify them) gathers the required information from the transferred messages between the target tag and a legitimate reader and stores them in a table T_L. The learning phase of the proposed attack is as follows:

1) Assuming that the reader's history of the shared parameters are $(ID, (IDS^r, IDS^{r-1}, IDS^{r-2}, \ldots, IDS^1), (K^r, K^{r-1}, K^{r-2}, \ldots, K^1))$ and the tag's history of the shared parameters are $(ID, IDS^r, IDS^{r-1}, K^r, K^{r-1})$.
2) For $c \in \{1, 2, \ldots, u\}$:
   a) Assume the legitimate reader R, sends Hello to the target tag T.
   b) T replies with its IDS^{r+c-1}.
   c) The adversary $\mathcal{A}$ blocks it, stores it in row $c-1$, and sends a random number to the reader.
   d) The legitimate reader R will not find a record and sends another Hello to the target tag T.
   e) T replies with its IDS^{r-1}.
   f) R generates two random numbers n_1^c and n_2^c and computes $A^c \| B^c \| C^c$ and sends it to the tag, where $A^c = P_x(n_1^c, K^{r-1})$, $B^c = P_x(n_2^c, K^{r-1} \oplus n_1^c)$, and $C^c = P_x(K^{r-1^*} \oplus n_2^c \oplus n_1^c, K^{r-1})$, where $K^{r-1^*}_c = P_x(K^{r-1} \oplus n_2^c, n_1^c)$.
   g) The adversary eavesdrops $A^c \| B^c \| C^c$.
   h) T extracts n_1^c from A^c, extracts n_2^c from B^c, and evaluates the received value for C^c to authenticate R. The tag computes $D^c = P_x(K^{r-1^*}_c \oplus ID \oplus n_1^c, n_2^c)$ and sends it to R. In addition, T updates its new shared values as $K^{r+c} = K^{r-1^*}_c$ and $IDS^{r+c} = P_x(IDS^{r-1} \oplus n_2^c, n_1^c \oplus n_2^c)$, also keeps the history of the values K^r and IDS^r as the last recently authenticated values. Hence, the tag's parameters are $(ID, IDS^r, IDS^{r+c}, K^r, K^{r+c})$ in this step.
   i) The adversary stores the tuple A^c, B^c, C^c, D^c in row c of T_L and blocks D^c.
3) The adversary also stores IDS^{r-1}.
4) End.

Hence, at the end of the learning phase, the adversary has a table of u tuples $(IDS^{r+i}, A^i, B^i, C^i, D^i)$, for $1 \le i \le u$, last tuple which has no record of IDS^{r+u} and row 0 which only includes IDS^r, although an active adversary can send a message Hello to the tag to obtain that value also. Given those information, the adversary recovers the secret key K^{r-1}, bit by bit. To this end and to recover the first bit of K^{r-1}, i.e. $(K^{r-1})_0$, the adversary assumes that $(K^{r-1})_0 = 1$. If it is so then, given that $A^j = P_x(n_1^j, K^{r-1})$ for any $j \in \{1, \ldots, u\}$, the adversary can extract $(n_1^j)_0$ as $(n_1^j)_0 = (A^j)_0 \oplus (K^{r-1})_0 = \overline{(A^j)_0}$. On the other hand, $B^j = P_x(n_2^j, K^{r-1} \oplus n_1^j)$. Hence, if $(n_1^j)_0 = 0$, we have $(K^{r-1})_0 \oplus (n_1^j)_0 = 1$ and $(n_2^j)_0 = (B^j)_0 \oplus (K^{r-1})_0 \oplus (n_1^j)_0 = \overline{(B^j)_0}$. We know that $C^j = P_x(K^{r-1^*}_j \oplus n_2^j \oplus n_1^j, K^{r-1})$ and $D^j = P_x(K^{r-1^*}_j \oplus ID \oplus n_1^j, n_2^j)$, where $K^{r-1^*}_j = P_x(K^{r-1} \oplus n_2^j, n_1^j)$ and we assumed $(K^{r-1})_0 = 1$. Next, if $(n_2^j)_0 = 1$ then:

$$(D^j)_0 \oplus (C^j)_0 = (K^{r-1^*}_j \oplus ID \oplus n_1^j)_0 \oplus (K^{r-1})_0 \oplus (K^{r-1^*}_j \oplus n_2^j \oplus n_1^j)_0 \oplus (n_2^j)_0 = \overline{ID_0} \quad (4)$$

Note that we assumed $(n_1^j)_0 = 0$. Hence, the adversary can extract $ID_0 = (D^j)_0 \oplus (C^j)_0 \oplus 1$. Assuming that the assumption is correct, i.e. $(K^{r-1})_0 = 1$, then the adversary will receive an identical value of ID_0 for all tuples in T_L that are satisfying $(n_1^j)_0 = 0$ and $(n_2^j)_0 = 1$, which its probability is $\frac{1}{4}$; otherwise for any tuple it receives a random bit-value as ID_0, which can be used as a countermeasure to filter wrong guess for $(K^{r-1})_0$. Hence, if the adversary received identical value for ID_0, for all tuples in T_L for them are satisfying $(n_1^j)_0 = 0$ and $(n_2^j)_0 = 1$, assumes that $(K^{r-1})_0 = 1$, otherwise assumes $(K^{r-1})_0 = 0$. The success probability of the adversary to recover the correct value of $(K^{r-1})_0$ is determined as $1 - (1 - \frac{1}{2})^{\frac{u}{4}-1}$. For $u = 36$ the success probability of extracting the correct value of $(K^{r-1})_0$ is $1 - 2^{-16} = 0.996$.

Given $(K^{r-1})_0$ and T_L, it is possible to extract other bits of K^{r-1} step by step. For example, to extract $(K^{r-1})_{l-1}$ we have two cases, depending on the value of $(K^{r-1})_0$ in the previous steps, as follows:

1) **Case 1** $(K^{r-1})_0 = 1$**:** if it is so, then the adversary can extract $(n_1^j)_0$ as $(n_1^j)_1 = (A^j)_1 \oplus (K^{r-1})_1 = \overline{(A^j)_1}$. On the other hand, $B^j = P_x(n_2^j, K^{r-1} \oplus n_1^j)$. Hence, if $(n_1^j)_0 = 0$, we have $(K^{r-1})_0 \oplus (n_1^j)_0 = 1$ and $(n_2^j)_0 = (B^j)_0 \oplus (K^{r-1})_0 \oplus (n_1^j)_0 = \overline{(B^j)_0}$. If $(n_2^j)_0 = 1$ then $(K^{r-1})_0 \oplus (n_1^j)_0 = 0$ and $(B^j)_{l-1} = (K^{r-1})_{l-1} \oplus (n_2^j)_0$ and for any $i \neq j$ such that $(n_1^j)_0 = 1$ and $(n_1^i)_0 = 1$ we have $(B^j)_{l-1} \oplus (B^i)_{l-1} = (n_2^j)_0 \oplus (n_2^i)_0$. Hence, we can divide the tuples of T_L for which $(n_1^j)_0 = 1$ into two sets S^0 and S^1, where namely for any tuple in $(IDS^{r+i}, A^i, B^i, C^i, D^i) \in S^0$ we have $(B^i)_{l-1} = 0$ and for any tuple $(IDS^{r+i}, A^i, B^i, C^i, D^i) \in S^1$ the $(B^i)_{l-1} = 1$ and any set has identical value for $(n_2)_0$. In addition, we know that $IDS^{r+i} = P_x(IDS^{r-1} \oplus$

$n_2^i, n_1^i \oplus n_2^i$) and we know the values of IDS^{r-1} and also $(n_1^i)_0$. For $(n_1^i)_0 = 1$, if $(n_2^i)_0 = 0$ then:

$$(IDS^{r+i})_0 = (IDS^{r-1})_0 \oplus (n_2^i)_0 \oplus (n_1^i)_0 \oplus (n_2^i)_0 = (IDS^{r-1})_0 \oplus (n_1^i)_0 \quad (5)$$

This equation helps to determine the value of $(n_2^i)_0$ for any tuple for which $(n_1^i)_0 = 1$ and $(IDS^{r+i})_0 \neq (IDS^{r-1})_0 \oplus (n_1^i)_0$, i.e. in this case with the probability of '1' we have $(n_2^i)_0 = 1$. Hence, if there is any tuple in S^0 for which $(IDS^{r+i})_0 \neq (IDS^{r-1})_0 \oplus (n_1^i)_0$ it means that for all tuples in S^0 the value of $(n_2)_0 = 0$ and for all tuples in S^1 the value of $(n_2)_0 = 1$ and vice versa. In this way, we know whole the bits of n_2.
On the other hand, given $(n_2)_0$ for any tuple in S^0 and S^1 and given that for any tuple in those sets $(B^j)_{l-1} = (K^{r-1})_{l-1} \oplus (n_2^j)_0$, determining $(K^{r-1})_{l-1}$ is trivial.

2) **Case 2** $(K^{r-1})_0 = 0$**:** if it is so, then for any tuple $(IDS^{r+i}, A^i, B^i, C^i, D^i) \in T_L$, we have $(A^i)_{l-1} = (K^{r-1})_{l-1} \oplus (n_1^i)_0$ and for any $i \neq j$ we have $(A^j)_{l-1} \oplus (A^i)_{l-1} = (n_1^j)_0 \oplus (n_1^i)_0$. Hence, we can divide the tuples of T_L into two sets T^0 and T^1, where namely for any tuple in $(IDS^{r+i}, A^i, B^i, C^i, D^i) \in T^0$ we have $(A^i)_{l-1} = 0$ and for any tuple $(IDS^{r+i}, A^i, B^i, C^i, D^i) \in T^1$ we have $(A^j)_{l-1} = 1$ and any set has identical value for $(n_1)_0$. Next, for example we assume $(n_1^i)_0 = 1$ for any tuple $(IDS^{r+i}, A^i, B^i, C^i, D^i) \in T^1$ and $(n_1)_0 = 0$ for any tuple in T^0. In this case, when $(K^{r-1})_0 \oplus (n_1^i)_0 = 1$ we have $(n_2^i)_0 = (B^i)_0 \oplus (K^{r-1})_0 \oplus (n_1^i)_0 = \overline{(B^i)_0}$. Similar to Case 1, we know that $IDS^{r+i} = P_x(IDS^{r-1} \oplus n_2^i, n_1^i \oplus n_2^i)$ and we know the values of IDS^{r-1} and also $(n_1^i)_0$. For $(n_1^i)_0 = 1$, if $(n_2^i)_0 = 0$ then we can use Equation 1 to verify the correctness of our guess for the value of $(n_1)_0$ in T^1. More precisely, if there is a tuple in T^1, for which $(n_2^i)_0 = 0$ and $(IDS^{r+i})_0 \neq (IDS^{r-1})_0 \oplus (n_1^i)_0$, the guessed value for $(n_1)_0$ in T^1 was wrong and we should change the assumption, i.e. we should assume $(n_1)_0 = 1$ for any tuple in T^1 and $(n_1)_0 = 0$ for any tuple in T^0. Similarly, we should also do the checking for T^0 also. In this way, we can determine the value of $(n_1)_0$ for any tuple in T^L. Given that $(A^i)_{l-1} = (K^{r-1})_{l-1} \oplus (n_1^i)_0$, determining $(K^{r-1})_{l-1}$ will be trivial. In addition, we can use an approach almost similar to Case 1 to determine $(n_2)_0$ of any tuple.

So far, we could determine $(K^{r-1})_0$ and $(K^{r-1})_{l-1}$ and for any tuple $(IDS^{r+i}, A^i, B^i, C^i, D^i) \in T_L$, we know $(n_1^i)_0$ and $(n_2^i)_0$. Given $(K^{r-1})_0$, the possible locations of $(n_1^i)_0$ will be deterministic. For example, if $(K^{r-1})_0 = 0$, then for any tuple in T_L, we have $(A^i)_{l-1} = (K^{r-1})_{l-1} \oplus (n_1^i)_0$. Hence, if $(K^{r-1})_1 = 0$, for any tuple in T_L, we have $(A^i)_{l-2} = (K^{r-1})_{l-2} \oplus (n_1^i)_1$; otherwise if $(K^{r-1})_1 = 1$, we have $(A^i)_0 = (K^{r-1})_0 \oplus (n_1^i)_1$. In any case, the adversary guesses a value for $(K^{r-1})_1$, categorizes the tuples based on $(n_1^i)_1$, e.g. T^1 and T^0, guesses the value of $(n_1^i)_1$ in the set T^1 for example, categorizes the tuples based on $(n_2^i)_1$, e.g. S^1 and S^0, guesses the value of $(n_2^i)_1$ in the set S^1 for example, verifies the correctness of the guesses using Equation 1 and filters the wrong guesses. This approach could be repeated to reveal whole bits of K^{r-1}. Assuming that $l = 128$ and $u = 36$, the parameter length, the success probability of the adversary to recover whole bits of K correctly will be 0.606. When we increase the complexity of the learning phase to $u = 68$, the success probability of the adversary to recover whole bits of K correctly will be 0.998 and for $u = 35$ the success probability will be 0.55.

Given K^{r-1} from the above attack and a tuple in T_L, the adversary can easily extract the static ID also. Please note that ID is a constant value and can be used to trace the target tag. Hence, EGP does not provide desired anonymity/privacy that was claimed by the designers. In addition, given the only secret parameters of the protocol, i.e. ID and K, it is possible to impersonate the tag, the reader or desynchronize them.

V. DISCUSSION

In this article, we analysed the security of three recent protocols, i.e., Son *et al.*, IoV-SMAP, and EGP. Son *et al.* is a bilinear pairings based protocol and uses blockchain to propose an authentication protocol for cloud-based telecare medical information system (TMIS). On the other hand, IoV-SMAP is a hash based scheme which has been proposed as a message authentication protocol for Internet of Vehicles (IoV). Among them EGP is an ultra-lightweight protocol that uses very efficient component to provide desired security for low cost tags. However, our security analysis demonstrates important security flaws on these protocols. This analysis highlights once again that message structures could be even more important than the used primitives. More precisely, employing secure primitives does not guarantee the protocol's security and details of the transferred messages are very important. While the vulnerability of EGP could be expected, due to used primitives and also the previous attacks on the similar designing such as RAPP, however, the other two protocols could perform much better because they are using sound cryptography primitives i.e. one-way hash functions and bilinear pairings. Hence, we suggest to consider the guidelines below on designing any cryptography protocols:

- It may not be possible to design a secure ultra-lightweight protocol. Many previously broken schemes such as RAPP, SASI, R^{2}AP, RCIA, KMAP, SLAP and UMAPSS are evidences for this advice.
- If the protocol does not include timestamp then all parties should contribute to the protocol's randomness. Otherwise it should be possible to do replay attack and even desynchronization attack if the protocol entities update their shared parameters, e.g. similar to EGP.
- Insider attacker is an important class of adversaries and should be seriously considered in designing most of the protocols, especially for sensitive applications such as Internet of Medical Things (IoMT). While we can avoid some attacks by considering a registration phase

over a secure channel for example, however, the transferred messages over such channel is accessible for the privileged insider adversary. Hence, designing a secure protocol in this scenario could be more challenging.

- Despite of the used cryptography primitives, the details of all transferred messages should be taken into account. The information leakage may come from the combination of the messages from a single or multiple sessions.
- Users generally select their usernames and passwords from a limited set of words. That set could be more limited if we have some side information related to the user, e.g. its nationality, gender, age hobbies and etc. Hence, considering a fully random password or ID is not a realistic assumption in most of the cases. Hence, designers should protect properly information related to the IDs and passwords that are stored in a memory, e.g. a smart card or a device memory.
- In many applications, edge devices can be physically accessed by the adversary and compromised to read their secret parameters. In addition, in a large scale network we cannot trust all participants and we should always consider the possibility of existing malicious nodes. Such nodes may provide the adversary with extra information related to the network and groups in the network. The designers should ensure that existing such a participant cannot affect the security of other users.
- Untraceability is a crucial property for many applications, e.g., IoMT and IoV. However, any user dependent constant or link-able parameter(s) could be used to trace the user. Hence, transformed messages should not be distinguishable from random values for the adversary. An insurance for this purpose could be an accurate security proof in random oracle model, e.g. [44].

VI. CONCLUSION

In this paper, we analyzed the security of three recent protocols for constrained environments successfully by proposing important attacks against them. The analysed protocols belong to different designing strategies and based on different cryptographic primitives.

EGP was among the latest ultra-lightweight protocols in literature, to the best of our knowledge, which can be considered as a member of the RAPP family of authentication protocols, due to its structure. This study along with other related studies on RAPP family specifically and other ultra-lightweight protocols in general, e.g. SecLAP [47] and the recent design by Sidorov *et al.* [48] and their cryptanalysis [31], [32], demonstrate that it may not be possible to design a secure ultra-lightweight without using a sound cryptography foundation. Although it may not be possible to use some cryptographic components such as strong public key infrastructures for passive tags, however, recent advances in symmetric cryptography and also the current ongoing competition for lightweight cryptography, i.e. the NIST LWC (Lightweight Cryptography (LWC) Standardization) [49] should be a promising direction to be considered to design secure protocols for the constrained environments.

On the other hand, the proposed protocol by Son *et al.* used very strong cryptographic component known as bilinear pairings and the proposed protocol by Yu *et al.*, IoV-SMAP, is based on one-way hash function. Hence, this study shows that using a sound cryptographic component may not be enough and designers should pay more attention to the structure of the messages while proposing a scheme to minimize information leakages.

Finally, we hope the community pay more attention to the security analysis to avoid proposing easy to break schemes.

REFERENCES

[1] Ö. Aydin, G. Dalkiliç, and C. Kösemen, "A novel grouping proof authentication protocol for lightweight devices: GPAPXR+," *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 28, no. 5, pp. 3036–3051, Sep. 2020.

[2] *EPCglobal Tag Data Standards Version 1.4*, EPCglobal, Brussels, Belgium, Jun. 2008.

[3] H.-Y. Chien, "SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity," *IEEE Trans. Dependable Secure Comput.*, vol. 4, no. 4, pp. 337–340, Oct. 2007.

[4] P. Peris-Lopez, J. C. H. Castro, J. M. Estévez-Tapiador, and A. Ribagorda, "Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol," in *Proc. 9th Int. Workshop Inf. Secur. Appl. (WISA)*, in Lecture Notes in Computer Science, Jeju Island, South Korea, vol. 5379, K. Chung, K. Sohn, and M. Yung, Eds. Berlin, Germany: Springer, Sep. 2008, pp. 56–68.

[5] G. Avoine, X. Carpent, and J. Hernandez-Castro, "Pitfalls in ultralightweight authentication protocol designs," *IEEE Trans. Mobile Comput.*, vol. 15, no. 9, pp. 2317–2332, Sep. 2016.

[6] G. Avoine and X. Carpent, "Yet another ultralightweight authentication protocol that is broken," in *Proc. 8th Int. Workshop Radio Freq. Identificat. Secur. Privacy Issues (RFIDSec)*, in Lecture Notes in Computer Science, Nijmegen, The Netherlands, vol. 7739, J. Hoepman and I. Verbauwhede, Eds. Berlin, Germany: Springer, Jul. 2012, pp. 20–30.

[7] G. Avoine, X. Carpent, and B. Martin, "Privacy-friendly synchronized ultralightweight authentication protocols in the storm," *J. Netw. Comput. Appl.*, vol. 35, no. 2, pp. 826–843, Mar. 2012.

[8] R. C.-W. Phan, "Cryptanalysis of a new ultralightweight RFID authentication protocol—SASI," *IEEE Trans. Dependable Secure Comput.*, vol. 6, no. 4, pp. 316–320, Oct. 2009.

[9] P. D'Arco and A. De Santis, "On ultralightweight RFID authentication protocols," *IEEE Trans. Dependable Secure Comput.*, vol. 8, no. 4, pp. 548–563, Jul. 2011.

[10] D. F. Barrero, J. C. Hernández-Castro, P. Peris-Lopez, D. Camacho, and M. D. R-Moreno, "A genetic tango attack against the david-prasad RFID ultra-lightweight authentication protocol," *Expert Syst.*, vol. 31, no. 1, pp. 9–19, Feb. 2014.

[11] Y. Tian, G. Chen, and J. Li, "A new ultralightweight RFID authentication protocol with permutation," *IEEE Commun. Lett.*, vol. 16, no. 5, pp. 702–705, May 2012.

[12] N. Bagheri, M. Safkhani, P. Peris-Lopez, and J. E. Tapiador, "Weaknesses in a new ultralightweight RFID authentication protocol with permutation-RAPP," *Secur. Commun. Netw.*, vol. 7, no. 6, pp. 945–949, Jun. 2014.

[13] Z. Ahmadian, M. Salmasizadeh, and M. R. Aref, "Desynchronization attack on RAPP ultralightweight authentication protocol," *Inf. Process. Lett.*, vol. 113, no. 7, pp. 205–209, Apr. 2013.

[14] S.-H. Wang, Z. Han, S. Liu, and D.-W. Chen, "Security analysis of RAPP an RFID authentication protocol based on permutation," Cryptol. ePrint Arch., Tech. Rep. 2012/327, 2012. [Online]. Available: https://eprint.iacr.org/2012/327

[15] X. Zhuang, Y. Zhu, and C. Chang, "A new ultralightweight RFID protocol for low-cost tags: R^2 AP," *Wireless Pers. Commun.*, vol. 79, no. 3, pp. 1787–1802, 2014.

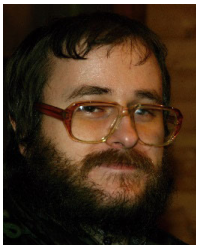
[16] U. Mujahid, M. Najam-ul-Islam, and M. A. Shami, "RCIA: A new ultralightweight RFID authentication protocol using recursive hash," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 1, Jan. 2015, Art. no. 642180.

[17] U. Mujahid, M. Najam-ul-Islam, and S. Sarwar, "A new ultralightweight RFID authentication protocol for passive low cost tags: KMAP," *Wireless Pers. Commun.*, vol. 94, no. 3, pp. 725–744, Jun. 2017.
[18] H. Luo, G. Wen, J. Su, and Z. Huang, "SLAP: Succinct and lightweight authentication protocol for low-cost RFID system," *Wireless Netw.*, vol. 24, no. 1, pp. 69–78, Jan. 2018.
[19] Y. Liu, M. F. Ezerman, and H. Wang, "Double verification protocol via secret sharing for low-cost RFID tags," *Future Gener. Comput. Syst.*, vol. 90, pp. 118–128, Jan. 2019.
[20] M. Safkhani and N. Bagheri, "Generalized desynchronization attack on UMAP: Application to RCIA, KMAP, SLAP and SASI$^+$ protocols," IACR Cryptol. ePrint Arch., Tech. Rep., 2016, p. 905. [Online]. Available: https://eprint.iacr.org/2016/905
[21] S. Son, J. Lee, M. Kim, S. Yu, A. K. Das, and Y. Park, "Design of secure authentication protocol for cloud-assisted telecare medical information system using blockchain," *IEEE Access*, vol. 8, pp. 192177–192191, 2020.
[22] S. Yu, J. Lee, K. Park, A. K. Das, and Y. Park, "IoV-SMAP: Secure and efficient message authentication protocol for IoV in smart city environment," *IEEE Access*, vol. 8, pp. 167875–167886, 2020.
[23] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.
[24] M. Hosseinzadeh, J. Lansky, A. M. Rahmani, C. Trinh, M. Safkhani, N. Bagheri, and B. Huynh, "A new strong adversary model for RFID authentication protocols," *IEEE Access*, vol. 8, pp. 125029–125045, 2020.
[25] M. Khalid, U. Mujahid, M. Najam-ul-Islam, H. Choi, I. Alam, and S. Sarwar, "Ultralightweight resilient mutual authentication protocol for IoT based edge networks," *J. Ambient Intell. Humanized Comput.*, early access, pp. 1–12, Jan. 2021. [Online]. Available: https://link.springer.com/article/10.1007/s12652-020-02732-2
[26] M. Khalid, U. Mujahid, and N.-U.-I. Muhammad, "Ultralightweight RFID authentication protocols for low-cost passive RFID tags," *Secur. Commun. Netw.*, vol. 2019, pp. 1–25, Jul. 2019.
[27] M. Adeli and N. Bagheri, "MDSbSP: A search protocol based on MDS codes for RFID-based internet of vehicle," *J. Supercomput.*, vol. 77, no. 2, pp. 1094–1113, Feb. 2021.
[28] I. A. Kamil and S. O. Ogundoyin, "A lightweight mutual authentication and key agreement protocol for remote surgery application in tactile internet environment," *Comput. Commun.*, vol. 170, pp. 1–18, Mar. 2021.
[29] Y. Zheng and C.-H. Chang, "Secure mutual authentication and key-exchange protocol between PUF-embedded IoT endpoints," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.
[30] T.-F. Lee and W.-Y. Chen, "Lightweight fog computing-based authentication protocols using physically unclonable functions for internet of medical things," *J. Inf. Secur. Appl.*, vol. 59, Jun. 2021, Art. no. 102817.
[31] C. Trinh, B. Huynh, J. Lansky, S. Mildeová, M. Safkhani, N. Bagheri, S. Kumari, and M. Hosseinzadeh, "A novel lightweight block cipher-based mutual authentication protocol for constrained environments," *IEEE Access*, vol. 8, pp. 165536–165550, 2020.
[32] M. Safkhani, S. Rostampour, Y. Bendavid, and N. Bagheri, "IoT in medical & pharmaceutical: Designing lightweight RFID security protocols for ensuring supply chain integrity," *Comput. Netw.*, vol. 181, Nov. 2020, Art. no. 107558.
[33] M. Adeli, N. Bagheri, and H. R. Meimani, "On the designing a secure biometric-based remote patient authentication scheme for mobile healthcare environments," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 2, pp. 3075–3089, Feb. 2021.
[34] S. Rostampour, M. Safkhani, Y. Bendavid, and N. Bagheri, "ECCbAP: A secure ECC-based authentication protocol for IoT edge devices," *Pervas. Mobile Comput.*, vol. 67, Sep. 2020, Art. no. 101194.
[35] B. Malakreddy, "ECC based multifactor authentication and key generation system for IoT healthcare," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 11, pp. 5026–5032, 2021.
[36] D. Rangwani, D. Sadhukhan, and S. Ray, "Cryptanalysis of a robust ECC-based provable secure authentication protocol with privacy preserving for industrial Internet of Things," in *Cloud Security*. Boca Raton, FL, USA: CRC Press, 2021, pp. 76–87.
[37] L. D. Tsobdjou, S. Pierre, and A. Quintero, "A new mutual authentication and key agreement protocol for mobile client—Server environment," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1275–1286, Jun. 2021.
[38] Y.-H. Chuang and Y.-M. Tseng, "CAKE: Compatible authentication and key exchange protocol for a smart city in 5G networks," *Symmetry*, vol. 13, no. 4, p. 698, Apr. 2021.
[39] T.-Y. Wu, Y.-Q. Lee, C.-M. Chen, Y. Tian, and N. A. Al-Nabhan, "An enhanced pairing-based authentication scheme for smart grid communications," *J. Ambient Intell. Humanized Comput.*, early access, pp. 1–13, Jan. 2021. [Online]. Available: https://link.springer.com/article/10.1007/s12652-020-02740-2
[40] D. Abbasinezhad-Mood, A. Ostad-Sharif, M. Nikooghadam, and S. M. Mazinani, "Novel certificateless chebyshev chaotic map-based key agreement protocol for advanced metering infrastructure," *J. Supercomput.*, early access, pp. 1–29, Jan. 2021. [Online]. Available: https://link.springer.com/article/10.1007/s11227-020-03552-z
[41] C.-M. Chen, X. Deng, W. Gan, J. Chen, and S. K. H. Islam, "A secure blockchain-based group key agreement protocol for IoT," *J. Supercomput.*, early access, pp. 1–23, Feb. 2021. [Online]. Available: https://link.springer.com/article/10.1007/s11227-020-03561-y
[42] V. S. Naresh, V. V. L. D. Allavarpu, and S. Reddi, "Blockchain privacy-preserving smart contract centric multiple multiparty key agreement over large WANETs," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 2, p. e4165, Feb. 2021.
[43] Z. Xu, W. Liang, K.-C. Li, J. Xu, and H. Jin, "A blockchain-based roadside unit-assisted authentication and key agreement protocol for internet of vehicles," *J. Parallel Distrib. Comput.*, vol. 149, pp. 29–39, Mar. 2021.
[44] S. Akleylek and K. Seyhan, "A probably secure bi-GISIS based modified AKE scheme with reusable keys," *IEEE Access*, vol. 8, pp. 26210–26222, 2020.
[45] K. Seyhan, T. N. Nguyen, S. Akleylek, K. Cengiz, and S. K. H. Islam, "Bi-GISIS KE: Modified key exchange protocol with reusable keys for IoT security," *J. Inf. Secur. Appl.*, vol. 58, May 2021, Art. no. 102788.
[46] P. Ravi, V. K. Sundar, A. Chattopadhyay, S. Bhasin, and A. Easwaran, "Authentication protocol for secure automotive systems: Benchmarking post-quantum cryptography," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Seville, Spain, Oct. 2020, pp. 1–5.
[47] S. F. Aghili, H. Mala, P. Kaliyar, and M. Conti, "SecLAP: Secure and lightweight RFID authentication protocol for medical IoT," *Future Gener. Comput. Syst.*, vol. 101, pp. 621–634, Dec. 2019.
[48] M. Sidorov, M. T. Ong, R. V. Sridharan, J. Nakamura, R. Ohmura, and J. H. Khor, "Ultralightweight mutual authentication RFID protocol for blockchain enabled supply chains," *IEEE Access*, vol. 7, pp. 7273–7285, 2019.
[49] NIST, Information Technology Laboratory Computer Security Resource Center. (2013). *Lightweight Cryptography (LWC) Standardization: Round 1 Candidates*. [Online]. Available: https://csrc.nist.gov/News/2019/lightweight-cryptography-round-1-candidates

AMIR MASOUD RAHMANI received the B.S. degree from Amir Kabir University, Tehran, in 1996, the M.S. degree from the Sharif University of Technology, Tehran, in 1998, and the Ph.D. degree from IAU University, Tehran, in 2005, all in computer engineering. He is currently a Professor of computer engineering. He is the author/coauthor of more than 350 publications in technical journals and conferences. His research interests include distributed systems, the Internet of Things, and evolutionary computing.

MOKHTAR MOHAMMADI received the B.Sc. degree in computer engineering from Shahed University, Tehran, Iran, in 2003, the M.Sc. degree in computer engineering from Shahid Beheshti University, Tehran, in 2012, and the Ph.D. degree in computer engineering from the Shahrood University of Technology, Shahrood, Iran, in 2018. He is currently with the Department of Information Technology, Lebanese French University, Erbil, Iraq. His current research interests include signal processing, time-frequency analysis, and machine learning.

SHIMA RASHIDI was born in Iran, in 1989. She received the B.E. and M.E. degrees in computer science from the University of Tabriz, Tabriz, Iran, in 2011 and 2013, respectively. She is currently pursuing the Ph.D. degree with the University of Science and Technology, Tehran, Iran. She is currently an Assistant Lecturer with the University of Human Development, Kurdistan Region, Sulaymaniyah, Iraq. Her main areas of research interests include text mining, semi supervised learning, and social network analysis.



JAN LANSKY received the M.Sc. and Ph.D. degrees in computer science and software systems from Charles University, Prague, Czech Republic, in 2005 and 2009, respectively. He has been a Professor with Department of Computer Science and Mathematics, Faculty of Economic Studies, University of Finance and Administration, Prague, since March 2009. Since September 2014, he has been the head of the department. His research interests include cryptocurrencies, text compression, and databases.

STANISLAVA MILDEOVA received the degree from the University of Economics, Prague. She has been an Associate Professor and the Deputy Head of the Department of Informatics and Mathematics, Faculty of Economic Studies, University of Finance and Administration, Prague, Czech Republic, since 2017. In her work, she focuses on applied informatics and systems science with a focus on systems dynamics. She has been the Editor-in-Chief of the Scopus Journal *Acta Informatica Pragensia*.

MASOUMEH SAFKHANI received the Ph.D. degree in electrical engineering from the Iran University of Science and Technology, in 2012, with a focus on security analysis of RFID protocols. She is currently an Associate Professor with the Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran. She is the author/coauthor of over 70 technical articles in information security and cryptology in major international journals and conferences. Her current research interests include security analysis of lightweight and ultra-lightweight protocols, targeting constrained environments, such as RFID, the IoT, VANET, and WSN.

SARU KUMARI received the Ph.D. degree in mathematics from Chaudhary Charan Singh University, Meerut, India, in 2012. She is currently an Assistant Professor with the Department of Mathematics, Chaudhary Charan Singh University. She has published more than 133 research articles in reputed international journals and conferences, including 115 publications in SCI-indexed journals. Her current research interests include information security and applied cryptography. She is a technical program committee member for many international conferences. She has served as a Lead/Guest Editor of four special issues in SCI journals of Elsevier, Springer, and Wiley. She is on the Editorial Board of more than 12 journals of International repute, including seven SCI journals.

MEHDI HOSSEINZADEH received the B.S. degree in computer hardware engineering from Islamic Azad University, Dezfol Branch, Iran, in 2003, and the M.Sc. and Ph.D. degrees in computer system architecture from the Science and Research Branch, Islamic Azad University, Tehran, Iran, in 2005 and 2008, respectively. He is the author/coauthor of more than 150 publications in technical journals and conferences. His research interests include SDN, information technology, data mining, big data analytics, e-commerce, e-marketing, and social networks.