applied sciences

MDPI

Article

Light-Weighted Password-Based Multi-Group Authenticated Key Agreement for Wireless Sensor Networks

Mao-Sung Chen [1], I-Pin Chang [2,*] and Tung-Kuan Liu [1]

[1] Institute of Engineering Science and Technology, National Kaohsiung University of Science and Technology, Kaohsiung City 80778, Taiwan; sjm071977@gmail.com (M.S.C.); tkliu@nkust.edu.tw (T.-K.L.)

[2] Department of Digital Applications, University of Kang Ning, Tainan 708, Taiwan

* Correspondence: ipinchang0315@gmail.com; Tel.: +886-6-255-2500 (ext. 33300)

Received: 8 August 2019; Accepted: 4 October 2019; Published: 14 October 2019

Abstract: Security is a critical issue for medical and health care systems. Password-based group-authenticated key agreement for wireless sensor networks (WSNs) allows a group of sensor nodes to negotiate a common session key by using password authentication and to establish a secure channel by this session key. Many group key agreement protocols use the public key infrastructure, modular exponential computations on an elliptic curve to provide high security, and thus increase sensor nodes' overhead and require extra equipment for storing long-term secret keys. This work develops a novel group key agreement protocol using password authentication for WSNs, which is based on extended chaotic maps and does not require time-consuming modular exponential computations or scalar multiplications on an elliptic curve. Additionally, the proposed protocol is suitable for multiple independent groups and ensures that the real identities of group members cannot be revealed. The proposed protocol is not only more secure than related group key agreement protocols but also more efficient.

Keywords: sensor networks; chaotic maps; group authentication; extended chaotic maps

1. Introduction

A security association which manages security in a network layer is an important matter and it involves the establishment of a shared security key between two end points to support secure associations [1]. Wireless sensor networks (WSNs) consist of a large number of sensor nodes, which cannot support heavy computations, extensive communications or extensive storage and have limited bandwidth. They can be applied in many environments, such as medical monitors, military reconnaissance and communication, and others. WSNs are deployed to allow a legitimated user to login to the network and access data. The sensor node authentication has become one of the important security issues [2]. Group authenticated key agreements for WSNs enable a group of sensor nodes to authenticate each other and to establish a common key for securely communicating over public sensor networks. Group authenticated key agreement protocols typically fall into two categories, which are group key agreement protocols without public keys and group key agreement protocol using public keys. The former realize authentication and negotiates a group key using shared weak passwords or a shared long-term secret key [3–6], while the latter realize authentication and negotiate a common group key using public key systems [7–9]. Most group key agreement protocols that use public keys have higher security than those without. However, they depend on time-consuming modular exponential computations and scalar multiplications on an elliptic curve, and thus are not suitable for sensor networks. Recently, several group-authenticated key agreement approaches have been presented.

Unfortunately, most of these protocols were developed for two communication entities (two-party) or three communication entities (three-party), and can only be extended to a group key agreement protocol with difficulty. Thus, most authenticated key agreement protocols are difficult to extend to multi-group authenticated key agreement for WSNs.

A multi-group key agreement protocol for WSNs allows communicating entities (sensor nodes) to belong to multiple groups, and enables each group to establish an independent group session key. A key hypergraph [9–11] is a graph where each vertex represents a party and each hyper-edge represents a relation among parties who to share a key. For instance, group members SN_1, SN_2, SN_3, SN_4 and SN_5 involve groups $\{SN_1, SN_2, SN_3\}$, $\{SN_3, SN_4, SN_5\}$, and $\{SN_1, SN_2, SN_3, SN_4, SN_5\}$ and establish group keys used for secure communication. Then its key hypergraph can be denoted as $G = \{V, E\}$, where $V = \{SN_1, SN_2, SN_3, SN_4, SN_5\}$ is a finite set of vertices and $E = \{e_1 = \{SN_1, SN_2, SN_3\}, e_2 = \{SN_3, SN_4, SN_5\}, e_3 = \{SN_1, SN_2, SN_3, SN_4, SN_5\}\}$ is a set of subsets of V, as presented in Figure 1.

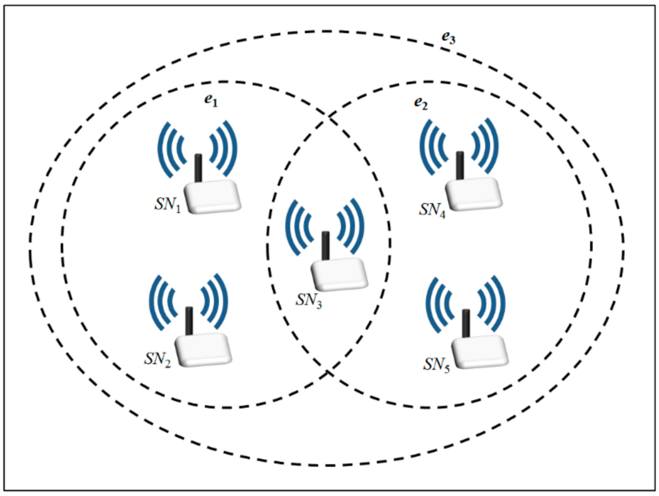

Figure 1. In a key hypergraph, one sensor node is allowed to belong to multiple independent groups.

Key management issues are also considered to have a major impact on the security scale of WSNs [12]. Recently, several group-authenticated key agreement approaches have been presented for WSNs. For example, in 2007, Jeong and Lee [9] proposed a group-authenticated key agreement protocol that uses a public key system to build a session key; these group key approaches can be extended for hypergraphs and are suitable for use with multiple groups [9–11].

Users also need extra storage, such as radio frequency identification (RFID) tags, flash drives, smart cards and so on, to store public/private key pairs. In 2006, Abdalla et al. [3] developed a password-based group-authenticated key exchange that can be executed in a constant number of rounds. In the same year, Dutta and Barua [13] proposed a password-based encrypted group-authenticated key agreement protocol. Although these approaches do not require the maintenance of public key systems, all communicating users share the same password so these protocols do not protect the privacy of users. In 2013, Lee et al. [14] proposed a password-based group-authenticated key agreement protocol for the integrated electronic patient record (EPR) information system, which enabled users to have their own passwords. A multi-server authentication protocol based on dynamic identity is proposed by Sood et al. [15]. Amin et al. [16] demonstrated that Xue et al.'s protocol [17] is not protected against the user anonymity problem and cannot resist user impersonation and session key discloser attack. In 2017, Lin et al. [18] applied an extended chaotic map to present password-less group authentication key agreement which improves the computation efficiency for the simple group password-based authenticated key agreement (SGPAKE) proposed by Lee et al [19]. Although most limitations in the field of security have been overcome, the above protocols require many time-consuming modular exponential computations or scalar multiplications on elliptic curves and so are inefficient and unsuitable for use in many practical scenarios. Moreover, most of them are difficult to extend to multiple groups.

Recently, a number of key agreement protocols based on chaotic maps were proposed, which have improved computational efficiency. Using Chebyshev chaotic maps has been shown to be more efficient than cryptography using modular exponential computations and scalar multiplications on elliptic curves [20–29]. However, Chebyshev chaotic maps and their enhancement are affected by the discrete logarithm problem and the Diffie–Hellman problem [30–33]. In addition, most of them were developed for two communication entities (two-party) or three communication entities (three-party), and can only be extend to group key agreement protocol with difficulty.

In our analysis, we present a novel password-based multi-group authenticated key agreement protocol for WSNs that was based on the extended chaotic map-based Diffie–Hellman problem. The main contributions of this paper are:

(1). The proposed protocol enables one sensor node to belong to several mutually independent groups and ensures group key security. Additionally, the real identities of group members cannot be revealed.

(2). Accordingly, the proposed protocol is suitable for multiple groups and ensures users' anonymity. It overcomes not only the limitations of previously proposed protocols and has a lower computational cost, but also offers greater security and is suitable for WSNs.

The remainder of this paper is organized as follows. The primitives used are described in Section 2. The proposed extended chaotic map-based multi-group authenticated key agreement protocol is illustrated in Section 3. In Section 4, we presented the security analysis and overall comparison. The conclusions are drawn in Section 5.

2. Preliminaries

This section lists notations and describes the underlying primitives used in this paper. The underlying primitives include Chebyshev polynomials, enhanced Chebyshev chaotic maps, the extended chaotic map-based discrete logarithm and Diffie–Hellman problems [30–33] which are described as follows and Table 1 lists the symbol system applied by the proposed solution.

Table 1. The symbol system applied by the proposed solution.

Symbol	Definition
SN_i	The sensor node for $i = 1, 2, \ldots, n$.
ID_i	The identity of sensor node i (SN_i.)
pw_i	The password of sensor node i (SN_i.)
AS	The trusted authentication server.
$h(.)$	One-way hash function.
A→B: M	A sends messages (M) to B by a common channel.
A⇒B: M	A sends message (M) to B by a secure channel.
$M_1 \| M_2$	Message 1(M_1) concatenates to message 2(M_2).

2.1. Chebyshev Polynomials

The Chebyshev polynomials of degree n are defined as:

$$\begin{cases} T_0(x) = 1; \\ T_1(x) = x;\ \text{and} \\ T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \text{for } n \ \geq 2, \end{cases} \tag{1}$$

and the first few Chebyshev polynomials are

$$\begin{cases} T_2(x) = 2x^2 - 1, \\ T_3(x) = 4x^3 - 3x, \\ T_4(x) = 8x^4 - 8x^2 + 1. \end{cases} \quad (2)$$

2.2. Semigroup Property

We have $T_{\mathrm{r}}(T_s(x)) = T_{rs}(x)$ for different r and s, where $-1 \le x \le 1$. The core idea of semi-group is similar to the Diffie–Hellman problem. Semi-group implies that there is not a specific order for r and s. This property comes from Chebyshev polynomials. However, $-1 \le x \le 1$ is not enough to provide high security in terms of the diversity of x, and Zhang extends the mapping range from $(-1,1)$ to $(-\infty, \infty)$ [33]. In other words, the scheme with a semi-group property has similar security to that of the Diffie–Hellman key exchange [34].

2.3. Enhanced Chebyshev Polynomials

In order to enhance the property of the Chebyshev chaotic map, Zhang [19] proved that the semi-group property holds for Chebyshev polynomials defined on interval $(-\infty, +\infty)$. This paper uses the following enhanced Chebyshev polynomials:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \ mod \ p \ \ for \ n \ge 2. \quad (3)$$

The enhanced Chebyshev polynomials meet the semi-group property. Then,

$$T_r(T_s(x)) \equiv T_{rs}(x) \equiv T_s(T_r(x)) \ mod \ p. \quad (4)$$

2.4. Extended Chaotic Map-Based Discrete Logarithm (ECM-DL) Problem

Given x, y, and p, it is not computationally feasible to find the satisfied integer r,

$$\mathrm{y} = T_r(x) \ mod \ p. \quad (5)$$

2.5. Extended Chaotic Map-Based Diffie-Hellman (ECM-DH) Problem

Given $T_u(x)$ *mod p*, $T_v(x)$ *mod p*, $T(\bullet)$, x, and p, where $u, v \ge 2$, $x \in (-\infty, +\infty)$, and p is a large prime number, the calculations are not feasible.

$$T_{uv}(x) \equiv T_u(T_v)) \equiv T_v(T_u)) \ mod \ p. \quad (6)$$

3. Proposed Multi-Group Authenticated Key Agreement Protocol for WSNs

This section presents a group authenticated key agreement protocol using extended chaotic maps for hypergraphs. The proposed protocol enables one user to belong to several independent groups, ensures group key security, and protects the real identities of group members. The proposed protocol is composed of four phases, which are the initialization phase, registration phase, the authentication and key agreement phase and the password change phase, and it is implemented as follows.

3.1. Initialization Phase

Step 1: The authentication server *AS* randomly selects *mk* as its master key.
Step 2: *AS* computes $pk_s = T_{mk}(x)$ *mod p*, where x is a random number and p is a large prime number.
Step 3: *AS* publishes parameters $(pk_s, T(.), x, p)$.

3.2. Registration Phase

Step 1: SN_i⇒S:{$\mathrm{ID_i}$, $\mathrm{pw_i}$}

The sensor node SN_i chooses his/her identity ID_i and password pw_i, and sends {ID_i, pw_i, $Group_i$} to AS over a secure channel, where $Group_i$ = (G_{i1}, G_{i2}, … ,G_{iN}) and G_{i1}, G_{i2}, … ,G_{iN} are groups that SN_i belong to.

Step 2: Upon receiving the register message from SN_i, The trusted authentication server (AS) computes $HID_i = h(ID_i \| mk)$, $Q_i = h(ID_i \| pw_i)$ and stores (HID_i, Q_i, $Group_i$).

3.3. Authentication and Key Agreement Phase

This phase, as shown in Figure 2, enables sensor nodes SN_i for $i = 1,2, \dots ,n$ to authenticate each other and to negotiate session keys for each group with the help of AS. First, sensor node SN_i sends its password pw_i to AS, which is encrypted with a secret key of SN_i and AS. After AS successfully authenticates SN_i, AS assists these sensor nodes in agreeing a common secret key as their group session key. The details are worked as follows.

Step 1. $\mathrm{SN_i} \rightarrow \mathrm{AS} : \mathrm{M_{i,1}} = \{\mathrm{DID_i}, \mathrm{X_i}, \mathrm{C_i}, \mathrm{T_i}\}$

Each sensor node $\mathrm{SN_i}$ chooses a nonce $\mathrm{r_i}$, computes $\mathrm{K_1} = \mathrm{T_{r_i}(pk_S)}$ mod p, $\mathrm{DID_i{=}K_1 \oplus ID_i}$, $\mathrm{X_i} = \mathrm{T_{r_i}(x)}$ mod p, $\mathrm{Q_i{=}h(ID_i \| pw_i)}$, $\mathrm{C_i = h(DID_i \| Q_i \| K_1 \| X_i \| C_i, \| T_i)}$, where $\mathrm{T_i}$ is the current timestamp, and sends $\mathrm{M_{i,1}} = \{\mathrm{DID_i}, \mathrm{X_i}, \mathrm{C_i}, \mathrm{T_i}\}$ to AS.

Step 2. $\mathrm{AS} \rightarrow \mathrm{SN_i} : \mathrm{M_{i,2}} = \{\mathrm{Y_{i\text{-}1}}, \mathrm{Y_{i+1}}, \mathrm{HGP_{i,m}}, \mathrm{C_S}, \mathrm{T_S}\}$

After receiving $M_{i,1}$, AS checks the validity of T_i. If successful, AS computes $K_1{'} = T_{mk}(X_1)\ mod\ p$, $ID_i{'} = DID_i \oplus \mathrm{K_1}{'}$, $HID_i{'} = h(ID_i{'} \| \mathrm{m_k})$, retrieves ($Q_i$, $Group_i$) by $HID_i{'}$, and checks $C_i = h(DID_i \| Q_i \| K_1{'} \| X_i \| T_i)$. If successful, AS chooses a nonce r_S, computes $Y_i = T_{rs}(X_i)\ mod\ p$, constructs a group identity GID_{im} = (DID_1, DID_2, … ,DID_i, … ,DID_j, …) by using sensor nodes' temporal identity DID_i and calculates $HGP_{i,m} = h(K_1{'} \| G_{im} \| T_S) \oplus GID_{im}$, $CS_i = h(K_1{'} \| Q_i \| Y_{i\text{-}1} \| y_{i+1} \| GID_{im} \| T_S)$ for $i = 1,2, \dots ,n$, where T_S is the current timestamp and SN_i is a group member of G_{im} for $m = 1,2, \dots ,N$, and sends $M_{i,2} = \{Y_{i\text{-}1}, Y_{i+1}, HGP_{i,m}, C_S, T_S\}$ to U_i.

Step 3. $\mathrm{SN_i} \rightarrow * : \mathrm{M_{i,3}} = \{\mathrm{DID_i}\ ,\mathrm{W_{i,m}}\}$

SN_i checks $\mathrm{T_S}$, calculates $GID_{im} = h(K_1 \| G_{im} \| T_S) \oplus\ HGP_{i,m}$ and verifies $CS_i = h(K_1 \| Q_i \| Y_{i\text{-}1} \| y_{i+1} \| GID_{im} \| T_S)$. If successful, SN_i computes $Z_{i\text{-}1} = T_{r_i}\ (Y_{i\text{-}1})\ mod\ p$, $Z_i = T_{r_i}\ (Y_{i+1})\ mod\ p$ and $W_{i,m} = Z_i\ /\ Z_{i\text{-}1}$, and broadcasts $M_{i,3} = \{DID_i\ ,W_{i,m}\}$.

Step 4. $\mathrm{SN_i} \rightarrow * : \mathrm{M_{i,4}} = \{\mathrm{DID_i}\ ,\mathrm{Auth_{i,m}}\}$

After receiving $M_{i,3}$ for $j \neq i$, if SN_i computes $sk_{i,m} = (Z_i)^n \times (W_{i+1})^{n\text{-}1} \times (W_{i+2})^{n\text{-}2} \times \dots \times (W_{i\text{-}1})^1$ and key confirmation $Auth_{i,m} = h(DID_i \| sk_{i,m} \| GID_{im} \| T_S)$, and broadcasts $M_{i,4} = \{DID_i\ ,Auth_{i,m}\}$.

Step 5. Finally, SN_i authenticates SN_j by checking $Auth_{i,m} = h(DID_i \| sk_{i,m} \| GID_{im} \| T_S)$ for $j \neq i$, and computes $sk_m = h(GID_{im} \| sk_{i,m})$ for the group G_{im}.

3.4. Password Change Phase

A legal sensor nodes SN_i changes its password by performing the following steps.

Step 1. $\mathrm{SN_i} \rightarrow \mathrm{AS} : \mathrm{M_{i,1}} = \{\mathrm{DID_i}, \mathrm{X_i}, \mathrm{C_i}, \mathrm{T_i}\}$

$\mathrm{SN_i}$ chooses a nonce $\mathrm{r_i}$, computes $\mathrm{K_1} = \mathrm{T_{r_i}(pk_S)}$ mod p, $\mathrm{DID_i} = \mathrm{K_1 \oplus ID_i}$, $\mathrm{X_i} = \mathrm{T_{r_i}(x)}$ mod p, $\mathrm{Q_i = h(ID_i \| pw_i)}$, $\mathrm{Q_{i\_new} = h(ID_i \| pw_{i\_new})}$, $\mathrm{D_i = h(K_1 \| T_i) \oplus Q_{i\_new}}$, $\mathrm{E_i = h(DID_i \| Q_i \| Q_{i\_new} \| K_1 \| X_i \| T_i)}$, where $\mathrm{T_i}$ is the current timestamp, and sends $\mathrm{M_{i,1}} = \{\mathrm{DID_i}, \mathrm{X_i}\ ,\mathrm{D_i}, \mathrm{E_i}, \mathrm{T_i}\}$ to AS.

Each sensor node $\mathrm{SN_i}$ chooses a nonce $\mathrm{r_i}$, computes $\mathrm{K_1} = \mathrm{T_{r_i}(pk_S)}$ mod p, $\mathrm{DID_i} = \mathrm{K_1 \oplus ID_i}$, $\mathrm{X_i} = T_{r_i}\ \mathrm{(x)}$ mod p, $\mathrm{Q_i = h(ID_i \| pw_i)}$, $\mathrm{C_i = h(DID_i \| Q_i \| K_1 \| X_i \| C_i, \| T_i)}$, where $\mathrm{T_i}$ is the current timestamp, and sends $\mathrm{M_{i,1}} = \{\mathrm{DID_i}, \mathrm{X_i}, \mathrm{C_i}, \mathrm{T_i}\}$ to AS.

Step 2. $\mathrm{AS} \rightarrow \mathrm{SN_i} : \mathrm{M_{i,2}} = \{\mathrm{V_{S_i}}, \mathrm{T_S}\}$

After receiving $M_{i,1}$, AS checks the validity of T_i. If successful, AS computes $K_1{'} = T_{mk}(X_1)\ mod\ p$, $ID_i{'} = DID_i \oplus K_1{'}$, $HID_i{'} = h(ID_i{'} \| \mathrm{m_k})$, retrieves ($Q_i$, $Group_i$) by $HID_i{'}$, computes $Q{'}_{i\_new} = D_i \oplus h(K_1 \| T_i)$ and checks $E_i = h(DID_i \| Q_i \| Q{'}_{i\_new} \| K{'}_1 \| X_i \| T_i)$. If successful, AS replaces Q_i with $Q{'}_{i\_new}$, and calculates

$V_{S_i} = h(K_1{}'||Q_i||Q'_{i\_new}||K'_1||X_i||T_s)$, where T_s is the current timestamp, and sends $M_{i,2} = \{V_{S_i}, T_S\}$ to SN_i. Finally, SN_i makes sure that AS has updated SN_i's verification data in S's database by validating T_s and checking $V_{S_i} = h(K_1||Q_i||Q_{i\_new}||K'_1||X_i||T_s)$.

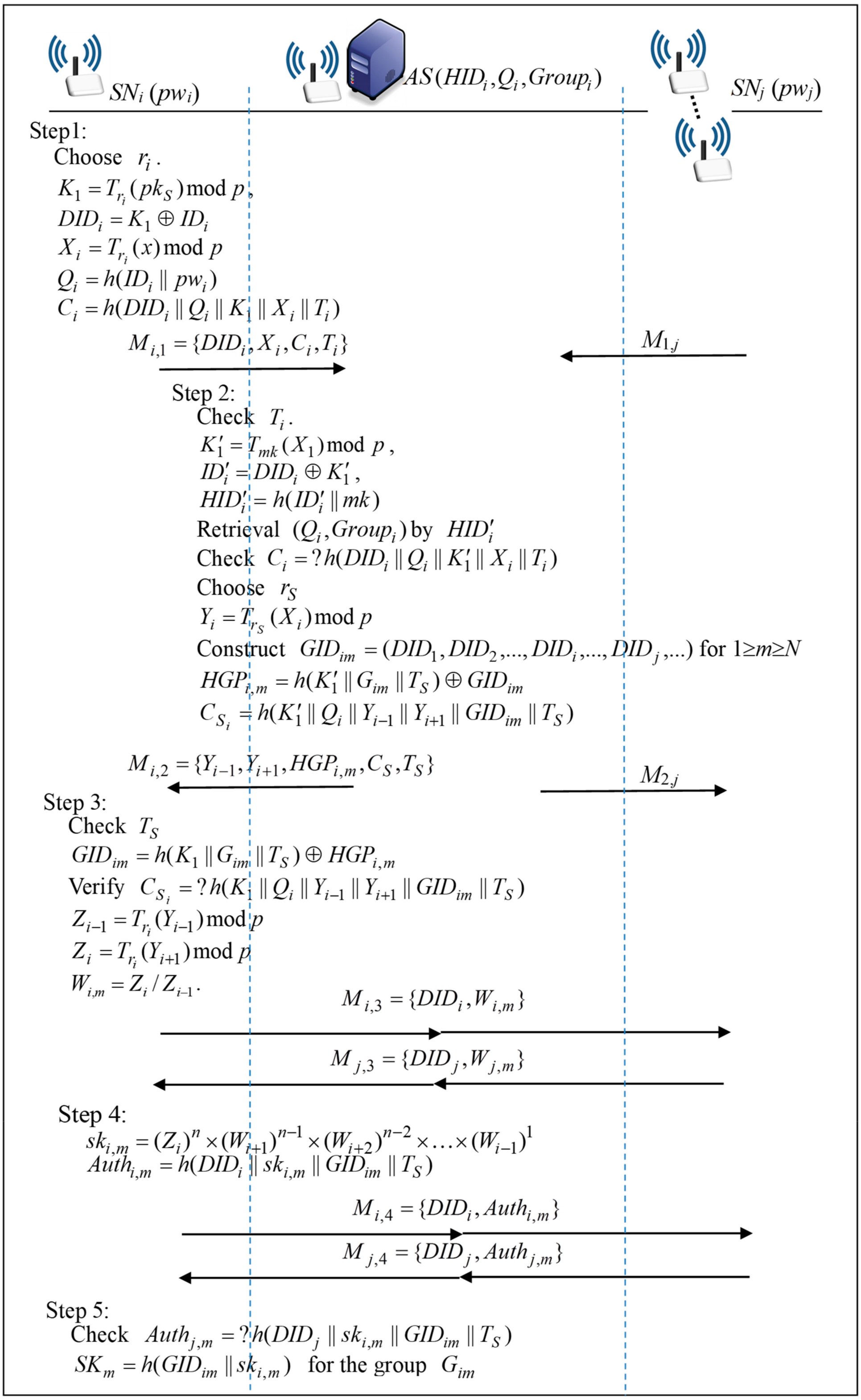
$SN_i\ (pw_i)$
$AS(HID_i, Q_i, Group_i)$
$SN_j\ (pw_j)$
Step1:
Choose r_i.
$K_1 = T_{r_i}(pk_S) \bmod p$,
$DID_i = K_1 \oplus ID_i$
$X_i = T_{r_i}(x) \bmod p$
$Q_i = h(ID_i \| pw_i)$
$C_i = h(DID_i \| Q_i \| K_1 \| X_i \| T_i)$
$M_{i,1} = \{DID_i, X_i, C_i, T_i\}$
$M_{1,j}$
Step 2:
Check T_i.
$K_1' = T_{mk}(X_1) \bmod p$,
$ID_i' = DID_i \oplus K_1'$,
$HID_i' = h(ID_i' \| mk)$
Retrieval $(Q_i, Group_i)$ by HID_i'
Check $C_i = ?\, h(DID_i \| Q_i \| K_1' \| X_i \| T_i)$
Choose r_S
$Y_i = T_{r_S}(X_i) \bmod p$
Construct $GID_{im} = (DID_1, DID_2, \ldots, DID_i, \ldots, DID_j, \ldots)$ for $1 \geq m \geq N$
$HGP_{i,m} = h(K_1' \| G_{im} \| T_S) \oplus GID_{im}$
$C_{S_i} = h(K_1' \| Q_i \| Y_{i-1} \| Y_{i+1} \| GID_{im} \| T_S)$
$M_{i,2} = \{Y_{i-1}, Y_{i+1}, HGP_{i,m}, C_S, T_S\}$
$M_{2,j}$
Step 3:
Check T_S
$GID_{im} = h(K_1 \| G_{im} \| T_S) \oplus HGP_{i,m}$
Verify $C_{S_i} = ?\, h(K_1 \| Q_i \| Y_{i-1} \| Y_{i+1} \| GID_{im} \| T_S)$
$Z_{i-1} = T_{r_i}(Y_{i-1}) \bmod p$
$Z_i = T_{r_i}(Y_{i+1}) \bmod p$
$W_{i,m} = Z_i / Z_{i-1}$.
$M_{i,3} = \{DID_i, W_{i,m}\}$
$M_{j,3} = \{DID_j, W_{j,m}\}$
Step 4:
$sk_{i,m} = (Z_i)^n \times (W_{i+1})^{n-1} \times (W_{i+2})^{n-2} \times \ldots \times (W_{i-1})^1$
$Auth_{i,m} = h(DID_i \| sk_{i,m} \| GID_{im} \| T_S)$
$M_{i,4} = \{DID_i, Auth_{i,m}\}$
$M_{j,4} = \{DID_j, Auth_{j,m}\}$
Step 5:
Check $Auth_{j,m} = ?\, h(DID_j \| sk_{i,m} \| GID_{im} \| T_S)$
$SK_m = h(GID_{im} \| sk_{i,m})$ for the group G_{im}

Figure 2. The proposed multi-group authenticated key agreement protocol for wireless sensor networks (WSNs).

4. Security Analysis

The security analyses on the correctness, session key security, perfect forward security, mutual authentication, and privacy protection are provided; it also resists password guessing, known-key attacks, and sensor node capture attacks.

4.1. Correctness

All legal users have the same secret $sk_{i,m}$ since SN_i computes

$$\begin{aligned} sk_{i,m} &= (Z_i)^n \cdot (W_{i+1,m})^{n-1} \cdot (W_{i+2,m})^{n-2} \cdot \ldots \cdot (W_{i-1,m})^1 \\ &= (Z_i)^n \cdot (\tfrac{Z_{i+1}}{Z_i})^{n-1} \cdot (\tfrac{Z_{i+2}}{Z_{i+1}})^{n-2} \cdot \ldots \cdot (\tfrac{Z_{i-1}}{Z_{i-2}})^1 \\ &= Z_1 \cdot Z_2 \cdot Z_3 \cdot \ldots \cdot Z_n \\ &= T_{r_1 \cdot r_2 \cdot r_S}(x) \mathrm{mod} p \cdot T_{r_2 \cdot r_3 \cdot r_S}(x) \mathrm{mod} p \cdot T_{r_3 \cdot r_4 \cdot r_S}(x) \mathrm{mod} p \cdot \ldots \cdot T_{r_n \cdot r_1 \cdot r_S}(x) \mathrm{mod} p. \end{aligned}$$

Thus, these sensor nodes can obtain a common session key $SK_m = h(GID_{im} \| sk_{i,m})$ for the group G_{im}.

4.2. Session Key Security

Given T_{r_i} (x_o) *mod p* (= $T_{r_i r_s}$ (x) *mod p*) and $T_{r_{i+1}}$ (x_o) *mod p* (= $T_{r_{i+1} r_s}$ (x) *mod p*), where x_0 denotes $T_{r_s}(x)$ *mod p*, $Y_i = T = T_{r \bullet r_{i+1}}(x_o)$ *mod p* (= $T_{r_i r_{i+1} r_s}(x)$ *mod p*) cannot be determined, because of the ECM-DH problem. The values of $r_1, r_2, \ldots, r_n$ and r_s are randomly selected and mutually independent in each protocol execution, so the secret $sk_{i,m}$ and the session key SK_m fail to be determined without knowledge of r_s and r_i for $1 \leq i \leq n$, where $sk_{i,m} = T_{r_1 \cdot r_2 \cdot r_S}(x) \cdot T_{r_2 \cdot r_3 \cdot r_S}(x) \cdot \ldots \cdot T_{r_n \cdot r_1 \cdot r_S}(x) \mathrm{mod} p$ and $SK_m = h(GID_{im} \| sk_{i,m})$ for the group G_{im}. Hence, the session key security is based on the ECM-DH problem and is therefore considered not computationally feasible.

4.3. Perfect Forward Security

In the proposed protocol, since $r_1, r_2, \ldots, r_n$ and r_s are randomly selected and independent among protocol executions, a compromised password pw_i does not yield any previous session keys $SK_m = h(GID_{im} \| sk_{i,m})$ for G_{im}, where $sk_{i,m} = T_{r_1 \cdot r_2 \cdot r_S}(x) \mathrm{mod} p \cdot T_{r_2 \cdot r_3 \cdot r_S}(x) \mathrm{mod} p \cdot \ldots \cdot T_{r_n \cdot r_1 \cdot r_S}(x) \mathrm{mod} p$. The session key security is based on the ECM-DH problem. Accordingly, the proposed protocol provides perfect forward security.

4.4. Mutual Authentication

In the proposed group key agreement scheme, only legal sensor node SN_i who has the correct ID_i and pw_i can compute $C_i = h(DID_i \| Q_i \| K_1 \| X_i \| C_i, \| T_i)$, where $Q_i = h(ID_i \| pw_i)$. *AS* then authenticates sensor node by checking $C_i = h(DID_i \| Q_i \| K_1 \| X_i \| C_i, \| T_i)$. Also, sensor node authenticates *AS* by checking $CS_i = h(K_1 \| Q_i \| Y_{i\text{-}1} \| Y_{i+1} \| GID_{im} \| T_S)$. Additionally, only legal SN_i in $G_{i,m}$ can compute $sk_{i,m} = T_{r_1 \cdot r_2 \cdot r_S}(x) \cdot T_{r_2 \cdot r_3 \cdot r_S}(x) \cdot \ldots \cdot T_{r_n \cdot r_1 \cdot r_S}(x) \mathrm{mod} p$. Then, SN_i authenticates SN_j by checking $Auth_{i,m} = h(DID_i \| sk_{i,m} \| GID_{im} \| T_S)$ for $j \neq i$. Therefore, the participants of the proposed protocol authenticate each other.

4.5. Privacy Protection

In the proposed protocol, DID_i implicitly involves the identity of SN_i, ID_i, where $DID_i = K_1 \oplus ID_i$. Attackers cannot derive ID_i from DID_i because ID_i is protected by K_1 and the security of K_1 (= T_{rmk} *mod p*) is based on the ECM-DH problem. Additionally, the group identity GID_{im} = (DID_1, DID_2, … ,DID_i, … ,DID_n) is protected by $h(K_1 \| G_{im} \| T_s)$ (or K_1). No one can derive GID_{im} from the revealed message HGP_{im}, where $HGP_{im} = h(K_1 \| G_{im} \| T_s) \oplus GID_{im}$. Another group member SN_i cannot recognize the group members of G_{im} to which SN_i does not belong. Thus, the proposed protocol ensures users' privacy protection.

4.6. Resistance to Undetectable On-Line Password-Guessing Attacks

In the proposed protocol, an adversary SN_i^* cannot compute the correct $C_i = h(DID_i \| h(ID_i \| pw_i) \| K_1 \| X_i \| T_i)$ without SN_i's identity ID_i, where $K_1 = T_{r_i}(pk_S)\ mod\ p$, $DID_i = K_1 \oplus ID_i$, $X_i = T_{r_i}(x)\ mod\ p$ and T_i is the timestamp, and so such an adversary fails to send out $M_{i,1} = \{DID_i, X_i, C_i, T_i\}$ in Step 1. Additionally, SN_i^* who has ID_i and is disguised as SN_i guesses a password pw_i^*, computes $C_i^* = h(DID_i \| Q_i^* \| K_1 \| X_i \| T_i)$, where $Q_i^* = h(ID_i \| pw_i^*)$ and sends $M^*_{i,1} = \{DID_i, X_i, C_i^*, T_i\}$ to S in Step 1. After receiving $M^*_{i,1}$, AS will detect this failed password-guessing by checking $C_i = h(DID_i \| Q_i \| K_1' \| X_i \| T_i)$ in Step 2, where $K'_1 = T_{mk}(X_1)\ mod\ p$, $ID_i' = DID_i \oplus \mathrm{K}_1'$, $HID'_I = h(ID'_i \| mk)$, $Q_i = h(ID_i \| pw_i)$. Therefore, the proposed protocol is secure against undetectable on-line password-guessing attacks.

4.7. Resistance to Off-Line Password-Guessing Attacks

In the authentication and key agreement phase of the proposed protocol, only messages $C_i = h(DID_i \| Q_i \| K_1 \| X_i \| C_i \| T_i)$ in $M_{i,1}$ and $CS_i = h(K_1 \| Q_i \| Y_{i-1} \| Y_{i+1} \| GID_{im} \| T_S)$ in $M_{i,2}$ contain password pw_i, where $Q_i = h(ID_i \| pw_i)$. However, pw_i is protected by K_1, and the one-way property of hash functions. Similarly, in the password change phase of the proposed protocol, only messages $D_i = h(K_1 \| T_i) \oplus Q_{i\_new}$ and $E_i = h(DID_i \| Q_i \| Q_{i\_new} \| K_1 \| X_i \| T_i)$ in $M_{i,1}$ and $CS_i = h(K_1 \| Q_i \| Y_{i-1} \| Y_{i+1} \| GID_{im} \| T_S)$ in $M_{i,2}$ contain password pw_i, where $K_1 = T_{r_i}(pk_S)\ mod\ p$, $DID_i = K_1 \oplus ID_i$, $Q_i = h(ID_i \| pw_i)$ and $Q_{i\_new} = h(ID_i \| pw_{i\_new})$. However, pw_i and pw_{i_new} are protected by K_1 and the one-way property of hash functions. No information helps to confirm the correctness of the guessed passwords, so off-line password-guessing attacks are unsuccessful against the proposed protocol.

4.8. Known-Key Security

The session keys $SK_m = h(GID_{im} \| sk_{i,m})$, generated in various runs, are mutually independent, where $sk_{i,m} = T_{r_1 \cdot r_2 \cdot r_S}(x) \cdot T_{r_2 \cdot r_3 \cdot r_S}(x) \cdot \ldots \cdot T_{r_n \cdot r_1 \cdot r_S}(x) \mathrm{mod} p$, since $r_1, r_2, \ldots, r_n$ and r_S are randomly selected by $SN_1, SN_2, \ldots, SN_n$ and AS, respectively, and are independent across protocol executions. Thus, the proposed group key agreement protocol exhibits known-key security.

4.9. Resistance to Sensor Node Capture Attacks

In the proposed scheme, each sensor node SN_i has its secrets (ID_i, pw_i). An attacker A who has captured SN_j and obtained ID_j cannot derive other sensor node SN_i's secrets (ID_i, pw_i), and thus cannot impersonate SN_i and AS.

5. Performance Analyses and Comparisons

The performance of the proposed protocol in communication was compared with that of related approaches. Table 2 presents a performance comparison of the group authenticated key agreement (GAKA) protocols of Abdalla et al. [3], Kim et al. [7], Boyd and Nieto [8], and Dutta and Barua [13] and the protocol that was proposed herein, where T_{chao} denotes the time required to execute a Chebyshev chaotic map operation; T_{sym} denotes the time required to execute a symmetric encryption/decryption operation; T_{exp} denotes the time required to execute a modular exponential operation, $T_{sign/veri}$ denotes the time required to execute a signing/verifying operation in the public key system, and $T_{chao} < T_{sym} < T_{exp}$ ($\approx T_{sign/veri}$) [35,36].

Table 2. Comparisons of other related protocols and the proposed protocol.

Protocols	P1	P2	P3	P4	P5
Abdalla et al. [3]	$3nT_{exp} + 3nT_{sym}$	All users share a password	Yes	No	No
Dutta and Barua [13]	$3nT_{exp} + (n+3)T_{sym}$	All users share a password	Yes	No	No
Kim et al. [7]	$2nT_{sign/veri} + nT_{exp}$	PKI based	No	Yes	No
Boyd and Nieto [8]	$nT_{sign/veri} + (2n-2)T_{exp}$	PKI based	No	No	No
Lee et al. [14]	$4nT_{exp} + nT_{sym}$	A private password	Yes	No	No
Proposed GAKA	$3nT_{chao}$	A private password	Yes	Yes	Yes

P1: computations; P2: mutual authentication; P3: no user's public key; P4: for multiple groups; P5: providing users privacy protection.

The first comparison concerned computations. These GAKA protocols [3,7,8,13,14] require many time-consuming modular exponential computations or scalar multiplications on elliptic curves to realize authentication and negotiate group keys. Only the proposed GAKA protocol was developed using extended chaotic map operations and did not have a heavy computational burden. Thus, the proposed GAKA protocol was more efficient than the other GAKA protocols.

The second comparison concerned the realization of user authentication in each protocol. The protocols of Kim et al. [7] and Boyd and Nieto [8] realize authentication using users' public keys. The GAKA protocols of Abdalla et al. [3], Dutta and Barua [13], and Lee et al. [14] as well as the proposed GAKA protocol realize authentication using users' passwords. However, in the GAKA protocols of Abdalla et al. [3] and Dutta and Barua [13], all users share the same password so their protocols do not ensure users' privacy.

The third comparison concerned whether the protocol required the maintenance of users' public keys. The protocols of Kim et al. [7] and Boyd and Nieto [8] employ users' public keys, and thus require extra equipment to store long-term secret keys and the results of time-consuming exponential computations in clients. The GAKA protocols of Abdalla et al. [3], Dutta and Barua [13], and Lee et al. [14] as well as the proposed GAKA protocol are password-based authentication protocols. Each user remembers only his weak password without the need for any extra equipment to store long-term secret keys.

The fourth comparison involved whether the protocol was suitable for hypergraphs. The GAKA protocols [3,7,8,13] consider only a single group, and are difficult to extend to multiple groups. The protocol of Kim et al. [7] and the proposed GAKA protocol enable communicating entities to belong to multiple groups, and so are effective for hypergraphs.

The final comparison involved whether the protocol provided the anonymity of users. The GAKA protocols [3,7,8,13,14] reveal users' identities, and fail to protect user privacy. Only the proposed GAKA protocol did not reveal users' identities, and so protected users' anonymity.

Table 3 lists the simulation environment, including used hardware/software specifications and algorithms. Figure 3 illustrates simulation results for the response time of related protocols and the proposed one for $n = 5, 10, 15, \ldots, 30$. Due to the use of extended Chebyshev chaotic map operations, the proposed protocol required less response time than related protocols.

Table 3. Simulation environment.

Hardware/Software Specification
Intel CPU i7 CPU 3.2GHz 8G Memory Windows 10 Scala programming language
Used Algorithms
Asymmetric en/decryption algorithm: RSA Symmetric en/decryption algorithm: AES Extended Chebyshev chaotic maps

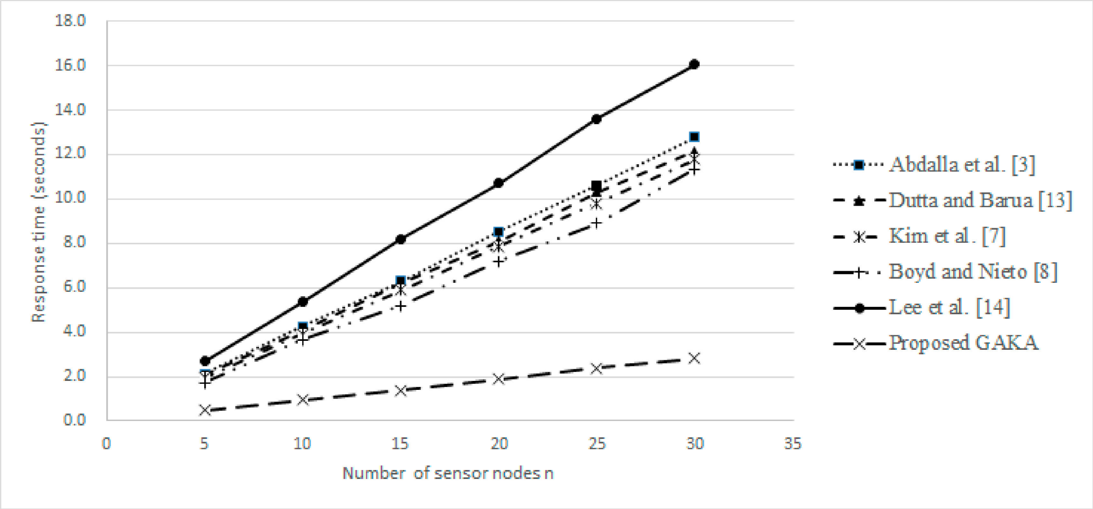

Figure 3. The response time of related protocols and the proposed one.

6. Conclusions

This work presented an efficient and secure group authenticated key agreement protocol for WSNs, which enabled sensor nodes to belong to multiple independent groups. The proposed protocol used extended chaotic map operations, did not require time-consuming computations, and thus was more computationally efficient than other group-authenticated key agreement protocols. Moreover, it did not require the maintenance of users' public keys or extra equipment for storing a long-term secret key, and resisted potential attacks and provided more functionality than comparable approaches. The proposed protocol is not only suitable for WSNs, but also can be implemented in the current environment involving database systems, file sharing systems, broadcasting radio/TV systems, and others.

Author Contributions: Conceptualization, M.-S.C., I.-P.C. and T.-K.L.; Methodology, M.-S.C. and I.-P.C.; Writing—Original draft preparation, M.-S.C. and I.-P.C.; Writing—Review and editing, M.-S.C., I.-P.C. and T.-K.L.; Funding acquisition, I.-P.C.

Funding: This study was supported in part by grants from the Ministry of Science and Technology of the Republic of China (Grant No. 104-2221-E-426-001).

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

References

1. Blaze, M. Trust Management and Network Layer Security Protocols. In *International Workshop on Security Protocols*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 109–118.
2. Hsieh, W.B.; Leu, J.S. A dynamic identity user authentication scheme in wireless sensor networks. In Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC), Sardinia, Italy, 1–5 July 2013; pp. 1132–1137.

3. Abdalla, M.; Bresson, E.L.; Chevassut, O.; Pointcheval, D. Password-based group key exchange in a constant number of rounds. In *Public Key Cryptography—PKC 2006*; Springer: Berlin/Heidelberg, Germany, 2006.
4. Tang, Q.; Choo, K.K.R. Secure password-based authenticated group key agreement for data-sharing peer-to-peer networks. In *International Conference on Applied Cryptography and Network Security ACNS 2006*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 162–177.
5. Pecori, R. A comparison analysis of trust-adaptive approaches to deliver signed public keys in P2P systems. In Proceedings of the 7th International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 27–29 July 2015.
6. Pecori, R.; Veltri, L. 3AKEP: Triple-authenticated key exchange protocol for peer-to-peer VoIP applications. *Comput. Commun.* **2016**, *85*, 28–40. [CrossRef]
7. Kim, H.-J.; Lee, S.-M.; Lee, D.-H. Constant-round authenticated group key exchange for dynamic groups. In *Advances in Cryptology—ASIACRYPT 2004 LNCS 3329*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 245–259.
8. Boyd, C.; Nieto, J.M.G. Round-optimal contributory conference key agreement. In *Public Key Cryptography—PKC 2003 LNCS 2567*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 161–174.
9. Jeong, I.; Lee, D. Key agreement for key hypergraph. *Comput. Secur.* **2007**, *26*, 452–458. [CrossRef]
10. Voloshin, V.I. *Introduction to Graph and Hypergraph Theory*; Nova Science Publishers: New York, NY, USA, 2009.
11. Bretto, A. *Hypergraph Theory*; Springer: Berlin, Germany, 2013.
12. Gandino, F.; Celozzi, C.; Rebaudengo, M. A Key Management Scheme for Mobile Wireless Sensor Networks. *Appl. Sci.* **2017**, *7*, 490. [CrossRef]
13. Dutta, R.; Barua, R. Password-based encrypted group key agreement. *Int. J. Netw. Secur.* **2006**, *3*, 30–41.
14. Lee, T.F.; Chang, I.P.; Wang, C.C. Simple group password-based authenticated key agreements for the integrated EPR information system. *J. Med. Syst.* **2013**, *37*, 9916. [CrossRef]
15. Sood, S.K.; Sarje, A.K.; Singh, K. A secure dynamic identity based authentication protocol for multi-server architecture. *J. Netw. Comput. Appl.* **2011**, *34*, 609–618. [CrossRef]
16. Amin, R.; Kumar, N.; Biswas, G.P.; Iqbal, R.; Chang, V. A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment. *Future Gener. Comput. Syst.* **2018**, *78*, 1005–1019. [CrossRef]
17. Xue, K.; Hong, P.; Ma, C. A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. *J. Comput. Syst. Sci.* **2014**, *80*, 195–206. [CrossRef]
18. Lin, T.H.; Tsung, C.K.; Lee, T.F.; Wang, Z.B. A round-efficient authenticated key agreement scheme based on extended chaotic maps for group cloud meeting. *Sensors* **2017**, *17*, 2793. [CrossRef]
19. Lee, T.F.; Wen, H.A.; Hwang, T. A weil pairing-based round-efficient and fault-tolerant group key agreement protocol for sensor networks. In *Sensor Network Operations*; IEEE Press: Piscataway, NJ, USA, 2006; pp. 571–579.
20. Xiao, D.; Liao, X.; Deng, S. Using time-stamp to improve the security of a chaotic maps-based key agreement protocol. *Inf. Sci.* **2008**, *178*, 1598–11602. [CrossRef]
21. Han, S.; Chang, E. Chaotic map based key agreement with/out clock synchronization. *Chaos Solitons Fractals* **2009**, *39*, 1283–1289. [CrossRef]
22. Xiao, D.; Liao, X.; Deng, S. A novel key agreement protocol based on chaotic maps. *Inf. Sci.* **2007**, *177*, 136–1142. [CrossRef]
23. Guo, X.; Zhang, J. Secure group key agreement protocol based on chaotic hash. *Inf. Sci.* **2010**, *180*, 4069–4074. [CrossRef]
24. Gong, P.; Li, P.; Shi, W. A secure chaotic maps-based key agreement protocol without using smart cards. *Nonlinear Dyn.* **2012**, *70*, 2401–2406. [CrossRef]
25. Niu, Y.; Wang, X. An anonymous key agreement protocol based on chaotic maps. *Commun. Nonlinear. Sci. Numer. Simulat.* **2011**, *16*, 1986–1992. [CrossRef]
26. Farash, M.S.; Attari, M.A. Cryptanalysis and improvement of a chaotic map-based key agreement protocol using chebyshev sequence membership testing. *Nonlinear Dyn.* **2014**, *76*, 1203–1213. [CrossRef]
27. 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*, 58. [CrossRef]

28. Lee, T.-F. Efficient three-party authenticated key agreements based on Chebyshev chaotic map-based diffie-hellman assumption. *Nonlinear Dyn.* **2015**, *81*, 2071–2078. [CrossRef]
29. Lee, T.-F.; Lin, C.-Y.; Lin, C.-L.; Hwang, T. Provably secure extended chaotic map-based three-party key agreement protocols using password authentication. *Nonlinear Dyn.* **2015**, *82*, 29–38. [CrossRef]
30. Kocarev, L.; Tasev, Z. Public-key encryption based on Chebyshev maps. In Proceedings of the IEEE International Symposium on Circuits and Systems 3, Bangkok, Thailand, 25–28 May 2003.
31. Mason, J.C.; Handscomb, D.C. *Chebyshev Polynomials*; Chapman & Hall/CRC: Boca Raton, FL, USA, 2003.
32. Bergamo, P.; D'Arco, P.; Santis, A.; Kocarev, L. Security of public-key cryptosystems based on Chebyshev polynomials. *IEEE Trans. Circuits Syst. I* **2005**, *52*, 1382–1393. [CrossRef]
33. Zhang, L. Cryptanalysis of the public key encryption based on multiple chaotic systems. *Chaos Solitons Fractals* **2008**, *37*, 669–674. [CrossRef]
34. Wang, X.; Zhao, J. An Improved Key Agreement Protocol based on Chaos. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 4052–4057. [CrossRef]
35. Wu, S.; Chen, K. An efficient key-management scheme for hierarchical access control in e-medicine system. *J. Med. Syst.* **2012**, *36*, 2325–2337. [CrossRef] [PubMed]
36. Cheng, Z.Y.; Liu, Y.; Chang, C.C.; Chang, S.C. Authenticated RFID security mechanism based on chaotic maps. *Secur. Comm. Netw.* **2013**, *6*, 247–256. [CrossRef]