Enhanced Authenticated Key Agreement for Surgical Applications in a Tactile Internet Environment

Tian-Fu Lee 1, Xiucai Ye 2, Wei-Yu Chen 1 and Chi-Chang Chang 3,4,*

1 Department of Medical Informatics, Tzu Chi University, No. 701, Zhongyang Road, Sec. 3, Hualien 970, Taiwan
2 Department of Computer Science, University of Tsukuba, Tsukuba 3058577, Japan
3 Department of Medical Informatics, Chung Shan Medical University, No. 110, Section 1, Jianguo North Road, South District, Taichung City 402, Taiwan
4 Department of Information Management, Ming Chuan University, No. 5 De Ming Rd., Taoyuan City 333, Taiwan
* Correspondence: changintw@gmail.com

Abstract: The Tactile Internet enables physical touch to be transmitted over the Internet. In the context of electronic medicine, an authenticated key agreement for the Tactile Internet allows surgeons to perform operations via robotic systems and receive tactile feedback from remote patients. The fifth generation of networks has completely changed the network space and has increased the efficiency of the Tactile Internet with its ultra-low latency, high data rates, and reliable connectivity. However, inappropriate and insecure authentication key agreements for the Tactile Internet may cause misjudgment and improper operation by medical staff, endangering the life of patients. In 2021, Kamil et al. developed a novel and lightweight authenticated key agreement scheme that is suitable for remote surgery applications in the Tactile Internet environment. However, their scheme directly encrypts communication messages with constant secret keys and directly stores secret keys in the verifier table, making the scheme vulnerable to possible attacks. Therefore, in this investigation, we discuss the limitations of the scheme proposed by Kamil scheme and present an enhanced scheme. The enhanced scheme is developed using a one-time key to protect communication messages, whereas the verifier table is protected with a secret gateway key to mitigate the mentioned limitations. The enhanced scheme is proven secure against possible attacks, providing more security functionalities than similar schemes and retaining a lightweight computational cost.

Keywords: Tactile Internet; 5G; authentication; key agreement; surgery; robotic arm

1. Introduction

The fifth generation (5G) network provides fast speeds, high data rates, very low latency, and reliable connections for intelligent devices, sensors, and actuators, as well as the ability to communicate through a single device, such as a smartphone. When 5G technology matures, it will provide 100 Gbps coverage, 10 GB/s peak data rates, and more than 100 billion smart device connections to the entire Internet of Things [1]. The high capacity and speed of the 5G network will provide many opportunities for the IoT environment. The Tactile Internet (TI) represents a future development goal with respect to the Internet of Things (IoT), including human–machine interaction and machine–machine interaction, which will enable real-time collaboration and innovative applications in the industrial, social, and commercial fields of the Internet [2,3].

The Tactile Internet will use 5G URLLC (ultra-reliable and low-latency communication) functionality to provide users with ultra-fast Internet so that haptic interaction can be realized through visual feedback [3]. This visual feedback relates to audio–visual interaction, real-time control of robotic systems and actuators, and real-time control of the human body and the environment around it. With the increasing availability of high-speed...
Internet connections, such low-latency functions will lead to enhanced human–machine (tactile) interactions that can be transmitted to the other end of the world in real time [1,3,4]. However, such messages may face security or performance risks once they are transmitted. Therefore, any unauthorized access may lead to an unplanned or unexpected surgery, which could lead to adverse consequences or even death.

The open nature of Tactile Internet connections makes them vulnerable to a variety of security attacks, including replay, denial of service, man-in-the-middle, differential privacy, error data injection, impersonation, and modification attacks, as well as malicious software attacks, requiring secure Tactile Internet access. The remote surgery application establishes a secure user authentication protocol, which allows authorized and registered surgeons to authenticate each other and to generate a shared secure session key for secure and reliable communications with others.

1.1. The Model of a Tactile Internet Remote Surgery Application

Figure 1 illustrates a simple model of a Tactile Internet remote surgery application. A hospital operating room includes robotic arms with tactile sensors and actuators; gateways, such as access points (APs); and patients to be operated on. A remote surgeon controls the robotic arm using instructions provided by a mobile device (or multiple mobile devices) and receives the results of the operation on the screen. All devices must be registered with a trusted institution (TA).

![Figure 1. A simple model of a Tactile Internet remote surgery application.](image_url)

1.2. Related Works

The Tactile Internet can allow doctors to perform accurate, remote surgery more urgently than ever before. The transmission of the data would require the surgical manipulator to move the scalpel with a delay of less than 1 ms to allow the scalpel to move in the correct direction. To obtain the real-time status of the patient, high-resolution organ images and medical equipment data must also be sent back to doctors within 1 ms. Recently, many authenticated key agreement approaches have been developed for remote medical systems. For example, in 2018, Amin et al. [5] proposed a robust and anonymous patient monitoring system based on wireless medical sensor networks to provide secure access to patient data in WMSN environments. In the same year, Wu et al. [6] developed a lightweight and robust authentication scheme for personalized healthcare systems using wireless medical sensor networks and demonstrated that their scheme meets common security requirements and prevents attackers from tracking users. Using wireless medical sensor networks, Chandrakar [7] presented a secure remote user authentication protocol...
for healthcare monitoring that provides privacy, data security, and user authentication to access real-time health information over an insecure channel. Kaur et al. [8] presented a protocol in 2020 that provides the surgeon, robotic arm, and trusted authority (TA) with secure communications, leveraging the advantages of elliptic curve cryptography (ECC) and biometrics. In 2020, Nykvist et al. [9] developed and implemented a lightweight, portable IDS over wireless networks and evaluated throughput, power consumption, and response time. In 2021, Bolton et al. [10] discussed and considered potential data security and privacy issues that may arise when large amounts of data are processed and stored in the cloud. Additional research on the use of the Tactile Internet in remote surgery [8,11,12] provides important background information about the use of the Tactile Internet in remote surgery. For example, Wazid et al. [12] presented a generalized authentication model that can be used to perform authentication among communicating parties to ensure secure remote surgery in the TI environment. In 2021, Kamil et al. [11] proposed an authentication and key agreement (AKA) scheme for a Tactile Internet remote surgery application using lightweight cryptographic operations, such as the one-way hash function and bitwise exclusive OR (XOR), making the scheme ultra-lightweight and suitable for the Tactile Internet environment. However, the proposed scheme directly encrypts communication messages with the constant secret keys of the remote surgeon and the long-life secret key of the robotic arm, directly storing secret keys of the robotic arm in the gateway database; therefore, the scheme cannot resist robotic arm compromise attacks and stolen verifier attacks. Additionally, the scheme proposed by Kamil et al. misuses exclusive OR operations, preventing its correct execution.

1.3. Our Motivation

Many AKA schemes have been recently developed for a Tactile Internet for remote surgery. However, most of these schemes are subject to limitations in terms of security and efficiency. Performance improvement and security considerations are two major factors associated with the Tactile Internet because inappropriate and insecure authentication key agreements for the Tactile Internet may cause misjudgment and improper operation by medical staff, endangering the life of patients.

1.4. Our Contributions

In this investigation, we discuss the limitations of the scheme proposed by Kamil et al., including the failure to resist potential attacks and incorrect execution. In order to overcome these limitations, we investigation develop an enhanced authenticated key agreement scheme based on the scheme proposed by Kamil et al. for the Tactile Internet environment. The enhanced scheme adopts a one-time key to protect communication messages such that the adversary cannot derive valuable information from previous messages and protects secret keys of robotic arms with a secret gateway key. Thus, the enhanced scheme requires more computations and response time than the protocol proposed by Kamil et al. However, the enhanced scheme solves the previous limitations, provides improved functionality, and retains a low computational cost. The contributions of this study are summarized as follows.

1. In this investigation, we develop an efficient and secure authenticated key agreement scheme based on the scheme proposed by Kamil et al. for the Tactile Internet environment.
2. The enhanced scheme adopts a one-time key to protect communication messages and stores the secret keys of robotic arms, which are encrypted the secret gateway key, in the gateway database to overcome the limitations of the previous scheme.
3. Burrows–Abadi–Needham (BAN) logic provides mutual authentication and session key security through its authentication proof. The heuristic security analyses of the enhanced scheme are presented to verify other security requirements.
4. Compared with related schemes, the enhanced scheme avoids the limitations of pervious schemes, providing improved security properties and retaining low computational cost.
1.5. Organization of Paper

The rest of the paper is organized as follows. In Section 2, we introduce the scheme proposed by Kamil et al. and discuss its weaknesses. In Section 3, we introduce an enhanced authenticated key agreement scheme for the Tactile Internet environment. In Section 4, we analyze the security and performance of the enhanced scheme. Finally, in Section 5, we present our conclusions.

2. Preliminary

In this section, we review the authentication and key agreement scheme proposed by Kamil et al. and discuss its limitations. The notations used in this paper are elaborated in Table 1.

2.1. Review of the Scheme of Kamil et al.

In 2020, Kamil et al. [11] proposed an authentication and key agreement scheme using the Tactile Internet for remote surgery. Prior to the announcement, they discussed Tactile Internet technology in remote surgery, the potential of network architecture for the Internet of Thing (IoT), and the security issues of Tactile Internet technology in remote surgery.

The scheme proposed by Kamil et al. comprises four entities: a trusted authority (TA), remote surgeons, gateways, and robotic arms. Gateways act as system administrators and serve as central authentication points. Without BS, other entities would never be able to trust each other in the authentication and key agreement scheme. Kamil et al.’s scheme consists of the following phases: registration of the gateway and robotic arm, registration of the user, the authentication and key agreement phase, the password update phase, the addition of the dynamic robotic arm, and the revocation phase.

Table 1. Notations.

| Notation | Description |
|----------|-------------|
| TA       | Trusted authority |
| Gi       | Gateway i    |
| RMj      | Robotic arm  |
| Sk       | Remote surgeon |
| RIDi     | Identity of gateway |
| RIDj     | Identity of robotic arm |
| RIDk     | Identity of Sk |
| ||      | Concatenation operation |
| TSx      | Timestamp at instant |
| ΔT       | Allowable network transmission delay x |
| ⊕        | Bitwise exclusive OR (XOR) operation |
| h(·)     | Hash function |
| K        | Session key |
| PW       | Password of Sk |
| A        | Adversary |

2.1.1. Gateway and Robotic Arm Registration Phase

Before placing the gateway and robot (or robotic arm) in the hospital operating room, they must register with the TA. These devices are generated and preloaded with secrets. The registration process is performed by the TA through the following steps.

Step 1: \( TA \Rightarrow Gi : M_1 = (RID_i, D_i, RID_j, D_j) \).

The trust authority (TA) first chooses a unique identity \( RID_{TA} \) and a one-way hash function operation \( h : \{0,1\}^* \rightarrow Z_q^* \) for itself. Next, the TA chooses \( RID_i \) and \( RID_j \) as the identities of the gateway \( Gi \) and a robotic arm \( RM_j \), respectively, picks a secret \( s \in Z_q^* \), and computes \( D_i = h(s, RID_{TA}, RID_i) \) and \( D_j = h(s, RID_{TA}, RID_j) \). Finally, the TA stores \( (RID_i, D_i, RID_j, D_j) \) and sends \( M_1 \) to \( Gi \) through a secure channel.

Step 2: \( Gi \Rightarrow RM_j : M_2 = (RID_j, D_j) \).

After gateway \( Gi \) receives \( M_2 \), \( Gi \) stores \( (RID_i, D_i, RID_j, D_j) \) and sends \( M_2 \) to \( RM_j \).
2.1.2. User Registration Phase

In this stage, when the remote surgeon wants to use the robotic arm for remote surgery, they first need to register with the TA. The process is as follows.

Step 1: \( S_k \Rightarrow TA : M_3 = (D_k, HPW_k) \).

The remote surgeon (\( S_k \)) first picks an identity (\( RID_k \)), a password (\( PW_k \)), and a random nonce (\( B_k \)) and computes \( D_k = h(RID_k, B_k) \) and \( HPW_k = h(PW_k, B_k) \). Next, \( S_k \) sends \( M_3 \) to the TA using a secure channel.

Step 2: \( TA \Rightarrow S_k : M_4 = (a, b, h(\cdot)) \).

When the TA receives \( M_3 \), the TA stores \( M_3 \) and then computes \( a = h(C, D_k) \oplus h(D_k, HPW_k) \) and \( b = C \oplus h(RID_k, D_k) \). After the TA stores \( (a, b, h(\cdot)) \) into the memory of a mobile device, it sends the mobile device to the surgeon through a secure channel.

Step 3: \( M_4 \) to the TA using a secure channel.

When \( S_k \) receives the mobile device, \( S_k \) uses a smart card to compute \( A_1 = h(PW_k, RID_k) \oplus B_k \) and \( A_2 = h(B_k, HPW_k, D_k) \). Next, \( S_k \) stores \( A_1 \) and \( A_2 \) in the smart card.

2.1.3. User Login Phase

First, \( S_k \) must input his/her identity or password into the mobile device in order to access the service of robotic arms for remote surgery. Upon successful verification, the mobile device sends a login request message to the gateway (\( G_i \)). The login process is as follows.

\( S_k \) first inputs his identity (\( RID_k \)) and password (\( PW_k \)) and compute \( B_k = A_1 \oplus h(PW_k, RID_k) \), \( D_k = h(RID_k, B_k) \), \( HPW_k = h(PW_k, B_k) \), and \( A_2 = h(B_k, HPW_k, D_k) \) to verify \( A_2 \). The mobile device checks whether \( A_2 \) is the same as \( A_2 \). If so, the identity and password of the surgeon are verified by the smart card. Otherwise, the session is aborted.

2.1.4. Authentication and Key Agreement Phase

In this phase, in order to perform remote surgery in an emergency, the remote surgeon needs to use the robotic arm to perform remote surgery on the patient through the authorization of the gateway device. The mutual authentication and key agreement process of the scheme proposed by Kamil et al. is described as follows.

Step 1: \( S_k \Rightarrow G_i : M_1 = (A_4, A_5, A_6, TS_1) \).

The mobile device of the remote surgeon (\( S_k \)) first picks a random nonce (\( R_3 \)) and a timestamp (\( TS_3 \) and computes \( A_3 = a \oplus h(D_k, HPW_k) \), \( A_4 = b \oplus TS_1 \), \( A_5 = h(R_k, A_3, TS_1) \), and \( A_6 = (R_k || A_5) \oplus A_3 \). Next, the remote surgeon sends a login request message (\( M_1 \)) to \( G_i \).

Step 2: \( G_i \Rightarrow RM_j : M_2 = (A_7, A_8, A_9) \).

After \( G_i \) receives the authentication request message (\( M_1 \)), \( G_i \) computes \( C^* = A_4 \oplus h(RID_i, D_i) \oplus TS_1 \) using the identity of gateway \( RID_i \) and \( D_i \) (\( A_3^* = C^* \oplus D_i \)) and computes \( R_j^* \) to obtain the random number (\( R_j^* \)) of the remote surgeon. Then, \( G_i \) checks the freshness of the message by verifying whether \( TR_1 - TS_1 \leq \Delta T \), where \( TR_1 \) is the time at which the message is received, \( TS_1 \) is the time at which it was sent, and \( \Delta T \) is the transmission delay. If the timestamp is legal, \( G_i \) computes \( A_9^* = h(R_j^*, C^*, R_i, TS_2) \) to verify whether \( A_9 \) is the same as \( A_9^* \). If the verification is successful, the gateway (\( G_i \)) is authenticated by \( G_i \). Then, \( G_i \) chooses a random nonce (\( R_i \)) and a timestamp (\( TS_2 \) and computes \( A_7 = C^* \oplus h(RID_i, D_i, R_i, R_j^*, TS_2) \), \( A_8 = D_i \oplus (R_i || R_j^* || TS_2) \), and \( A_9 = h(RID_i, D_i, C^*, R_i, TS_2) \). Finally, \( G_i \) sends \( M_2 \) to the robotic arm (\( RM_j \)).

Step 3: \( RM_j \Rightarrow G_i : M_3 = (A_{10}, A_{11}) \).

Upon receiving the tuple (\( A_{7}, A_{8}, A_{9} \)), \( RM_j \) computes \( R_i^* || R_j^* || TS_2 = A_8 \oplus D_i \) to obtain the random numbers \( R_i^* \) and \( R_j^* \), where \( R_i^* \) belongs to the gateway and \( R_j^* \) belongs to the remote surgeon, and checks the freshness of the message by verifying whether \( TR_2 - TS_2 \leq \Delta T \), where \( TR_2, TS_2 \), and \( \Delta T \) are the time at which the message was sent, the arrival time of the message, and the transmission delay, respectively. If the freshness
of timestamp is verified, \( RM_i \) computes \( C^{**} = A_7 \oplus h(RID_j, D_j, R_i^*, R_k^*, TS_2) \) and \( A_9^* = h(RID_j, D_j, C^{**}, R_i^*, TS_2) \). Finally, \( RM_i \) verifies whether \( A_9^* \) is the same as \( A_9 \). If verification is successful, the gateway is authenticated by \( RM_i \). Next, \( RM_i \) chooses a random number \( (R_i) \) and a timestamp \( (TS_3) \) and computes the session key \( K_1 = h(R_i^*, R_k^*, R_i) \). \( A_{10} = h(R_i^*, R_k^*, RID_j, D_j, TS_3) \), and \( A_{11} = R_i^* \oplus (R_i \parallel TS_3) \). Finally, \( RM_i \) sends \( M_3 \) to \( G_i \) through a public channel.

Step 4: \( G_i \) sends \( M_4 = (A_8, A_{12}, A_{13}) \).

When \( G_i \) receives \( M_3, G_i \) computes \( R_i^* \parallel TS_3 = A_{11} \oplus R_i \) to obtain the random number of \( RM_i \) using the random number of \( G_i \) and timestamp \( TS_3 \), and checks the freshness of the message by verifying whether \( T_R - TS_3 \leq \Delta T \), where \( T_R, TS_3 \), and \( \Delta T \) are the time at which the message was sent, the arrival time of the message, and the transmission delay, respectively. If the freshness of the timestamp is legal, \( G_i \) computes the session key \( K_2 = h(R_i^*, R_i^*, R_i^*) \) and \( A_{10}^* = h(R_i^*, R_k^*, K_2, RID_j, D_j, TS_3) \). \( G_i \) checks whether \( A_{10}^* \) is the same as \( A_{10} \). If so, the robotic arm \( (RM_i) \) is authenticated by \( G_i \). Next, \( G_i \) computes \( A_{12} = h(K_2, R_i^*, R_k^*, A_{13}, S_k) \) and \( A_{13} = (R_i \parallel R_i^* \parallel TS_4) \oplus R_k^* \) and sends \( M_4 \) to \( S_k \) where \( TS_4 \) is the timestamp.

Step 5: Verification of the remote surgeon.

When \( S_k \) receives \( M_4, S_k \) first computes \( R_i^* \parallel R_i^{**} \parallel TS_4 = A_{13} \oplus R_k \) using the random number \( (R_k) \) and then checks the freshness of the message by verifying whether \( T_R - TS_4 \leq \Delta T \), where \( T_R, TS_4 \), and \( \Delta T \) are the time at which the message was sent, the arrival time of the message, and the transmission delay, respectively. If the timestamp is fresh, \( S_k \) computes the session key \( K_3 = h(R_i^*, R_i^{**}, R_k) \) and \( A_{12} = h(K_3, R_i^*, R_k^{**}, A_{13}, TS_4) \) to verify \( A_{12} \). If the verification is successful, \( G_i \) and \( RM_i \) are authenticated by \( S_k \).

The mutual authentication of the remote surgeon and the robotic arm requires the assistance of the gateway for remote authentication. Additionally, secure communication during remote surgery is achieved with the secret session key, \( K = K_1 = K_2 = K_3 \).

2.1.5. Password Updating Phase

In this phase, when the remote surgeon thinks that his password has been leaked, for security reasons, he can change his password at any time. The password renewal phase is as follows.

The remote surgeon \( (S_k) \) inputs his original password \( (PW_k^*) \) and identity \( (RID_k^*) \) into the mobile device, and the mobile device computes \( B_k^* = A_1 \oplus h(PW_k^*, RID_k^*) \), \( HPW_k^* = h(PW_k^*, B_k^*) \), \( D_k^* = h(RID_k^*, B_k^*) \), and \( A_2^* = h(B_k^*, HPW_k^*, D_k^*) \) to check whether \( A_2^* \) is the same as \( A_2 \). If the verification is successful, the password and identity of the surgeon are verified. Next, the card reader prompts \( S_k \) to input a new password \( (PW_k^{new}) \) and a nonce \( (B_k^{new}) \). Then, it computes \( HPW_k^{new} = h(PW_k^{new}, B_k^{new}) \), \( D_k^{new} = h(RID_k^*, B_k^{new}) \), \( A_1^{new} = h(PW_k^{new}, RID_k^*) \oplus B_k^{new} \), \( A_2^{new} = h(B_k^{new}, HPW_k^{new}, D_k^{new}) \), and \( a^{new} = a \oplus h(D_k^{new}, HPW_k^{new}) \oplus h(D_k^{new}, HPW_k^{new}) \). Finally, the mobile device replaces \( a, A_1, \) and \( A_2 \) with \( a^{new}, A_1^{new}, \) and \( A_2^{new} \), respectively.

2.1.6. Dynamic Robotic Arm Addition Phase

After placing these robotic arms in the operation room, additional robots may be required for improved service delivery. The following steps are required.

The TA first chooses a new identity \( (RID_k^*) \) and computes \( D_k^* = h(s, RID_{TA}, RID_k^*) \). The TA stores \( (RID_k^*, D_k^*) \) in the memory of the new robotic arm and sends the tuple to the gateway \( (G_i) \) through a secure channel. When \( G_i \) receives the tuple \( (RID_k^*, D_k^*) \), \( G_i \) stores it in its repository.

2.1.7. Revocation Phase

When the remote surgeon’s mobile device is stolen by an attacker, the attacker can reuse the data from the mobile device, thus impersonating the legitimate doctor. The same method is applied to the robot arm; the attacker can analyze the sensitive information in
the robotic arm and compute the session key to execute an attack. In addition, attackers can swap out a robotic arm with a cloned robotic arm, which can lead to life-threatening conditions in patients who require medical attention. The proposed scheme involves two revocation processes: revocation of compromised mobile devices and revocation of compromised robotic arms.

1. Revocation of Smart Card: Steps can be taken to prevent compromised mobile devices from gaining access to the network. The TA first chooses a new identity (RID\text{new}) and computes D\text{new} = h(s, RID_{TA}, RID_{new}). Next, the TA sends the tuple (RID\text{new}, D\text{new}) to Gi. When Gi receives (RID\text{new}, D\text{new}), Gi replaces (RID\text{i}, D\text{i}) with (RID\text{new}, D\text{new}) and stores it in its database.

2. Revocation of Robotic Arm: Suppose RID\text{i} is the identity of the malicious or compromised robot. In order to prevent the malicious or damaged robotic arm from being verified by the remote surgeon and accessing the network, the following steps are performed in order to log off the manipulator. The TA computes Π = (RID\text{i} || D\text{i}) ⊕ h(RID\text{i}, D\text{i}) and sends (Π, rev\_eq) to Gi, where rev\_eq is the revocation request. When Gi receives the tuple (Π, rev\_eq), Gi computes RID\text{j} || D\text{j} = Π ⊕ h(RID\text{i}, D\text{i}). Finally, Gi deletes the tuple (RID\text{i}, D\text{i}) from its database.

2.2. Limitations of the Authenticated Key Agreement Proposed by Kamil et al.

The authenticated key agreement scheme proposed by Kamil et al. directly encrypts communication messages between the gateway and the remote surgeon with the constant secret keys of the remote surgeon and directly encrypts communication messages between the gateway and the robotic arm with the long-life secret key of the robotic arm so that an attacker who has captured a robotic arm can derive secret keys of the remote surgeon from previous messages and successfully impersonate the remote surgeon and the robotic arm. The attacker can successfully compute session keys from previous messages to decrypt communication messages between the remote surgeon, the gateway, and the robotic arm to trick legal participants. Additionally, the scheme of Kamil et al. directly stores secret keys of robotic arms, so an attacker who has stolen the verifier table can successfully impersonate the robotic arm. Accordingly, the scheme proposed by Kamil et al. cannot resist robotic arm compromise attacks and stolen verifier attacks. Moreover, the scheme proposed by Kamil et al. misuses exclusive OR operations, preventing its correct execution.

Below, we discuss the limitations of the scheme proposed by Kamil et al. in detail.

2.2.1. Failure to Resist Robotic Arm Compromise Attacks

1. Scenario I: Impersonation of a surgeon.

In the scheme proposed by Kamil et al., when a robotic arm (RM\text{j}) is compromised, an attacker (A) can obtain RID\text{j} and D\text{j}. The attacker (A) obtains A\text{8} from previous communication messages and computes R\text{i} || R\text{k} || TS\text{2} = A\text{8} ⊕ D\text{j} to obtain the random secrets (R\text{i}) of the gateway (Gi) and R\text{k} of the remote surgeon (S\text{k}). Next, A computes C = A\text{7} ⊕ h(RID\text{j}, D\text{j}, R\text{i}, R\text{k}, TS\text{2}) to obtain the random secret (C) of TA. A obtains previous communication messages (A\text{4}, A\text{5}, A\text{6}, TS\text{1}) of S\text{k} and computes B = A\text{4} ⊕ TS\text{1}, A\text{3} = (R\text{i} || A\text{5}) ⊕ A\text{6}(= h(C, D\text{i})). A can compute A\text{4} = B ⊕ TS\text{1}, A\text{5} = h(R\text{k}, A\text{3}, TS\text{1}) and A\text{6} = R\text{i} || A\text{5} ⊕ A\text{3} and send out a service request (M\text{1} = (A\text{4}, A\text{5}, A\text{6}, TS\text{1})) to impersonate S\text{k}, where R\text{k} is a nonce selected by A, and TS\text{1} is the current timestamp.

Upon receiving M\text{4} = (A\text{8}, A\text{12}, A\text{13}) from Gi, A can compute R\text{j}∗ || R\text{j}∗∗ || TS\text{4} = A\text{13} ⊕ \overline{R\text{k}} and the session key (K\text{3} = h(R\text{j}∗, R\text{j}∗∗, \overline{R\text{k}})) shared with Gi and RM\text{i} and successfully impersonate the surgeon (S\text{k}). Therefore, the scheme proposed by Kamil et al. fails to resist robotic arm compromise attacks.

2. Scenario II: Impersonation of a gateway.

According to the analyses of Scenario I, the attacker (A) can easily derive A\text{3}(= h(C, D\text{i})), the random secret (C) from previous communication messages. Upon receiving M\text{1} = (A\text{4}, A\text{5}, A\text{6}, TS\text{1}) from S\text{k}, A computes h(RID\text{i}, D\text{i}) = A\text{4} ⊕ C ⊕ TS\text{1} and R\text{k} || A\text{5} =
A_6 \oplus A_3. Then, A chooses a nonce (\( \tilde{R}_i \)) and picks the current timestamp (\( \tilde{T}_S_2 \)) and then computes \( A_7 = C \oplus h(RID_j, D_j, \tilde{R}_i, R_i^4, \tilde{T}_S_2) \), \( A_8 = D_j \oplus \{ \tilde{R}_i || R_i^4 || \tilde{T}_S_2 \} \), and \( A_9 = h(RID_j, D_j, C, \tilde{R}_i, \tilde{T}_S_2) \) and sends \( M_2 = (A_7, \tilde{A}_8, \tilde{A}_9) \) to \( R_M_j \).

Upon receiving \( M_3 = (A_{10}, A_{11}) \), A computes \( R_i^* || TS_3 = A_{11} \oplus \tilde{R}_i \) and the session key (\( K_2 = h(\tilde{R}_i, R_i^4, R_i^j) \)) shared with \( G_i \) and \( R_M_j \). Next, A computes \( \tilde{A}_{12} = h(K_2, \tilde{R}_i, R_i^4, \tilde{A}_8, TS_4) \) and \( \tilde{A}_{13} = (\tilde{R}_i || R_i^4 || \tilde{T}_S_4) \oplus R_i^* \) and sends \( \tilde{M}_4 = (\tilde{A}_8, \tilde{A}_{12}, \tilde{A}_{13}) \) to \( S_k \), where \( TS_4 \) is the current timestamp. A successfully impersonates the gateway (\( G_i \)); therefore, the scheme proposed by Kamil et al. fails to resist robotic arm compromise attacks.

3. Scenario III: Violation of session key security.

According to the analyses of Scenario I, the attacker (A) can easily derive \( A_3(= h(C, D_j)) \), the random secret (C) from previous communication messages. First, A impersonates \( S_k \) to compute \( A_4 = B \oplus TS_1^j, A_5 = h(\tilde{R}_i, A_3, TS_1) \), and \( A_6 = \tilde{R}_i || A_5 \oplus A_3 \) and to send a service request (\( \tilde{M}_1 = (A_4, A_5, A_6, \tilde{TS}_1) \)) to \( G_i \), where \( \tilde{R}_i \) is a nonce selected by \( A_4 \) and \( \tilde{T}_S_1 \) is the current timestamp.

Then, A eavesdrops on communications between \( G_i \) and another robotic arm (\( R_M_j' \)) and obtains \( M_2 = (A_7, \tilde{A}_8, \tilde{A}_9) \) and \( M_3 = (A_{10}, A_{11}) \) where \( RID_j' \) is the identity of \( R_M_j' \), \( D_j' \) is the secret key of \( R_M_j' \), \( A_7 = C' \oplus h(RID_j', D_j', \tilde{R}_i, \tilde{R}_i, \tilde{T}_S_2) \), \( A_8 = D_j' \oplus \{ R_i^* || \tilde{R}_i || TS_2 \} \), \( A_9 = h(RID_j', D_j', C, \tilde{R}_i, TS_2) \), \( A_{10} = h(R_i^4, R_i^j, K_1, RID_j', D_j', TS_3) \), and \( A_{11} = R_i^4 \oplus \{ R_i^* || TS_3 \} \).

Upon receiving \( M_4 = (A_6, A_{12}, A_{13}) \) from \( G_i \), where \( A_{12} = h(K_2, \tilde{R}_i, R_i^4, A_8, TS_4) \) and \( A_{13} = (R_i^* || R_i^4 || \tilde{T}_S_4) \oplus \tilde{R}_i \), A can compute \( R_i^* || R_i^4 || TS_4 = A_{13} \oplus \tilde{R}_i \) and the secret key of \( R_M_j' \), \( D_j' = A_8 \oplus (R_i^* || \tilde{T}_S_2) \).

Although the attacker (A) does not have \( R_M_j' \)'s identity (\( RID_j' \)), A can still monitor other communications between \( S_k, G_i \), and some robotic arms (\( R_M_j' \)). A computes \( R_i^* || TS_2 = (A_8 \oplus D_j') \) and verifies whether \( TS_2 \) is a current timestamp. If successful, A makes sure that \( R_M_j' \) is \( R_M_j \) and \( R_1 \) is \( R_i \) from \( G_i \) and that \( R_2 \) is \( R_k \) from \( S_k \).

Then, A computes \( (R_i^* || R_i^4 || TS_4) = A_{13} \oplus \tilde{R}_i \). Accordingly, A can obtain the session key (\( K = h(R_i^*, R_k, R_j) \)) of \( S_k, G_i, \) and \( R_M_j' \) to decrypt communication messages between \( S_k, G_i \), and \( R_M_j' \) to perform man-in-the-middle attacks and modification attacks and to trace \( R_M_j' \).

2.2.2. Failure to Resist Stolen Verifier Attacks

In the register phase of the scheme proposed by Kamil et al., the gateway (\( G_i \)) stores \( RID_j \) and \( D_j \) for each robotic arm (\( R_M_j \)). An attacker who has stolen the verifier table can impersonate the robotic arm (\( R_M_j \)), as it obtains the secrets (\( RID_j, D_j \)) of \( R_M_j \) and has the same ability as \( R_M_j \).

2.2.3. Failure to Execute Correctly

In the scheme proposed by Kamil et al., the surgeon (\( S_k \)) cannot correctly compute \( A_6 = (R_k \| A_5) \oplus A_3 \) in Step 1. Because \( (R_k \| A_5) \) is longer than \( A_3 \), where \( A_3 = h(C, D_j) \) and \( A_5 = h(R_k, A_3, TS_1) \), \( S_k \) cannot directly execute an exclusive OR operation of \( (R_k \| A_5) \) and \( A_3 \). Similar problems also occur in that \( G_i \) cannot correctly compute \( A_8 = D_j \oplus (R_i^* \| R_i^4 \| TS_2) \) in Step 2, \( R_M_j \) cannot correctly compute \( A_{11} = R_i^* \oplus (R_i^* \| TS_3) \) in Step 3, and \( G_i \) cannot correctly compute \( A_{13} = R_i \| R_i^* \| TS_4 \oplus R_i^* \) in Step 4.

3. Enhanced Authenticated Key Agreement Scheme for Tactile Internet Environment

In this section, we develop an enhanced AKA scheme based on the AKA scheme proposed by Kamil et al. for the Tactile Internet environment. In order to overcome the limitations of the AKA scheme proposed by Kamil et al., the enhanced scheme adopts a one-time key to protect communication messages such that an attacker who captures the robotic arm cannot derive valuable information from previous messages to perform impersonation attacks. To avoid stolen verifier attacks, \( G_i \) does not directly store the secret
key \((D_j)\) of \(RM_j\) in its database and protects \(D_j\) with the secret key \((D_i)\) of \(G_i\). Even if the attacker steals the verification table, he/she still cannot obtain the secret key \((D_j)\) of \(RM_j\) to successfully impersonate \(RM_j\).

A number of phases are involved in the enhanced scheme, including registration of gateways and robotic arms, registration of remote surgeons, login of remote surgeons, authentication and key agreement, updating of passwords, adding dynamic robotic arms, and revocation. Because the password updating phase, dynamic robotic arm addition phase, and revocation phase of the enhanced scheme are similar to the scheme proposed by Kamil et al., they are not discussed here. Below, we provide a detailed description of the gateway and robotic arm registration phase, the remote surgeon registration phase, the remote surgeon login phase, the authentication phase, and the key agreement phase. Figure 2 shows a flow chart of the enhanced scheme.

**Figure 2.** Flow chart of the enhanced scheme.

### 3.1. Registration Phase of Gateway and Robotic Arms

This phase provides the registration process for the gateway and robotic arms with the TA, as shown in Figure 3. The registration process is as follows.

**Step 1:** \(TA \Rightarrow G_i : M_1 = (RID_{i}, D_i, RID_{j}, D_j)\).

The trust authority (TA) at first chooses a unique identity \((RID_{TA})\) and a one-way hash function operation \((h : \{0,1\}^* \rightarrow Z_q^* )\). Next, the TA chooses \(RID_i\) and \(RID_j\) as the identities of the gateway \((G_i)\) and the robotic arm \((RM_j)\), respectively, picks a secret \((s \in Z_q^*)\), and computes \(D_i = h(s, RID_{TA}, RID_{i})\) and \(D_j = h(s, RID_{TA}, RID_{j})\). Finally, the TA stores \((RID_{i}, D_i, RID_{j}, D_j)\) and sends \(M_1\) to \(G_i\) through a secure channel.

**Step 2:** \(G_i \Rightarrow RM_j : M_2 = (RID_{j}, D_j)\).

After the gateway \((G_i)\) receives \(M_2\), \(G_i\) computes \(CD_j = h(RID_j \parallel D_i) \oplus D_j\) and stores \((RID_{j}, D_j, RID_{j}, CD_j)\). Finally, \(G_i\) sends \(M_2\) to \(RM_j\).
3.2. User Registration Phase

In this phase, the remote surgeon (S_r) registers with the trusted authority (TA). Each surgeon (S_r) has a smart card with the information of the surgeon. The registration process of the remote surgeon is shown in Figure 4.

Figure 4. Registration phase of the remote surgeon of the proposed scheme.

Step 1: S_r \Rightarrow TA : M_1 = (RID_k, D_k, HPW_k).

The remote surgeon (S_r) first picks his/her own identity (RID_k), password (PW_k), and a random number B_k and computes D_k = h(RID_k, B_k) and HPW_k = h(PW_k, B_k). Finally, S_r sends M_1 to the TA through a secure channel.

Step 2: TA \Rightarrow S_r : M_2 = (TID_k, a, h(\cdot)).

After receiving M_1, the TA first picks a random identity (TID_k) and computes \( a = h(TID_k, D_k) \oplus h(D_k, HPW_k) \). Then, the TA stores (a, TID_k) in the memory of a mobile device and sends it to S_r through a secure channel. Upon receiving the mobile device, S_r computes \( A_1 = h(PW_k, RID_k) \oplus B_k \) and the verification message, \( VM_1 = h(B_k, HPW_k, D_k) \). Then, S_r stores A_1, VM_1, TID_k, and a in the smart card.
3.3. Login, Authentication, and Session Key Agreement Phase

In order to perform remote operations in case of an emergency, the remote surgeon (S) needs to log in to a smart card and send a verification message to access the gateway (G). The gateway (G) sends a verification message to the robot after the remote surgeon has been identified. The robot passes the authentication message to the remote surgeon via the gateway. Finally, the gateway, remote coverage, and robotic arm establish a session key for the current login session. The authentication and key agreement of the proposed protocol is shown in Figure 5, and the details are summarized below:

Step 1: \( S_k \rightarrow G_i : M_1 = (TID_i, A_3, VM_{M2}, TS_1) \).

The remote surgeon (\( S_k \)) inputs his/her \( RID_k \) and \( PW_k \) into the mobile device; then, mobile device computes \( B_k = A_1 \oplus h(RID_k, PW_k) \) to obtain the random number \( (B_k) \) and computes \( D_k = h(RID_k, B_k) \), \( HPW_k = h(PW_k, B_k) \), and \( VM_1^0 = h(B_k, HPW_k, D_k) \) to verify \( VM_1^0 = ?VM_1 \). If successful, the mobile device picks the current timestamp \( (TS_1) \) and a random number \( (R_k) \) and computes \( A_2 = a \oplus h(D_k, HPW_k) \) and \( A_3 = h(A_2, HPW_k) \oplus R_k \) and verification the message, \( VM_2 = h(R_4, A_2, TS_1) \). Finally, \( S_k \) sends \( M_1 \) to the gateway (\( G_i \)).

Step 2: \( G_i \rightarrow RM_j : M_2 = (TID_j, A_4, A_5, VM_{M3}, TS_2) \).

When \( G_i \) receives \( M_1 \), \( G_i \) checks whether the timestamp \( (TR_1 - TS_1) \) is less than \( \Delta T \). If successful, \( G_i \) computes \( A_2^g = h(TID_k, D_1) \), \( R_k^g = A_3 \oplus h(A_2^g, TS_1) \), and \( VM_2^g = h(R_k^g, A_2^g, TS_1) \) to verify \( VM_2^g = ?VM_2 \). If successful, \( G_i \) picks a random number \( (R_i) \) and the current timestamp \( (TS_2) \) and computes \( D_j = h(RID_j \parallel D_j) \oplus CD_j \) to obtain the \( D_j \) of \( RM_j \), then computes \( A_4 = h(D_j, TS_2, 0) \oplus R_i \), \( A_5 = h(D_j, TS_2, 1) \oplus R_k^g \), and a verification message, \( VM_3 = h(RID_j, D_j, TID_k, R_i, TS_2) \), where \( D_j \) is the secret of the robotic arm, and \( TS_2 \) ensures the freshness of messages.

Step 3: \( RM_j \rightarrow G_i : M_3 = (A_6, VM_{M4}, TS_3) \).

After receiving \( M_2 \) from \( G_i \), \( RM_j \) checks whether the timestamp \( (TR_2 - TS_2) \) is less than \( \Delta T \). If successful, \( RM_j \) computes \( R_k^g = A_4 \oplus h(D_j, TS_2, 0), R_k^g = A_5 \oplus h(D_j, TS_2, 1) \), and \( VM_2^g = h(RID_j, D_j, TID_k, R_k^g, TS_2) \) to verify \( VM_2^g = ?VM_3 \). If successful, \( RM_j \) picks a random number \( (R_i) \) and the current timestamp \( (TS_3) \) and computes the session key \( (K_1 = h(R_i^g, R_k^g, R_j)) \), \( A_6 = h(R_k^g, TS_3) \oplus R_i \), and the verification message \( VM_4 = h(R_i^g, R_k^g, K_1, RID_j, D_j, TS_3) \). Then, \( RM_j \) sends \( M_3 \) to \( G_i \).

Step 4: \( G_i \rightarrow S_k : M_4 = (A_7, A_8, A_9, VM_{M5}, TS_4) \).

When \( G_i \) receives \( M_3 \), \( G_i \) checks whether the timestamp \( (TR_3 - TS_3) \) is less than \( \Delta T \). If successful, \( G_i \) computes \( R_k^g = A_6 \oplus h(R_i, TS_3), K_2 = h(R_k^g, R_i^g, R_i^g) \), and the verification message \( VM_4^g = h(VM_{M4}^g, A_6^g, R_k^g, K_2, RID_j, D_j, TS_3)) \) to verify \( VM_4^g = ?VM_4 \). If successful, \( G_i \) picks the current timestamp \( (TS_4) \) and computes \( A_7 = h(A_2, TS_4, 0) \oplus R_i, A_8 = h(A_2, TS_4, 1) \oplus R_j^g, TID_k^{new} = h(A_2, K_2), A_2^{new} = h(TID_k^{new}, A_7), A_9 = h(A_2, TS_4, 2) \oplus A_2^{new}, \) and \( VM_5 = h(K_2, A_2^{new}, TS_4) \). Finally, \( G_i \) sends \( M_4 \) to \( S_k \).

Step 5: Update TID_k and \( a \) in \( S_k \).

After \( S_k \) receives \( M_4 \), \( S_k \) checks whether the timestamp \( (TR_4 - TS_4) \) is less than \( \Delta T \). If successful, \( S_k \) computes \( R_k^g = h(A_2, TS_4, 0) \oplus A_7 \) and \( R_j^{**} = h(A_2, TS_4, 1) \oplus A_8 \) to obtain the random number \( (R_k^g) \) of \( G_i \) and the random number \( (R_j^{**}) \) of \( RM_j \). Next, \( S_k \) computes the session key \( (K_3 = h(R_k^g, R_j^{**}, R_k)) \), \( A_2^{new} = A_9 \oplus h(A_2, TS_4, 2) \), and \( TID_k^{new} = h(A_2, K_3) \). Then, \( S_k \) computes \( VM_4^{g} = h(K_3, A_2^{new}, TS_4) \) to verify \( VM_4^{g} = ?VM_4 \). If successful, \( S_k \) computes \( a^{new} = A_2^{new} \oplus h(D_k, HPW_k) \) and updates \( a \) and \( TID_k \) via \( a^{new} \) and \( TID_k^{new} \) in the smart card.
4. Security and Performance Analysis

An analysis and comparison of the performance and security of the enhanced scheme are provided in this section.
4.1. Authentication Proof of the Proposed Scheme Using BAN Logic

BAN logic [13] is used in this subsection to verify that the proposed scheme satisfies the session key security and mutual authentication requirements. Table 2 lists the notations of BAN logic.

Table 2. BAN logic notations and respective abbreviations [13].

| Notation | Abbreviation |
|----------|--------------|
| $P \equiv X$ | Entity $P$ believes statement $X$ |
| $P \implies X$ | $P$ has jurisdiction over statement $X$ |
| $P \sim X$ | $P$ once said $X$ |
| $P \triangleleft X$ | $P$ sees $X$ |
| $\langle X \rangle_K$ | Formula $X$ is encrypted by key $K$ |
| $P \not\triangleleft Q$ | $P$ and $Q$ communicate via shared key $K$ |
| $P \to Q : m$ | $P$ sends the message ($m$), and $Q$ receives it |
| $\#X$ | Message $\#X$ is freshly generated |

4.1.1. Inference Rules of BAN Logic

Below, we present a list of the rules and logical postulates of BAN logic [13].

**Rule 1.** $\frac{P \equiv P', Q \not\triangleleft \langle X \rangle_K}{P \equiv Q \implies \langle X \rangle_K}$: If entity $P$ believes that secret $K$ is shared with $Q$ and sees message $X$ is encrypted using $K$, then $P$ believes that $Q$ once said $X$.

**Rule 2.** $\frac{P \equiv \#(X), Q \equiv \#X}{P \equiv Q \implies \#X}$: If entity $P$ believes that $X$ is fresh and entity $Q$ once said $X$, then $P$ believes that $Q$ believes $X$.

**Rule 3.** $\frac{P \equiv Q \implies X, Q \equiv X}{P \equiv X}$: If entity $P$ believes that $Q$ has jurisdiction over $X$ and $Q$ believes $X$, then $P$ believes that $X$ is true.

**Rule 4.** $\frac{P \equiv \#(X), Q \equiv \#X}{P \equiv \#(X, Y)}$: If entity $P$ believes that $X$ is fresh and $Q$ believes $X$, then $P$ believes that $X$ is shared between entities $P$ and $Q$.

**Rule 5.** $\frac{P \equiv \#(X, Y)}{P \equiv \#(X', Y)}$: If entity $P$ believes that $X$ is fresh, then $P$ believes in the freshness of $(X', Y)$.

4.1.2. Goals of Authentication and Key Agreement

In this subsection, we demonstrate that the proposed scheme satisfies the following goals to ensure its security according to the above assumptions and postulates.

**Goal 1:** $G_i \equiv S_k \equiv G_i^{K_k} S_k$.

**Goal 2:** $G_i \equiv RM_j \equiv G_i^{K_k} RM_j$.

**Goal 3:** $RM_j \equiv G_i \equiv RM_j^{K_k} G_i$.

**Goal 4:** $S_k \equiv G_i \equiv S_k^{K_k} G_i$.

**Goal 5:** $S_k \equiv RM_j \equiv S_k^{K_k} RM_j$.

**Goal 6:** $RM_j \equiv S_k \equiv RM_j^{K_k} S_k$.

4.1.3. Idealized Form

The proposed scheme is transformed into an idealized form in the following manner.

**M1.** $(S_k \rightarrow G_i) : TID_k, A_3 : \langle R_k \rangle_{h(A_2^*, TS_1)}; VM_2 : h(R_k, A_2, TS_1), TS_1$.

**M2.** $(G_i \rightarrow RM_j) : TID_k, A_4 : \langle R_k \rangle_{h(D_j, TS_2)}; A_5 : \langle R_k \rangle_{h(D_j, TS_1)}$.

**M3.** $(RM_j \rightarrow G_i) : A_6 : \langle R_j \rangle_{h(R_j^*, TS_3)}; VM_4 : h(K_1, RID_j, D_j, TS_3), TS_3$.

**M4.** $(G_i \rightarrow S_k) : A_7 : \langle R_i \rangle_{h(A_2^*, TS_4)}; A_8 : \langle R_i \rangle_{h(A_2^*, TS_4)}; A_9 : \langle A_2^{new} \rangle_{h(A_2^*, TS_4)}$.

**VM5:** $h(K_2, A_2^{new}, TS_4), TS_4$. 


4.1.4. Assumptions

According to the following assumptions, in this subsection, we prove that the proposed scheme satisfies the security properties.

- **AS₁**: \( G_i \vdash h(R_k, A_2, TS_1) \).
- **AS₂**: \( G_i \vdash h(K_1, RID_j, D_j, TS_3) \).
- **AS₃**: \( G_i \vdash G_i \xrightarrow{A_2} S_k \).
- **AS₄**: \( S_k \vdash S_k \xrightarrow{A_2} G_i \).
- **AS₅**: \( G_i \vdash G_i \xrightarrow{D_j} RM_j \).
- **AS₆**: \( RM_j \equiv RM_j \xrightarrow{D_j} G_i \).
- **AS₇**: \( RM_j \equiv h(RID_j, D_j, TID_k, R_i, TS_2, R_k^*) \).
- **AS₈**: \( S_k \equiv h(K_2, A_2^{new}, TS_4) \).
- **AS₉**: \( S_k \equiv G_i \implies R_j \).
- **AS₁₀**: \( S_k \equiv RM_j \implies R_j \).
- **AS₁₁**: \( G_i \equiv S_k \implies R_k \).
- **AS₁₂**: \( G_i \equiv RM_j \implies R_j \).
- **AS₁₃**: \( RM_j \equiv G_i \implies R_j \).
- **AS₁₄**: \( RM_j \equiv S_k \implies R_k \).

4.1.5. Verification

Based on the above assumptions and the logic of BAN, the following confirms the correctness of the proposed scheme. By using Message \( M_1 \),

\[ G_i \vdash (TID_k, A_3 : \langle R_k \rangle_{h(A_2, TS_1)}, VM_2 : h(R_k, A_2, TS_1), TS_1) \]

From Rule 1 and **AS₃**, \( V_3; G_i \equiv S_k \approx R_k \).

From Rule 2 and **AS₁**, \( V_2; G_i \equiv S_k \equiv R_k \).

Then, from Rule 3 and **AS₁₁**, \( V_3; G_i \equiv S_k \).

According to Rule 4, **AS₁** and \( V_2 \),

\[ V_4; G_i \equiv G_i \xrightarrow{K_k} S_k \]

Further, using Rule 2, **AS₁** and \( V_1 \),

\[ V_5; G_i \equiv S_k \equiv G_i \xrightarrow{K_k} S_k \] \( \text{Goal 1} \)

Similarly, by using Message \( M_3 \),

\[ G_i \vdash (A_6 : \langle R_j \rangle_{h(R_j, TS_3)}, VM_4 : h(K_1, RID_j, D_j, TS_3), TS_3) \]

From Rule 1 and **AS₅**, \( V_6; G_i \equiv RM_j \approx R_j \).

From Rule 2 and **AS₂** and \( V_6 \),

\[ V_7; G_i \equiv RM_j \equiv R_j \]

From Rule 3 and **AS₁₂**, \( V_8; G_i \equiv R_j \).

According to Rule 4, **AS₂** and \( V_7 \),

\[ V_9; G_i \equiv G_i \xrightarrow{K_k} RM_j \]

Using Rule 2, **AS₂** and \( V_6 \), we have

\[ V_{10}; G_i \equiv RM_j \equiv G_i \xrightarrow{K_k} RM_j \] \( \text{Goal 2} \)

By using Message \( M_2 \),

\[ RM_j \vdash (TID_k, A_4 : \langle R_j \rangle_{h(D_j, TS_2, 0)}, A_5 : \langle R_k^* \rangle_{h(D_j, TS_2, 1)}, VM_5 : h(RID_j, D_j, TID_k, R_i, TS_2, R_k^*), TS_2) \]

From Rule 1 and **AS₆**, \( V_{11}; RM_j \equiv G_i \approx R_i \).

From Rule 2 and **AS₇**, \( V_{12}; RM_j \equiv G_i \equiv R_i \).

Then, from Rule 3 and **AS₁₃**, ...
V_{13}: RM_j \models R_i.
According to Rule 4, AS_7 and V_{12},
V_{14}: RM_j \models RM_j^{K_i} G_i.
Further, using Rule 2, AS_7 and V_{11},
V_{15}: RM_j \models G_i \models RM_j^{K_i} G_i. \quad \text{Goal 3}
Similarly, by using Message \textbf{M}_4,
S_k \not\in \{ A_j : \langle R_i \rangle_{h(A_i^j, TS_k)}, A_8 : \langle R_i \rangle_{h(A_i^j, TS_k)}, A_9 : \langle A_2^{new} \rangle_{h(A_i^j, TS_k)} \}
VM_j : h(A_2^{new}, TS_k), TS_k).
From Rule 1 and AS_6,
V_{16}: S_k \models G_i \models R_i.
From Rule 2 and AS_8,
V_{17}: S_k \models G_i \models R_i.
Then, from Rule 3 and AS_9,
V_{18}: S_k \models R_i.
According to Rule 4, AS_8 and V_{17},
V_{19}: S_k \models S_k^{K_i} G_i.
Further, using Rule 2, AS_8 and V_{16},
V_{20}: S_k \models G_i \models S_k^{K_i} G_i. \quad \text{Goal 4}
By using Message \textbf{M}_4,
V_{21}: S_k \models RM_j \models R_j.
From Rule 2 and AS_2,
V_{22}: S_k \models RM_j \models R_j.
Then, from Rule 3 and AS_10,
V_{23}: S_k \models R_j.
According to Rule 4, AS_2 and V_{22},
V_{24}: S_k \models S_k^{K_i} RM_j.
Further, using Rule 2, AS_2 and V_{21},
V_{25}: S_k \models RM_j \models S_k^{K_i} RM_j. \quad \text{Goal 5}
By using Message \textbf{M}_2,
V_{26}: RM_j \models S_k \models R_k.
From Rule 2 and AS_7,
V_{27}: RM_j \models S_k \models R_k.
Then, from Rule 3 and AS_14,
V_{28}: RM_j \models R_k.
According to Rule 4, AS_7 and V_{27},
V_{29}: RM_j \models RM_j^{K_i} S_k.
Further, using Rule 2, AS_7 and V_{26},
V_{30}: RM_j \models S_k \models RM_j^{K_i} S_k. \quad \text{Goal 6}
The proof is concluded.

4.2. Security Analysis

The security requirements of the enhanced scheme are discussed in this subsection. The enhances scheme uses the properties of the scheme proposed by Kamil et al. [9]. The arguments of some security requirements, including provision of strong anonymity; session key establishment; perfect forward secrecy; and resistance to replay attacks, impersonation attacks, offline user login credentials guessing attacks, insider attacks, mobile device loss attacks, and denial of service attacks, are similar to those in the scheme proposed by Kamil et al. and are therefore not discussed here. These security requirements include resistance to robotic arm compromise attacks and resistance to stolen verifier table attacks, as described below.

4.2.1. Resistance to Robotic Arm Compromise Attacks

In the enhanced scheme, even if the attacker compromises the robotic arm (RM_{i}) and obtains (RID_{j}, D_{i}) from RM_{j}, the attacker cannot indirectly obtain information about
remote surgeons and the gateway \((G_i)\). Additionally, because the \((RID_j, D_j)\) of each robotic arm is independent, as destroying a robotic arm, the attacker can communicate with \(S_k\), but it does not affect the security of \(S_k\’s\) communication with other robotic arms. The same is true for the gateway. Therefore, the proposed scheme is resilient against robot compromise attack.

4.2.2. Resistance to Stolen Verifier Attacks

In the enhanced scheme, the gateway \((G_i)\) stores \((RID_j, CD_j)\) instead of \((RID_j, D_j)\), where \(CD_j = D_j \oplus h(RID_j \parallel D_j)\), \(D_j\) is the secret key of \(RM_j\), and \(D_i\) is the secret key of \(G_i\). The verifier table does not contain \(G_i\’s\) secret key \((D_i)\). Then, an attacker who has stolen the verifier table cannot derive \(D_j\) from \((RID_j, CD_j)\) without \(D_i\), and it is difficult to impersonate \(RM_j\). Therefore, the enhanced scheme is resilient against stolen verifier table attacks.

4.3. Functionality Comparison

Table 3 compares the enhanced AKA scheme with related AKA schemes in term of security functionality. The enhanced AKA scheme provides more security requirements than related AKA schemes and is secure against potential attacks. Furthermore, it can resist robotic arm compromise attacks and stolen verifier table attacks.

Table 3. Functionality comparisons.

| Security Attribute                                      | [11] | [5] | [6] | [7] | [14] | [15] | [16] | Our AKA |
|--------------------------------------------------------|------|-----|-----|-----|------|------|------|--------|
| Provision of strong anonymity                          | O    | O   | X   | O   | X    | O    | O    | O      |
| Provision of session key establishment                 | O    | -   | O   | -   | O    | O    | O    | O      |
| Provision of perfect forward secrecy                   | O    | O   | O   | O   | O    | O    | O    | O      |
| Resistance to replay attacks                           | O    | X   | X   | X   | O    | O    | O    | O      |
| Resistance to impersonation attacks                    | O    | X   | O   | O   | O    | O    | O    | O      |
| Resistance to offline user login credentials guessing attack | O    | X   | O   | O   | O    | O    | O    | O      |
| Resistance to insider attacks                          | O    | -   | O   | O   | O    | O    | O    | O      |
| Resistance to mobile device loss attacks               | O    | X   | O   | O   | O    | O    | O    | O      |
| Resistance to denial of service attacks                | O    | O   | O   | O   | O    | O    | O    | O      |
| Resistance to robotic arm compromise attacks           | X    | O   | O   | O   | O    | O    | O    | O      |
| Resistance to stolen verifier attacks                  | X    | O   | O   | X   | X    | O    |      |        |

O: the property is satisfied, X: the property is not satisfied; -: the property is not considered.

4.4. Performance Comparisons

Table 4 shows comparisons between the enhanced AKA scheme and related AKA schemes in terms of computational cost, where \(T_h\) denotes the execution time of a one-way hash function, \(T_e\) denotes the execution time of a point multiplication based on ECC, and \(T_f\) denotes the execution time of a fuzzy extractor. The experiment is run on an Intel CPU i3-3220 3.3 Ghz, RAM 4096 MB, Windows 7 Professional 64-bit, Eclipse Java Mars and Java SE 1.8. The hash function uses SHA-1, the point multiplication is based on ECC with a 16-bit key, and the fuzzy extractor refers to [11,17].

The scheme proposed by Kamil et al. [11] requires 20 hash operations, the scheme proposed by Amin et al. [5] requires 37 hash operations, the scheme proposed by Wu et al. [6] requires 34 hash operations, the scheme proposed by Chandrakar [7] requires 29 hash operations, the scheme proposed by Guo et al. [14] requires 36 hash operations, and our enhanced scheme requires 35 hash operations. The scheme proposed by Soni et al. [15] requires 31 hash operations, 6 point multiplications based on ECC, and 11 fuzzy extractor operations. The scheme proposed by Li et al. [16] requires 20 hash operations and 8 point multiplications based on ECC. Both these schemes ([15,16]) require time-consuming point multiplications based on ECC. The enhanced AKA scheme adopts a one-time key to protect communication messages and protects the verifier table with the \(G_i\’s\) secret...
key, so it requires more computations and response time than the AKA protocol proposed by Kamil et al. However, the enhanced AKA scheme addresses the limitations of the scheme proposed by Kamil et al., providing improved functionality while retaining a low computational cost.

Table 4. Computation cost comparison.

| Scheme | Mobile Device/User | Gateway | Sensor Node/Robotic Arm | Total/Response Time |
|--------|-------------------|---------|-------------------------|-------------------|
| [11]   | 8T_h              | 8T_h    | 4T_h                    | 20T_h/240 ms.     |
| [5]    | 12T_h             | 19T_h   | 6T_h                    | 37T_h/444 ms.     |
| [6]    | 11T_h             | 17T_h   | 6T_h                    | 34T_h/408 ms.     |
| [7]    | 11T_h             | 13T_h   | 5T_h                    | 29T_h/348 ms.     |
| [14]   | 13T_h             | 17T_h   | 6T_h                    | 36T_h/432 ms.     |
| [15]   | 13T_h + 3T_e + 13T_f | 11T_h + 3T_e | 7T_h | 31T_h + 6T_e + 13T_f/1645 ms. |
| [16]   | 8T_h + 3T_e      | 8T_h + 3T_e | 4T_h + 2T_e | 20T_h + 8T_e/776 ms. |
| Our AKA| 13T_h             | 16T_h   | 6T_h                    | 35T_h/420 ms.     |

5. Conclusions

In this paper, we addressed the limitations of the AKA scheme proposed by Kamil et al. for a Tactile Internet environment, including failure to resist robotic arm compromise attacks, failure to resist stolen verifier attacks, and failure to execute correctly. In order to address these limitations, an enhanced AKA scheme based the scheme proposed by Kamil et al. was developed by adopting a one-time key to protect communication messages and protecting the verifier table with a gateway secret key. Although the enhanced scheme requires more computations than the AKA protocol proposed by Kamil et al. it retains a low computational cost and provides more security features. Therefore, the enhanced AKA scheme is suitable for the Tactile Internet environment.

Author Contributions: Formal analysis, X.Y.; Funding acquisition, T.-F.L.; Investigation, W.-Y.C.; Methodology, X.Y.; Software, W.-Y.C.; Supervision, T.-F.L.; Validation, C.-C.C.; Visualization, W.-Y.C.; Writing—original draft, X.Y. and C.-C.C.; Writing—review & editing, T.-F.L. and C.-C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Science and Technology Council under grants MOST 109-2221-E-320-003, MOST 110-2221-E-320-005-MY2, MOST 110-2221-E-040-004-MY2 and TCRP109001. The authors thank Ted Knoy for his editorial support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References
1. Maier, M.; Chowdhury, M.; Rimal, B.P.; Van, D.P. The tactile internet: Vision, recent progress, and open challenges. *IEEE Commun. Mag.* 2016, 54, 138–145. [CrossRef]
2. Shafiq, A.; Ayub, M.F.; Mahmood, K.; Sadiq, M.; Kumari, S.; Chen, C.-M. An Identity-Based Anonymous Three-Party Authenticated Protocol for IoT Infrastructure. *J. Sens.* 2020, 2020, 8829319. [CrossRef]
3. Simsek, M.; Aijaz, A.; Dohler, M.; Sachs, J.; Fettweis, G. 5G-enabled tactile internet. *IEEE J. Sel. Areas Commun.* 2016, 34, 460–473. [CrossRef]
4. Fettweis, G.P. The tactile internet: Applications and challenges. *IEEE Veh. Technol. Mag.* 2014, 9, 64–70. [CrossRef]
5. Amin, R.; Islam, S.H.; Biswas, G.P.; Khan, M.K.; Kumar, N. A robust and anonymous patient monitoring system using wireless medical sensor networks. *Future Gener. Comput. Syst.* 2018, 80, 483–495. [CrossRef]
6. Wu, F.; Li, X.; Sangaiah, A.K.; Xu, L.; Kumari, S.; Wu, L.; Shen, J. A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks. *Future Gener. Comput. Syst.* 2018, 82, 727–737. [CrossRef]
7. Chandrakar, P.A. Secure Remote User Authentication Protocol for Healthcare Monitoring Using Wireless Medical Sensor Networks. *Int. J. Ambient Comput. Intell.* 2019, 10, 96–116. [CrossRef]

8. Kaur, K.; Garg, S.; Kaddoum, G.; Guizani, M. Secure authentication and key agreement protocol for tactile internet-based tele-surgery ecosystem. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6.

9. Nykvist, C.; Larsson, M.; Sodhro, A.H.; Gurtov, A. A lightweight portable intrusion detection communication system for auditing applications. *Int. J. Commun. Syst.* 2020, 33, e4327. [CrossRef]

10. Bolton, T.; Dargahi, T.; Belguith, S.; Al-Rakhami, M.; Sodhro, A. On the Security and Privacy Challenges of Virtual Assistants. *Sensors* 2021, 21, 2312. [CrossRef]

11. Kamil, I.A.; Ogundoyin, S.O. A lightweight mutual authentication and key agreement protocol for remote surgery application in Tactile Internet environment. *Comput. Commun.* 2021, 170, 1–18. [CrossRef]

12. Wazid, M.; Das, A.K.; Lee, J.H. User authentication in a tactile internet based remote surgery environment: Security issues, challenges, and future research directions. *Pervasive Mob. Comput.* 2019, 54, 71–85. [CrossRef]

13. Burrows, M.; Abadi, M.; Needham, R.M. A logic of authentication. *Proc. R. Soc. Lond. A Math. Phys. Sci.* 1989, 426, 233–271.

14. Guo, H.; Xu, Y.G.T.; Zhang, X.; Ye, J. A secure and efficient three-factor multigateway authentication protocol for wireless sensor networks. *Ad Hoc Netw.* 2019, 95, 101965. [CrossRef]

15. Soni, P.; Pal, A.K.; Islam, S.H. An improved three-factor authentication scheme for patient monitoring using WSN in remote health-care system. *Comput. Methods Programs Biomed.* 2019, 182, 105054. [CrossRef] [PubMed]

16. Li, X.; Peng, J.; Obaidat, M.S.; Wu, F.; Khan, M.K.; Chen, C. A secure three-factor user authentication protocol with forward secrecy for wireless sensor network systems. *IEEE Syst. J.* 2020, 4, 39–50. [CrossRef]

17. He, D.; Kumar, N.; Lee, J.-H.; Sherratt, R.S. Enhanced three-factor security protocol for consumer USB mass storage devices. *IEEE Trans. Consum. Electron.* 2014, 60, 30–37. [CrossRef]