applied sciences

MDPI

Article

An Integrated Two-Stage Medical Pre-Checkup and Subsequent Validation Key Agreement Authentication Mechanism

Tsung-Hung Lin * and Ming-Te Chen

Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 41170, Taiwan; mtchen@ncut.edu.tw

* Correspondence: duke@ncut.edu.tw

Received: 30 January 2020; Accepted: 6 March 2020; Published: 10 March 2020

Abstract: In the global village era, several competitions require pre-checkups for the participants who are qualified to participate that must be passed before the competition, so the accuracy of the checkup data must be confirmed and must not be leaked or tampered with. This is a new challenge to the accuracy of medical checkups data in the information and communication era. How to protect the rights of participants and the non-repudiation of participants are the main issues of this study. We have designed a two-phase user identity embedding and authentication scheme for pre-checkups and subsequent validations. A participant's private key is added to the physical examination data, and the identity of the examinations data is confirmed by the contestant before the competitions. Our work integrates lightweight Exclusive-OR (XOR) operations, fuzzy extractor biometric personal passwords, and a fixed-length hash operation accords with post-quantum operations to solve the problem of two-stage medical pre-checkup and subsequent validation key agreement authentication. The random oracle authentication mechanism proves the security of the protocols, and the security analysis proves that the protocols can resist the vulnerability attacks.

Keywords: random oracles; key agreement; authentication; pre-checkup

1. Introduction

With the efforts of countries around the world, advances in communications and technology are obvious to all. With the development of electronics and mobile technology, communication technology has evolved into a portable ubiquitous generation. Especially driven by Industry 4.0, cloud computing has become one of the best choices for data processing and storage. The sensing and monitoring data of the Internet of Things is also continuously transmitted and stored in the devices of cloud computing. Physical examinations or doctor consultations can use telemedicine in addition to requiring patients to go to the hospital in person. The storage of patient medical records has also changed from manual handwriting to electronic input, and from various hospitals to cloud storage systems. Instant messaging data for telemedicine consultations, or instant messaging data stored and read from cloud servers, may expose private data or be tampered with by malicious people. Therefore, many security technologies with cloud computing as the core issue have been proposed, such as several different protocols provided in the literature [1–5].

In such a global village era, several competitions require pre-checkups for the participants, and must be passed prior to being entitled to participate in competitions, so the accuracy of the checkup data must be confirmed and must not be leaked or tampered with. In addition, issues such as the adoption of a specimen related to the law and the ex-post evidence, the process of taking the specimen must be confirmed by the party. However, most of the studies on communication security focus on a

one-time conference for symmetric key agreement. Therefore, how to design a mechanism that can be used for two-phase communication to apply to the two phases of specimen acquisition and subsequent validation has become our main research focus. The key design of this two-phase communication is that, before the second meeting, the required messages for the second conference cannot be derived from the information transmitted during the first conference communication. This is similar to the post-quantum era. Even if the computing speed of computers increases, the calculation functions cannot be cracked quickly.

The purpose of this study is to design an integrated two-phase pre-checkup and subsequent validation key agreement authentication mechanism (TCVK), in the use of regular checkups records server (RCRS) for protecting the security of participants' records and making sure that their data are not lost or tampered with. The process is constructed by three main roles—participants, RCRS, and checkups stations. The records of the participants are collected and processed through these checkup stations, and then encrypted by participants. Finally, each participant could enter the system and check records simply using his own smart card and biometric. With the help of our protocols, the participants' authentication checking process will no longer be as time and effort consuming as before and the cost can be reduced drastically. Besides, our solution may avoid the disputes happening in participant subsequent validation process and dispel the suspicion of unfair judges. The scenario of our TCVK is indicated in Figure 1.

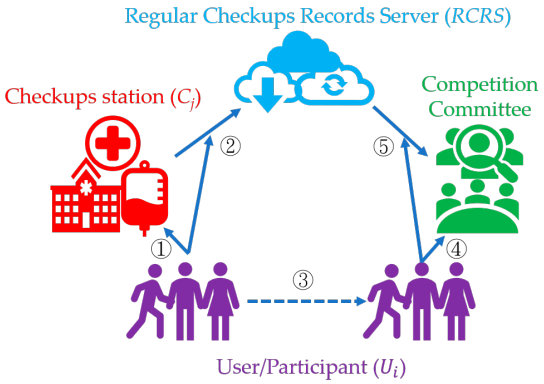

Figure 1. A scenario of our proposed two-phase pre-checkup and subsequent validation key agreement authentication mechanism (TCVK).

The study of our integrated two-phase TCVK is organized in the following manner. Section 2 describes the related works. Section 3 shows the proposed schemes and Section 4 shows random oracle model security proof of the proposed protocols. Section 5 makes a security analysis of the proposed scheme. Section 6 states performance comparisons of the proposed scheme. Finally, our conclusion is written in Section 7.

2. Related Work

In the era of telematics and telemedicine, a large amount of sensitive information is being transmitted in a public network environment. To protect the information privacy of these transmissions, much information security research has proposed various solutions. These schemes include symmetric encryption [6–9] and asymmetric encryption. Owing to the high maintenance cost required by the asymmetric public key method, some symmetric encryption systems have been popularly used recently [10–12]. The symmetric encryption must generate the shared key required for the conference immediately, as the zero-knowledge-based key exchange mechanism came into being. In the key exchange process, it must be ensured that the transfer messages are not tampered with or resented. Therefore, some researchers use the timestamp mechanism to ensure the security of the communication using the irreversibility of time [13–15]. However, some researchers believe that the computer's

timestamps cannot guarantee consistency. The computer's computing speed is very fast, and a tiny time error will cause the key agreement to fail. As a result, many scholars have started to use one-time random numbers instead of timestamps [10–12].

Traditional public key cryptosystems (TPKCs) and elliptic curve cryptosystems (ECCs) are key exchange systems commonly used by many researchers. Recently, in order to reduce the computational cost of communication preparation and the advent of the post-quantum era, many researchers have focused on lightweight and time-independent security. Therefore, key exchange mechanisms using hash functions and Exclusive-OR (XOR) operations have been proposed. Lightweight certification can achieve better execution efficiency than TPKC and ECC designs, so it has become a new design requirement. Instead of the traditional password directly input method, a more secure and unique biometric authentication method is often used for offline key verification in recent works [16–18].

Recently, the fuzzy extractor has replaced the hash function corresponding to a single result of the dynamic range, which is a biometric tool for user recognition and inspection process [19–21]. The fuzzy extractor allows users to use their biometric characteristics as keys. When users enter their biometric characteristics into the extractor, the extractor will use the generation algorithm Gen $(B_i) = (X_B, P_B)$ to randomly generate a fixed string of words (Gen for generate and Rep for reproduce), where secret key X_B is a word string and extracts a public key P_B that can be stored. An example of a biometric fuzzy extractor is shown in Figure 2. Our study also uses a combination of smart cards and participant certifications. If it equipped with powerful Central Processing Unit, Random Access Memory, and Input / Output device a smart card can deal with more data processing tasks. The uniqueness of a participant's identification can be confirmed using the computing system installed in the smart cards [22].

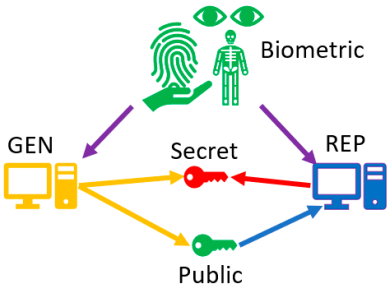

Figure 2. An example of biometric fuzzy extractor. Gen, generate; Rep, reproduce.

In the algorithm design of secure information communication protocols, in order to prove that the algorithms used to exchange messages in the public network are secure, researchers usually use probabilistic assumptions and verification. In 1993, Mihir Bellare and Phillip Rogaway (1993) first published a rigorous method of cryptographic proof using mathematical abstraction random oracles [23]. The main issue is to strengthen the proof using random oracles when weak password assumptions cannot be used to prove the password hash function. In contrast to the security in the standard cryptographic model, each hash function is replaced with a random oracle in the random oracle model to prove that the system is secure. A function mapping each possible query to a fixed random response from its output domain—that is, a mathematical function is chosen uniformly at random—is a random oracle [23].

Lin [24] designed a special medical examination case for athletes in 2019. When athletes need a physical examination before the game, they need to go to the on-site checkpoint for a physical examination, and then report to the competition committee within a limited time period. According to Lin's protocol, a malicious person can calculate a session key by combining multiple transmissions. The information in the communication will also be intercepted and eavesdropped by the malicious personnel, which will cause the confidential information of the athletes to be leaked by the malicious personnel and tamper with the medical examination data [24]. Additionally, this method uses a fixed

block string and then generates the corresponding value from the one-type single hash function. It can also be maliciously attacked by a birthday attack or a meet-in-the-middle attack. Another disadvantage of Lin's design approach is that online verifications must be performed with the medical checkpoints of the physical examination of many players at the same time before the game [24].

Our study avoids the possible disadvantages of Lin's method [24] and integrates lightweight XOR operations, fuzzy extractor biometric personal passwords, and a fixed-length hash operation accords with post-quantum operations to solve the problem of two-stage medical examination data protection and future data verification. The random oracle authentication mechanism proves the security of the protocols, and the security analysis proves that the protocol can resist the vulnerability attack.

3. Our TCVK Mechanism

Our proposed TCVK scheme is composed of four phases: user/participant (U_i) registration phase, checkups stations (C_j) registration phase, pre-checkups phase, and subsequent validation phase. There are three roles, namely, user/participant (U_i), regular checkups records server (RCRS), and checkups station (C_j) that will be introduced into our scenario.

3.1. User/Participant Registration Phase

In this phase, a participant U_i registers to the regular checkups records server (RCRS) in the first-time registration. Each U_i possesses a smart card that includes a configured identity ID_i from *RCRS* and an ex-factory number r_i. Then, U_i performs the following steps to complete the user's registration work.

S1 U_i imprints biometric B_i on the sensor device of *RCRS*. Then, *RCRS* computes $\mathrm{Gen}(B_i) = (X_B, P_B)$, where Gen(.) is a generating function of the fuzzy extractor and (X_B, P_B) are secret key and public key tuple, as illustrated in Figure 2. Additionally, *RCRS* computes $UA_i = \mathrm{h}(ID_i \oplus X_B)$, $UC_i = UA_i \oplus \mathrm{h}(r_i \oplus X_s)$, $D_i = r_i \oplus ID_i \oplus X_s$, and $E_i = UA_i \oplus \mathrm{h}(ID_i \oplus X_s)$, where h(.) is a one-way hash function. Then, *RCRS* stores $\{ID_i, D_i\}$ in *RCRS*'s database for verification later. To compute $UX_i = UA_i \oplus D_i$, and then stores $\{ID_i, UX_i, UC_i, E_i, \mathrm{Rep}(.), \mathrm{h}(.)\}$ on the smart card of U_i.

3.2. Checkups Station Registration Phase

Checkups station (C_j) must be registered as a valid *RCRS* member before examinations.

S1 Checkups station makes a request $\{ID_{Cj}\}$ to the *RCRS* via a secure channel.

S2 When *RCRS* has received the request, it creates a ID_{Cj} for C_j and computes the share token $RCC_j = \mathrm{h}(ID_{Cj} \oplus X_S)$ with ID_{Cj} using *RCRS*'s secret key X_S. Then, it creates a key X_{RCCj} for ID_{Cj} and then stores tuple $\{(ID_{Cj}, X_{RCCj} \oplus X_S)\}$ in its database. Finally, it forwards $\{ID_{Cj}, RCC_j, X_{RCCj}\}$ to C_j.

S3 When the checkups station C_j receives the tuple $\{ID_{Cj}, RCC_j, X_{RCCj}\}$ from *RCRS*, it keeps this sec ret.

3.3. Pre-Checkups Phase

When a registered user U_i attempts to forward physical examination records to *RCRS*, it must be authenticated by the checkups station C_j first. After successfully completing the key agreement among U_i, C_j, and *RCRS*, U_i can forward the encrypted medical examination records to *RCRS* via C_j.

S1 U_i inserts the smart card into the card reader and imprints biometric B_i. Then, it retrieves $\mathrm{Rep}(B_i, P_B) = X_B$ and computes $UA_i = \mathrm{h}(ID_i \oplus X_B)$.

S2 U_i produces a random nonce number R_U and computes $M_0 = \mathrm{h}\left(R_U \oplus UA_i \oplus UC_i\right)$ and $M_1 = R_U \oplus UA_i \oplus E_i$. Finally, U_i forwards intermediate messages $\{ID_i, M_0, M_1\}$ to C_j by public channel.

S3 While the C_j receives $\{ID_i, M_0, M_1\}$ from U_i, C_j selects a random nonce number R_C and computes $M_2 = R_C \oplus RCC_j$ and $M_3 = \mathrm{h}(ID_{Cj} \oplus R_G)$. Furthermore, C_j sends $\{M_0, M_1, M_2, M_3, ID_{Cj}, ID_i\}$ to *RCRS* by public channel.

S4 After RCRS receives the message $\{M_0, M_1, M_2, M_3, ID_{Cj}, ID_i\}$ from C_j. RCRS first retrievals D_i from RCRS's database using $\mathrm{ID_i}$and computes $r_i = D_i \oplus ID_i \oplus X_s$, $R_U = M_1 \oplus \mathrm{h}(ID_i \oplus X_s)$, and $R_C = M_2 \oplus \mathrm{h}(ID_{Cj} \oplus X_s)$. Then, RCRS checks whether M_0 is equal to $\mathrm{h}(R_U \oplus \mathrm{h}(r_i \oplus X_s))$ and whether M_3 is equal to $\mathrm{h}(ID_{Cj} \oplus R_C)$. If the above are valid, the RCRS continues to deal with the requisition. On the contrary, the session process aborted.

S5 *RCRS* produces a random nonce number R_R and computes $SK = \mathrm{h}(R_U \oplus R_C \oplus R_R)$, $M_4 = R_R \oplus R_U \oplus \mathrm{h}(X_{RCCj} \oplus X_S)$, $M_5 = \mathrm{h}(SK \oplus M_4)$, and $M_6 = R_C \oplus \mathrm{h}(ID_i \oplus X_s)$. Then, *RCRS* sends $\{M_4, M_5, M_6\}$ back to C_j. *RCRS* now owns the session key *SK* for this key agreement.

S6 If C_j has received the message $\{M_4, M_5, M_6\}$ from *RCRS*, then G_j retrieves $X_{RCCj} \oplus X_S$ from database using ID_{Cj}. C_j computes $SK = \mathrm{h}(M_4 \oplus \mathrm{h}(X_{RCCj} \oplus X_S) \oplus R_C)$ and $M_5^* = \mathrm{h}(SK \oplus M_4)$. C_j will verify whether or not M_5^* equals M_5. If both are the same, then mutual authentication and session key agreement are completed. On the contrary, the session process is aborted. C_j now owns the session key *SK* for this key agreement.

By the above steps, the key agreement process has finished and the secure tunnel between C_j and *RCRS* is created. Then, C_j produces an encrypted examination record UR_i from the checkups station. C_j computes $M_7 = RCC_j \oplus R_C \oplus M_6 = RCC_j \oplus \mathrm{h}(ID_i \oplus X_s)$. C_j encrypts the UR_i using $\mathrm{M_7}$ to form the $M_8 = \mathrm{E}_{M_7}(UR_i)$ and sends $\{M_8\}$ to RCRS by this secure session tunnel. Then, *RCRS* stores encrypted examination data of U_i to the database.

3.4. Subsequent Validation Phase

The user U_i has to pass subsequent validations to connect to the pre-checkup records, and then the competition committee will subsequently either validate U_i or not. The proposed scheme is carried out in the following steps.

S1 U_i inserts the smart card into the card reader and imprints biometric B_i to the fuzzy extractor. Then, it computes $\mathrm{Rep}(B_i, P_B) = X_B^*$, $UA_i = \mathrm{h}(ID_i \oplus X_B^*)$.

S2 U_i produces a random nonce number R_U, and computes $N_0 = \mathrm{h}(UX_i \oplus UA_i \oplus ID_i)$ and $N_1 = R_U \oplus UA_i \oplus UR_i$. Then, U_i sends $\{N_0, N_1, ID_i\}$ to *RCRS* via public channel.

S3 When the RCRS received the message $\{N_0, N_1, ID_i\}$ from U_i, RCRS could find D_i in the database using ID_i. Then, whether or not N_0 equals $\mathrm{h}(D_i \oplus ID_i)$ is checked using $UA_i = UX_i \oplus D_i$. If both are the same, then RCRS retrieves $r_i = D_i \oplus ID_i \oplus X_s$ and $R_U^* = N_1 \oplus \mathrm{h}(r_i \oplus X_s)$. On the contrary, the session process is aborted.

S4 RCRS produces a random nonce number R_R and computes $N_2 = R_R \oplus \mathrm{h}(r_i \oplus X_s)$ and $N_3 = \mathrm{h}(SK \oplus R_U)$. After preparing them, RCRS sends $\{N_2, N_3\}$ to U_i. Then, *RCRS* gets the session key $SK = \mathrm{h}(R_U \oplus R_R)$.

S5 After U_i received message $\{N_2, N_3\}$, U_i retrieves $R_R = N_2 \oplus (UA_i \oplus UR_i)$. U_i gets session key $SK* = \mathrm{h}(R_U \oplus R_R)$ and then checks whether or not $N_3 = \mathrm{h}(SK \oplus R_U)$. If both are the same, then the session key agreement process has finished and mutual authentication is built. On the contrary, the session process is aborted. By the above steps, the key agreement process has finished and the secure tunnel is built.

S6 When the RCRS has received $\{ID_i, ID_{Cj}\}$, it computes $N_4 = \mathrm{h}(ID_{Cj} \oplus X_s) \oplus \mathrm{h}(ID_i \oplus X_s)$. RCRS decrypts UR_i using N_4, where $M_7 = \mathrm{h}(ID_{Cj} \oplus X_S) \oplus \mathrm{h}(ID_i \oplus X_s) = N_4$.

4. Random Oracles Proof for the Security of Our Protocols

If the function output requires a strong randomness assumption, random oracles can be used as an ideal alternative to the cryptographic functions. We employ some security definitions in the following proposed scenarios and proofs.

Definition 1. *Partner.*

We will define the partner functions here. First, we suppose that each player p_i has its corresponding instance $\Pi^k{}_i$ in the k-th session, where $i \in I$, $I \in \{U, RCRS\}$, and $k \in N$. Besides, we also assume that the player p_i's partner is the j's instance, where $j \in I$, $I \in \{U, RCRS\}$ and $k \in N$. From above definitions, we also defined what the partners are if each of them satisfied the following definitions.

1. *p_i's session is equal to p_j's session in the k-th session, that is, $ssid^k{}_i = ssid^k{}_j$.*
2. *Each partner's instance is matched the corresponding partner's instance, that is, p_i's instance $\Pi^k{}_i \equiv \Pi^k{}_j$.*

Definition 2. *Queries.*

*In the following, we give some definitions about query types that an attacker could use to make this request to ask the simulator respectively. By the way, we also model that the attacker's ability may control all communication during the simulation of the pre-checkups phase and subsequent validations phase of the proposed scheme. We defined in a "**Game**" that an attacker could ask query types as follows.*

1. *Send(i, k, M) (or Send(j, k, M))query: an attacker could impersonate some player and forward the message M to the instance $\Pi^k{}_i$ in the k-th session, where $i \in I$ and $k \in N$.*
2. *Reveal(i, k) (or Reveal(j, k)) query: an attacker could obtain the session key from the instance $\Pi^k{}_i$ in the k-th session, where $i \in I$ and $k \in N$.*
3. *Corrupt(i) (or Corrupt(j)): the instance $\Pi^k{}_i$'s secret key is exposed to the attacker.*
4. *Test(i, k) (or Test(j, k)): an attacker could guess the real session key with non-negligible advantage. If the attacker makes this type of query to the simulator, then the simulator could make a coin flipped by b. If b equals to 1, the simulator will output real session key $SK_{i,j}{}^k$ in the k-th session, where i, $j \in I$, and $k \in N$. Otherwise, it gives the random string chosen from $\{0,1\}^*$ to the attacker. Then, the attacker has to guess whether or not the session key is the real one. Besides, the attacker only could be allowed to make this type of query to the "fresh" instance of each player.*

Definition 3. *Freshness.*

If the following situations occur, an instance $\Pi^k{}_i$ is "fresh".

1. *$\Pi^k{}_i$ owns the session key and the attacker does not query the player $\Pi^k{}_i$ who is p_j's instance, Reveal(i, k).*
2. *If there is a player p_j, its instance and partner are both $\Pi^k{}_i$. Then, none of the attackers query the p_j and $\Pi^k{}_i$ that owns the same session key, Reveal(j, k).*
3. *An insider attacker created by the opponent cannot be for player i or j, where $\{i, j\} \in I$ and $I \in \{U, RCRS\}$.*

Definition 4. *Forward Secure (FS).*

In our proposed scheme, we define that our scheme satisfied "forward secure" if there exists an attacker that could not guess the session key successfully with a non-negligible advantage with both instances in which they were asked the corrupted queries (i.e., Corrupt(i) or Corrupt(j)).

Theorem 1. *We assume that there exists h to be a hash function that satisfies the random oracle (RO) assumptions. Then, we claim that our proposed scheme (AD) is a user authentication scheme with forward secure (FS), that is, if AD is forward secure, then*

$$Adv^{FS}_{AD,\,A,\,C}(\theta,\ t'') \le \left(I^2 q_h 2^{3l}\left(Adv_{PC,\,h,\,RO}(\theta,\ t)\right)\right) + \left(I^2 q_h 2^{2l}\left(Adv_{SV,\,h,\,RO}(\theta,\ t')\right)\right)$$

where t′ is the maximal game time including an attacker perform its own execution time in the subsequent validations (SV) phase, t is the maximal game time including an attacker distinguish the real session key in the pre-checkups (PC) phase, t′′ is the maximal game time in the above phases, I is the upper bound of the number of players, θ is the security parameter of the proposed scheme, Z_n^* is the l-bit length prime number filed, and q_h is the upper bound of hash query number in the above game.

Proof. In the beginning, we consider that there exists an attacker A that attempts to attack our proposed scheme (AD) against the forward secure in the above definition. Then, we defined that the following equation will hold:

$$Pr[b = b'] \leq Pr[b = b' \text{ in the Pre checkups Phase }] + Pr[b = b' \text{ in Subsequent validations phase}]$$

where b and b' are the coin *flips* chosen by the simulator and the attacker, correspondingly. □

Then, we consider the above two situations in the following cases.

1. In the pre-checkups phase.

In this pre-checkups (PC) phase, we assume that there exists an attacker, D, whose job is to distinguish the real session key in this phase. The simulator that we assume to be A begins to prepare system parameters including the instance of players $\{i, j\} \in I$ and $I \in \{U, RCRS, C\}$ in the k-th session, where C is the checkups station in this phase with the $k \in N$ under the security parameter θ.

- After preparing the above parameters for building the environment, A also prepares the above query types in order to respond to D's query. Before the simulation starts, A also generates the corresponding key pairs for each player $\{i, j\} \in I$ and $I \in \{U, RCRS, C\}$, where C is the checkups station. The following are the simulation steps.
- In the beginning, D would make a *Send* (i, k, ID_i) query to the A. When A has received this type of query, it forwards to the hash oracle and the hash oracle has to compute the UA_i with the secret key's help X_B, that is, $UA_i = h(ID_i \oplus X_i)$. A also prepares the hash oracle simulation of each message in this pre-checkups phase. The hash oracle would record the tuple $(i,\ ID_i,\ UA_i,\ M_0, M_1, k)$ in the k-th session.
- In the checkups station, the simulator also records the communication message $\left(j,\ M_0, M_1, M_2, M_3, ID_{Cj}, ID_i,\ k,\ R_C\right)$. From the above message simulation, we could see that A would be able to handle this query type with the help of random oracle and the secret key.
- If D makes a Reveal(i) query, A could reply to D according to the secret key X_i generated in the beginning. In order to compute whether the session key of A is the desired one, A also asks the random oracle to generate the hash value of R_U and R_R from a random oracle. However, A does not know the real value of R_U and R_R. After receiving the hash value from A, D could compute $SK_{i,j}^t$ by assigning the received hash value, where $t \neq k \in N$ and $\{i, j\} \in I$ and $I \in \{U, RCRS, C\}$.
- After the above query training, A makes the Test(i) query to the simulator D. We assume that A has chosen some instances to attack that $i = i^*$ and $j = j^*$ in the k-th session. In this time, D starts to coin flip to output b. If b is 1, the simulator generates the real session key $SK_{i^*,j^*}^k = h(R_U \oplus R_R \oplus R_C)$, where $R_U,\ R_R$, and R_C are random numbers in the Z_n^* with *l-bit* length and $\{i^*,\ j^*\} \in I$ and $I \in \{U, RCRS, C\}$. Otherwise, A outputs the random string from $\{0, 1\}^*$. When D has received the tuple from A, its work is to distinguish whether or not this tuple is a real session key.

We assume that if the attacker D could distinguish this tuple with a non-negligible advantage $Adv_{AD,\,h,\,RO}(\theta,\,t)$. Then, the following equation will hold.

$$\begin{aligned}
&Adv_{CD,\,h,\,RO}(\theta,\,t)\\
&\geq Pr[D(Z_n^*, PC,\,h,\,RO)=1\Big|SK^k_{i^*,j^*}=h(R_U\oplus R_R\oplus R_C)\Big]\\
&\quad -\Pr\Big[D(Z_n^*, PC,\,h,\,RO)=1\Big|SK^k_{i^*,j^*}\leftarrow\{0,1\}^*\Big]\\
&\geq \frac{1}{I^2q_h2^{3l}}Pr[A(\cdot)=1\Big|SK^k_{i^*,j^*}\ \textit{is real in the Test query}\ ]\\
&\quad -\Pr\Big[A(\cdot)=1\Big|SK^k_{i^*,j^*}\ \textit{is a random string in the Test query}\Big]\\
&\geq \frac{1}{I^2q_h2^{3l}}(Pr[b=b'\textit{in the}\ \text{Pre checkups phase}]\\
&\quad -Pr[b\neq b'\textit{in the}\ \text{subsequent validations phase}])\\
&\geq \frac{1}{I^2q_h2^{3l}}(2(Pr[b=b'\textit{in the}\ \text{Pre checkups phase}])-1).
\end{aligned}$$

Finally, the following equation will hold

$$Pr[b=b'\textit{in the}\ \text{Pre checkups phase}]\leq (I^2q_h2^{3l}\Big(Adv_{PC,\,h,\,RO}(\theta,\,t)\Big)+1)/2.$$

2. In the subsequent validation phase.

In this subsequent validation (SV) phase, we consider the following situation. In this phase, we assume that there is an attacker C whose job is to distinguish the real session key after gathering enough training information. The simulator that we assume to be F begins to prepare system parameters including the instance of players $\{i, j\} \in I$ and $I \in \{U, RCRS\}$ in the k-th session, where $k \in N$ under the security parameter θ and each player's key pair. The attacker C could also make queries as follows.

- Send (i, k, ID_i) query: When the attacker makes the send query to the simulator F, F will prepare the (i, ID_i) for the further simulation usage. Then, F forwards (i, ID_i) to the attacker C.
- Hash query (i, k, ID_i): When the attacker makes the hash query of instance $\Pi^k{}_i$ with the ID_i. The simulator F will prepare the random oracle to reply to the result UA_i to C, where UA_i is computed from random oracle with the help of ID_i and the instance's secret key X_i.
- Reveal(i) query: If C makes a Reveal(i) query, F could reply to C according to the hash value (i, $h(R_U \oplus R_R)$, R_U, R_R, k), where R_U and R_R are random numbers in the Z_n^* with l length bits and they are chosen by player U_i and $RCRS$ in the k-th session, respectively.
- Corrupt(i) query: If C makes a Corrupt(i) query, F could reply to C according to the secret key value X_i.
- Finally, if C makes a Test(i) query to F, then F prepares in the following. First, we assume that the instance $i = i'$ and the instance $j = j'$ in the kth session are chosen by attacker C, where each of them is a fresh instance of player, respectively. In this time, F also prepares the session key to respond to the attacker C. It depends on the coin flips by the simulator F with the output b. If b is 1, then F computes $SK^k_{i',j'} = h(R_U \oplus R_R)$, where R_U and R_R are random numbers and $\{i',\ j'\} \in I$ and $I \in \{U, RCRS\}$. Otherwise, F outputs a random string from $\{0, 1\}^*$. When C has received the tuple from F, its work is to distinguish whether this tuple is real session key or not.

We assume that the attacker F could distinguish this tuple with a non-negligible advantage $Adv_{SV,\,h,\,RO}(\theta,\,t')$. Then, the following equations will hold.

$$\begin{aligned}
&Adv_{SV,\,h,\,RO}(\theta,\,t')\\
&\geq Pr\Big[C(Z_n^*, SV,\,h,\,RO)=1\Big|SK_{i^*,j^*}^k=h(R_U\oplus R_R)\Big]\\
&\quad -\Pr\Big[C(Z_n^*, SV,\,h,\,RO)=1\Big|SK_{i^*,j^*}^k\leftarrow\{0,1\}^*\Big]\\
&\geq \tfrac{1}{I^2q_h2^{2l}}Pr\Big[F(\cdot)=1\Big|SK_{i^*,j^*}^k \textit{ is real one in the Test query }\Big]\\
&\quad -\Pr\Big[F(\cdot)=1\Big|SK_{i^*,j^*}^k \textit{ is a random in the Test query}\Big]\\
&\geq \tfrac{1}{I^2q_h2^{2l}}(Pr[b=b' \text{ in the Subsequent validations phase}]\\
&\quad -Pr[b\neq b' \text{in the Subsequent validations phase}]\\
&\quad \geq \tfrac{1}{I^2q_h2^{2l}}(2(Pr[b=b' \textit{in the } \text{Subsequent validations phase}])-1)
\end{aligned}$$

Finally, the following equation will hold

$$Pr[b=b' \textit{in the } \text{Subsequent validations phase}]\leq (I^2q_h2^{2l}\Big(Adv_{SV,\,h,\,RO}(\theta,\,t')\Big)+1)/2$$

From the above two cases, we summarize the attacker's advantage to break the system with the following equation.

$$\begin{aligned}
&Adv_{AD,\,A,\,C}^{FS}(\theta,\,t'')=Pr[b=b']\leq\ Pr[b\ =\ b' \text{ in the Pre checkups phase }]\\
&+\ Pr[b\ =\ b' \text{ in Subsequent validations phase}]\\
&Adv_{AD,\,A,\,C}^{FS}(\theta,\,t'')=Pr[b=b']\\
&\qquad \leq \Pr[b=b' \text{ in the Pre checkups phase}]\\
&\qquad +Pr[b=b' \text{in the subsequent validations phase}]\\
&Adv_{AD,\,A,\,C}^{FS}(\theta,\,t'')=Pr[b=b']\\
&\qquad \leq \Big(I^2q_h2^{3l}\Big(Adv_{PC,\,h,\,RO}(\theta,\,t)\Big)+1\Big)/2+(I^2q_h2^{2l}\Big(Adv_{SV,\,h,\,RO}(\theta,\,t')\Big)+1)/2\\
&Adv_{AD,\,A,\,C}^{FS}\Big(\theta,\,t''\Big)=Pr[b=b']\\
&\qquad \leq \Big(I^2q_h2^{3l}\Big(Adv_{PC,\,h,\,RO}(\theta,\,t)\Big)\Big)+(I^2q_h2^{2l}\Big(Adv_{SV,\,h,\,RO}(\theta,\,t')\Big))
\end{aligned}$$

5. Security Analysis

This section describes some well-known security defenses analyses for our proposed scheme.

5.1. Privileged Insider Attack

The RCRS secret key Xs is known only by the RCRS itself. In our proposed scheme, all participants, including checkups stations, have never shown their secret keys to others, and the proof in the previous section confirms that the keys cannot be derived from the communication process. Therefore, according to the definition of attack mode, we know that our TCVK scheme can indeed resist privileged insider attacks.

5.2. Perfect Forward Secrecy Attack

In our proposed scheme, the RCRS secret key Xs is not related to the RCRS session key SK. The session key is not created by the RCRS internal key Xs. In addition, session keys are randomly created by each legitimate participant from each session. Using the session key for a limited time and then encrypting each key with an irreversible hash function, the attacker cannot find the correct rules and guess the correct session key. That is, the key for each conference session is only related to a random number, and it is impossible to find the rules at any time. Therefore, we confirm that the proposed TCVK scheme provides perfect forward secrecy.

5.3. Checkups Station Impersonation Attack

If an attacker tries to impersonate C_j by transmitting a request message $\{ID_i, M_0, M_1\}$ to U_i and obtains message $\{M_4, M_5, M_6\}$, where $M_0 = \mathrm{h}(R_U \oplus UA_i \oplus UC_i)$, $M_1 = R_U \oplus UA_i \oplus E_i$, $M_4 = R_R \oplus R_U \oplus \mathrm{h}(X_{RCCj} \oplus X_S)$, $M_5 = \mathrm{h}(SK \oplus M_4)$, and $M_6 = R_C \oplus \mathrm{h}(ID_i \oplus X_s)$. In order to compute M_4 and M_5, the attacker must find R_U and R_R using a shared key X_{RCCj}, where $(R_U \oplus R_R) = M_4 \oplus h(X_{RCCj} \oplus X_S)$. In communication, if C_j is illegal, it will get the wrong value, and at that time, RCRS will immediately recognize and terminate this illegal authentication phase. From the above description, we confirm that the proposed TCVK scheme can resist the checkups station impersonation attack.

5.4. User/Participant Impersonation Attack

If an attacker tries to impersonate a legitimate user U_i by sending a request message $\{ID_i, M_0, M_1\}$, including $M_0 = \mathrm{h}(R_U \oplus UA_i \oplus UC_i)$, $UA_i = \mathrm{h}(ID_i \oplus X_B)$ and $UC_i = UA_i \oplus \mathrm{h}(r_i \oplus X_s)$, to the RCRS. Attackers cannot obtain user biometrics and cannot calculate $\mathrm{h}(r_i \oplus X_s)$. In addition, RCRS checks its ID_i through its own database to confirm its legitimacy. Illegal data will interrupt communication. From the above description, our proposed TCVK scheme can resist participant impersonation attacks.

5.5. Offline Password Guessing Attack

When a participant's smart card is lost or stolen, an attacker can try to brute force the owner's password to log in to the system. However, in this study, the smart card does not required entering or storing any password, it only needs biometric characteristics through fuzzy extraction, and does not directly store any private keys. Therefore, the attacker cannot obtain the biometric characteristics of the smart card owner via a smart card and will not be able to apply the fuzzy extractor to pass the password verification at any phase of our proposed scheme. Therefore, our proposed scheme can resist offline password guessing attacks.

5.6. Stolen Smart Card Attack

Similarly, when a participant's smart card is lost or stolen. An attacker can obtain information from the smart card, which only has $\{ID_i, UX_i, UC_i, E_i, Rep(.), h(.)\}$, where $UX_i = UA_i \oplus D_i$, $UC_i = UA_i \oplus \mathrm{h}(r_i \oplus X_s)$. Then, the attacker has to invert the value of UX_i or UC_i to obtain the secret value. However, because of the characteristics of the hash function, inverting the values of UX_i or UC_i is computationally unfeasible in the polynomial time. Hence, our proposed scheme can resist stolen smart card attacks.

5.7. Session Key Security

Similar to privileged internal attacks, the RCRS key Xs is known only by the RCRS itself. In our proposed TCVK scheme, the session key SK is related to the random number generated by each legitimate participant in the pre-checkups phase and subsequent verification phase. Owing to the

characteristics of the hash function, it is not feasible to calculate the session key SK in polynomial time to obtain a random value. Therefore, our proposed scheme provides session key security.

5.8. Man-in-the-Middle Attack

According to the definition of a man-in-the-middle attack, an attacker can disguise itself as a terminal, and each participant in the session cannot identify it as a real terminal. In fact, a man-in-the-middle attack is a mutual authentication attack. In our proposed scheme, RCRS saves the authentication data of legitimate users in a database and performs mutual identity verification. If an attacker tries to pretend to be a real terminal, then, without the RCRS authentication record, other steps cannot be performed and any information can be obtained. Therefore, our proposed TCVK scheme can resist man-in-the-middle attacks.

5.9. Tampering Attack

We assume that the user forwards the tampered message $\{ID_i, M_0, M_1\}$ during the pre-checkups phase. RCRS receives $\{ID_i, M_0, M_1\}$ and then obtains D_i by retrieving ID_i in the database. If RCRS cannot map the corresponding D_i, it will find that these messages have been tampered with. Additionally, RCRS will also check if M_0 is equal to $h(R_U \oplus h(r_i \oplus X_s))$, and then use M_1 to calculate M_0 to solve R_U. In other words, the attacker cannot reverse the real R_U by tampering with the message. In another case, the message $\{ID_{Cj}, M_2, M_3\}$ transmitted by the checkups station is tampered with and forwarded to the RCRS. RCRS will also check if M_3 is equal to $h(ID_{Cj} \oplus R_C)$ and then use M_2 to calculate M_3 to solve R_C.

Therefore, if users and checkups stations forward these tampered messages to RCRS, RCRS can check that these messages may have been tampered with by an attacker. We assume that RCRS then sends these tampered messages $\{M_4, M_5, M_6\}$ to the checkups stations during the verification phase, which will use ID_{Cj} and $X_{RCCj} \oplus X_S$ to look in the database to confirm whether or not M_5^* equals M_5. According to the above description, our proposed scheme can resist tampering attacks.

6. Performance Comparisons

This section shows a security analysis comparison among Ali et al.'s [25] scheme (Ali [25]) and Chen et al.'s [26] scheme (Chen [26]) compared with our proposed TCVK scheme. Functionality and performance comparisons are presented in the following.

6.1. Functionality Comparisons

This subsection shows functionality comparisons among Ali [25], Chen [26], and the proposed TCVK scheme in Table 1. Providing secure communication protocols is a consistent design goal for researchers. In this article, we replace timestamp annotations with one-time random numbers. Our method avoids time inconsistencies and prevents most common malicious attacks.

6.2. Efficacy Comparisons

This subsection demonstrates the efficiency comparisons of Ali [25], Chen [26], and the proposed TCVK scheme. Ali [25] applies symmetric encryption and decryption operations, and Chen [26] adopts lightweight operations. Our article applies two-stage lightweight operations. According to the experimental data, the proposed scheme includes three main communication parties—participant/user, RCRS/server, and checkups station/gateway node (GWN). Table 2 shows an efficacy comparison table of authentication and key agreement phase, where T_H means the operating time of hash operation, T_X means the operating time of XOR operation, T_C means the operating time of string concatenation operation, and T_S means the operating time of symmetric encryption and decryption. Although the operating time of the concatenation operation is light, the parameter length will greatly affect the operating time of the hash function.

Table 1. Functionality comparisons. Two-phase pre-checkup and subsequent validation key agreement authentication mechanism (TCVK).

Property	Ali [25]	Chen [26]	TCVK
P1	YES	YES	YES
P2	NO	NO	YES
P3	NO	YES	YES
P4	NO	YES	YES
P5	YES	YES	YES
P6	YES	YES	YES
P7	YES	YES	YES
P8	NO	YES	YES
P9	YES	NO	YES

P1: fuzzy extractor; P2: no timestamp; P3: session key security; P4: perfect forward secrecy attack; P5: offline password guessing attack; P6: replay attack; P7: checkups station impersonation attack; P8: insider attack; P9: real identity.

Table 2. Efficacy comparisons. Regular checkups records server (RCRS).

	Participant (User)	RCRS (Server)	Checkups Station (GWN)	Sensor Nodes	Total
Ali [25]	$2T_H + 6T_C + 1T_S$	$1T_X + 4T_H + 13T_C + 2T_S$	$1T_X + 8T_H + 18T_C + 3T_S$	$1T_X + 4T_H + 10T_C + 1T_S$	$5T_X + 22T_H + 42T_C + 6T_S$
Chen [26]	$2T_X + 7T_H + 16T_C$	$4T_X + 12T_H + 21T_C$	$4T_X + 9T_H + 33T_C$ -	$3T_X + 6T_H + 19T_C$	$13T_X + 34T_H + 89T_C$
TCVK-PC	$5T_X + 2T_H$	$14T_X + 6T_H$	$6T_X + 4T_H$	—	$25T_X + 12T_H$
TCVK-FV	$5T_X + 2T_H$	$8T_X + 3T_H$	—	—	$13T_X + 5T_H$
TCVK-Total	$10T_X + 4T_H$	$22T_X + 9T_H$	$6T_X + 4T_H$	—	$38T_X + 17T_H$

$T_{H\text{-}}$ means the operating time of hash operation. T_X means the operating time of XOR operation. $T_{C\text{-}}$ means the operating time of string concatenation operation. T_S means the operating time of symmetric encryption and decryption.

In this article, we apply only lightweight operations XOR and hash functions. In our pre-checkups phase (TCVK-PC), our protocol requires a computation cost of $25T_X + 12T_{H\text{-}}$, and in the subsequent verification phase (TCVK-FV), our protocol requires a computation cost of $13T_X + 5T_{H\text{ -}}$. Statistics show that the hash function operation time of our proposed scheme totals $17T_{H\text{-}}$, which is better than that of Ali [25] and Chen [26]. Comparing the length of the computation time, the longest of the three is the encryption and decryption operation time, and the shortest is the bitwise XOR operation time. The length of the string also affects the operation time of the hash function. Therefore, in contrast, our method obviously has better performance even if it involves two stages of computation time.

7. Conclusions

Our TCVK uses a cloud computing network to design an integrated two-stage medical examination and verification key agreement authentication scheme for data storage and verification. To ensure the fairness of the competition and the rights of the participants, which the participants can fully control and verify, the correct checkups information will be encrypted using the participant's key and stored in the cloud server in an encrypted manner. Before participants are qualified to participate in competitions, participants will decrypt the encrypted checkup data and submit it to the committee of the competition. Through our agreement, neither party can refuse to acknowledge the correctness of these checkups data. We also use random oracles to prove in detail the security of our designed protocols. Additionally, in security analysis and performance comparison, we also prove that our proposed protocol is secure, fast, and able to resist many types of malicious attacks.

Author Contributions: Conceptualization, T.-H.L.; Formal analysis, T.-H.L. and M.-T.C.; Methodology, T.-H.L. and M.-T.C.; Supervision, T.-H.L.; Validation, M.-T.C.; Writing—Original draft, T.-H.L.; Writing—Review & editing, T.-H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported in part by grants from the Ministry of Science and Technology of the Republic of China (Grant No. MOST 108-2218-E-167-003-MY2).

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

References

1. Zhang, Q.; Cheng, L.; Boutaba, R. Cloud computing: State-of-the-art and research challenges. *J. Internet Serv. Appl.* **2010**, *1*, 7–18. [CrossRef]
2. Armbrust, M.; Stoica, I.; Zaharia, M.; Fox, A.; Griffith, R.; Joseph, A.D.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; et al. A view of cloud computing. *Commun. ACM* **2010**, *53*, 50–58. [CrossRef]
3. Zhou, J.; Lin, X.; Dong, X.; Cao, Z. PSMPA: Patient Self-Controllable and Multi-Level Privacy-Preserving Cooperative Authentication in Distributedm-Healthcare Cloud Computing System. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *26*, 1693–1703. [CrossRef]
4. Zhou, J.; Cao, Z.; Dong, X.; Xiong, N.; Vasilakos, A.V. 4S: A secure and privacy-preserving key management scheme for cloud-assisted wireless body area network in m-healthcare social networks. *Inf. Sci.* **2015**, *314*, 255–276. [CrossRef]
5. Sajid, A.; Abbas, H. Data Privacy in Cloud-assisted Healthcare Systems: State of the Art and Future Challenges. *J. Med. Syst.* **2016**, *40*, 155. [CrossRef] [PubMed]
6. Paterson, K.G.; Price, G. A comparison between traditional public key infrastructures and identity-based cryptography. *Inf. Secur. Tech. Rep.* **2003**, *8*, 57–72. [CrossRef]
7. Wang, P.; Lin, J.; Jing, J.; Xie, Y. Mediated Hierarchical Identity-Based Combined Public Key Schemes. In Proceedings of the 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, Jinggangshan, China, 2–4 April 2010; pp. 614–618.
8. Tseng, Y.-M.; Jan, J.-K. ID-based cryptographic schemes using a non-interactive public-key distribution system. In Proceedings of the 14th Annual Computer Security Applications Conference, Scottsdale, AZ, USA, 7–11 December 1998; pp. 237–243.
9. Noh, J.; Kim, J.; Kwon, G.; Cho, S. Secure key exchange scheme for WPA/WPA2-PSK using public key cryptography. In Proceedings of the 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, Korea, 26–28 October 2016; pp. 1–4.
10. Shen, J.; Zhou, T.; He, D.; Zhang, Y.; Sun, X.; Xiang, Y. Block Design-Based Key Agreement for Group Data Sharing in Cloud Computing. *IEEE Trans. Dependable Secur. Comput.* **2019**, *16*, 996–1010. [CrossRef]
11. Chen, F.M.; Lee, T.F. Enhancing dynamic identity-based authentication and key agreement using extended chaotic maps for telecare medicine information systems. *J. Qual.* **2018**, *25*, 153–165.
12. Lee, T.F.; Diao, Y.-Y.; Chen, F.M. An Improved Authenticated Key Agreement Protocol with Privacy Protection for Mobile Healthcare Systems with Wearable Sensors. *Int. J. Bus. Syst. Res.* **2019**. Accepted.
13. Gao, A.; Wei, W.; Shi, W. Efficient Password-Proven Key Exchange Protocol against Relay Attack on Ad Hoc Networks. In Proceedings of the 2010 IEEE Asia-Pacific Services Computing Conference, Hangzhou, China, 6–10 December 2010; pp. 469–475.
14. Song, I.-A.; Lee, Y.-S. Improvement of Key Exchange protocol to prevent Man-in-the-middle attack in the satellite environment. In Proceedings of the 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Australia, 5–8 July 2016; pp. 408–413.
15. Kulkarni, G.; Patel, B.; Laxkar, P. Time stamp based cross layer MANET security protocol. In Proceedings of the Third International Conference on Computational Intelligence and Information Technology (CIIT 2013), Mumbai, India, 18–19 October 2013; pp. 191–199.
16. He, D.; Wang, D. Robust Biometrics-Based Authentication Scheme for Multiserver Environment. *IEEE Syst. J.* **2014**, *9*, 816–823. [CrossRef]
17. Lou, D.-C.; Lee, T.-F.; Lin, T.-H. Efficient biometric authenticated key agreements based on extended chaotic maps for telecare medicine information systems. *J. Med. Syst.* **2015**, *39*, 1–10. [CrossRef] [PubMed]
18. Lin, T.-H.; Lee, T.-F. Secure Verifier-Based Three-Party Authentication Schemes without Server Public Keys for Data Exchange in Telecare Medicine Information Systems. *J. Med. Syst.* **2014**, *38*, 1–9. [CrossRef] [PubMed]
19. Dodis, Y.; Ostrovsky, R.; Reyzin, L.; Smith, A. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM J. Comput.* **2008**, *38*, 97–139. [CrossRef]

20. Aswin, V.; Deepak, S. Medical Diagnostics Using Cloud Computing with Fuzzy Logic and Uncertainty Factors. In Proceedings of the 2012 International Symposium on Cloud and Services Computing, Mangalore, India, 17–18 December 2012; pp. 107–112.
21. Becker, G.T. Robust Fuzzy Extractors and Helper Data Manipulation Attacks Revisited: Theory versus Practice. *IEEE Trans. Dependable Secur. Comput.* **2019**, *16*, 783–795. [CrossRef]
22. Baruni, K.; Helberg, A.; Nair, K.; Helberg, A.S. Fingerprint Matching on Smart Card: A Review. In Proceedings of the 2016 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 15–17 December 2016; pp. 809–813.
23. Bellare, M.; Rogaway, P. Random oracles are practical. In Proceedings of the CM Conference on Computer and Communications Security, Fairfax, VA, USA, 3–5 November 1993; pp. 62–73.
24. Lin, C.-J. A Secure Examination and Check-In System for Athletes. Master's Thesis, National Chin-Yi University of Technology, Taichung, Taiwan, 26 July 2019.
25. Ali, R.; Pal, A.K.; Kumari, S.; Karuppiah, M.; Conti, M. A secure user authentication and key-agreement scheme using wireless sensor networks for agriculture monitoring. *Future Gener. Comput. Syst.* **2018**, *84*, 200–215. [CrossRef]
26. Chen, M.; Lee, T.-F.; Pan, J.-I. An Enhanced Lightweight Dynamic PseudonymIdentity Based Authentication and Key AgreementScheme Using Wireless Sensor Networks forAgriculture Monitoring. *Sensors* **2019**, *19*, 1146. [CrossRef] [PubMed]