



Article

Fine-Auth: A Fine-Grained User Authentication and Key Agreement Protocol Based on Physical Unclonable Functions for Wireless Body Area Networks

Kaijun Liu ¹, Qiang Cao ^{1,*} , Guosheng Xu ¹  and Guoai Xu ²

¹ Key Laboratory of Trustworthy Distributed Computing and Service (MoE), Beijing University of Posts and Telecommunications, Beijing 100876, China; guoshengxu@bupt.edu.cn (G.X.)

² School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China

* Correspondence: scq@bupt.edu.cn

Abstract: Wireless body area networks (WBANs) can be used to realize the real-time monitoring and transmission of health data concerning the human body based on wireless communication technology. With the transmission of these sensitive health data, security and privacy protection issues have become increasingly prominent. Fine-grained authentication allows physicians to run authentication checks of another specific entity according to their identifying attributes. Hence, it plays a key role in preserving the security and privacy of WBANs. In recent years, substantial research has been carried out on fine-grained authentication. However, these studies have put considerable effort into WBAN performances, resulting in weakened security. This paper proposes a fine-grained user authentication and key agreement protocol based on physical unclonable functions (PUFs) while maintaining robust security and performance. This will allow physicians to perform mutual authentication and obtain key agreements with authorized body area sensor nodes according to their identity parameters, such as occupation type and title. We then provide comprehensive security and heuristic analyses to demonstrate the security of the proposed protocol. Finally, the performance comparison shows that the proposed protocol is more robust in security, cost-effective communication, and computational overheads compared to three leading alternatives.

Keywords: authentication; physical unclonable function (PUF); wireless body area networks (WBANs)



Citation: Liu, K.; Cao, Q.; Xu, G.; Xu, G. Fine-Auth: A Fine-Grained User Authentication and Key Agreement Protocol Based on Physical Unclonable Functions for Wireless Body Area Networks. *Appl. Sci.* **2023**, *13*, 12376. <https://doi.org/10.3390/app132212376>

Academic Editor: Ugo Vaccaro

Received: 29 September 2023

Revised: 6 November 2023

Accepted: 7 November 2023

Published: 15 November 2023



Copyright: © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Wireless body area networks (WBANs) [1] have been widely used in healthcare as a mature wireless communication technology. By deploying tiny body area sensor nodes and communication devices around the body, medical staff can monitor and transmit physiological parameters and health status data in real time, effectively enhancing people's quality of life.

As shown in Figure 1, WBANs involve collaboration among medical staff, gateway nodes, and body area sensor nodes (BASNs). Medical staff or physicians are the users and controllers of the system. They obtain the patient's physiological parameters by communicating with BASNs and performing the operation of diagnosis and treatment. As an intermediate device, the gateway node (GWN) bridges medical staff and BASNs and is responsible for data transmission, forwarding, and coordination. BASNs are equipped to collect patients' biological data, including heart rate, blood pressure, body temperature, etc., while allowing medical staff to access these data in real time. In addition, BASNs can receive instructions from medical staff to perform corresponding operations as needed. Medical staff can rapidly assess the patient's physical condition via this network topology model and perform corresponding medical operations.

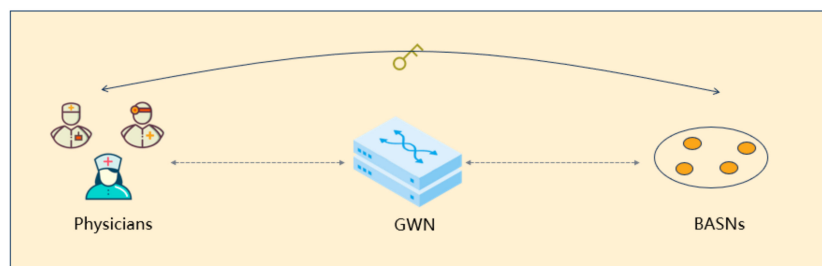


Figure 1. WBANs network topology.

However, the openness of WBANs communication will undoubtedly lead to illegal network intrusion. If adequate security protection measures for health data involving personal privacy are not implemented, serious consequences will arise, including personal privacy disclosure, data tampering, and unauthorized access [2,3]. Meanwhile, given the different identity authentication requirements for users of different professional levels, it is important to perform personalized authentication according to corresponding levels and permissions to ensure that each user can only access the resources for which they have permission. Naturally, realizing efficient, safe, and credible data transmission and personalized authentication mechanisms in WBANs is essential.

Fine-grained authentication [4] is an authentication technology designed to provide detailed and precise identity verification in order to identify and authorize users in detail according to the users' unique attributes and permissions. The advantage of this authentication technology is that it can provide a higher level of security and precise control and meet WBANs' requirements for real-time accuracy and personalization in the authentication process.

This paper aims to provide an efficient and reliable fine-grained authentication solution to ensure the privacy and security of medical data. By carrying out this study, we expect to provide a useful reference and guidance for developing fine-grained authentication technology in order to promote the expansion and application of authentication in WBANs.

1.1. Related Work

In the health field, fine-grained user authentication is an important security measure that aims to ensure that only authorized individuals or entities can gain authentication from other specific entities, maintaining personal privacy and data security. In this section, we summarize existing fine-grained authentication schemes for healthcare systems.

Chatterjee et al. pioneered an attribute-based fine-grained access control scheme to secure communication in the client-server architecture [5]. This groundbreaking scheme utilizes smart cards and biometric authentication for verification purposes. Furthermore, it enables the establishment of session keys to encrypt subsequent communications. However, it is important to highlight that the solution's communication overheads are relatively substantial, potentially impacting the overall user experience.

Wang et al. [6] introduced an access control with fog computing to achieve a more optimal balance between efficiency and security. This approach is well suited for a range of scenarios, including data storage, directory management, and file organization. However, Singh et al. in [7] reported that the scheme proposed by Wang et al. cannot achieve mutual authentication or resist device impersonation attacks.

Ogundoyin et al. recognized the sensitivity of medical data and consequently introduced a lightweight privacy-preserving authentication and fine-grained access control solution: PAASH [8]. They presented an elliptic curve cryptography (ECC)-based certificateless signature scheme. Simultaneously, they employed attribute-based encryption and signature technology to achieve precise access control. Nevertheless, Benil et al. [9] highlighted that the PAASH fails to counter impersonation, forgery, and modification attacks.

There are also works for securing communication in WBANs. The authors of Ali et al. in [10] offered a robust authentication and access control solution by using expensive

bilinear pairing. However, the password of the user can be effectively guessed by the adversary if there is no use of “modulus” operations.

Similarly, one research study [11] designed an E-health-oriented proposal that is relevant to authentication, key agreement, and access control. Furthermore, this study was the first to propose a method of transferring the ownership of patient information from the former physician to the new physician. Before the application of this charming proposal, forward secrecy and three-factor security should be applied.

For the scenario of wireless medical sensor networks, Yao et al. [12] proposed multiple solutions for user–server authentication, patient–server authentication, and user–patient authentication scenarios. However, the password verifier table is stored in the registration center and can face password-guessing attacks, which lead to the exposure of the user’s password.

1.2. Motivations and Contribution

Given the increasing adoption of WBANs in security-critical scenarios, the need to provide a fine-grained three-factor authentication solution increases. However, according to our performance analysis in Section 5, most existing fine-grained user authentication protocols commonly cannot balance security and performance:

1. Security: In terms of security, similar approaches exhibit the same security problems to greater and lesser degrees [10–12]: for example, the lack of mutual authentication, the inevitable smart card loss attacks, or the failure to provide forward secrecy.
2. Performance: From the view of storage, communication, and computation costs, existing solutions still require more resources in order to ensure the functionality of fine-grained authentication. However, WBANs are more resource-constrained than conventional networks, and a tiny body area sensor device cannot run extensive operations according to the published protocols.

Thereupon, we consider a robust and effective fine-grained user authentication scheme that can maintain a good balance between security and performance. The specific contributions of this study are as follows:

1. Fine-grained authentication protocol: We design a fine-grained authentication protocol for WBANs. This proposed scheme slows for mutual authentication among users with varying privileges and corresponding authorized BASNs while also facilitating the negotiation of a session key for encrypting subsequent data transmission.
2. Complete security analysis: The proposed protocol’s security is rigorously examined via heuristic and provable security analyses, which show that the proposed protocol attains multiple desired security properties and exhibits resilience against all known attacks.
3. Performance evaluation: Via a comparative assessment of storage, communication, and computational overheads for the proposed protocol and also other established methods, we show the advantages of the proposed protocol with respect to performance.

2. Preliminaries

In this section, we introduce preliminaries, which include the system model, adversary model, physically unclonable function, fuzzy extractor, and RSA cryptosystem, in order to ease the reader’s understanding of this study.

2.1. System Model

As shown in Figure 2, based on the standard single-gateway model [13], our system model consists of three entities: physicians at different occupational levels (A, B, and C), a gateway node (GWN), and a series of body area sensor nodes (BASNs).

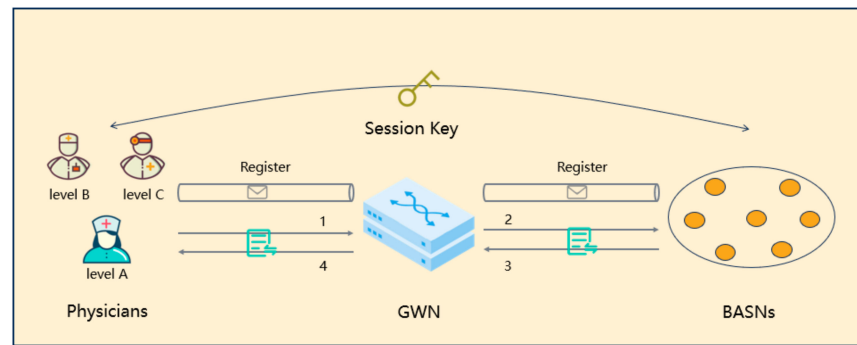


Figure 2. System model of the proposed scheme.

In the registration phase, users and BASNs register at the gateway, which corresponds to the registration process in Figure 2. At the same time, the gateway sets the user's fine-grained authentication parameters according to the user's occupation level and type in order to prepare for subsequent fine-grained authentication between users and BASNs. At the end of registration, the physicians (BASNs) retain real-time fine-grained authentication parameters (resp. secret value), and GWN stores the identity of BASNs.

In the login and authentication phase, if the user wants to access the data of some body area sensor nodes, they first need to initiate an authentication request in the gateway, which corresponds to process 1 in Figure 2. Next, the gateway authenticates the user. If authentication is successful, the gateway node sends the user's request to the BASN, which corresponds to process 2 in Figure 2. After the BASN receives the request, it first verifies the identity of the gateway, and if the authentication passes, it sends the authentication-related parameters to the gateway, which corresponds to process 3. Upon the gateway's receipt of data from BASN, it first authenticates the BASN and then updates relevant fine-grained authentication parameters according to the user's occupation; then, it sends the relevant authentication data to the user, which corresponds to process 4. Finally, the user negotiates a consistent session key with BASN via authentication data from the gateway node. The purpose of this phase is to ensure that only legitimate physicians can access the data resources of BASNs.

2.2. Adversary Model

The Dolev–Yao (DY) model, which portrays the capabilities of the adversary, has been widely used in formal and heuristic analyses with respect to the security of authentication protocols [14]. Currently, the latest research [15] has taken this a step further by consolidating adversary capabilities in order to comprehensively evaluate the authentication protocol. In this refined threat model, an adversary \mathcal{A} possesses six capacities (A-), which are as follows:

1. (A-1) \mathcal{A} can intercept, modify, insert, and delete any messages that are being transmitted through the open channel.
2. (A-2) \mathcal{A} can systematically enumerate all elements within the Cartesian product of the identity space and password space, which is denoted as $D_{id} \times D_{pw}$.
3. (A-3) \mathcal{A} is capable of obtaining previously established session keys between the physician and BASN.
4. (A-4) \mathcal{A} possesses the capability to acquire the secret key of the GWN in situations where the system eventually experiences failure.
5. (A-5) \mathcal{A} can breach some specific BASNs, extracting sensitive data stored within them. Furthermore, \mathcal{A} can manipulate the compromised BASN so that it can participate in subsequent communications involving the GWN, other users, and body area sensor nodes.
6. (A-6) \mathcal{A} could potentially register as either a legitimate user or even the role of the GWN administrator only if the security of the physician's password is evaluated

during the registration phase. Carrying out formal and heuristic analyses in Section 4, based on the DY adversary's capability, we can quantify the advantage of the adversary relative to their ability to bypass semantic security, and via heuristic analyses, we prove that the protocol can resist all kinds of attacks issued by the DY adversary.

2.3. Physical Unclonable Function

The physical unclonable function (PUF) [16] generates an output based on physical characteristics, such as delay, resistance, capacitance, or reflection properties. Since the output of the PUF is based on slight randomness and unevenness in the manufacturing process, it is difficult for an adversary to generate the same response sequence as the original PUF via copying or simulation. Therefore, the PUF has a high degree of security and protection in security systems, and it is widely used to protect sensitive information in cryptographic authentication protocols.

During the registration phase of the authentication protocol, the user or device generates a unique identifier or a unique key C via PUF and stores it securely. In the login phase, PUF generates a response: R , $R = PUF(C)$. Response R is compared with a previously registered identifier or key. If R matches the key, the login can be seen as a success, and the user is allowed to send the authentication request to the GWN. This PUF-based user-authentication mechanism takes advantage of the physical properties and unclonability of the PUF in order to ensure a secure, unique, and difficult-to-forge identity verification process. In the proposed protocol, we use a physical unclonable function $PUF_{sum}(\cdot)$, which configures the embedded trigger sum for the user, where the sum refers to the number of times that the user is allowed to try to use $PUF_{sum}(\cdot)$ in the event that the user forgets the secret key.

2.4. Fuzzy Extractor

The fuzzy extractor [17] is an important concept in cryptography, and it is especially suitable for correcting data inconsistencies caused by noise or changes. In the field of cryptographic authentication, fuzzy extractors are used to deal with the variability and noise that may be present when physical characteristics are collected to ensure authentication.

The fuzzy extractor works by converting irregular physical characteristics into a stable and consistent key or bit string. This stable key can be used for the following authentication. Importantly, the fuzzy extractor allows for the extraction of fixed and verifiable information without compromising the accuracy of physical characteristic identification in order to ensure stability and consistency during the login phase of the authentication protocol.

In this paper, the fuzzy extractor is utilized to mitigate the influence of noise during PUF execution. The system executes the PUF to obtain the R , $R = PUF(C)$ response during the registration process; then, it adds R to the $FE.Gen(R) = (K, hd)$ fuzzy extractor and stores the auxiliary string hd . During the login phase, the system can execute the PUF to obtain the current $R' = PUF(C')$ response and employ the hd stored in memory to determine $K' = FE.Rec(R', hd)$. If K' and K are not equal, this indicates that an unauthorized user attempted to initiate a login request, and the system dismisses this illegal login request.

2.5. RSA Cryptosystem

In the realm of public key cryptography, the RSA cryptosystem [18], founded on the intricacies of the large number factorization problem, is elucidated below. To aid comprehension, an example involving a message sender denoted as Sed who transmits a message m to a message receiver Rev is presented.

Initiation: The message receiver Rev selects two substantial prime numbers p and q . Subsequently, Rev computes $n = p \cdot q$ and Euler's totient function of n , which is denoted as $\varphi(n) = (p - 1) \cdot (q - 1)$. Next, Rev chooses an integer e that satisfies $\gcd(e, \varphi(n)) = 1$. The receiver then computes $d \equiv e^{-1} \pmod{\varphi(n)}$. The outcome is that Rev publicizes the public key (e, n) while keeping the private key d confidential.

Encryption: The message sender *Sed* takes the message *m* and performs an encryption operation $c = m^e \bmod n$ using *Rev*'s public key *e*. Consequently, *Sed* transmits the resultant cipher *c* to *Rev*.

Decryption: Upon the receipt of the cipher *c*, message receiver *Rev* employs private key *d* to decipher the message. This is accomplished via the computation $m = c^d \bmod n$.

In Section 3 (i.e., step L2 in the user login phase and step V10 in the authentication and key agreement phase), we provide the detailed method of using RSA to securely transmit secret values.

3. The Proposed Protocol

Aiming at the solving common security and storage problems of authentication protocols in WBANs, we propose a fine-grained user authentication method based on the physical unclonable function (PUF). Specifically, it includes seven phases, namely system initialization, body area sensor node and user registration, user login, authentication and key agreement, password update, and dynamic node addition. To promote the understanding of researchers, some notations used in the proposed protocol are explained in Table 1.

Table 1. Notations with related descriptions in the proposed protocol.

Notations	Descriptions	Notations	Descriptions
<i>GWN</i>	Gateway node	S_M	The set of BASN's identity
\oplus	XOR operation	$T_{reg}^{U_i}$	Registration timestamp of U_i
T_c	Current timestamp	PID_i	A pseudo-random identity of U_i
<i>GID</i>	<i>GWN</i> 's identity	$f_{U_i}(t)$	Authorization check polynomial
\parallel	Bit concatenation	(ID_i, PW_i)	The identity and password of U_i
MIS_j	The identity of MS_j	$FE.REP(\cdot)$	Fuzzy extraction and recovery function
$h(\cdot)$	Secure hash function	$A \implies B : M$	The message M is sent from A to B through a secure channel
U_i	<i>i</i> th user (medical staff)	$A \rightarrow B : M$	The message M is sent from A to B through a public channel
x_j	The secret value of MS_j	$PUF_{sum}(\cdot)$	The physically unclonable function with embedded <i>sum</i>
x, y	<i>GWN</i> 's long-term key pair	$\Delta T_{auth}^{U_i}$	Time threshold for U_i to be authorized in order to obtain authentication
MS_j	<i>j</i> th body area sensor node	$S_M^{U_i}$	The set of BASN's identity for U_i to be authorized in order to obtain authentication

3.1. System Initialization Phase

In the initialization phase, given a security parameter *n*, the gateway node *GWN* selects a long-term key pair $x, y \in \{0, 1\}^n$ and generates a unique identity *GID*. Then, *GWN* saves $\{x, y\}$ and publicizes identity *GID*.

3.2. Registration Phase

The registration phase comprises the following: In the terminal of the *GWN*, the user and BASN need to complete the registration of identity information and receive authentication parameters in order to be ready for future identity authentication and key agreement. Specifically, the registration phase includes the registration of the BASN and users.

3.2.1. Registration Phase of BASN MS_j

The registration of MS_j includes R11~R13:

R11: $MS_j \implies GWN : MIS_j$; the MS_j transmits identity MIS_j to gateway node *GWN* via a secure channel. Then, gateway *GWN* collects MIS_j and stores it in identity set $S_M = \{MIS_j\}$.

R12: $GWN \implies MS_j : \{x_j\}$; *GWN* calculates secret value $x_j = h(MIS_j \parallel x)$ for MS_j and also returns secret value x_j to MS_j via a secure channel.

R13: MS_j stores x_j secretly.

Note that the secure channel can be understood here as comprising the user and node devices that are within the same physical space (such as a computer room managed by a hospital administrator); then, the registration and sharing of secret values can be completed in a face-to-face manner.

3.2.2. Registration Phase for User U_i

The registration of user U_i includes R21~R23:

R21: $U_i \implies GWN : \{A_0\}$; user U_i transmits the calculated A_0 to GWN via a secure channel.

Specifically, U_i inputs the ID_i and PW_i of their own choice to the personal digital assistant (PDA). The PDA selects a random number $r \in [1, n - 1]$, where n is a system security parameter. Then, PDA calculates the following: hash value $A_0 = h(ID_i \oplus PW_i \parallel r) \bmod n_0$; n_0 is a large prime number.

R22: $GWN \implies U_i : \{\text{Registration Package (RP)}\}$. The GWN sends a registration package to U_i .

After GWN receives information A_0 from user U_i , it first records the registration timestamp $T_{reg}^{U_i}$, selects a pseudo-random identity PID_i , and computes $V_i = h(PID_i \parallel x)$ and $A_1 = V_i \oplus A_0$. Secondly, GWN determines the body area sensor node identity set $S_M^{U_i}$, which is authorized for the authentication of U_i , and binds the authorization check polynomial; the authorized authentication time threshold; and the authorization, authentication, and verification value for user U_i , where $S_M^{U_i}$ is a subset of S_M , the check polynomial is $f_{U_i}(t) = h(PID_i \oplus x) + \prod_{MIS_j \in S_M^{U_i}} (t - h(PID_i \oplus MIS_j))$, the time threshold is $\Delta T_{auth}^{U_i}$, and

the authentication verification value is $EID_i = T_{reg}^{U_i} \cdot (h(y \parallel PID_i))^{-1}$. Furthermore, GWN uses symmetrical algorithms (e.g., the well-known AES [19]) to generate symmetric ciphertext $F_{U_i}(t)$ and configures the physically unclonable function $PUF_{sum}(\cdot)$ with embedded trigger sum , where $F_{U_i}(t) = Enc_{h(x \oplus y)} [f_{U_i}(t), \Delta T_{auth}^{U_i}, EID_i]$, and sum refers to the number of times the user is allowed to try to use $PUF_{sum}(\cdot)$. Here, we set the maximum value to 3 and the initial value to 0. Finally, GWN sends the registration package (RP) to U_i , where RP stores parameters $PID_i, PUF_{sum}(\cdot), A_1, S_M^{U_i}$, and $F_{U_i}(t)$.

R23: After U_i receives RP, U_i updates A_1 and calculates A_2 as follows: At first, U_i inputs ID_i and PW_i to PDA, and PDA computes $V_i = A_0 \oplus A_1$ and $V_{ii} = PUF_{sum}(PW_i)$. PDA then uses $FE.GEN(\cdot)$ to compute the following: $(k_i, k_{ii}) = FE.GEN(V_{ii})$, $A_2 = h(V_i \parallel k_i \parallel S_M^{U_i}) \bmod n_0$. After that, PDA updates secret value $A_1 = V_i \parallel S_M^{U_i} \parallel k_{ii} \oplus h(ID_i \parallel V_{ii})$. Finally, PDA stores a series of values: $\langle PID_i, A_1, A_2, PUF_{sum}(\cdot), F_{U_i}(t) \rangle$.

During registration, the gateway node no longer issues smart cards to users, thereby avoiding offline password-guessing attacks, which result from smart card loss attacks. At the same time, the periodicity of modulo calculations makes it impossible for the adversary to guess passwords effectively in order to protect password security. Meanwhile, the gateway needs to encrypt these fine-grained authentication parameters to prevent users from tampering with them. Additionally, to ease the understanding of readers, Figure 3 summarizes the registration operation of the user and BASN.

3.3. Login Phase

In the login phase, the user needs to be verified via the PDA. Upon the PDA's authentication of the user identity's legitimacy, the user can log in via the PDA successfully; then, the PDA generates an authentication request for some specific BASN. Furthermore, the PDA transmits this authentication request to the GWN . The login phase includes three steps from L1 to L3:

L1: Firstly, U_i enters the identity and password (ID_i^*, PW_i^*) . Secondly, PDA uses the physically unclonable function $PUF_{sum}(\cdot)$ to verify the user's identity. Specifically, PDA computes $V_{ii}^* = PUF_{sum}(PW_i^*), V_i^* \parallel S_M^{U_i^*} \parallel k_{ii}^* = h(ID_i^* \parallel V_{ii}^*) \oplus A_1, k_i^* = FE.REP(V_{ii}^*, k_{ii}^*),$

and $A_2^* = h(V_i^* \parallel k_i^* \parallel S_M^{U_i^*}) \bmod n_0$. Thirdly, PDA compares whether A_2^* is equal to A_2 ; if $A_2^* = A_2$, the user's identity can be verified. Then, it proceeds to step L2 to continue the execution; otherwise, $PUF_{sum}(\cdot)$ will add 1 to the value of sum automatically when the user tries to enter another (ID_i^*, PW_i^*) again for the login. If the value exceeds the preset maximum value, the session will be terminated, and the user's account will be frozen until U_i re-registers.

L2: PDA runs a 1024-bit RSA cryptosystem to generate public key e_i and private key d_i for U_i , and PDA keeps d_i secret. Then, the PDA selects a random number $r_u \in [1, n - 1]$; chooses the identity of MS_j , which the user wants to acquire; extracts the timestamp T_1 ; and calculates the following parameters: $B_1 = h(V_i) \oplus (h(r_u \parallel T_1) \parallel e_i \parallel S_M^{U_i})$ and $B_2 = h(PID_i \parallel MIS_j \parallel h(r_u \parallel T_1) \parallel e_i \parallel S_M^{U_i})$, where e_i is the public key of U_i .

L3: $U_i \rightarrow GWN : \{F_{U_i}(t), PID_i, MIS_j, B_1, B_2, T_1\}$. PDA sends the requested information to GWN in a public channel.

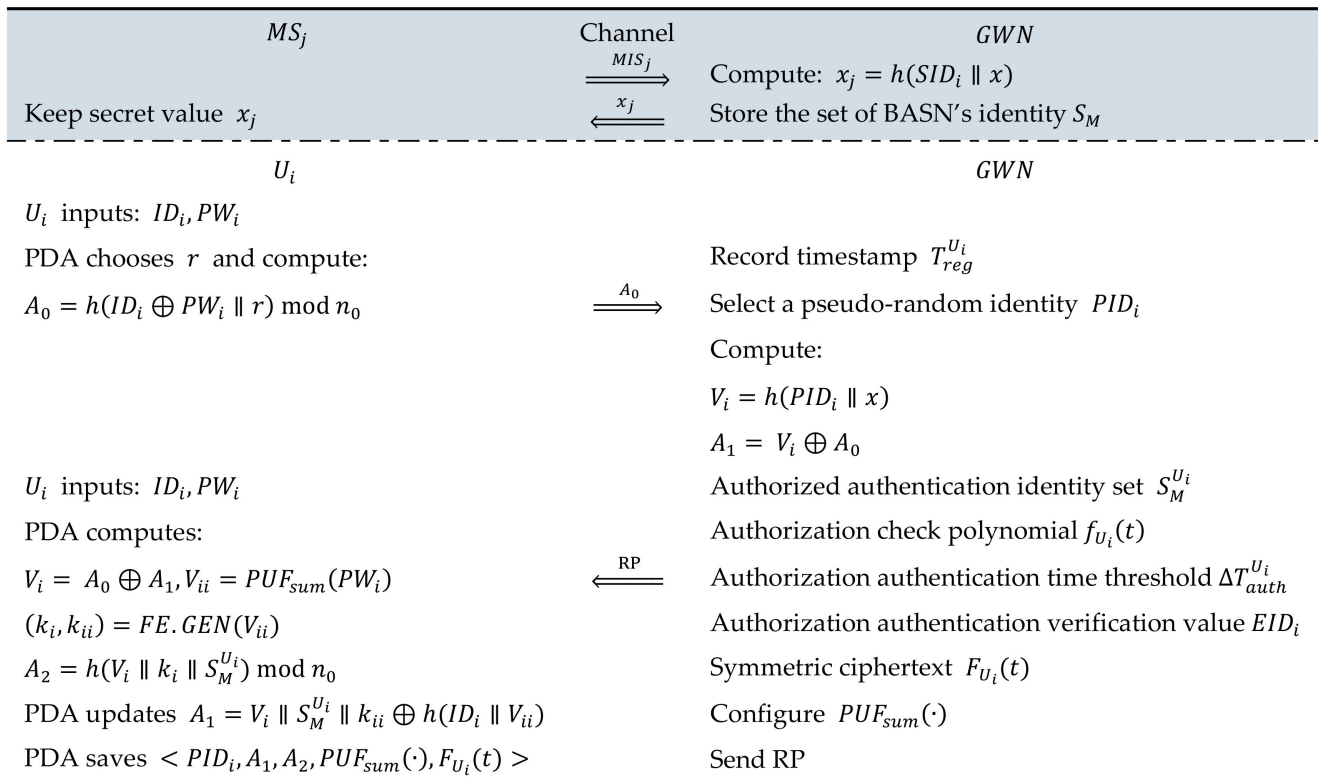


Figure 3. Registration of the user and BASN.

3.4. Authentication and Key Agreement Phase

In the authentication and key agreement phase, three entities (i.e., user, GWN, and BASN) first verify the identity of the communicating party when communicating with each other. At the same time, U_i and MS_j can negotiate a consistent session key. This phase includes ten steps from V1 to V10:

V1: Upon GWN's receipt of the requested information by user U_i , it first extracts the current timestamp T_c and checks whether the time gap between T_c and T_1 is less than time threshold ΔT . If not, GWN directly discards the request information; if so, GWN decrypts $F_{U_i}(t)$ and obtains $[f_{U_i}^*(t), \Delta T_{auth}^{U_i}, EID_i^*]$.

Then, GWN verifies whether U_i is authorized to obtain authentication with MS_j , for which its identity is MIS_j . GWN calculates $f_{U_i}^*(h(PID_i \oplus MIS_j))$ and $h(PID_i \oplus x)$ and checks whether the two values are equal. If so, GWN further judges whether EID_i^* belongs to case (a), case (b), or case (c):

- (a) $EID_i^* \cdot h(y \parallel PID_i) = h(PID_i \parallel y)$;
- (b) $|T_c - EID_i^* \cdot h(y \parallel PID_i)| > \Delta T_{auth}^{U_i^*}$;
- (c) $|T_c - EID_i^* \cdot h(y \parallel PID_i)| \leq \Delta T_{auth}^{U_i^*}$.

If it belongs to case (a), this means that U_i does not have authentication authority with respect to MS_j ; if it belongs to case (b), this means that the authentication authority of U_i exceeds the time threshold for U_i to be authorized; if it belongs to case (c), this means that the body area sensor node MS_j that U_i wants to access can be authorized for authentication within a valid period of time. GWN further computes $V_i^* = h(PID_i \parallel x)$, $h(r_u^* \parallel T_1) \parallel e_i^* \parallel S_M^{U_i^*} = B_1 \oplus h(V_i^*)$, and $B_2^* = h(PID_i \parallel MIS_j^* \parallel h(r_u^* \parallel T_1) \parallel e_i^* \parallel S_M^{U_i^*})$. Finally, GWN compares B_2^* and B_2 . If they are equal, i.e., $h(r_u^* \parallel T_1) = h(r_u \parallel T_1)$ and $e_i^* = e_i$, $S_M^{U_i^*} = S_M^{U_i}$, this means that U_i can be validated, and the operation proceeds to step V2; otherwise, GWN terminates this session.

V2: GWN selects random number $r_g \in [1, n - 1]$, extracts the current timestamp T_2 , and computes the following: $x_j = h(MIS_j \parallel x)$, $B_3 = h(x_j \parallel MIS_j) \oplus (r_g \parallel h(r_u \parallel T_1))$, $B_4 = (e_i \parallel h(V_i \parallel EID_i)) \oplus h(x_j \parallel r_g)$, and $B_5 = h(h(r_u \parallel T_1) \parallel r_g \parallel x_j \parallel h(V_i \parallel EID_i) \parallel T_2)$.

V3: $GWN \rightarrow MS_j : \{B_3, B_4, B_5, T_2\}$. GWN sends $\{B_3, B_4, B_5, T_2\}$ to the body area sensor node MS_j .

V4: MS_j firstly extracts timestamp T_c and checks whether the time gap between T_c and T_2 is less than time threshold ΔT . If so, MS_j recovers $r_g^*, h(r_u^* \parallel T_1), e_i^*, h(V_i^* \parallel EID_i^*) : r_g^* \parallel h(r_u^* \parallel T_1) = B_3 \oplus h(x_j \parallel MIS_j)$, and $e_i^* \parallel h(V_i^* \parallel EID_i^*) = B_4 \oplus h(x_j \parallel r_g^*)$. At the same time, MS_j computes $B_5^* = h(h(r_u^* \parallel T_1) \parallel r_g^* \parallel x_j \parallel h(V_i^* \parallel EID_i^*) \parallel T_2)$ and compares the value of B_5^* and B_5 . If the two values are equal, GWN's identity is verified, and MS_j proceeds to step V5; otherwise, MS_j terminates this session.

V5: MS_j selects random number $r_s \in [1, n - 1]$, extracts timestamp T_3 , and computes $r'_s = (r_s)^{e_i}$ via the RSA algorithm: $SK = h(h(r_u \parallel T_1) \parallel r_s \parallel h(V_i \parallel EID_i))$, $B_6 = MIS_j \oplus h(r_g)$, $B_7 = r'_s \parallel h(SK \parallel r_g) \oplus x_j$, $B_8 = h(r'_s \parallel h(SK \parallel r_g) \parallel x_j) \parallel T_3$, and $B_9 = h(SK \parallel r_g) \oplus x_j \oplus h(r'_s \parallel SK)$, where B_6, B_7, B_8 , and B_9 represent the intermediate parameters, and SK represents the session key of MS_j and U_i .

V6: $MS_j \rightarrow GWN : \{B_6, B_7, B_8, B_9, T_3\}$. MS_j sends information B_6, B_7, B_8, B_9 , and T_3 to GWN.

V7: GWN firstly checks the validity of the timestamp. Next, GWN uses the secret value r_g to compute x_j^*, r'_s^* : $MIS_j^* = B_6 \oplus h(r_g)$, $x_j^* = h(MIS_j^* \parallel x)$, $r'_s^* \parallel h(SK^* \parallel r_g^*) = B_7 \oplus x_j^*$, and $B_8^* = h(r'_s^* \parallel h(SK^* \parallel r_g^*) \parallel x_j^*) \parallel T_3$. Then, GWN checks the consistency of B_8^* and B_8 . If the two values are equal, GWN executes step V8; otherwise, it terminates this communication.

V8: GWN computes $h(r'_s \parallel SK) = B_9 \oplus h(SK \parallel r_g) \oplus x_j$ and updates the parameters as follows:

GWN updates the new pseudo-random identity PID_i^{new} for U_i ;

GWN updates $V_i^{new} = h(PID_i^{new} \parallel x)$;

GWN updates the authorization verification value EID_i^{new} , the authorized authentication time threshold $\Delta T_{auth}^{U_i^{new}}$, and the authorized authentication body area sensor node identity set $S_M^{U_i^{new}}$. Specifically, if the authentication authority of U_i needs to be revoked, then GWN computes $EID_i^{new} = h(PID_i^{new} \parallel y) \cdot (h(y \parallel PID_i^{new}))^{-1}$ and sets $\Delta T_{auth}^{U_i^{new}} = \text{null}$, $f_{U_i}^{new}(t) = \text{null}$, and $S_M^{U_i^{new}} = \text{null}$; otherwise, GWN computes $EID_i^{new} = (h(y \parallel PID_i^{new}))^{-1} \cdot T_c$, updates $\Delta T_{auth}^{U_i^{new}}$, and further updates the authorized authentication identity set $S_M^{U_i^{new}}$ according to situations (d), (e), (f), and (g):

- (d) If there is no change in the identity set of the body area sensor node, then $S_M^{U_i^{new}} = S_M^{U_i}$.

- (e) If there is a newly added identity set S_M^{add} with respect to the body area sensor node, then $S_M^{U_i^{new}} = S_M^{U_i} + S_M^{add}$.
- (f) If identity set S_M^{del} is removed from the body area sensor node, then $S_M^{U_i^{new}} = S_M^{U_i} - S_M^{del}$.
- (g) If cases (e) and (f) occur simultaneously, then $S_M^{U_i^{new}} = S_M^{U_i} - S_M^{del} + S_M^{add}$.

GWN updates the value of $f_{U_i}^{new}(t)$, where $f_{U_i}^{new}(t) = h(PID_i^{new} \oplus x) + \prod_{MIS_j \in S_M^{U_i^{new}}} (t - h(PID_i \oplus MIS_j))$.

GWN updates $F_{U_i}^{new}(t) = Enc_{h(x \oplus y)} [f_{U_i}^{new}(t), \Delta T_{auth}^{U_i^{new}}, EID_i^{new}]$.

GWN computes $B_{10} = (V_i^{new} \parallel S_M^{U_i^{new}} \parallel h(V_i \parallel EID_i)) \oplus V_i$ and $B_{11} = PID_i^{new} \parallel r'_s \oplus h(V_i^{new} \parallel h(r_u \parallel T_1))$ and further computes $B_{12} = h(V_i^{new} \parallel h(r'_s \parallel SK) \parallel S_M^{U_i^{new}})$.

V9: $GWN \rightarrow U_i : \{F_{U_i}^{new}(t), B_{10}, B_{11}, B_{12}\}$; GWN sends related information $F_{U_i}^{new}(t)$, B_{10}, B_{11}, B_{12} to U_i .

V10: U_i uses V_i to recover V_i^{new*} and $S_M^{U_i^{new*}}, h(V_i^* \parallel EID_i^*)$, i.e., $(V_i^{new*} \parallel S_M^{U_i^{new*}} \parallel h(V_i^* \parallel EID_i^*)) = B_{10} \oplus V_i$. Then, U_i uses private key d_i and then computes $PID_i^{new*} \parallel r'_s = B_{11} \oplus h(V_i^{new*} \parallel h(r_u \parallel T_1)), r_s^* = (r'_s)^{d_i}, SK^* = h(h(r_u \parallel T_1) \parallel r_s^* \parallel h(V_i^* \parallel EID_i^*))$, and $B_{12}^* = h(V_i^{new*} \parallel h(r'_s \parallel SK^*) \parallel S_M^{U_i^{new*}})$. If B_{12}^* equals B_{12} , U_i accepts session key SK and completes the authentication; otherwise, U_i rejects the session key. After accepting the session key, the PDA computes parameters $A_1^{new} = (V_i^{new} \parallel S_M^{U_i^{new}}) \oplus h(ID_i \oplus PW_i)$ and $A_2^{new} = h(V_i^{new} \parallel k_i \parallel S_M^{U_i^{new}}) \bmod n_0$. Finally, the PDA updates the value from $\{PID_i, A_1, A_2, F_{U_i}(t)\}$ to $\{PID_i^{new}, A_1^{new}, A_2^{new}, F_{U_i}^{new}(t)\}$. Additionally, Figure 4 summarizes the login, authentication, and key agreement operations of the user and BASN.

3.5. Password Update Phase

U_i can update the password by following steps U1~U2 below without interacting with GWN:

U1: At first, U_i enters (ID_i^*, PW_i^*) . PDA then uses $PUF_{sum}(\cdot)$ to verify the identity of U_i . Specifically, PDA computes $V_{ii}^* = PUF_{sum}(PW_i^*), V_{ii}^* \parallel S_M^{U_i^*} \parallel k_{ii}^* = h(ID_i^* \parallel V_{ii}^*) \oplus A_1, k_i^* = FE.REP(V_{ii}^*, k_{ii}^*)$, and $A_2^* = h(V_i^* \parallel k_i^* \parallel S_M^{U_i^*}) \bmod n_0$. If A_2^* is equal to A_2 , PDA continues to run step U2; otherwise, this session is terminated.

U2: Via the new password PW_i^{new} , which is chosen by U_i , the PDA computes the new parameters: $V_{ii}^{new} = PUF(PW_i^{new}), (k_i^{new}, k_{ii}^{new}) = FE.GEN(V_{ii}^{new}), A_1^{new} = V_i \parallel S_M^{U_i} \parallel k_{ii}^{new} \oplus h(ID_i \parallel V_{ii}^{new})$, and $A_2^{new} = h(V_i \parallel k_i^{new} \parallel S_M^{U_i}) \bmod n_0$. Finally, the PDA updates the value of $\{A_1, A_2\}$ to $\{A_1^{new}, A_2^{new}\}$.

3.6. Dynamic Increase in Sensor Nodes

In order to adapt or meet the continuous medical needs of WBANs, the addition of new body area sensor nodes is undoubtedly necessary. When a new body area sensor node S_t joins WBANs, S_t only needs to initiate a registration request as in Section 3.2.1. After S_t is successfully registered, GWN broadcasts S_t 's identity SID_t and stores SID_t in identity set S_M .

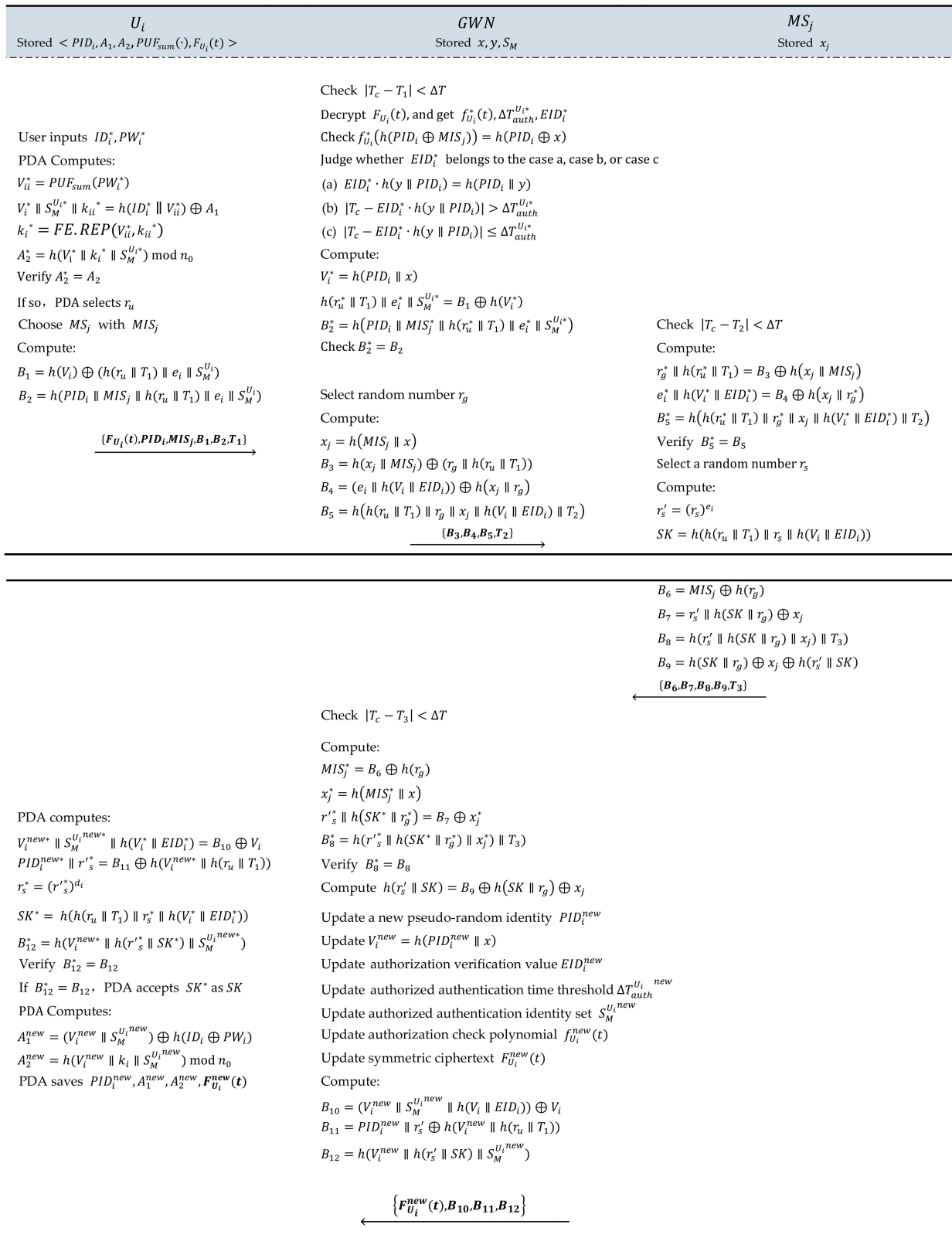


Figure 4. Login, authentication, and key agreement.

4. Security Analysis of the Proposed Protocol

In this section, we carry out the security analysis of the protocol, which includes formal security and heuristic security analyses. Given the DY adversary’s capabilities, via formal

security analysis, we can demonstrate that the adversary does not have a significant and strong advantage with respect to breaking the semantic security of the session key in our protocol. Then, via heuristic analysis, one can observe that the proposed protocol not only fulfills the desired attributes but also demonstrates resilience against a multitude of known attacks [20,21].

4.1. Formal Security Proof

Formal security analysis is an effective method for demonstrating the semantic security of the proposed protocol. It employs formal mathematical techniques to prove whether the proposed protocol satisfies the desired security properties within a specific security model. In this type of analysis, the initial step involves defining a formal model that describes participants, furthering the DY adversary's capabilities. Subsequently, security properties need to be precisely defined. Lastly, the desired security properties are proven to be satisfied within the defined security model via mathematical, logical deduction.

4.1.1. Basis for Security Proof

In this study, three primary participants within our protocol P are identified as follows: a physician denoted as U_i , a gateway node referred to as GWN , and a body area sensor node labeled as MS_j . Before initiating the simulation, the simulator selects the RSA cryptosystem, employing two large prime numbers p and q with equal bit lengths, i.e., $|p| = |q| = n$. Subsequently, U_i selects a set of personal information $\{ID_i, PW_i\}$. Simultaneously, GWN generates a long-term key pair $\{x, y\}$, and MS_j owns an identity–secret key pair $\{MIS_j, x_j\}$.

During the proof process, three entities will instantiate U_i , GWN , and MIS_j , with their respective instances denoted as $\Pi_{U_i}^u$, Π_{GWN}^g , and $\Pi_{MS_j}^m$. For the sake of simplicity, these instances can collectively be marked as Π^t when distinguishing them is unnecessary. Moreover, each instance is treated as an oracle. This implies that if a message input is valid, invalid, or null, the oracle's state accordingly is an acceptance, rejection, or " \perp ", respectively, where " \perp " indicates that there is no response to the input.

Subsequently, we introduce certain terms that are pertinent to this proof:

Accepted State: An instance Π^t reaches an accepted state upon receiving the final expected protocol message. Notably, the ordered concatenation of all exchanged messages (both sent and then received) shapes the session identifier for the current session of Π^t .

Partnering: Instances Π^{t_1} and Π^{t_2} are considered partnered if they simultaneously meet the following criteria: (a) both are in an accepted state, (b) mutual authentication has occurred, and they share an identical session identifier.

Adversary: In this context, adversary \mathcal{A} can interact solely with honest entities by initiating query oracles and controlling the simulator. \mathcal{A} aims to compromise the security of authentication messages and re-construct the session key within protocol P . The queries that \mathcal{A} can launch include the following:

- Execute ($\Pi_{U_i}^u, \Pi_{GWN}^g, \Pi_{MS_j}^m$): This query allows \mathcal{A} to simulate the entire authentication process and access exchanged messages between U_i , GWN , and MS_j .
- Send (Π^t, m): \mathcal{A} can send message m and conduct an active attack on instance Π^t . If l is valid and Π^t has received m , the simulator responds to \mathcal{A} with the computation of m ; otherwise, this query is terminated.
- Reveal (Π^t): This query results in revealing the session key calculated by Π^t and its partner to adversary \mathcal{A} .
- Corrupt ($\Pi_{U_i}^u, \alpha$): In this query, \mathcal{A} can obtain authentication factors associated with U_i based on value α . Specifically, the oracle exposes the password to \mathcal{A} when $\alpha = 0$ and exposes the data stored in the registration package to \mathcal{A} when $\alpha = 1$.
- Corrupt (Π_{GWN}^g): In this query, \mathcal{A} can gain access to the long-term key x possessed by GWN .
- Corrupt ($\Pi_{MS_j}^m$): This query enables \mathcal{A} to obtain the secret value of MS_j .

Freshness: An instance of $\Pi_{u_i}^u$, Π_{GWN}^s , or $\Pi_{MS_j}^m$ is deemed fresh if the session key between U_i and MS_j remains undisclosed to \mathcal{A} via the aforementioned reveal query.

Test (Π^t): This query assesses the semantic security of session key SK . In this query, \mathcal{A} can make only one inquiry. Considering protocol \mathcal{P} , instance Π^t can only either be $\Pi_{u_i}^u$ or $\Pi_{MS_j}^m$. Formally, if test (Π^t) has been queried before, the query outputs “ \perp ” (null). Otherwise, the oracle flips an unbiased coin b . If $b = 1$, test (Π^t) provides the real session key to \mathcal{A} ; if $b = 0$, test (Π^t) yields a random string with the same length as the real session key and sends it to \mathcal{A} .

Semantic Security: Given a protocol \mathcal{P} , probabilistic polynomial time (PPT) adversary \mathcal{A} requests new instances for a series of queries, including execute query, send query, corrupt query, and test query. \mathcal{A} endeavors to compromise protocol \mathcal{P} by guessing the value of b in the test query and returns a guessed value b^* . Let $\text{Succ}(\mathcal{A})$ denote \mathcal{A} 's successful guess of b^* as b , i.e., $b^* = b$. Then, the advantage of \mathcal{A} successfully breaking the semantic security of protocol \mathcal{P} concerning the session key is defined as $\text{Adv}_{\mathcal{A}}^{\mathcal{P}} = 2\text{Pr}[\text{Succ}(\mathcal{A})] - 1$.

4.1.2. Security Proof

In this section, we set up a total of eight games to simulate the semantic security of the adversary's ability to break the session key from different perspectives. Among these simulated games, the only difference is that the latter game provides more information to the adversary; the former and latter games are indistinguishable to the adversary. In each game, the simulator responds to queries from the adversary, who, in turn, obtains different information to increase their advantage of interfering with semantic security. Finally, based on the advantage of the adversary in each game, the total advantage of the adversary in interfering with the semantic security of the session key can be quantified.

Theorem 1. Let \mathcal{P} be the proposed protocol, $|\mathcal{D}|$ be the space of password, and n be the system's security parameter. Then, PPT adversary \mathcal{A} breaks \mathcal{P} with a negligible advantage $\text{Adv}_{\mathcal{A}}^{\mathcal{P}, \mathcal{D}}$ by making a series of queries, including q_e execute query, q_s send query, q_h hash query, and q_p PUF query, where $\text{Adv}_{\mathcal{A}}^{\mathcal{P}, \mathcal{D}}$ encounters the following.

$$\text{Adv}_{\mathcal{A}}^{\mathcal{P}, \mathcal{D}} \leq \frac{q_h^2 + 6q_s}{2^{l_1}} + \frac{(q_s + q_e)^2}{p} + \frac{q_p^2 + 2q_p}{2^{l_2}} + 2 \left(C' q_{send}^s + \text{Adv}_{\mathcal{A}}^{\text{RSA}}(n) \right)$$

Proof. We now demonstrate and prove that the adversary's advantage in breaking the semantic security of the session key is factually negligible due to the involvement of **Game₁** with **Game₈**. Succ_i is set to be the event during which \mathcal{A} guesses b in the test query of Game_k successfully, where $k = 1, 2, \dots, 8$.

Game₁: This game simulates a real attack by a random oracle. Bit b is then randomly chosen at the beginning of this game. Thus, we obtain the following:

$$\text{Adv}_{\mathcal{A}}^{\mathcal{P}, \mathcal{D}} = 2\text{Pr}[\text{Succ}_1] - 1 \tag{1}$$

Game₂: This game shapes hash list Ω_h . For example, \mathcal{A} initiates a hash query $h(\gamma)$ and hash oracle Θ_h takes γ to retrieve Ω_h . If there is a retrieved hash value, $h(\gamma)$, in Ω_h , Θ_h responds to the hash value. Otherwise, a random string ψ will be sent to \mathcal{A} ; meanwhile, (γ, ψ) is stored in Ω_h .

Using the known list in this game, \mathcal{A} performs a test query to distinguish the real session key and the random string. Factually, given $SK = h(h(r_u \parallel T_1) \parallel r_s \parallel h(V_i \parallel EID_i))$, only secret values including U_i 's r_u, V_i , and MS_j 's r_s essentially comprise SK . Hence, \mathcal{A} has no way of computing SK and cannot distinguish whether $b = 0$ or $b = 1$ other than making guesses.

Thus, compared to **Game**₁, \mathcal{A} 's chance of winning this game does not empower \mathcal{A} 's advantage.

$$\Pr[Succ_2] = \Pr[Succ_1] \tag{2}$$

Game₃: In this game, the active attack is modeled based on **Game**₂. \mathcal{A} can execute the send query and hash query to try to persuade a participant to accept a forged message. Thus, \mathcal{A} 's advantage may be enhanced by finding the collision that generates a valid message compared with **Game**_{1/2}. That is, if the following collisions occur, this game is aborted:

- (i) A collision can be found in the hash values or PUF's outputs, and the probability is $\frac{q_h^2}{2^{l_1+1}}$ or $\frac{q_p^2}{2^{l_2+1}}$, where l_1 and l_2 denote the length of output by the hash function and PUF, respectively.
- (ii) Another collision that can be found is relative to the choice of random numbers r_u, r_g , and r_s , where the probability is $\frac{(q_s+q_e)^2}{2^p}$.

Thus, we have the following:

$$|\Pr[Succ_3] - \Pr[Succ_2]| \leq \frac{q_h^2}{2^{l_1+1}} + \frac{q_p^2}{2^{l_2+1}} + \frac{(q_s + q_e)^2}{2^p} \tag{3}$$

Game₄: In this game, \mathcal{A} desires to guess B_2, B_5, B_8 , and B_{12} without asking the hash query.

We can obtain:

$$|\Pr[Succ_4] - \Pr[Succ_3]| \leq \frac{q_s}{2^{l_1}} \tag{4}$$

Game₅: In this game, \mathcal{A} tries to guess A_2 without asking the hash query. Similarly, we can obtain the following:

$$|\Pr[Succ_5] - \Pr[Succ_4]| \leq \frac{q_s}{2^{l_1}} \tag{5}$$

Game₆: In this game, via the corrupt $(\Pi_{u_i}^u, \alpha)$ query, \mathcal{A} computes A_2 . There are two cases we need to consider:

Case 1, i.e., corrupt $(\Pi_{u_i}^u, \alpha = 0)$: with respect to "fuzzy keywords + honeywords", the probability that \mathcal{A} guesses a physician's password is no greater than $C' q_{send}^s$ [22–24];

Case 2, i.e., corrupt $(\Pi_{u_i}^u, \alpha = 1)$: the probability that \mathcal{A} guesses the values of A_2 is less than $\frac{q_s}{2^{l_1}}$.

Therefore, we obtain the following:

$$|\Pr[Succ_6] - \Pr[Succ_5]| \leq C' q_{send}^s + \frac{q_s}{2^{l_1}} \tag{6}$$

Game₇: In this game, \mathcal{A} initiates a corrupt $(\Pi_{MS_j}^m)$ query to compromise body area sensor node MS_j , and then \mathcal{A} further obtains secret values x_j and r'_s . However, \mathcal{A} cannot obtain r_s from r'_s since there is no PPT solution for solving the difficulty of the large number factorization problem [18].

Therefore, we can yield the following:

$$|\Pr[Succ_7] - \Pr[Succ_6]| \leq Adv_{\mathcal{A}}^{RSA}(n) \tag{7}$$

Game₈: This game simulates the attack where \mathcal{A} tries to calculate the session key, which means that \mathcal{A} no longer queries the oracle's execute query, send query, and corrupt query. However, similarly to the analysis in **Game**₇, \mathcal{A} cannot compute r_s from r'_s . In other

words, \mathcal{A} 's advantage in this game is equal to the advantage in **Game**₇. Thus, we can have the following:

$$\Pr[Succ_8] = \Pr[Succ_7] \tag{8}$$

Ultimately, we can observe that \mathcal{A} has no un-negligible advantage greater than $\frac{1}{2}$; thus, $\Pr[Succ_8] = \frac{1}{2}$.

From Equation (1) to Equation (8) and with respect to the triangular inequality, we yield the following:

$$Adv_{\mathcal{A}}^{P,D} = 2\Pr[Succ_1] - 1 = 2\Pr[Succ_8] - 1 + 2(\Pr[Succ_1] - \Pr[Succ_8]) \leq \frac{q_h^2 + 6q_s}{2^1} + \frac{(q_s + q_e)^2}{p} + \frac{q_p^2}{2^2} + 2(C' q_{send}^{s'} + Adv_{\mathcal{A}}^{RSA}(n))$$

In sum, we can conclude that adversary \mathcal{A} does not have a significant and strong advantage, $Adv_{\mathcal{A}}^{P,D}$, in breaking the semantic security of the session key in our protocol. \square

4.2. Heuristic Analysis

The heuristic method [25,26] eschews the use of intricate formulas, making it remarkably straightforward. This approach proves to be both highly efficient and uncomplicated, enabling a succinct yet all-encompassing security analysis of the scheme. In this section, via the heuristic analysis, one can observe that our solution not only fulfills the desired attributes but also demonstrates resilience against a multitude of known attacks.

4.2.1. Mutual Authentication

The proposed scheme can attain mutual authentication since U_i and GWN authenticate each other bidirectionally by checking if $B_2^* = B_2$ and $B_{12}^* = B_{12}$, respectively. Similarly, with MS_j checking whether $B_5^* = B_5$ and GWN verifying that $B_8^* = B_8$, GWN and MS_j can authenticate each other successfully.

4.2.2. Session Key Agreement

The session key agreement means that no one can solely pre-compute the session key without interacting with another entity. Factually, in the proposed protocol, $SK = h(h(r_u \parallel T_1) \parallel r_s \parallel h(V_i \parallel EID_i))$ contains an indispensable part from U_i (secret parameter r_u) and MS_j (secret parameter r_s); thus, our scheme meets this well-defined attribute.

4.2.3. Forward Secrecy

Forward secrecy holds if the past built session keys are still secure on the condition that the long-term secret holds; i.e., GWN 's x is corrupted. As a matter of fact, suppose the following: the adversary knows x , they can obtain PID_i from the open channel and then compute $V_i = h(PID_i \parallel x)$, and they can obtain $h(r_u \parallel T_1)$. However, an important consideration is that they cannot retrieve r_s due to the difficulty of large number factorization in RSA [18]. That is, we can retain forward secrecy.

4.2.4. User Anonymity

User anonymity mainly comprises user identity protection, which prevents the adversary from obtaining the user's identity, and user un-traceability, which guarantees that the adversary cannot decide upon who the communicating user is, nor does it allow them to distinguish whether two instances of data interaction are from the same communicating user.

For the first form of identity protection, on the one hand, during the registration phase, U_i only submits A_0 to GWN ; thus, it cannot directly extract identity information for the adversary even if GWN could be destroyed. On the other hand, PID_i cannot be used to deduce the identity of the user during the authentication phase; thus, the adversary cannot capture the user's identity, ID_i . As for the un-traceability of another user, the randomness of

PID_i breaks the statistical property, which effectively confuses the adversary in determining whether two sessions are from the same communicating user.

4.2.5. Password-Guessing Attack

In this attack, the adversary tries to guess the password via the physical unclonable function (PUF), in which the PUF generates an inherently unclonable output for a given input. That is, the adversary prepares a guessed password PW_i^{gue} ; then, the PDA computes A_2^{gue} using the same operations as in the login phase. Even if A_2^{gue} may be equal to A_2 , “fuzzy keywords + honey words”, by inducing the modulus operation, cannot help the adversary in determining whether the guessed password is correct.

4.2.6. Body Area Sensor Node Impersonation Attack

This attack [25] gives the adversary, i.e., the legitimate inside user, an opportunity: The user could obtain the body area sensor node’s secret key x_j and create a faulty session key for the new physician. However, this attack makes no sense in our scheme. Factually, this adversary cannot extract this secret x_j from B_7, B_8 , and B_9 because they do not possess the secret value r_g of GWN. Therefore, the proposed protocol resists sensor node impersonation attacks.

4.2.7. De-Synchronization Attack

Generally, after the session key is established, U_i , GWN, and MS_j have no need to update any parameters; thus, the de-synchronization attack is impossible. In our proposed protocol, U_i needs to update the parameters for the next authentication. Then, U_i checks whether B_{12}^* is equal to B_{12} . Luckily, the checking operation can detect this attack in a timely manner. That is, the occasion in which $B_{12}^* \neq B_{12}$ holds implies that this attack interferes with the normal update of parameters, and the user only asks GWN to run the update operation of B_{12} again.

4.2.8. Replay Attack

The replay attack comprises the following: the adversary usually sends old messages to pass the verification of entities and re-computes the session key. However, in each session, U_i , GWN, and MS_j choose random numbers r , r_u , r_g , and r_s , respectively, to ensure the freshness and independence of exchanged messages. As a result, the adversary can neither calculate the correct session key based on the replayed message nor can they pass the authentication of U_i .

4.2.9. Privileged Insider Attack

In order to prevent the adversary (even corrupted GWN) from using privileged insider attacks and extracting the identity information of legitimate users during the registration phase, U_i only submits A_0 ($A_0 = h(ID_i \oplus PW_i \parallel r) \bmod n_0$), which encapsulates ID_i relative to GWN, rather than bare string ID_i , and the adversary cannot obtain the real ID_i .

4.2.10. Node Capture Attack

Even if it is possible to assume that the adversary has the node’s secret x_j value and retrieve r_g and $e_i, h(r_u \parallel T_1), h(V_i \parallel EID_i)$, this adversary cannot re-calculate session key SK unless they can effectively solve the difficulty of large number factorization in RSA [18] in obtaining another important value r_s .

4.2.11. Denial of Service (DoS) Attack

In the proposed scheme, even if the adversary may render BASN unavailable by replaying old messages B_3, B_4, B_5 , and T_2 repeatedly, BASN firstly verifies whether the time gap meets $|T_c - T_2| > \Delta T$ or not. If it does, BASN directly terminates this session. Furthermore, even if the adversary updates timestamp T_2 to obtain $|T_c - T_2| < \Delta T$, BASN also ignores this session due to the following verification failure of value B_5 , where B_5 can

The protocol shown in [11] falls short of achieving EC_7 , EC_8 , and EC_9 . Specifically, neither the physician nor the sensor node can authenticate the gateway, aligning with EC_7 . Also, the protocol of [11] is susceptible to user impersonation attacks, which is in line with EC_8 . Additionally, it falters in attaining forward security, corresponding to EC_9 .

For the protocol in [12], the server or GWN in [12] retains many more password-related parameters, which threatens the security of passwords (EC_5). As for EC_7 , mutual authentication (EC_7) cannot be met because the messages do not guarantee that BASN can realize mutual authentication with respect to the GWN. Additionally, the users in [12] directly submit their bare identities to the GWN or the registration center in order to complete the registration phase, and once the gateway is corrupted by the adversary, the anonymity (EC_1) of the user will not be respected.

Only our proposal fulfills all the stipulated security prerequisites. It is evident that our proposal exhibits resistance against known attacks, enabling the attainment of optimal security and usability objectives. Notably, since no smart card has been used in our proposal, our proposal can meet criterium EC_{10} naturally.

5.3. Storage, Communication, and Computation Cost Comparisons

In order to provide a comprehensive evaluation of storage and communication overheads, Table 4 provides reasonable reference lengths for all components.

Table 4. The lengths of all terms involved in storage and communication costs.

Symbols	Bits	Symbols	Bits
Module n_0	32	ECC point p	160
Counter c	32	Hash value h	160
Threshold value t	16	Secret key value x	160
Timestamp T	32	Random/once r	160
User's/BASN's identity	128	Symmetric ciphertext size enc	256
BASN's identity set $S_M/S_M^{U_i}$	32	Public reproduction parameter τ_i	128

Simultaneously, to ascertain computational costs during the login and verification phases, we executed the RSA algorithm with a key length of 1024 bits on a 12th-generation intel core i7-12700 H with 16 G memory; we report that the elapsed time with respect to 1024-bit RSA modular exponentiation is 0.63 ms. For other cryptographic functions, based on the results from [10–12,28,29], the time required for the SHA-1 hash function is 0.00069 ms [28], the PUF function requires 0.43 ms [29], and symmetric encryption/decryption and the bio-hash function demand 0.1303 ms and 0.01 ms, respectively [11]. ECC point multiplication requires 0.0018 ms [12], and the fuzzy extractor function and bilinear pairing require 2.226 ms and 5.811 ms, respectively [10].

Then, we provide Table 5, which presents a comparative analysis covering the storage, communication, and computational overheads consumed in all compared schemes.

Table 5. Storage, communication, and computation costs in the login and authentication phase.

Schemes	Ref.	Storage Cost: Bits			Communication Cost: Bit			Computation Cost: ms		
		U_i	GWN	BASN	U_i	GWN	BASN	U_i	GWN	BASN
Ali et al.	[10]	1328	288	2336	1056	800	1280	10.27	0.005	5.81
Aghili et al.	[11]	1057	322	128	1408	1856	352	0.16	0.14	0.003
Yao et al.	[12]	576	288	1024	1888	3488	2144	0.03	0.03	0.90
Ours	---	864	352	160	1920	3616	1664	3.28	0.20	0.64

In our protocol, our total storage overhead is the smallest at 1376 bits, and the storage overhead of Ali et al. [10] is the largest at 3952 bits. The storage overhead of each scheme

increases in the order of 1376 bits, 1507 bits, 1888 bits, and 3952 bits. Moreover, our proposal has obvious advantages in terms of storage overheads. As for communication overheads, the user, GWN, and BASN costs are 1920 bits, 3616 bits, and 1664 bits, respectively, with corresponding computation times of 3.28 ms, 0.20 ms, and 0.64 ms. It is evident that our solution boasts the lowest cumulative storage overhead compared to [10–12]. Simultaneously, the consumed times of the user and BASN in our scheme are 3.28 ms and 0.64 ms, respectively, which can reduce the user's and BASN's computation cost by 68.1% and 83.8% compared to the scheme reported in [10].

In summary, our proposal outperforms others in terms of optimal security, superior storage and communication efficiency, and competitively efficient computational overheads. Other schemes, to varying degrees, require improvements in terms of security, communication overheads, or computational overheads.

6. Conclusions

With respect to high-security-requirement WBAN scenarios, we first introduced the authentication model of WBANs. Then, based on the PUFs, we proposed a fine-grained user authentication and key agreement protocol for WBANs. The proposed protocol does not need to allocate smart cards for users, and it can provide fine-grained user authentication and authorization. In the final security and performance analysis, the proposed protocol demonstrates advantages in terms of overall performance, and it is expected to significantly improve the security, efficiency, and availability of user authentication in WBANs. Regarding future studies, we will concentrate on blockchain-based authentication schemes in order to avoid single-point failure in a centralized GWN.

Author Contributions: Validation, methodology, and writing—original draft, K.L.; writing—review and editing, Q.C. and G.X. (Guoai Xu); validation, G.X. (Guosheng Xu). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China (No.: 2022YFB3104400) and the Fundamental Research Funds for the Central Universities under Grant No.: 2023RC69.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. O'Donovan, T.; O'Donoghue, J.; Sreenan, C.; Sammon, D.; O'Reilly, P.; O'Connor, K. A context aware wireless body area network (BAN). In Proceedings of the 2009 International Conference on Pervasive Computing Technologies for Healthcare, London, UK, 1–3 April 2009; pp. 1–8.
2. Wazid, M.; Das, A.K.; Kumar, N.; Rodrigues, J. Secure Three Factor User Authentication Scheme for Renewable-Energy-Based Smart Grid Environment. *IEEE Trans. Ind. Inform.* **2017**, *13*, 3144–3153. [[CrossRef](#)]
3. Halperin, D.; Heydt-Benjamin, T.S.; Ransford, B.; Clark, S.S.; Defend, B.; Morgan, W.; Fu, K.; Kohno, T.; Maisel, W.H. Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses. In Proceedings of the 2018 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 18–22 May 2018; pp. 129–142.
4. Liu, H.B.; Wang, Y.; Liu, J.; Yang, J.; Chen, Y.; Poor, H.V. Authenticating Users through Fine-Grained Channel Information. *IEEE Trans. Mob. Comput.* **2018**, *17*, 251–264. [[CrossRef](#)]
5. Chatterjee, S.; Roy, S.; Das, A.K.; Chattopadhyay, S.; Kumar, N.; Reddy, A.G.; Park, K.; Park, Y. On the Design of Fine Grained Access Control with User Authentication Scheme for Telecare Medicine Information Systems. *IEEE Access* **2017**, *5*, 7012–7030. [[CrossRef](#)]
6. Wang, X.F.; Wang, L.; Li, Y.; Gai, K. Privacy-Aware Efficient Fine-Grained Data Access Control in Internet of Medical Things Based Fog Computing. *IEEE Access* **2018**, *6*, 47657–47665. [[CrossRef](#)]
7. Singh, D.; Wazid, M.; Singh, D.P.; Das, A.K.; Joel, R. Embattle the Security of E-Health System Through A Secure Authentication and Key Agreement Protocol. In Proceedings of the 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco, 19–23 June 2023; pp. 1130–1135.

8. O Gundoyin, S.O.; Kamil, I.A. PAASH: A Privacy-Preserving Authentication and Fine-Grained Access Control of Outsourced Data for Secure Smart Health in Smart Cities. *J. Parallel Distrib. Comput.* **2021**, *155*, 101–119. [[CrossRef](#)]
9. Benil, T.; Jasper, J. Blockchain Based Secure Medical Data Outsourcing with Data Deduplication in Cloud Environment. *Comput. Commun.* **2023**, *209*, 1–13. [[CrossRef](#)]
10. Ali, Z.; Ghani, A.; Khan, I.; Chaudhry, S.A.; Islam, S.H.; Giri, D. A robust authentication and access control protocol for securing wireless healthcare sensor networks. *J. Inf. Secur. Appl.* **2020**, *52*, 2020. [[CrossRef](#)]
11. Aghili, S.F.; Mala, H.; Shojafar, M.; Peris-Lopez, P. LACO: Lightweight three-factor authentication, access control and ownership transfer scheme for e-health systems in IoT. *Future Gener. Comp. Syst.* **2019**, *96*, 410–424. [[CrossRef](#)]
12. Yao, H.L.; Yan, Q.; Fu, X.B.; Zhang, Z.; Lan, C. ECC-based lightweight authentication and access control scheme for IoT E-healthcare. *Soft Comput.* **2022**, *26*, 4441–4461. [[CrossRef](#)]
13. Wang, D.; Li, W.T.; Wang, P. Measuring Two-Factor Authentication Schemes for Real-Time Data Access in Industrial Wireless Sensor Networks. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4081–4092. [[CrossRef](#)]
14. Dolev, D.; Yao, A.C. On the Security of Public Key Protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [[CrossRef](#)]
15. Wang, C.Y.; Wang, D.; Tu, Y.; Xu, G.; Wang, H. Understanding node capture attacks in user authentication schemes for wireless sensor networks. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 507–523. [[CrossRef](#)]
16. Kumar, S.S.; Guajardo, J.; Maes, R.; Schrijen, G.J.; Tuyls, P. The Butterfly PUF: Protecting IP on Every FPGA. In Proceedings of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, Anaheim, CA, USA, 9 June 2018; IEEE: Piscataway, NJ, USA, 2008; pp. 67–70.
17. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Proceedings of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, Anaheim, CA, USA, 9 June 2018; pp. 523–540.
18. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [[CrossRef](#)]
19. Daemen, J.; Rijmen, V. *AES Proposal: Rijndael*; Technical Report; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2003.
20. Xie, Q.; Liu, D.; Ding, Z.; Tan, X.; Han, L. Provably Secure and Lightweight Patient Monitoring Protocol for Wireless Body Area Network in IoHT. *J. Healthc. Eng.* **2023**, *2023*, 4845850. [[CrossRef](#)]
21. Wu, F.; Li, X.; Xu, L.; Vijayakumar, P.; Kumar, N. A Novel Three-Factor Authentication Protocol for Wireless Sensor Networks with IoT Notion. *IEEE Syst. J.* **2021**, *15*, 1120–1129. [[CrossRef](#)]
22. Wang, D.; Wang, P. Two birds with one stone: Two-factor authentication with security beyond conventional bound. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 708–722. [[CrossRef](#)]
23. Wang, D.; Cheng, H.B.; Wang, P. Zipf’s Law in Passwords. *IEEE Trans. Inf. Forensic Secur.* **2017**, *12*, 2776–2791. [[CrossRef](#)]
24. Wang, D.; Zou, Y.K.; Dong, Q.Y. How to Attack and Generate Honeywords. In Proceedings of the 43rd IEEE Symposium on Security and Privacy (IEEE S&P), San Francisco, CA, USA, 23–25 May 2022; pp. 489–506.
25. Zou, S.H.; Cao, Q.; Wang, C.Y.; Huang, Z.; Xu, G. A Robust Two-Factor User Authentication Scheme-Based ECC for Smart Home in IoT. *IEEE Syst. J.* **2021**, *16*, 4938–4949. [[CrossRef](#)]
26. Qiu, S.M.; Wang, D.; Xu, G. Practical and Provably Secure Three-Factor Authentication Protocol Based on Extended Chaotic-Maps for Mobile Lightweight Devices. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 1338–1351. [[CrossRef](#)]
27. Wang, D.; Gu, Q.; Cheng, H. The request for better measurement: A comparative evaluation of two-factor authentication schemes. In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, Xi’an, China, 30 May–3 June 2016.
28. Wang, C.Y.; Wang, D.; Xu, G.; He, D. Efficient Privacy-Preserving User Authentication Scheme with Forward Secrecy for Industry 4.0. *Sci. China Inf. Sci.* **2022**, *65*, 112301. [[CrossRef](#)]
29. Kumar, D.; Jain, S.; Khan, A.; Pathak, P.S. An improved lightweight anonymous user authenticated session key exchange scheme for Internet of Things. *J. Am. Intell. Hum. Comp.* **2020**, *14*, 5067–5083. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.