


## Article

# Secure Authentication and Prescription Safety Protocol for Telecare Health Services Using Ubiquitous IoT

Zahid Mahmood <sup>1</sup>, Huansheng Ning <sup>1,\*</sup> , Ata Ullah <sup>2</sup> and Xuanxia Yao <sup>1</sup>

<sup>1</sup> School of Computer and Communication Engineering University of Science and Technology Beijing (USTB), Beijing 10008, China; b20140561@xs.ustb.edu.cn (Z.M.); kathy.yao@163.com (X.Y.)

<sup>2</sup> Department of Computer Science, National University of Modern Languages, Islamabad 44000, Pakistan; aullah@numl.edu.pk

\* Correspondence: ninghuansheng@ustb.edu.cn; Tel.: +86-10-6233-3015

Received: 16 September 2017; Accepted: 12 October 2017; Published: 16 October 2017

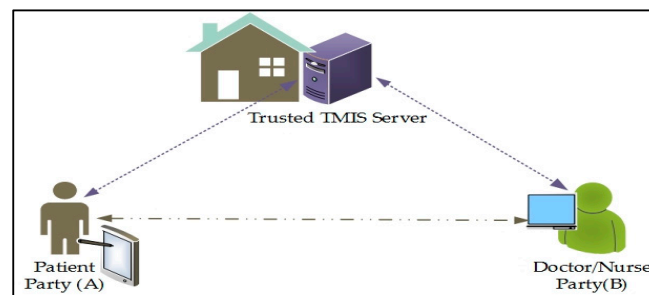
**Abstract:** Internet-of-Things (IoT) include a large number of devices that can communicate across different networks. Cyber-Physical Systems (CPS) also includes a number of devices connected to the internet where wearable devices are also included. Both systems enable researchers to develop healthcare systems with additional intelligence as well as prediction capabilities both for lifestyle and in hospitals. It offers as much persistence as a platform to ubiquitous healthcare by using wearable sensors to transfer the information over servers, smartphones, and other smart devices in the Telecare Medical Information System (TMIS). Security is a challenging issue in TMIS, and resourceful access to health care services requires user verification and confidentiality. Existing schemes lack in ensuring reliable prescription safety along with authentication. This research presents a Secure Authentication and Prescription Safety (SAPS) protocol to ensure secure communication between the patient, doctor/nurse, and the trusted server. The proposed procedure relies upon the efficient elliptic curve cryptosystem which can generate a symmetric secure key to ensure secure data exchange between patients and physicians after successful authentication of participants individually. A trusted server is involved for mutual authentication between parties and then generates a common key after completing the validation process. Moreover, the scheme is verified by doing formal modeling using Rubin Logic and validated using simulations in NS-2.35. We have analyzed the SAPS against security attacks, and then performance analysis is elucidated. Results prove the dominance of SAPS over preliminaries regarding mutual authentication, message integrity, freshness, and session key management and attack prevention.

**Keywords:** authentication; key agreement; telecare medical information system; anonymity; un-traceability

## 1. Introduction

With the development of computing technologies and the expansion of smart-devices and inter-network protocols, hospitals and healthcare organizations are adopting the Telecare Medical Information System (TMIS). The medical identification process has become more resourceful, trustworthy, and efficient by using TMIS. Some Telecare services have been projected in the modern age to ease the workload on the professionals, for example, by methods such as automated healthcare devices, distant health nursing, and patient's health monitoring for remote areas. It is considered as a time-saving, economical, and easy way to access health care facilities remotely. TMIS can lower societal and medical operating cost with improved quality and efficiency [1]. Patient authentication is mandatory to guard against data forgery, misuse, and falsifying the original information by

unauthorized active or passive parties. In TMIS, a trusted medical server is responsible for registering all participants, including doctors, nursing staff, and patients, as illustrated in Figure 1. The remote healthcare server keeps the advanced clinical information of the register patients and gives different administrations like healthcare education, doctors, doctor's facilities, remote medical aid, general healthcare, private care benefit, and the most important is medical prescription. Registered TMIS users for medical services can utilize smart cards to access medical devices and transmit collected statistics to the medical server using a public channel. An intruder may likewise find access over the public open channel. The adversary can eavesdrop, block service, modify, delete records, and can also reply to a message broadcasted over an open channel. Therefore, authentication is mandatory for providing information security, records honesty, and confidentiality of data. It is necessary to overcome all existing threats that are major hurdles and make TMIS trustworthy. A session key must be established for a short-time secure communication between two parties where a new key is established for every new session. We have considered various advanced cryptographic calculations like non-reversible one-way hash function, Rivest-Shamir-Adelman (RSA), Elliptic Curve Cryptography (ECC), and chaotic maps cryptosystem. Despite the fact that both ECC and RSA cryptosystems offer an equal level of security, ECC is additionally advantageous as compared to RSA with respect to computational efficiency [2].



**Figure 1.** Telecare Medical Information System (TMIS) architecture.

An Internet of Things (IoT) is composed of a vast number of small and large devices that are using the internet for sharing data across the nodes of different networks. In an IoT environment, TMIS can serve patients to save time by remotely accessing consultants and doctors by using the new generation of internet. IoT supports a collection of objects, devices, and networks in surroundings to develop a large number of smart applications. It can undeniably improve the identification and dissemination of information regarding emergency situations where appropriate medical aid is mandatory to save precious lives. IoT can also support integration mechanisms for all physical objects with embedded systems or cyber physical systems for monitoring patients at distant locations. It improves diagnosis at the patient's home for better usefulness and effectiveness regarding medical services [3]. IoT promises an attractive future networking prototype.

Secure authentication and session key establishment schemes enable secure communication for health care services. A three-party password authentication key exchange (3-PAKE) scheme provides mutual authentication between doctor-patient-TS at the same time and hides identities from the adversary [4]. It helps in maintaining a secure link by using a common session key for specific communication. Schemes [5–7] discuss secure session key establishment between participants but later [8–10] identified that these plans are vulnerable to man-in-the-middle attack and undetectable online and offline dictionary attacks for guessing passwords. ECC-based schemes are also explored to evaluate the applicability of efficient solutions with desired security strengths using small key sizes as compared to preliminaries.

The main problem and common deficiency in these schemes are that user anonymity is not provided because the user's identity is sent without encryption on an open channel. The participants'

credentials are insecure, causing a lack of anonymity and un-tractability [11]. The scheme is also vulnerable to identification-guessing attacks and tracking attacks. Specifically, a participant's identification might get exposed by an intruder when it gets disconnected from the internet. Furthermore, an adversary can tune into a selected user with the information the user provided within the login request message.

This paper presents a Secure Participant Authentication and Prescription Safety (SAPS) scheme for TMIS to accomplish participant's anonymity and un-traceability. The process begins when the patient registers with the trusted server that validates the communicating parties and then establishes the session key. We have used ECC in this study; results showed that an anonymous ECC-based presented technique has secure and well-organized authentication protocols with foolproof security along with user confidentiality protection which is practical for TMIS. Formal protocol secure analysis using Rubin Logic was used to evaluate their safety performance and reliability. Using the proposed protocol patient can get a medical prescription from physician securely and interacts with health service provider anonymously using TMIS keeping identities secret.

The rest of paper is organized as follows; Section 2 includes the previous work which has some related information regarding the present study and an overview of existing practices. The system model and problem statement are presented in Section 3. Section 4 explores the proposing anonymous SAPS scheme in details with formal steps. Formal analysis using Rubin Logic is presented along with security analysis in Section 5. The performance measures and results have been included in Section 6. Section 7 presents the conclusions of the work and a future roadmap.

## 2. Literature Review

This section explores some well-known existing schemes related to the security of TMIS and health care services. In this regard, different 3-PAKE and ECC-based schemes are reviewed to analyze the effectiveness, usefulness, and security strengths for providing reliable security solutions. Xie et al. have proposed an ECC-based efficient 3-PAKE scheme [12] that overcomes the flaws mentioned above but suffers from offline password guessing attacks. Che et al. have proposed modular exponentiation on an ECC-based 3-PAKE scheme [13] to make it more complicated for the attacker, but these operations require huge computational cost as compared to existing counterparts. Wu et al. have presented the concept of the secret password and smart card-based verification protocol [14] for TMIS by utilizing the discrete hard logarithm problem (DHLP). It pre-registers to stay away from the exponential mathematical computations during authentication stage. In three-party key exchanges (TPKE) [15], He et al. has found some flaws in [14] and presented an improved scheme that handles impersonation and insider attacks which made the existing scheme vulnerable. Later on, in scheme [16], Wei et al. explored both [14,15] schemes and identified the vulnerability of these schemes for offline password guessing attacks, dictionary attacks, and inability to maintain user un-traceability. To overcome these drawbacks, Wei et al. have improved protocols and presented a user authentication scheme for TMIS that can uphold different attacks. Moreover, Zhu et al. [17] also identified the offline password guessing attacks in these schemes. To overcome issues in [17], Pu et al. [18] has devised a new user authentication scheme applying smart card which is a user password-based identification that provides user anonymity. This scheme is based on an elliptic curve cryptosystem that provides the same security level as RSA with less key size.

Chen et al. [11] highlighted the client anonymity of Khan et al.'s technique [19] but it might be defenseless against insider attacks because all the lawful clients share the secret key. It presented an efficient and secret identity-based validation and secure protocol for TMIS that generates random identities for every exchange in a session to sustain a strategic distance between individual data about the client and the danger of identity theft attack and ensured that their protocol accomplished client privacy. The scheme is a successor and much preferable to the previous responses for use on mobile devices. The scheme does not provide anonymity of the user and is vulnerable for identity guessing and tracking attacks.

In [20] Kumar et al. has presented a smart device authentication scheme in a WSN environment and found that these schemes are suitable for TMIS and fulfill all prerequisites for medical device networks. Later on, He et al. [21] found that some special insider attacks in medical device networks such as offline password guessing attacks occurred and the system was unable to handle anonymity in [20]. A user anonymous authentication scheme has been presented to handle remote medical services applying in WSN to resolve this issues. Nam et al. [22] pointed out some flaws in [21] for user anonymity and smart card theft. By exploiting symmetric encryption and secure key management for message integrity, Xue et al. [23] presented a temporary credential-based secure key using a one-way hash function and XOR operations. It enhances security fundamentals without considerably expanding the memory requirements. In [24], Li et al. pointed out that scheme [23] is unable to protect against stolen-verifier attacks, denial of service attacks, smart card theft, and participant's signing attacks. Turkanovic et al. presented a secure hash function-based user prediction and secure key management protocol [25] that ensures security with low energy consumption. Amin et al. [26] identified that the scheme [25] is vulnerable to offline dictionary attacks, password guessing attacks, smart card theft, and has an inefficient authentication process.

IoT can be an appropriate approach to support health care systems by the technological advancements that enable the outlining of new advanced strategies for the treatment of many diseases, e.g., by the surveillance of chronic diseases to assist doctors to work out the best treatments, as projected by [27]. Due to the ubiquitous computational nature of IoT, all the TMIS entities may be monitored and managed continually. Mobile healthcare (m-Healthcare) is an associate economic model to give patients with the right of entry to resources, about past and present health records, blood pressure, and heart rate measurements. Additionally, hospitals and healthcare associations offer on-demand services hosted in the cloud, reducing the equipped costs and overcoming the constraints of standard medical treatment. On the contrary, the privacy of private healthcare information continues to be a challenge to be faced [28].

Rahimi et al. [29,30] introduced a more secure and powerful user authentication and key agreement technique for fitness-IoT structures which requires considerable processing power. It exploits the property of a sensible gateway in fog computing for critical and security services-associated organization. IoT technologies for healthcare achieve standardized medical care [31] that is frequently improved by further automating the tasks that can be performed without human interaction. In this experience, IoT-based healthcare permits remote monitoring and management of large amounts of medical data using cloud services. Yeh et al. [32] highlighted the idea of an ECC-based unique participant's verification protocol to get higher performance and security in a smart medical information system. ECC was first presented in [33,34] by utilizing a logarithm problem that can do a much better job with a smaller key length as compared to well-known existing schemes with larger key sizes [35]. By these assumptions, it is stated that ECC authentication schemes are very appropriate for resource-constrained and remotely accessible devices.

### 3. System Model and Problem Identification

TMIS requires strong security to provide a dependable user validation and secret key management system for an IoT-based medical environment as depicted in Figure 2. In TMIS, patients remotely present their healthcare services information to a trusted server using a medicinal gadget. Subsequently, accepting the medical records of patients in the network, the servers diagnose the issue and suggest medicines to the online patients. Such systems are rapidly growing in our society and hence require privacy of patient's data. Nonetheless, the private data transmitted over the internet using the public network are not ensured in many TMIS conditions. The system can be subject to a variety of attacks from external parties. Social insurance solutions linked with TMIS should fulfill critical security and protection necessities to ensure patient's medical records and prescriptions are secure, and that they are constantly provided validation, privacy, trustworthiness, and anonymity. Authentication restricts

the medical data from being accessed by malicious and dangerous attackers. The secret key is used to encrypt the data packet to guarantee the confidentiality of medical information during data exchange.

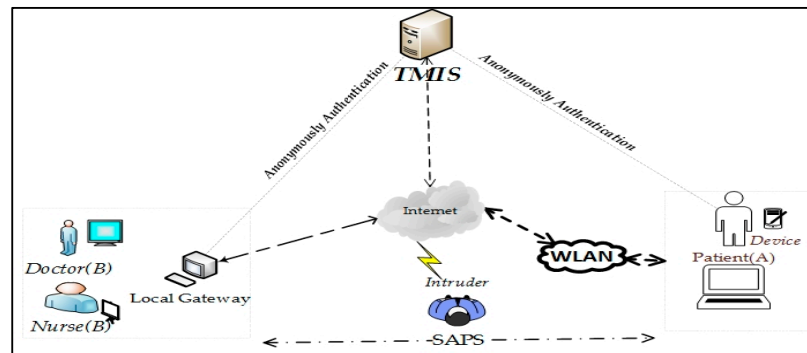


Figure 2. TMIS System Model.

In TMIS, when a patient wishes to become a new legitimate user, denoted as  $U_i$ , then the following steps are accomplished; validity of server-assigned timestamp and IDs. The user authenticated process is done after verifying smart card holding parameters to grant or reject authentication. Anonymity and un-traceability are two fundamental and attracting concerns during basic user privacy, where the former ensures secure user identity in transit, and no one else knows the exact IDs except the communication agents. The stronger property of privacy is un-traceability of the user, such that no one can find the sources of secret data during mutual authentication of the two parties by the trusted third party. Due to un-traceability, the outflow of client personal information in one session would be unusable for the adversary to recognize the user characteristics in another session [36].

### Security Model

In this section, the identification of the possible threat scenarios that can be launched by exploiting user identity is discussed. An adversary can expose the user ID through exhaustive offline guessing because it is usually shorter and has a certain format which can be easily guessed [37]. To secure participants' medicinal prescription, the identity must not be disclosed at intermediate nodes when the information is being exchanged over the uncertain channel. An intruder can attempt to get the private information of the participant from the system, causing a potential risk for confidential data. There is no check of the rightness of a client's old secret key amid the secret password change stage, implying that an individual [38] who has other individual's smart card can change secret code and bio-metric without giving the original password. Other conceivable assaults are participant impersonation attacks, brute-force attacks, or dictionary assaults. Furthermore, an illegal participant may misuse the other participant's personal information to conduct various login sessions and follow participant's exercises by phishing techniques [30]. Also, the breaching of user privacy and their routine activities may likewise encourage an unapproved user to follow the participant's login history and even their current location. After obtaining the password by off-line password guessing, the user can easily guess or deduce common shared secret keys between end users like  $A$  and  $B$  after intercepting the transmitted messages  $R_A$  and  $R_B$ .

An intruder can attempt to retrieve the identity of the legitimate user from their login message. As the user's identity is short, the adversary ( $A$ ) finds  $AId_i$  with polynomial time by executing exhaustive guessing. During multiple valid user sessions using the same unique password, forgery is possible [37]. The well-known threats that can be launched to breach the security of legitimate users are as follows.

- (i) Applying reverse engineering by monitoring user activities, the adversary extracts the privately stored data in the participant's device [38].



- (ii) During private information transmission over an open channel, there are possibilities of eavesdropping, intercepting, content modification, and reply attacking to influence overall communication.
- (iii) A user registers with a true identity but acts as a malicious user in the TMIS.
- (iv) To overcome low entropy issues, the password must be robust enough to defend against password guessing attacks.

Using two parameters, such as a secret password and unique identity, is more feasible and is cannot be guessed by an attacker in polynomial time. Considering this assumption, a malicious user can obtain private data from the location of data storage of the user and re-transmit all messages using the public channel. In case a security protocol is unable to handle these issues, then it cannot protect users from password guessing attacks (which is possible offline), replay attacks (by altering the original message), denial of services, man-in-the-middle attacks, and cannot provide seamless forward privacy.

By the above analysis, Chen's scheme [11] cannot survive tracking attacks and it has failed to uphold user un-traceability. It is defenseless against guessing attacks and tracking attacks. A smart card holds a secret number ( $r$ ) and this can be determined from the fix value of  $W = h(r||pw_i)$ . An adversary (A) can eavesdrop on a legal user's login request message  $R_m = h(Id_i, W)$  and get the value of  $W$  to subvert the privacy of the participant unless a new authentication session is established [37]. Moreover, tracking attacks are possible by monitoring different session of same user using  $W$ .

#### 4. Secure Authentication and Prescription Safety Protocol

A novel protocol for TMIS is presented to protect patient's privacy and satisfy the security requirements. Mutual authentication enables the communicating parties to verify each other's identities. The proposed Secure Authentication and Prescription Safety (SAPS) protocol demonstrates that it is dynamic and overcomes the above-highlighted flaws in the existing schemes. In SAPS, user authentication and verification are performed by the trusted server (TS). The patient and doctor/nurse are the sender and receiver nodes, respectively, in the proposed scenario to establish the secret keys using SAPS. In this section, all the protocol steps are elaborated on, along with a description of the message contents of a SAPS scenario. It includes a trusted server (TS), patient (A), and a doctor/nurse (B). ASAP provides a secure link for the Patient (A) to communicate securely with a physician to obtain a medical opinion. We highlight that ECC-based mutual authentication is secure against numerous significant attacks and improves the communication and memory requirements of authentication. Considering the positive characteristics of ECC, such as the shorter secret key size and computational efficiency, it is attractive to establish an ECC-based anonymous 3-PAKE protocol to protect TMIS users. By exploring existing literature and studies, there is no such system based on ECC that gives anonymous 3-PAKE protocol to authenticate users without knowing their private information publicly. It achieves its security benefits due to the hardness property of the EC Discrete Logarithm Problem (ECDLP). A list of useful notation for SAPS is listed in Table 1.

In the SAPS protocol, we have assumed that during the registration phase, the TS provides masked identities for both the patient and doctors at the point of registration. Participants provide some secret credentials, such as biometrics, to verify their original identities and record the time of service request. The TS uses the parties' secret credentials for future verification, keeping their original identities and private information secret. Besides this, the authentication protocol satisfies the following functions and security requirements to achieve credible and secure authentication and data sharing in TMIS.

**Table 1.** Abbreviations and acronyms for Secure Authentication and Prescription Safety (SAPS).

Notations	Descriptions
<i>TMIS</i>	Telecare medical information system
<i>E</i>	A large-order finite field on elliptic curve
<i>P</i>	EC generator of a large order <i>n</i>
<i>H(.)</i>	Digestive Hash Function.
<i>A</i>	Patient that is participant A in TMIS
<i>B</i>	Doctor/nurse that is user B in TMIS
<i>MAC</i>	Message Authentication Code
$E_{K_{A-TS}}$	Pre-Shared key between TS and User A
$E_{K_{B-TS}}$	Pre-Shared key between TS and User B
<i>TS</i>	Trusted Server as a trusted third party in TMIS
$Pw_P$	TS shared password for Patient
$Pw_{D,N}$	Doctor/Nurse password shared with TS
$ID_A, ID_B, ID_{TS}$	Masked Identities of A, B and TS respectively
$E_{K_{TS-A/B}}$	Temporary Encryption key between TS & Ends
<i>d</i>	Private/Public key of TS
$E_k(.), D_k(.)$	Using key (k) perform Encryption/Decryption
$T_1, T_2, T_3$	User (A, B, TS) Time Stamp
$N_1, N_2, N_3$	User (A, B, TS) Nonce No
$M_A, M_B$	Message at User A (Patient) & B (Doctor/Nurse)
$C_A, C_B$	Cipher Text at A and B

1. Person anonymity: In an authentication mechanism, despite the fact that an attacker extracts some and can eavesdrop on the shared message within the communication network, the legal participant's identity is kept anonymous from the intruder.
2. Identity proof: the process in which both the user and authentication server prove their identities before accessing each other. Numerous steps are performed to achieve mutual authentication to check the integrity of all transmitted messages.
3. Session key management: When the verification method is consummated, the consumer and server must present the consultation key to each other.
4. Password verification manner: If a person has entered an incorrect password within the authentication section, the password has to be detected earlier than the check phase.
5. Person cordiality: An authentication mechanism system provides a password change method through which an individual may facilely change their password without communicating with the server.
6. Robustness: An authenticated key acquisition mechanism has been engendered, and has to be resistant to extraordinary types of assaults, insider attacks, off-line password conjecturing attacks, replay assaults, and consumer impersonation assaults. Besides this, in the proposed protocol, the TS partially establishes a session key between each party. Using the secret credentials generated by the TS, the ends parties establish a session key for the particular session which helps to protect participant identities and ensure un-traceability.

During authentication, a new patient (A) and doctor (B) submit their original identity to the TS using a secure channel. After receiving their network joining request, the TS generates shadow-IDs for each participant and stores them in its database. The shadow-IDs of each participant is to establish anonymous joining of the patient and doctor and to keep prescriptions and private data secret. SAPS is explored in a stepwise manner as follows.

#### A. Step-I: Initialization by Patient (A)

At the beginning, Patient (A) chooses a random number  $R_p$  from a finite field and computes secret parameters. After that,  $X_A$  is calculated by multiplying random number  $R_p$  by an ECC-based generator  $P$  of large order  $n$ . Similarly,  $Y_A$  is the resultant of  $R_p$  and the TS's public key  $F$  that is

equal to  $dP$ , where  $d$  is random number a from finite field selected by TS. For level 1 encryption of security credentials, a hash of  $Y_A$  is taken to prepare key  $H_{Y_A}$ . Patient (A) prepares a message  $M_A$  that contains hash of IDs and  $PW_p$  as the patient's password and Message Authentication Code (MAC) is used for providing message integrity on the server side. Patient (A) calculates  $H(PW_p || ID_A || ID_B)$  and includes  $PW_p$  to keep it more secure. In our proposed scheme, an intruder is not able to get the IDs of A and B but if in any case these values are exposed, then the exact hash value cannot be calculated because of the missing  $PW_p$  that is held by the Patient (A) and TS only. For transmission to the server, the patient computes cipher text  $P_A$  which is encrypted by the patient's generated secret key  $H_{Y_A}$  as shown in step (iv). After that, a cipher text  $C_A$  is generated using a pre-established key  $K_{A-TS}$ . In step (vi), a temporary ID as  $ID_{A-T}$  of the patient is obtained by taking the hash of the  $H(X_A, P_A, N_1)$  and is used for the current session only. The new  $ID_{A-T}$  is never transmitted and can be calculated at the TS using  $H(X_A, P_A, N_1)$  where  $N_1$  can be extracted after decryption. It encrypts the parameters  $\{X_A, P_A, T_1\}$  using  $K_{A-TS}$  where,  $T_1$  is timestamp. Patient (A) transmits  $\{ID_{A-T}, C_A\}$  to trusted server {TS} for authentication.

- (i)  $X_A = R_p P$
- (ii)  $Y_A = R_p F$
- (iii)  $M_A = H(PW_p || ID_A || ID_B)$
- (iv)  $P_A = E_{H_{Y_A}}(ID_A || M_A || N_1 || MAC(M_A) || ID_B)$
- (v)  $C_A = E_{K_{A-TS}}(X_A || P_A || T_1)$
- (vi)  $ID_{A-T} = \{H(X_A, P_A, N_1)\}$

#### B. Step-II: Verification at Trusted Server

Upon receiving  $\{ID_{A-T}, C_A\}$  from Patient (A), the TS decrypts the cipher text  $C_A$  to get  $(X_A || P_A || T_1)$ . It also checks the message freshness by taking the difference from  $T_1$  to guard against replay attacks. After that, the TS computes the temporary key of the patient by multiplying the received  $X_A$  with  $d$  which was pre-generated by the TS as  $Y'_A = dX_A$ . To verify whether the message is original, the TS computes the Patient's (A) masked identity as  $R_p F = R_p dP = dX_A$ . It also decrypts  $P_A$  to obtain security credentials, including  $ID_A$ ,  $M_A$ ,  $N_1$ ,  $MAC(M_A)$ , and  $ID_B$ . The hash of these values is calculated as  $M'_A = H(PW_p || ID_A || ID_B)$  and is then compared to verify the equality of  $M_A$  and  $M'_A$  to ensure message integrity. Otherwise, the message is discarded. The  $(MAC(M_A))$  provides data integrity for  $M_A$ . The trusted server computes the following steps.

- (i) Decrypt  $C_A$  using  $K_{A-TS}$  to get  $\{(X_A || P_A || T_1)\}$
- (ii) Computer  $Y'_A = dX_A$
- (iii) Decrypts  $P_A$  using  $K_{H(Y_A)}$  to get  $\{ID_A, M_A, N_1, MAC(M_A), ID_B\}$
- (iv) Compute:  $M'_A = H(PW_p || ID_A || ID_B)$
- (v) If Verify  $(MAC'(M_A) \neq MAC(M_A))$  then discard
- (vi) If  $M_A$  NOT equals  $M'_A$  then discard message

#### C. Step-III: TS-based Mutual Authentication of B&A

After verification, the TS picks a random number  $R_{TS}$  and then computes  $Z_{TS} = H(ID_{TS} || ID_B || R_{TS})$  using identities of B and TS. It also generates a nonce number  $N_2$  to get its hash with identities of communicating parties A and B. After that, TS calculates the XOR of hash value with  $Z_{TS}$  to get a new temporary ID for B. The value of  $C_{TS}$  is obtained by encrypting  $(ID_A || Z_{TS} || T_2 || N_2)$  using the pre-established key  $K_{TS-B}$ . The TS transmits the temporary identity  $ID_{B-T}$  and cipher text  $C_{TS}$  to B.

- (i)  $Z_{TS} = H(ID_{TS} || ID_B || R_{TS})$
- (ii)  $ID_{B-T} = Z_{TS} \text{ XOR } \{H(ID_B || ID_A || N_2)\}$



$$(iii) \ C_{TS} = E_{K_{TS-B}}(ID_A || Z_{TS} || T_2 || N_2 || ID_{B \sim T})$$

$$TS \rightarrow B : \{ID_{TS}, C_{TS}\}$$

Doctor/Nurse (B) receives the message  $\{ID_{TS}, C_{TS}\}$  and decrypts it to get the other party's prescription details and TS validity by computing the set time stamp threshold value, nonce number, received masked-ID values, and decrypted message using the pre-share key from the TS. At each end, entity  $E_{K_{TS}}$  is used as a key to encrypt secure credentials in addition to MAC and the hash function application to make them more secure.

- (i) Decrypt using  $K_{TS-B}$  to get  $\{(ID_A || Z_{TS} || T_2 || N_2)\}$
- (ii) If  $\{Z_{TS} \text{ XOR } \{H(ID_B || ID_A || N_2)\}\}$  NOT Equals  $ID_{B \sim T}$  then discard
- (iii)  $X_B = R_B P, Y_B = R_B F$
- (iv)  $M_B = H(PW_B || ID_{TS} || ID_B)$
- (v)  $P_B = E_{H_{Y_B}}(ID_B || M_B || N_3 || \text{MAC}(M_B) || ID_{TS})$
- (vi)  $C_B = E_{K_{B-TS}}(X_B || P_B || T_3)$

$$B \rightarrow TS : \{ID_{B \sim T}, C_B\}$$

TS receives the message  $\{ID_{B \sim T}, C_B\}$  and decrypts it to get  $(X_B || P_B || T_3)$ . After that, the TS computes  $Y'_B = dX_B$  which is equal to  $dR_B P = R_B dP = R_B F = Y_B$  calculated at Doctor/Nurse (B). It further decrypts the  $P_B$  to get  $ID_B, M_B, N_3, \text{MAC}(M_B)$  and  $ID_{TS}$ , as illustrated in steps below. After that, the TS verifies the message's integrity by computing and comparing the hash of the message. Finally, it computes the common parameters  $CP_A$  and  $CP_B$  for both parties and forwards them to the Patient (A) and doctor/nurse (B) for session key computation.

- (i) Decrypt  $C_B$  to get  $[(X_B || P_B || T_3)]$
- (ii) Computes  $Y'_B = dX_B$
- (iii) Decrypt  $P_B$  to get  $[(ID_B || M_B || N_3 || \text{MAC}(M_B) || ID_{TS})]$
- (iv) Calculate  $M'_B = H(PW_B || ID_{TS} || ID_B)$
- (v) If  $M'_B$  NOT equals  $M_B$  then drop message
- (vi)  $CP_A = \{E_{H_{Y'_A}}(X_B || ID_A || ID_B || Y'_A || N_1)\}$
- (vii)  $CP_B = \{E_{H_{Y'_B}}(X_A || ID_A || ID_B || Y'_B || N_2)\}$

$$TS \rightarrow A : \{ID_{A \sim T}, CP_A\}$$

$$TS \rightarrow B : \{ID_{B \sim T}, CP_B\}$$

#### D. Step-IV: Participant Validation and Common Session Key Generation

Patient (A) decrypts  $CP_A$ , verified by its own nonce and MAC which provide integrity and validity of the TS and the message. The common parameters generated by the trusted server are transmitted securely on each end. Upon receiving the secret credentials, the participating parties first verify message integrity and authority by verifying  $Y'_A$  and  $Y'_B$ , respectively. After that, MAC, nonce, TS-ID, and the time stamp are also used for double-checking the source's integrity before processing secret credentials. After successful validation of both parties' identities and that of the TS, participants start to compute the common key. The proposed protocol along with stepwise execution is elaborated in Figure 3.

- a. Party (A) gets:  $\{X_B, ID_A, ID_B, Y'_A\}$
- b. Patient (A):  $S_{kA} = H(dX_B, ID_B, ID_A)$
- a. Party (B) gets:  $\{X_A, ID_A, ID_B, Y'_B\}$
- b. Doctor/Nurse (B):  $S_{kB} = H(dX_A, ID_B, ID_A)$

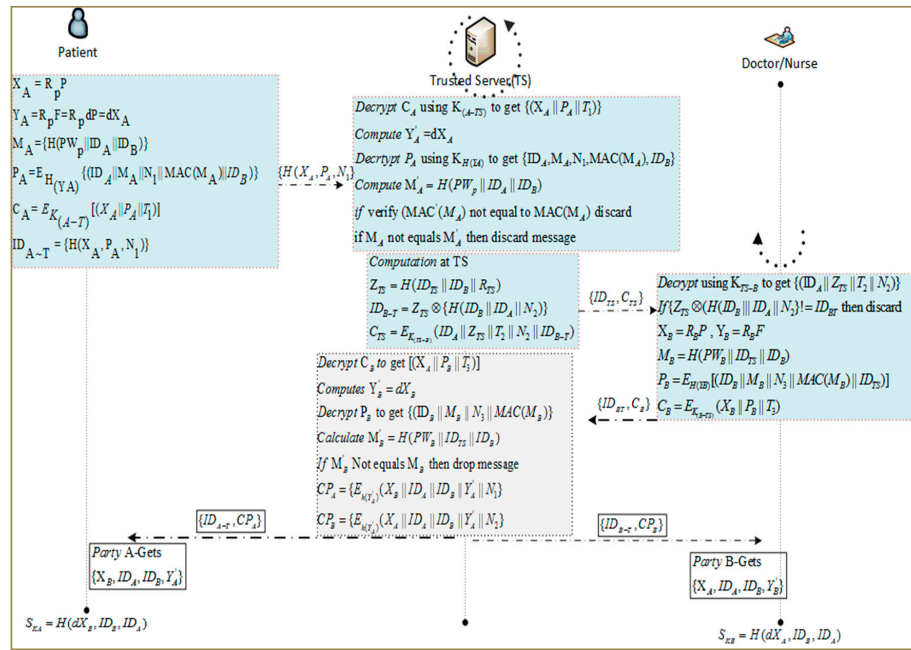


Figure 3. Secure authentication and prescription safety protocol steps.

The novelty of our study is relying upon the creation of secret credentials of the session key for multiparty computing using ECC and symmetric parameters which have less computation cost and are hard to compromise. Upon successful authentication of end parties and common session key generation, both ends share private data securely and efficiently. SAPS attains shared verification, better forward privacy, un-traceability, and participant anonymity. It can launch a secure data sharing connection between the end user and a trusted authority. It also ensures that various attacks, such as offline password guessing, untraceable online secret parameters guessing, confidential insider attacks, card theft attacks, and replay attacks. Intruders cannot enter the system, and the user remains protected at all times.

## 5. Formal Modeling and Analysis of SAPS

We have performed formal modeling using a method known as Nonmonotonic Cryptographic Protocol (NCP), also called Rubin Logic [39]. The analysis verification of the SAPS protocol is as per its previous terms and specification of NCP. NCP authenticates the proposed scheme as per the regular necessities of cryptographic procedures, considering parameters such as authenticity, data integrity, the freshness of received data, message encryption, and decryption, etc. This will also help identify the lack of certain properties in the presented scheme and for potential data compromising scenarios. This analysis is similar to the actual operation functionalities in a programming scenario. When we talk about Rubin Logic, the units are assigned specific roles, and a universal set of information is maintained. It also maintains current state of the parameters of users after each update operation. Global sets are accessible to all the member nodes and can be categorized into four types which are secret, observer, rule, and first sets. A detailed discussion of the formal specification for WSN protocols is provided in [26,29], along with appropriate case studies. Entity or node contains a local set that can be categorized into ownership, represented as POSS (), a set of beliefs known as belief BEL (), and to represent the behavior, set BL (). Detailed specification of these sets can be explored in [14,30]. Table 2 represents the local set for the application scenario of Rubin logic on SAPS-AN. The details of SAPS, its verification, and analysis provide a detailed overview of all sets maintained under the category of the local set. All the participating entities, including sender, receiver, and Trusted Server are separately maintained locally. An ownership set, POSS (entity), contains all the

parameters involved in encryption, decryption, and other processes accomplished in a local memory of each entity, as described in the section below. For the operations and input arguments that are performed in implementation steps, a Behavior List BL () is maintained. Local sets for the entities of the Trusted Server (TS), patient (A) as first party, and Doctor (B) as the second party are presented in Tables 2–4, respectively.

**Table 2.** Local set at (Patient) (A).

---

$POSS(A) = \{ID_A, P, R_p, K_{A-TS}, F, PW_p\}$   
 $BEL(A) = \{\#(ID_A), \#(P), \#(R_p), \#(K_{A-TS}), \#(F), \#(PW_p)\}$   
 $BL(A) =$   
 $Mul(R_p, P) \rightarrow X_A, Mul(R_p, F) \rightarrow Y_A$   
 $Hash(h(.); Concat(PW_p, ID_A, ID_B)) \rightarrow M_A$   
 $Concat(ID_A, M_A, N_1, MAC(M_A), ID_B) \rightarrow Q_A$   
 $Hash(h(.); Y_A) \rightarrow H_{Y_A}$   
 $Encrypt([Q_A, H_{Y_A}]) \rightarrow P_A$   
 $Encrypt([Concat(X_A, P_A, T_1), K_{A-TS}]) \rightarrow C_A$   
 $Hash(h(.); Concat(X_A, P_A, N_1)) \rightarrow ID_{A-T}$   
 $Send([ID_{A-T}, C_A])$  to TS and Update  $(ID_{A-T})$   
 $Receive(TS, (CP_A))$   
 $Decrypt([ID_{TS}, CP_A], H_{Y_A})$  and Split to get  $[X_B, ID_A, ID_B, Y'_A, N_1]$   
 $Hash(h(.); Concat(dX_B, ID_B, ID_A)) \rightarrow S_{kA}$

---

**Table 3.** Local Set at Trusted Server (TS).

---

$POSS(TS) = \{ID_{TS}, d, K_{(TS-A)}, K_{(TS-B)}\}$   
 $BEL(TS) = \{\#(ID_{TS}), \#(K_{(TS-A)}), \#(K_{(TS-B)})\}$   
 $BL(TS) =$   
 $Receive(A, (C_A))$   
 $Decrypt([ID_{A-T}, C_A], K_{TS-A})$  and Split to get  $[X_A, P_A, T_1]$   
 $check(T'_1 - T_1) \geq \Delta T$  then abort  
 $Hash(h(.); Mul(d, X_A)) \rightarrow H_{Y'_A}$   
 $Decrypt([P_A], H_{Y'_A})$  and Split to get  $[ID_A, M_A, N_1, MAC(M_A), ID_B]$   
 $Hash(h(.); Concat(PW_p, ID_A, ID_B)) \rightarrow M'_A$   
 $if MAC(M'_A) \text{ equals } MAC(M_A) \text{ else discard}$   
 $if M'_A \text{ NOT equals } M_A \text{ then discard}$

---

$Hash(h(.); Concat(ID_{TS}, ID_B, R_{TS})) \rightarrow Z_{TS}$   
 $Hash(h(.); Concat(ID_A, ID_B, N_2)) \rightarrow Q_{TS}$   
 $XOR(Z_{TS}, Q_{TS}) \rightarrow ID_{B-T}$   
 $Encrypt([Concat(ID_A, Z_{TS}, T_2, N_2, ID_{B-T}), K_{(TS-B)}]) \rightarrow C_{TS}$   
 $Send([ID_{TS}, C_{TS}])$  to B and Update  $(ID_{B-T})$

---

$Receive(B, (C_B))$   
 $Decrypt([ID_{B-T}, C_B], K_{TS-B})$  and Split to get  $[X_B, P_B, T_3]$   
 $check(T'_3 - T_3) \geq \Delta T$  then abort  
 $Hash(h(.); Mul(d, X_B)) \rightarrow H_{Y'_B}$   
 $Decrypt([P_B], H_{Y'_B})$  and Split to get  $[ID_B, M_B, N_3, MAC(M_B), ID_{TS}]$   
 $Hash(h(.); Concat(PW_B, ID_{TS}, ID_B)) \rightarrow M'_B$   
 $if MAC(M'_B) \text{ equals } MAC(M_B) \text{ else discard}$   
 $if M'_B \text{ NOT equals } M_B \text{ then discard}$

---

$Encrypt([Concat(X_B, ID_A, ID_B, Y'_A, N_1), H_{Y'_A}]) \rightarrow CP_A$   
 $Send([ID_{A-T}, CP_A])$  to A  
 $Encrypt([Concat(X_A, ID_A, ID_B, Y'_B, N_2), H_{Y'_B}]) \rightarrow CP_B$   
 $Send([ID_{B-T}, CP_B])$  to B

---

**Table 4.** Local set at Patient/Doctor (B).

$POSS(B) = \{ID_B, P, R_B, K_{B-TS}, F, PW_B\}$ $BEL(A) = \{\#(ID_B), \#(P), \#(R_B), \#(K_{(B-TS)}), \#(F), \#(PW_B)\}$ $BL(A) =$ Receive(TS, $\{C_{TS}\}$ ) Decrypt( $\{ID_{B-T}, C_{TS}\}K_{B-TS}$ ) and Split to get $[ID_A, Z_{TS}, T_2, N_2]$ Hash( $h(\cdot); Concat(ID_A, ID_B, N_2)$ ) $\rightarrow Q'_{TS}$ check( $XOR(Z_{TS}, Q'_{TS})$ ! equals $ID_{B-T}$ ) then abort Mul( $R_B, P$ ) $\rightarrow X_B$ , Mul( $R_B, F$ ) $\rightarrow Y_B$ Hash( $h(\cdot); Concat(PW_B, ID_{TS}, ID_B)$ ) $\rightarrow M_B$ Concat( $ID_B, M_B, N_3, MAC(M_B), ID_{TS}$ ) $\rightarrow Q_B$ Hash( $h(\cdot); Y_B$ ) $\rightarrow H_{YB}$ Encrypt( $\{Q_B, H_{YB}\}$ ) $\rightarrow P_B$ Encrypt( $\{Concat(X_B, P_B, T_3), K_{(B-TS)}\}$ ) $\rightarrow C_B$ Send( $\{ID_{B-T}, C_B\}$ ) to TS and Update( $ID_{B-T}$ )
Receive(TS, $\{CP_B\}$ ) Decrypt( $\{ID_{TS}, CP_B\} H_{YB}$ ) and Split to get $[X_A, ID_A, ID_B, Y'_B, N_2]$ Hash( $h(\cdot); Concat(dX_A, ID_B, ID_A)$ ) $\rightarrow S_{kB}$

### 5.1. SAPS Analysis and Verification

In this section, SAPS is analyzed for secure session key establishment for the patient to access their physicians anonymously for medical prescription, as discussed in section IV. In this scenario, the establishment request is initiated by the patient (A) by transmitting secret credentials generated using secure methodology established through the TS. After the sending operation, an update operation performed by the sender to refresh the security credentials for future sessions is as shown below.

- Concat( $ID_A, MA, N_1, MAC(MA), ID_B$ )  $\rightarrow QA$
- Encrypt( $\{QA, H_{YA}\}$ )  $\rightarrow P_A$
- Encrypt( $\{Concat(X_A, P_A, T_1), K_{(A-TS)}\}$ )  $\rightarrow C_A$
- Send( $\{ID_{A-T}, C_A\}$ ) to TS and Update( $ID_{A-T}$ )
- Update()

To observe, a list of associated factors, messages, nonce numbers, actual key, ciphers, pseudo-dynamic, finite field, and one-way hash values, are kept in a possession set at the sender node, i.e., with the patient (A). In this case, during the authentication for key establishment phase, the following steps are performed.

- $POSS(A) = \{ID_A, P, R_p, K_{A-TS}, F, PW_p\}$
- $BEL(A) = \{\#(ID_A), \#(P), \#(R_p), \#(K_{(A-TS)}), \#(F), \#(PW_p)\}$
- $BL(A) = Mul(R_p, P) \rightarrow X_A, Mul(R_p, F) \rightarrow Y_A$

After sending messages to the other party and the authentication request is sent to the trusted authority, the TS decrypts the received message to get the secret credentials necessary for verification and to explore the request message. The freshness of the message is checked by comparing the timestamp threshold value, nonce number, and masked IDs value with set values. If we fulfill the threshold parameter and verify the answer, then further process calculations are processed. Otherwise, the message is discarded. MAC is calculated for message integrity and compared to hash values to ensure integrity. The participant verification and message integrity steps performed at the TS are as follows.

- Receive(A,  $\{C_A\}$ )
- Decrypt( $\{ID_{A-T}, C_A\}K_{TS-A}$ ) and Split to get  $[X_A, P_A, T_1]$
- check( $T_1 - T_1 \geq \Delta T$ ) then abort
- Hash( $h(\cdot); Mul(d, X_A)$ )  $\rightarrow H_{Y'A}$
- Decrypt( $\{P_A\}H_{Y'A}$ ) and Split to get  $[ID_A, M_A, N_1, MAC(M_A), ID_B]$

- $\text{Hash}(h(.); \text{Concat}(\text{PW}_P, \text{ID}_A, \text{ID}_B)) \rightarrow M'_A$
- if  $\text{MAC}(M'_A)$  equals  $\text{MAC}(M_A)$  else discard
- if  $M'_A$  NOT equals  $M_A$  then discard

After Patient (A) verification, the TS calculates some parameters and generates a message for the other party that can provide medical services to the patient (A) as follows.

- $\text{Hash}(h(.); \text{Concat}(\text{ID}_{TS}, \text{ID}_B, \text{RTS})) \rightarrow Z_{TS}$
- $\text{Hash}(h(.); \text{Concat}(\text{ID}_A, \text{ID}_B, N_2)) \rightarrow Q_{TS}$
- $\text{XOR}(Z_{TS}, Q_{TS}) \rightarrow \text{ID}_{B \sim T}$
- $\text{Encrypt}(\{\text{Concat}(\text{ID}_A, Z_{TS}, T_2, N_2, \text{ID}_{B \sim T}), K(\text{TS}-B)\}) \rightarrow \text{CTS}$
- $\text{Send}(\{\text{ID}_{TS}, \text{CTS}\})$  to B and Update  $(\text{ID}_{B \sim T})$

Upon receiving the secret credentials sent by the TS, Doctor (B) verifies the message integrity and the TS as follows:

- $\text{Decrypt}(\{\text{ID}_{B \sim T}, \text{CTS}\}_{K_{B-TS}})$  and Split to get  $[\text{ID}_A, Z_{TS}, T_2, N_2]$
- $\text{Hash}(h(.); \text{Concat}(\text{ID}_A, \text{ID}_B, N_2)) \rightarrow Q'_{TS}$
- check  $(\text{XOR}(Z_{TS}, Q'_{TS}) \neq \text{ID}_{B \sim T})$  then abort

Following the authentication and verification of Party (B) by the TS, party (B) also computes some secret credentials and provides a reply message for the TS. The secret credentials are used by both parties to generate a secret key, and the transmission is secured using the shared key, as well as by calculating the hash, adding a nonce, and MAC. The steps performed at the physician's side for session key establishing are as follows:

- $\text{Mul}(R_B, P) \rightarrow X_B, \text{Mul}(R_B, F) \rightarrow Y_B$
- $\text{Hash}(h(.); \text{Concat}(P_{WB}, \text{ID}_{TS}, \text{ID}_B)) \rightarrow M_B$
- $\text{Concat}(\text{ID}_B, M_B, N_3, \text{MAC}(M_B), \text{ID}_{TS}) \rightarrow Q_B$
- $\text{Hash}(h(.); Y_B) \rightarrow H_{YB}$
- $\text{Encrypt}(\{Q_B, H_{YB}\}) \rightarrow P_B$
- $\text{Encrypt}(\{\text{Concat}(X_B, P_B, T_3), K(B-TS)\}) \rightarrow C_B$
- $\text{Send}(\{\text{ID}_{B \sim T}, C_B\})$  to TS and Update  $(\text{ID}_{B \sim T})$

For verification and checking of message integrity of party (B), the TS performs the same steps as Party (A). The TS securely generates secret credentials based on the information gathered from both ends, and distributes a common key which is partially completed. Secure parameter distribution steps are shown as below, which are transmitted securely to the participants.

- $\text{Encrypt}(\{\text{Concat}(X_B, \text{ID}_A, \text{ID}_B, Y'_A, N_1), H_{Y'A}\}) \rightarrow \text{CP}_A$
- $\text{Send}(\{\text{ID}_{A \sim T}, \text{CP}_A\})$  to A
- $\text{Encrypt}(\{\text{Concat}(X_A, \text{ID}_A, \text{ID}_B, Y'_B, N_2), H_{Y'B}\}) \rightarrow \text{CP}_B$
- $\text{Send}(\{\text{ID}_{B \sim T}, \text{CP}_B\})$  to B

Although data/credentials transmission is performed using strong encryption techniques by the TS to increase security and un-traceability of participants, the proposed scheme has a novel approach in its partial session key mechanism. After successful authentication and session key establishment, a "Forget Operation" performed at each participating entity will result in removing temporary values like nonce value, time stamp, temporary encryption key  $H^*$ , and MAC calculating parameters. These operations ensure security against forwarding secrecy and user traceability issues.

### 5.2. User Anonymity

The user's identity, ID, cannot be stored in plaintext at the user level, nor at the TS, and it can be transmitted via the login request. In our scheme, user identity is masked in both  $M_A$  and  $P_A$ , after the original identity registration, session request, and end-user joining request is sent to the TS in the encrypted form. The secret parameters are generated by party (A) and party (B) by choosing secret number  $d_A$ ,  $d_B$ , respectively, from  $E$  with large order  $n$  as shown in steps I–III. It is not feasible for any third party to get these secret parameters and it is almost impossible to recover messages  $M_A$ ,  $M_B$  to get identities without knowing the one-way hash function. It is not viable for an adversary to compute the original identities of the participant. The TS has private and public key pairs and the participants use their MAC for message integrity, masked-IDs, nonce number, and time stamp (T) for message freshness. At the trusted sever, a randomized symmetric encryption technique is used to conceal the random number  $\{R_{TS}\}$  generated by the TS instead of using the XOR. Before communicating the session request of party(A) to party(B), the TS first verifies the original identity of the requested party and the message freshness by computing the inverse functions as shown in Protocols II and III. It generates a temporary public ID of each participant instead of using their original identities. The public identity, say IDV, can run the protocol. Therefore, on the basis of the above analysis, it is impossible for the adversary to get the user's identity from the proposed protocol during authentication, forwarding, the joining request, or the session key computing procedure. Therefore, we know that the proposed protocol is able to provide user anonymity.

### 5.3. User Un-Traceability

In our scheme, each session-based communication contains parameters  $\{X_A, P_A, T_1\}$  based on  $X_A$  and  $P_A$  that are dynamic for each authentication/verification session of all participants and are different for each transmission. Moreover, a timestamp is appended to every session. The value of  $(X_A, P_A)$ ,  $Z_{TS}$  and  $X_B, P_B$ , as shown in the proposed protocols to fulfill the requirements for common session key establishment, are executed in a distributed manner. Consequently, the adversary is unable to figure out that two procedures have the same users involved. It ensures that our proposal accomplishes user un-traceability. According to the pattern of the projected protocol, the participant generates a new random number  $r \in Z_q^*$  to compute  $X_A$ ,  $Z_{TS}$  and  $X_B$  in each session. Due to the randomness of the secret parameters, the adversary cannot find and link among messages sent by the end parties or the trusted server, and therefore is unable to follow their actions. Therefore, the proposed protocol is able to provide un-traceability during authentication, session key management, and the following data transmission procedures.

### 5.4. Offline Password Guessing Attacks

In offline password guessing, the adversary tries to capture the entire communication between party (A) and the TS or between the TS and party (B), but is unable to get the password. The next adversary attack may be possible.

- i.  $\{X_A, P_A, T_1\} \rightarrow \{TS\}$  from party(A) to TS,  $\{ID_{TS}, Z_{TS}\}$  are TS parameters,  $(\{X_B, P_B\}, T_3)$  Party(B) to TS and  $\{CP_A, CP_B\}$  are the communications between A, B, and the TS known by the adversary. To commence the offline secret password estimating attack, the adversary tries to regenerate  $PW'_p$ , an inverse password of party (A) and computes  $M'_A$ . Still, if the adversary gets the IDs of both participants, it is not possible for the adversary to compute  $E_{H(Y_A)}$  and they are unable to verify the  $P_A$  as used in the protocols which contains the message and the identities of the participants.
- ii. Appropriate to the un-traceability of the Computational Diffie-Hellman assumption (CDH) a difficult adversary does not get  $\{Y_A = R_p F := R_p dP \rightarrow Y_A = dX_A F\}$  and it is impossible for the adversary to get the password.



- iii. If an adversary gets party B's password to compute the random  $R_{Ts}$  number generated by the TS, as shown in protocol-III, to compute the parameters after the received connection establishing request from party (A), it is not possible for the adversary to compute B's password without knowing  $Y_B$  because  $Y_B = \{d_B F\}$  and has ECDLP, as shown in step-3 of protocol-III.

As a result, the proposed protocol can oppose offline password guessing attacks. To deal with online password guessing attack, the TS detects the encounter during the message validation process and the freshness procedure at the beginning of the communication.

### 5.5. Perfect Forward Secrecy

In our scheme, the un-traceability of CDH plays a vital role in resisting guessing previous session keys by the adversary. The session key is generated using  $H(d_A, d_B P, ID_B, ID_A)$ , where  $d_A, d_B$  are nonce numbers chosen by both participants specially for the specific session and are different from the nonce numbers used during the authentication process, such as  $N_1, N_2, N_3$ . In the proposed protocol, the end parties compute the session key, SK, as  $H(dX_B, ID_B, ID_A)$  and  $H(dX_A, ID_B, ID_A)$ , where  $dX_A$  and  $dX_B$  are computed by randomly chosen numbers by both parties at their end with the help of the trusted server. Knowing the secret key of the server and the password does not help the adversary to compute a previously established session key, because the secret credentials used to compute the secure key are not based on the password and server's public key. If the adversary wants to know an old session key he /she must compute  $dX_A, dX_B$ . However, since the adversary does not know the  $R_p$  due to the hardness of ECDLP, the adversary cannot compute the secret credentials. Therefore, the proposed scheme provides perfect forward secrecy.

### 5.6. Replay Attack

Replay attacks can be launched while an attacker replays the original message parameters at some other time to impersonate any legal participant. In the proposed scheme, messages between the  $ID_A$ , TS, and  $ID_B$  are transmitted on the public channel. An attacker might try to use these conversations to launch a replay attack. However, in our protocol, replay attacks can be easily thwarted because an adversary cannot produce an updated timestamp. Both end parties and the TS verify message freshness with a threshold for timestamp and nonce number. If the difference exceeds this threshold, it will abort the session. It is not easy for an adversary to impersonate participants' reply messages because the nonce numbers  $N_1, N_2, N_3$  during the authentication process,  $d_A, d_B$  for key session key generation and  $R_{Ts}$  used by the TS are newly chosen for every session. For this scenario, an adversary cannot compute the  $\{X_A, P_A, T_1\}$ ,  $S_K = H(d_A X'_B, ID_B, ID_A)$ , and  $(\{X_B, P_B\}, T_3)$  communication patterns of the entire network.

### 5.7. Forgery Attacks and Impersonation

In our scheme, TS has a pair of public-private secret keys  $(d, F = dP)$ , and if an adversary attempts to impersonate the participants or the TS and sends a message to the TS acting as the participants or tries to establish a connection with any party, they will remain unable to verify themselves. For the verification process, it is necessary for an adversary to know the password or private key of the TS,  $d$ .

### 5.8. Man in the Middle Attack

An adversary may intercept the messages sent between the parties and the TS and replace them with their own messages. These forged messages need to be verified by the TS before getting the session key. However, it is not feasible as the adversary does not know the TS secret key and secret passwords of the parties. So, our scheme resists a man in middle attack.

## 6. Results and Analysis

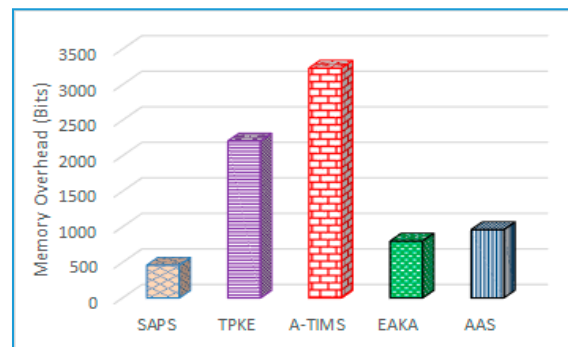
We have simulated the SAPS protocol by deploying nodes for the patient, doctor/nurse, and the trusted server using TCL script in NS 2.35. The simulation parameter and system setting defined in Table 5. We have separately configured the patient and doctor/nurse for low power devices. The server is configured for high residual and transmission power. In the TCL file, communication messages are initiated and traffic sources are also configured along with packet sizes. Moreover, C/C++ files are developed for providing device-level functionalities, including send, receive, hash, encrypt, and decrypt functions. Results are extracted from the trace file by executing the AWK scripts. We have used a 160-bit key along with  $f(x) = x^{167} + x^9 + x^7 + x^5 + x^3 + 1$  where the tuple  $T = (m, f, (x)a, b, G, n, h)$ . The performance of SAPS is evaluated regarding storage, computation, and communication costs for the base scheme TPKE [15], A-TMIS [16], EAKA [40], and AAS [41]. A list of simulation parameters is shown in Table 4.

**Table 5.** Simulation Parameters.

Parameters	Values
Network Field	1300 × 1300 m
Initial Energy at Smart Device	1000 Joules
Tx Power at Smart Device	0.819 Joules
Receiving Power	0.049 Joules
Queue Type	Queue/DropTail/PriQue
Max Packet in Queue	55 Packets
Agent Trace	ON
Router Trace	ON
Number of Nodes	500 Nodes

### 6.1. Storage Overhead

In the proposed protocol, we outlined the storage overhead and memory requirement, including the public/private key of the TS and the communication parties, IDs, random number, timestamp, and resultant length of the session key, as shown in Figure 4. In the view of the used parameters between the patient (A), TS, and Doctor (B), we have analyzed and compared the storage cost of TPKE [15], the smart card keeps  $\{ID, B, p\}$  and thus the storage overhead is  $(160 + 1022 \times 2) = 2208$  bits. A-TMIS [16] stores  $(1600 + 1024 \times 3) = 3232$  bits and an efficient authentication and key agreement EAKA protocol [40] requires  $(160 \times 5) = 800$  bits in the smart card for basics parameters. The AAS [41] scheme improves the protocol for TMIS and consumes 960-bits storage to process authentication and authorization which contains  $\{P, u, B, R, p, q\}$ .



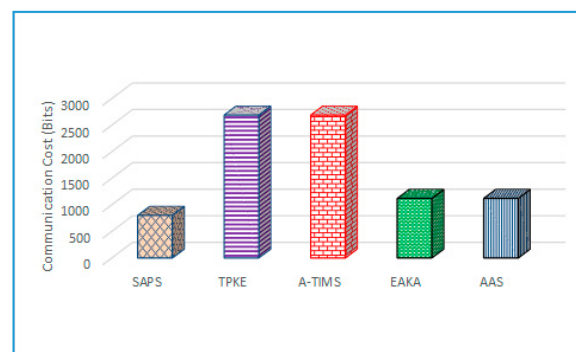
**Figure 4.** Storage cost for Secure Authentication and Prescription Safety (SAPS).

In our SAPS protocol, the end parties need to keep  $\{P_W, ID, X_A, N_i, T_i, E_{K_{A-TS}}\}$  for identification, authentication, and session key establishment. From an evaluation point of view, it has been supposed

that secret identity, one-way hash operation, and timestamps are 160-bits in size, whereas the ECC recommended size by NIST for a key is 160-bits and 64-bit for the secret key for encrypting passwords generated using the TS's shared credentials. So, the total storage overheads of proposed scheme are  $(160 + 160 \times 2) = 480$  bits, which is stored at both ends and the TS has more computation and storage capabilities. In our scenario, more parameters such as ID masking, public/private key and pseudorandom generation processes are accomplished at the TS.

## 6.2. Communication Overhead

To appraise the message exchanging overhead of the presented protocol, the data that is transmitted between the participating parties and the TS during the identification and session key generation phase need to be considered. It is identified that the secret key of size 160-bit using ECC can yield equal security to a 1024-bit RSA secret key. For the evaluation phase, we believe that the resultant one-way hash function, participant's identities, timestamp, and nonce number are 160 bits in length. The end party sends three parameters which are  $\{XA, PA, T1\}$  to the TS for authentication, and their lengths are  $(160 \times 3) = 480$  bits. On the other side, in the party verification phase, the TS sends an encrypted packet  $CP_A$  using a 160-bit ECC key to the end party for session key establishment and its length is  $(160 \times 2) = 320$  bits. On the basis of these computations in the proposed SAPS protocol, Patient (A) and Doctor (B) have the same communication cost during their end user confirmation and secure key generation processes. On the other hand, existing ECC- and RSA-based schemes have greater communication overheads, as depicted in Figure 5.



**Figure 5.** Communication overheads during authentication.

TPKE [15] is based on RSA which requires 1344 bits for the authentication method and 1344 bits for server to end party communication. Similarly, A-TMIS [16] consumes the same communication cost as TPKE for the whole procedure. In the login phase of Xu et al.'s protocol for EAKA [40], the server requires 640 bits of communication cost, and the server to the user requires 480-bits. AAS [41] requires 640 bits and 480 bits communication from the user to server and the server to user, respectively. By the above analysis, the proposed scheme has less communication cost by ensuring user anonymity.

## 6.3. Computation Complexity

Based on the simulation results and defined parameters, a 5 MHz frequency is required to compute the 160-bit elliptic curve, where 5 MHz is for one small data module in which multiple modules are included in the calculation of the elliptic curve and related mathematical operations. To elaborate in detail, we have defined some notations to describe the function of protocol like,  $T_E$ ,  $T_{EPM}$ ,  $T_{EDs}$ ,  $T_H$ ,  $T_X$  which are the time for performing an exponential operation, time for performing an elliptic curve point multiplication operation, time for computing EC point multiplication function, symmetric encryption/decryption operation time, the one way hash function computation time, and XOR operation, respectively.

According to [42,43], the computation time for an exponential operation is 0.522 ms, EC point multiplication process consumes 0.063075 ms, one-way hash operation implementation time is 0.0005 ms, and the encryption/decryption operation time is 0.0087 ms. On the user's side, the proposed scheme has lower computational overheads, as shown in Figures 6 and 7. Our SAPS protocol is more suitable for a mobile scenario when compared to existing schemes, because it has fewer rounds needed for authentication. Computing complexity for the adversary is high, whereas fewer communication rounds make it more efficient and secure.

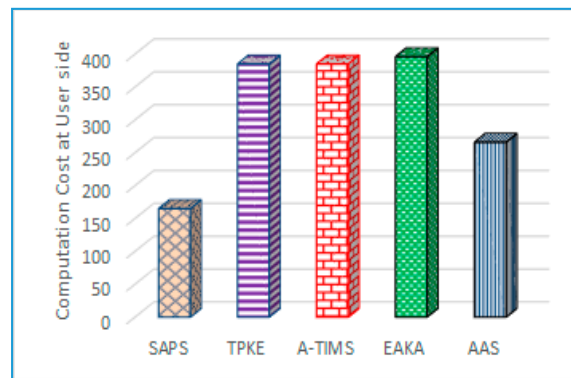


Figure 6. Computation cost at user side.

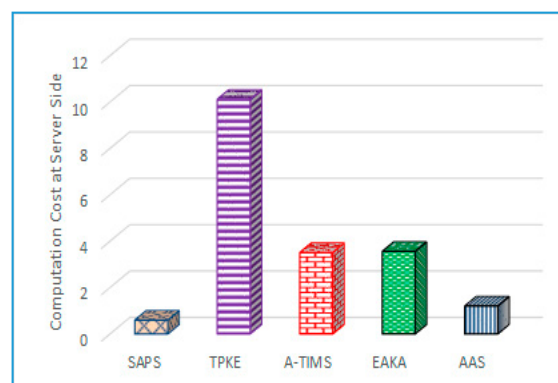


Figure 7. Computation cost at server side.

#### 6.4. Resilience

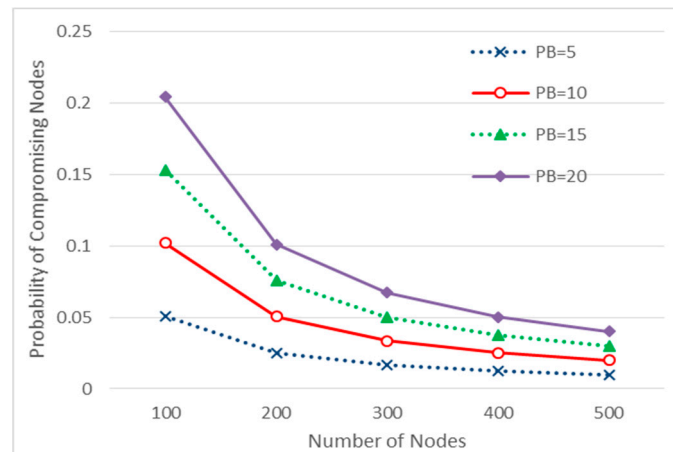
During secure communication between the patient and doctor, devices from different regions communicate using intermediate devices. The chance of compromised devices in the path exists. The probability  $Pr_\beta$  that an intermediate node is compromised is given in Equation (1), where  $N$  is the number of devices in the network and  $\beta$  is number of devices compromised by adversary. The term  $N-2$  represents that sender and receiver are considered uncompromised, whereas  $N-3$  means to exclude one more intermediate device which is a direct neighbor of the sender.

$$Pr_\beta = 1 - \left( \frac{N-3}{\beta} \right) / \left( \frac{N-2}{\beta} \right) = \frac{\beta}{N-2} \quad (1)$$

Figure 8 elucidates the scenario for measuring the impact of compromising intermediate devices by calculating the probability  $Pr_\beta$  when the number of devices in the network varies from 100 to 500. In the case of 300 devices, the probability is 0.0167, 0.033, 0.050, and 0.067 for  $\beta = 5, 10, 15$ , and 20 compromised devices, respectively. If an intermediate device is compromised, then data and security credentials stored in that device are exposed. Moreover, the probability  $Pr_\omega$  that a particular

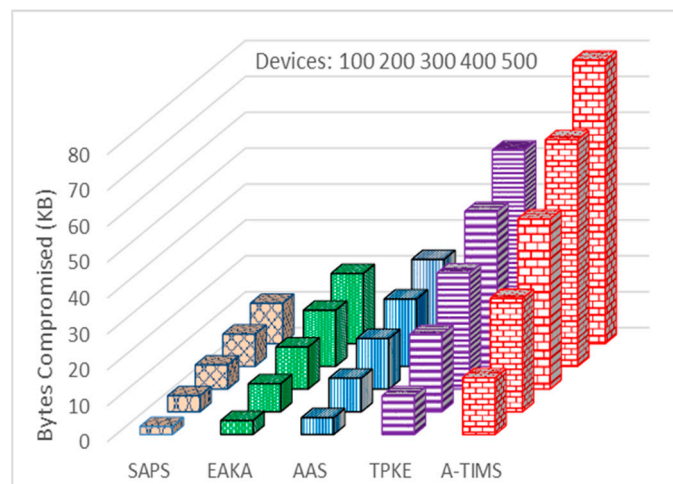
session key and its related credentials are compromised is given in Equation (2), where  $\omega$  credentials are compromised out of a total of  $M$  devices in the network.

$$Pr_{\omega} = 1 - \left( \frac{M-1}{\omega-1} \right) / \left( \frac{M}{\omega} \right) = \frac{\omega}{M} \quad (2)$$



**Figure 8.** Probability of compromised intermediate device.

Figure 9 shows that different security credentials stored in devices can be revealed to intruders when the devices are compromised. We have considered the scenario where the number of devices is varied from 100 to 500, and the number of compromised nodes is varied from 5 to 25, respectively. In the case of 400 devices in the network where 20 of them are compromised, the number of compromised bytes are 15.62 KB, 18.75 KB, 43.12 KB, and 63.12 KB for EAKA, AAS, TPKE, and A-TIMS respectively. Our proposed SAPS method dominates and achieves better resilience against node capture attack by revealing only 8.98 KB.



**Figure 9.** Bytes revealed upon device compromise.

## 7. Conclusions

A secure three-party key establishment technique for TMIS is presented to secure patients' medical prescriptions. It uses ECC for end-user anonymity, using the secure authentication and prescription safety (SAPS) protocol that can establish a secure connection between patients and a

doctor/nurse without revealing their secret identities. The proposed scheme achieves anonymity and un-traceability of the participants. The SAPS protocol has been analyzed by applying Rubin Logic to verify security, user anonymity, and un-traceability of participants during session key generation for secure information sharing between the doctor and the patient in TMIS. Public attacks and their countermeasures are analyzed at each level. For validation of SAPS, we have performed simulation using NS-2.35 to compare the performance of SAPS for storage, communication, and computation overheads, as compared to other methods to prove its suitability for the ubiquitous TMIS network. The storage cost reduction of the proposed scheme at the user side and at the server side is approximately 38% and 41%, respectively, as compared with the mean of the four latest existing techniques. The average computation overhead reductions at both the user side and the server side are 37% and 49%, respectively, as compared with the mean average of existing four schemes. Our future work will aim to analyze the impact of chaotic map-based keying for multi-party authentication between doctors and patients to get a medical prescription for common diseases without revealing either identities on an open network.

**Acknowledgments:** This work was funded by the National Natural Science Foundation of China (61471035), and Fundamental Research Funds for the Central Universities (06105031, 06500010). In particular, it was supported by Cybermatics and Cyberspace International Science and Technology Cooperation Base.

**Author Contributions:** Zahid Mahmood and Ata Ullah conceived and designed the experiments; Zahid Mahmood performed the experiments; Huansheng Ning analyzed the data and overall proposed protocols, models, structure and flow of entire paper; Xuanxia Yao and Ata Ullah performed the formal modeling and verification of proposed scheme. Zahid Mahmood contributed reagents/materials/analysis tools and also wrote the paper along with literature.

**Conflicts of Interest:** The authors declare that there is no conflict of interest in this research.

## References

1. Wu, Z.-Y.; Tseng, Y.-J.; Chung, Y.; Chen, Y.-C.; Lai, F. A reliable user authentication and key agreement scheme for web-based hospital-acquired infection surveillance information system. *J. Med. Syst.* **2012**, *36*, 2547–2555. [[CrossRef](#)] [[PubMed](#)]
2. Giri, D.; Maitra, T.; Amin, R.; Srivastava, P. An efficient and robust rsa-based remote user authentication for telecare medical information systems. *J. Med. Syst.* **2015**, *39*, 145. [[CrossRef](#)] [[PubMed](#)]
3. Xie, Q.; Zhang, J.; Dong, N. Robust anonymous authentication scheme for telecare medical information systems. *J. Med. Syst.* **2013**, *37*, 9911. [[CrossRef](#)] [[PubMed](#)]
4. Abdalla, M.; Fouque, P.-A.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. *IEEE Proc. Inf. Secur.* **2006**, *153*, 27–39. [[CrossRef](#)]
5. Chung, H.-R.; Ku, W.-C. Three weaknesses in a simple three-party key exchange protocol. *Inf. Sci.* **2008**, *178*, 220–229. [[CrossRef](#)]
6. Guo, H.; Li, Z.; Mu, Y.; Zhang, X. Cryptanalysis of simple three-party key exchange protocol. *Comput. Secur.* **2008**, *27*, 16–21. [[CrossRef](#)]
7. Lu, R.; Cao, Z. Simple three-party key exchange protocol. *Comput. Secur.* **2007**, *26*, 94–97. [[CrossRef](#)]
8. Huang, H.F. A simple three-party password-based key exchange protocol. *Int. J. Commun. Syst.* **2009**, *22*, 857–862. [[CrossRef](#)]
9. Nam, J.; Paik, J.; Kang, H.-K.; Kim, U.M.; Won, D. An off-line dictionary attack on a simple three-party key exchange protocol. *IEEE Commun. Lett.* **2009**, *13*, 205–207.
10. Phan, R.C.-W.; Yau, W.-C.; Goi, B.-M. Cryptanalysis of simple three-party key exchange protocol (S-3PAKE). *Inf. Sci.* **2008**, *178*, 2849–2856. [[CrossRef](#)]
11. Chen, H.-M.; Lo, J.-W.; Yeh, C.-K. An efficient and secure dynamic id-based authentication scheme for telecare medical information systems. *J. Med. Syst.* **2012**, *36*, 3907–3915. [[CrossRef](#)] [[PubMed](#)]
12. Xie, Q.; Dong, N.; Tan, X.; Wong, D.S.; Wang, G. Improvement of a three-party password-based key exchange protocol with formal verification. *Inf. Technol. Control* **2013**, *42*, 231–237. [[CrossRef](#)]
13. Wu, S.; Chen, K.; Pu, Q.; Zhu, Y. Cryptanalysis and enhancements of efficient three-party password-based key exchange scheme. *Int. J. Commun. Syst.* **2013**, *26*, 674–686. [[CrossRef](#)]



14. Wu, Z.-Y.; Lee, Y.-C.; Lai, F.; Lee, H.-C.; Chung, Y. A secure authentication scheme for telecare medicine information systems. *J. Med. Syst.* **2012**, *36*, 1529–1535. [[CrossRef](#)] [[PubMed](#)]
15. Debiao, H.; Jianhua, C.; Rui, Z. A more secure authentication scheme for telecare medicine information systems. *J. Med. Syst.* **2012**, *36*, 1989–1995. [[CrossRef](#)] [[PubMed](#)]
16. Wei, J.; Hu, X.; Liu, W. An improved authentication scheme for telecare medicine information systems. *J. Med. Syst.* **2012**, *36*, 3597–3604. [[CrossRef](#)] [[PubMed](#)]
17. Zhu, Z. An efficient authentication scheme for telecare medicine information systems. *J. Med. Syst.* **2012**, *36*, 3833–3838. [[CrossRef](#)] [[PubMed](#)]
18. Pu, Q.; Wang, J.; Zhao, R. Strong authentication scheme for telecare medicine information systems. *J. Med. Syst.* **2012**, *36*, 2609–2619. [[CrossRef](#)] [[PubMed](#)]
19. Khan, M.K.; Kim, S.-K.; Alghathbar, K. Cryptanalysis and security enhancement of a ‘more efficient & secure dynamic ID-based remote user authentication scheme’. *Comput. Commun.* **2011**, *34*, 305–309.
20. Kumar, P.; Lee, S.-G.; Lee, H.-J. E-SAP: Efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks. *Sensors* **2012**, *12*, 1625–1647. [[CrossRef](#)] [[PubMed](#)]
21. He, D.; Kumar, N.; Chen, J.; Lee, C.-C.; Chilamkurti, N.; Yeo, S.-S. Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. *Multimed. Syst.* **2015**, *21*, 49–60. [[CrossRef](#)]
22. Nam, J.; Choo, K.-K.R.; Han, S.; Kim, M.; Paik, J.; Won, D. Efficient and anonymous two-factor user authentication in wireless sensor networks: Achieving user anonymity with lightweight sensor computation. *PLoS ONE* **2015**, *10*, e0116709. [[CrossRef](#)] [[PubMed](#)]
23. Xue, K.; Ma, C.; Hong, P.; Ding, R. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. *J. Netw. Comput. Appl.* **2013**, *36*, 316–323. [[CrossRef](#)]
24. Li, C.-T.; Weng, C.-Y.; Lee, C.-C. An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks. *Sensors* **2013**, *13*, 9589–9603. [[CrossRef](#)] [[PubMed](#)]
25. Turkanović, M.; Brumen, B.; Hölbl, M. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. *Ad Hoc Netw.* **2014**, *20*, 96–112. [[CrossRef](#)]
26. Amin, R.; Islam, S.H.; Biswas, G.; Khan, M.K.; Li, X. Cryptanalysis and enhancement of anonymity preserving remote user mutual authentication and session key agreement scheme for e-health care systems. *J. Med. Syst.* **2015**, *39*, 140. [[CrossRef](#)] [[PubMed](#)]
27. Whitmore, A.; Agarwal, A.; Da Xu, L. The Internet of Things—A survey of topics and trends. *Inf. Syst. Front.* **2015**, *17*, 261–274. [[CrossRef](#)]
28. Dinh, H.T.; Lee, C.; Niyato, D.; Wang, P. A survey of mobile cloud computing: Architecture, applications, and approaches. *Wirel. Commun. Mob. Comput.* **2013**, *13*, 1587–1611. [[CrossRef](#)]
29. Moosavi, S.R.; Gia, T.N.; Nigussie, E.; Rahmani, A.-M.; Virtanen, S.; Tenhunen, H.; Isoaho, J. Session Resumption-Based End-to-End Security for Healthcare Internet-of-Things. In Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), Liverpool, UK, 26–28 October 2015; pp. 581–588.
30. Moosavi, S.R.; Gia, T.N.; Rahmani, A.-M.; Nigussie, E.; Virtanen, S.; Isoaho, J.; Tenhunen, H. SEA: A secure and efficient authentication and authorization architecture for IoT-based healthcare using smart gateways. *Procedia Comput. Sci.* **2015**, *52*, 452–459. [[CrossRef](#)]
31. Rahmani, A.M.; Gia, T.N.; Negash, B.; Anzanpour, A.; Azimi, I.; Jiang, M.; Liljeberg, P. Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach. *Futur. Gener. Comput. Syst.* **2017**, *78*, 641–658. [[CrossRef](#)]
32. Yeh, H.-L.; Chen, T.-H.; Liu, P.-C.; Kim, T.-H.; Wei, H.-W. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* **2011**, *11*, 4767–4779. [[CrossRef](#)] [[PubMed](#)]
33. Miller, V.S. Use of Elliptic Curves in Cryptography. In *Conference on the Theory and Application of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 1985; pp. 417–426.
34. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [[CrossRef](#)]
35. Hankerson, D.; Menezes, A.J.; Vanstone, S. *Guide to Elliptic Curve Cryptography*; Springer Science & Business Media: New York, NY, USA, 2006.

36. Li, X.; Qiu, W.; Zheng, D.; Chen, K.; Li, J. Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards. *IEEE Trans. Ind. Electron.* **2010**, *57*, 793–800.
37. Ying, Z.; Chiou, S.-Y.; Liu, J. An Efficient Privacy Authentication Scheme Based on Cloud Models for Medical Environment. In Proceedings of the 2015 18th International Conference on Network-Based Information Systems (NBIS), Taipei, Taiwan, 2–4 September 2015; pp. 628–633.
38. Zhang, L.; Zhu, S.; Tang, S. Privacy protection for telecare medicine information systems using a chaotic map-based three-factor authenticated key agreement scheme. *IEEE J. Biomed. Health Inf.* **2017**, *21*, 465–475. [[CrossRef](#)] [[PubMed](#)]
39. Rubin, A.D.; Honeyman, P. Nonmonotonic Cryptographic Protocols. In Proceedings of the Computer Security Foundations Workshop VII (CSFW 7), Franconia, NH, USA, 14–16 June 1994; pp. 100–116.
40. Xu, X.; Jin, Z.P.; Zhang, H.; Zhu, P. A Dynamic ID-Based Authentication Scheme Based on ECC for Telecare Medicine Information Systems. In *Applied Mechanics and Materials*; Trans Tech Publ: Zürich, Switzerland, 2014; pp. 861–866.
41. Islam, S.H.; Khan, M.K. Cryptanalysis and improvement of authentication and key agreement protocols for telecare medicine information systems. *J. Med. Syst.* **2014**, *38*, 135. [[CrossRef](#)] [[PubMed](#)]
42. He, D.; Kumar, N.; Khan, M.; Lee, J.-H. Anonymous two-factor authentication for consumer roaming service in global mobility networks. *IEEE Trans. Consum. Electron.* **2013**, *59*, 811–817. [[CrossRef](#)]
43. Jiang, Q.; Ma, J.; Li, G.; Yang, L. An efficient ticket based authentication protocol with unlinkability for wireless access networks. *Wirel. Pers. Commun.* **2014**, *77*, 1489–1506. [[CrossRef](#)]



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).