IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 19, 2018, accepted May 14, 2018, date of publication May 24, 2018, date of current version June 20, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2840131

Distributed Multiparty Key Management for Efficient Authentication in the Internet of Things

ZAHID MAHMOOD[1,2], (Student Member, IEEE), ATA ULLAH[1,3], (Member, IEEE), AND HUANSHENG NING[1,2], (Senior Member, IEEE)

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2] Beijing Engineering Research Center for Cyberspace Data Analysis and Applications, Beijing 100083, China
[3] Department of Computer Science, National University of Modern Languages, Islamabad 44000, Pakistan

Corresponding author: Huansheng Ning (ninghuansheng@ustb.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61471035 and Grant 61774014.

ABSTRACT With the inclusion of mobile devices and ubiquitous connectivity of smart devices in Internet of Things, secure key management is mandatory to ensure privacy for information exchange. In this regard, the multiparty key establishment schemes achieve better security strength by taking shared parameters from neighboring member nodes to calculate the key. The similar multiparty mechanism can be adopted among other hierarchical nodes, including head node, server and gateway node. Moreover, session keys can also be set up in a similar manner. The main problem in multiparty password-based authentication schemes is the computation of extensively hard problem that limits it to three parties and N-party is quite more complex or infeasible. This paper presents a novel distributed multiparty keying scheme where chaotic maps are used to provide one-way hashing and Chebyshev polynomial are used for establishing a common multiparty key. In this paper, Phase-I covers keying among trusted server and group heads and Phase-II elaborates the key establishment among smart devices and their group heads. The scheme is verified through the formal specification and security analysis using Rubin Logic for inter-group key establishment scenario. We have validated the intra-group and inter-group key establishment by doing extensive simulations in NS 2.35. Moreover, a test bed is setup for group head to server level authentication and key establishment. Results prove the supremacy of our scheme as compared with preliminaries in terms of computation cost, communication cost, and resilience.

INDEX TERMS Chaotic maps, multiparty-key, chebyshev polynomials, smartness, key establishment.

I. INTRODUCTION

Smart and wearable devices are playing a vital role in wireless communication especially in emerging Internet of Things (IoT) scenario. The smart devices are vulnerable to various security attacks due to availability of confined resources like memory, energy, computation and communication. Moreover, the chances of vigorous attacks increases when smart devices are continuously connected to internet [1]. It arises the need for dependable cryptographic solutions to ensure secure information exchange. In this regard, group head (GH) based approaches are considered more applicable to achieve authentication, privacy, data integrity, backward and forward secrecy. Group based information exchange has been progressively used for multimedia based social communication, group based games, streaming live videos, IP-TV [2], edge and cloud based distributed application scenarios [3]. For controlled access to such valuable services and application, a secure access control mechanism is mandatory between user and service providers. It demands secret keys among group members to exchange secret messages using encryption and decryption functions. A session key is generated whenever messages are shared for a specific time interval or session.

Multiparty key is considered strong enough to assure privacy because it is computed by taking parameters from member nodes instead of only two parties including sender and receiver [4]. In this case, Authentication Key Exchange (AKE) can be divided into 2-parties, 3-parties and N-Parties AKE schemes. Multiparty scenario requires more challenges due to reliability on neighboring devices as compared to existing security schemes for IoT [5], [6]. For security schemes, chaos theory is used in dynamic systems that are sensitive to little change in initial conditions. It is used for pseudo-randomness in cryptographic operations during past few years. The behavior is also apparent in weather, climate, traffic, and networking where a little change in system results in abrupt changes [7], [8]. Security schemes are used in authentication [9], confused maps in symmetric, asymmetric encryption [10], digestive hash capacities [11] and independent sub grouping proofs for hierarchical protection [12]. Similarly, chaotic maps are used to generate one-way hash values that are nearly impossible to predict by analyzing initial values. For key establishment, chebyshev polynomials (CPs) are used that comprises of a collection of polynomials in a sequence that is indexed in order of their degrees. These are applicable in approximation theory that can also be utilized in security solutions to provide the best approximation to a continuous function. CPs are considered to be the better in terms of keeping a balance between computing efficiency and providing affordable security [13]. It improves mutual authentication and attains session key establishment by using time-synchronization [14]. It provide validation and reduce computation overheads but none of these can individually secure client's data and interaction scenarios.

The main problem during password-based Multi-party authenticated key agreement systems is the extensive modular exponential calculation for RSA based schemes or scalar exponential for ECC curve based schemes [15]. The password-AKE (PAKE) protocols use modular exponential and scalar exponentiation on ECC curve for only three-party scenario [16]. It needs extensive computations and becomes more complex for more than three or multi party scenarios. Due to the fact, most of the schemes have considered 3-PAKE instead of N-party authentication. In [17], the ECC based solution is found efficient for computing and security but hash function has non-negligible cost. Moreover, these schemes have used Diffie–Hellman (DH) based key establishment schemes [9] that can be vulnerable to man-in-the-middle attack during secure authentication.

This paper presents a secure and efficient **D**istributed **M**ultiparty **K**ey (DMK) establishment scheme that uses Chaotic maps and Chebyshev polynomials for cryptographic operations. In Phase-I, initially all GHs calculate and share CP with a trusted server that authenticates each GH by calculating a new CP using semi-group property of CP. After that, a session key is established by taking secret parameters from all GH where TS calculates a common CP and shares with all GH to calculate a session key by using existing secret key. Moreover, chaotic map based one-way hash functions are also used for integrity protection. In Phase-II, the member nodes are registered with their appropriate GHs. The trusted server takes parameters from multi-parties to generate a common key and exchange with sender and receiver. Moreover, an intergroup session key establishment scheme is presented for communication across member of neighboring groups. The proposed scheme is verified using Rubin-Logic for the inter-group key establishment scenario. We have performed simulations using NS-2.35 for the phase-II whereas the protocols for GH to server level in phase-I are validated on a testbed using ModPow function and BinInteger class in Java. We have evaluated DMK against security attacks. Results elucidate the supremacy of our DMK scheme over counterparts for comparing computation cost, communication cost and resilience.

Remaining sections of the paper are as follows: Section 2 revisits interrelated multiparty key establishment schemes. Section 3 provides preliminaries for chebyshev polynomials. Section 4 elaborates the proposed Distributed Multiparty Keying scheme and formal modeling using Rubin Logic is explored in section 5. In section 6, Security analysis is a performed. Simulation and results are provided in Section 7 and Section 8 concludes our work.

II. PRELIMINARIES – CHEBYSHEV POLYNOMIAL

In this section, we quickly present Chebyshev disorderly maps. The trusted server $\mathrm{T_S}$ picks its main key $\mathrm{m_k}$ and creates random number $x \in (-\infty, +\infty)$. At that point, $\mathrm{T_S}$ chooses a restricted hash capacity $\mathrm{h}(\cdot)$ and $\mathrm{H}(\cdot)$, and secure symmetric encryption and decoding calculations $\mathrm{E}_{\mathrm{k}}(\cdot)$ and $\mathrm{D}_{\mathrm{k}}(\cdot)$, separately. In Chaotic Map-based Discrete Logarithm Problem (CMDLP), for a large prime p where $n \in \mathbb{N}, \alpha \in \mathrm{Z_p}$ and $\mathrm{T_n(x)} = ((2\mathrm{xT_{n-1}(x)}) - \mathrm{T_{n-2}(x)})\mathrm{mod\,N}$ is a CP sequence generator. This hypothesis can be precisely defined by two experimentations named $\mathrm{A}_{\alpha,p}^{\mathrm{CMDLP-REAL}}(\mathrm{A})$ and $\mathrm{A}_{\alpha,\mathrm{p}}^{\mathrm{CMDLP-RAND}}(\mathrm{A})$. An attacker A try to access messages transmitted between two ends like $\{m_1, m_2\}$ occupy random oracle for implementing. It attempts to acquire the key when the hash function is near to random oracle model. If u, v and w are drawn randomly from model (1, p+1), then the probability of success rate is on providing $\mathrm{Tu}(\alpha), \mathrm{Tv}(\alpha)$ and $\mathrm{Tuv}(\alpha)$ for $\mathrm{A}_{\alpha,p}^{\mathrm{CMDLP-REAL}}(\mathrm{A})$, and for $\mathrm{A}_{\alpha,\mathrm{p}}^{\mathrm{CMDLP-RAND}}(\mathrm{A})\mathrm{Tu}(\alpha), \mathrm{Tv}(\alpha)\&\mathrm{T}w(\alpha)$. According to these experiments, an intruder W who is able to invert the hash function and can also solve the CMDLP which is hard problem. It exposes the user $\mathrm{ID_i}$ and the key SK between user Ui and group head $(\mathrm{G_i})$. However, according to the definition pf CMDHP hard problem, inverting hash function is computationally infeasible, $\mathrm{A}_{\alpha,p}^{\mathrm{CMDLP-REAL}}(\mathrm{A})$, for $\mathrm{A}_{\alpha,p}^{\mathrm{CMDLP-RAND}}$ (t1) $\leq \varepsilon$ the small value $\varepsilon > 0$.

Suppose n is a digit and x is a variable with the interval between -1 to 1. In [18], $\mathrm{T}_n\mathrm{x} = \cos(ncos^{-1}(\mathrm{x}))$ which represents a CP. For plotting $\mathrm{T_n}: \mathrm{R} \rightarrow \mathrm{R}$ of n degree, the CPs hold the semi-group property as $\mathrm{T_r(T_s(x))} = \mathrm{T_s(T_r(x))}$ as given in (1) [19]. The improved CPs are utilized as a part of proposed convention as $\mathrm{T_n(x)} = (2\mathrm{xT_{n-1}(x)}) - \mathrm{T_{n-2}(x)(modN)}$ when $n \geq 2, x \in (-\infty, +\infty)$. In this case, N is a prime number and polynomial is computed as $\mathrm{T_{rs}(x)} = \mathrm{T_r(T_s(x))} = \mathrm{T_s(T_r(x))}$. For the composition scenario, $\mathrm{T_s(T_r(x))} = \mathrm{T_{sr}(x)}$ holds for CPs. In second case with x and y, it is hard enough to find the number s when $\mathrm{T_s(x)} = \mathrm{y}$. It is known as Chaotic Discrete Logarithm

Problem based on chaotic maps or CMBDLP.

$$T_r(T_s(x)) = \cos\left(\left(r\cos^{-1}(s\cos^{-1}(x))\right) = \cos\left(rs\cos^{-1}\right)(x)\right)$$
$$= T_s(T_r(x)) = T_{sr}(x) \qquad (1)$$

III. RELATED WORK

We have explored different existing schemes that utilize authentication a nd key management in smart devices based networks. With the advent and rapid development of IoT, smart devices are experiencing sensing bottlenecks. It arises the need for managing the sensing and secure communication capabilities to provide dependable solutions. A large number of smart devices from physical world map to cyber world to provide meaningful services for the betterment of human life. It includes wearable devices, environmental sensing, smart sensing, smart vehicles and smart industry. A heterogeneous network of smart devices play an instrumental role in enormous number of innovative applications in human-centered IoT scenario where security becomes more critical. In such networks, the end node smart devices are resource constrained in terms of processing, storage, sensing and communication. In contrary, the cluster head nodes are more resource rich for monitoring and managing the cluster of smart devices. In upper layer, the trusted server handle this heterogeneous network with smart devices and cluster heads interlinked via a strong, efficient and trusted hub. The server manages overall network in the system like smart industry network, environment network, smart living network and smart services network. By exploring and analyzing related existing schemes, we present the distributed multiparty based group key management schemes in the following sections.

1) PASSWORD AUTHENTICATION BASED SCHEMES

Password based schemes involve the secrets or passwords sharing from neighboring member nodes to generate the secret keys. Farash *et al.* [13] has presented a two-party password key management scheme. It authenticates participants and set up a session key over an unreliable channel where every client communicates to recall password from the trusted server. The primary methodology is that client either uses an open key or symmetric one. It can also be used in combination of both [20]. Otherwise, it can select the public key of server or any symmetric key or none of these [21]. The symmetric cryptosystem requires developing a key administration based system and compelling more complex and cumbersome computing cost to develop an agreement. An anonymity based authentication scheme is presented to guard against user tracking attacks [22]. Kwon *et al.* [23] have presented a practical 3-PAKE (P3-PAKE) scheme by taking decisional DH supposition as security parameters for the varied environment, the productivity of validated key trade must address one of the central considerations. By using the mobile environment, conventional 3-PAKE protocols face two common problems; (i) Server and users may have a different domain, and there are chances of shared key compromised, (ii) Conventional 3-PAKE needs higher computational and communication cost during session key establishment that causes overheads for low power devices.

Lu [24] has proposed multi-party password-authentication key exchange (M-PAKE) scheme for the mobile environment based on ECC. The most significant benefit of this scheme is that each user saves a single password that is maintained at trusted server. Moreover, the node can have a number of secret passwords stored at neighboring trusted parties for future authentication. In this scheme, n users act as participants and $n+1$ communicating parties including a trusted server S. By generating users' password secretly at server, session key can be generated by each user. In this scheme, the server holds password of all participants and then distributed to the users by using the secure channel as illustrated in figure 1. It can guard against the brute force and the word reference assaults when clients pick strong password values to ensure enough entropy. In practice, most participants choose simple passwords that are hard to secure against various attacks.

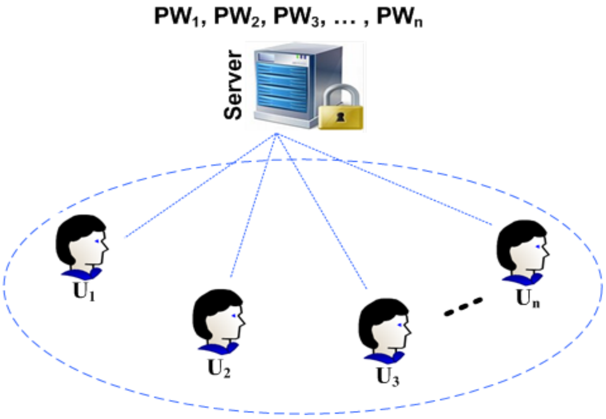


FIGURE 1. MPAKE scheme.

By applying this scheme, every user needs to memorize its password which causes offline and dictionary attack. On the other hand single-node-failure, single-node efficiency and single-node dependency are the main factors for system effectiveness. In case of n users, the communication and storage costs increases during setup, registration and session key generation phases. Due to partitioned exponential or scalar repetition on an elliptic curve, this scheme requires extensive computational cost. A single centralized server cannot handle such a complex network. In case of extension phase, new node authentication mechanism is not very efficient. With the advancement of the resource constraint smart devices network, existing solutions are not suitable for low-control gadgets [24], [25].

A. ASYMMETRIC SCHEMES FOR AUTHENTICATION

During the asymmetric approaches, a public-private keys pair is used for the encryption and decryption which is considered to be extensive for low power devices. In contrary, ECC is one of the mechanism for providing asymmetric solutions for low power smart device. Ruhul Amin and Biswas [26] has presented an analysis of various existing key management schemes that include the user authentication and key establishment. It explores that asymmetric key-based cryptosystem including elliptic curve, El-gamma and RSA require complex computations and need more storage space, so these are not yet appropriate for the resource constraint devices and networks as compared to symmetric schemes. In [27], the main shortcoming is using certificate's principal which causes extra computational overhead for resource-constraint smart devices. Moreover, the bilinear map based pairing function uniquely encode two original members into a single one but consumes high costs. In [28], a light-weight authentication scheme is presented for smart phone and wearable devices in IoT. The phone generates a timestamp embedded flag that is based on a pseudo random number. The wearable device extracts the security credentials from message to update its timestamp embedded pseudo-random flag. It utilizes the yoking proofs for simultaneous identifications performed by the cloud server.

Hasimoto-Beltran *et al.* [29] has implemented AES and chaotic map based encryption on a different platforms. It applies the constant rate for various data streaming traffic sources to analyze performance for energy consumption, computation and communication costs. By their results, chaos encryption is 200% efficient in term of CPU usage and 260 % time faster than AES. It consumes 32 Watt-hours of battery energy for 800 GigaBytes of data processing and AES consumes same power for 280 GigaBytes data only.

B. CHAOTIC MAPS AND CHEBYSHEV POLYNOMIAL BASED SCHEMES

Chaotic maps provide one-way hash to ensure the integrity of security credentials exchanged over the network. Chaotic maps holds the superb diffusion and effusion properties that are essential requirement of a cryptosystem. Chaotic operations have dependence on initial conditions or characteristic parameters for acquiring varying outputs. It also ensures topological transitivity and periodicity to provide strong cryptographic operations. Chaos develop secure ciphers to offer robustness in opposition to many common attacks that can occur in conventional ciphers. Therefore, chaos idea has been successfully implemented to cryptography for a few years. Chebyshev polynomials are used to generate secret keys between the communicating parties. It achieves the defensive privacy of data throughout communication over public channels in variety of networks and. More reliable, resilient and dependable cryptographic schemed are presented by researchers to cope with emerging era of facts generation and rapidly growing smart devices based communications. For the productive execution of unrestrained arrangements for random numbers and time Synchronization against fake timestamps [30] for the framework parameters. The unverifiable deterministic elements of disorderly circles are the prominent properties that are investigated for safe transmission of information [27]. It also misuses the properties of chaotic flow. Moreover, the confusion based cryptography is classified into simple and computerized. The former depends on the strategies of control and synchronization of confusion. The later involves the auto-generated credentials as well. In this scenario, the block ciphers can utilize chaos based cryptosystems along with cycles of chaotic maps [31] and chaotic round functions [32]. Additionally, stream ciphers use the chaos based pseudo-arbitrary bit generators [33].

Chaos-based cryptosystems are used in chaotic covering [34], chaotic instruction [35] and confused exchanging [9]. In advanced distributed cryptosystems, dynamic frameworks are used where the secret keys should be dynamic and secure. It has dependability over the chaotic map based one-way hash functions. In these scenarios, the lattice based cryptosystems are considered for secret key distribution and access control. The chebyshev polynomial based keys are never exposed over the network and only the key credentials are exchanged. It enhance the security strength as the key is never transmitted over the path where confidential information can be exchanged. Hao *et al.* [36] has proposed an efficient chaotic map based authentication (ECMA) scheme for the medical information system. These schemes improve security as well as efficiency for the session key generation and authentication process. Li *et. al.* [37] has improved Lee's scheme and proposed a secure chaotic map based password authentication (CMPA) protocol for telecare medical system with user anonymity. This scheme handled service misuse attack and mitigated the drawbacks in existing scheme. Contemporary cryptographic strategies are based on the basic mathematical models or algebraic concepts whereas chaotic concepts is another paradigm, which appears promising and hard. It is based on nonlinear dynamics that are considered in crypto-graphical operations based on mathematical mechanisms including diffusion, confusion, and dependence. Security schemes use the message-embedded mechanism to generate strong chaotic ciphers [38]. During cryptanalysis of chaotic operation, the strength and weaknesses of the existing and chaotic cryptographic operations are compared. It explores that chaotic map operations and Chebyshev polynomials are suitable for smart devices based networks. This evaluation encourages to develop secure key management systems using chaotic maps and CPs which provides better data security in an efficient manner for smart devices. Cheng *et al.* has proposed a Chaos-based Group Key Agreement (CGKA) scheme [39] for distributed network environment to provide security and efficiency in Guo *et al.* [40] scheme titled Group Keying using Chaotic Hash (GKCH). CGKA is suitable for collaborative group based communication. In [41] several techniques are presented for the computation and the CP problems are solved to achieve efficiency. CP based computations require small key size, less extensive computations and reduction in energy, memory and storage as compared to ECC and RSA.

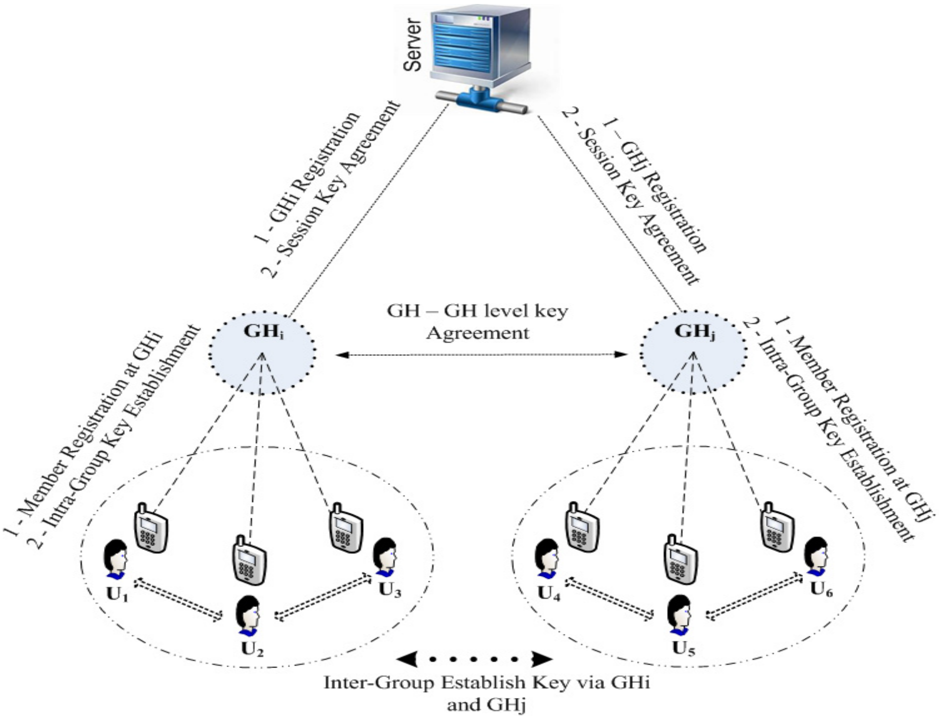

Server
1 - GHi Registration
2 - Session Key Agreement
1 – GHj Registration
2 - Session Key Agreement
GHi
GHj
GH – GH level key Agreement
1 - Member Registration at GHi
2 - Intra-Group Key Establishment
1 - Member Registration at GHj
2 - Intra-Group Key Establishment
U1
U2
U3
U4
U5
U6
Inter-Group Establish Key via GHi and GHj

FIGURE 2. Distributed multi-party key computing architecture.

IV. DISTRIBUTED MULTIPARTY KEY (DMK) ESTABLISHMENT SCHEME

In this section, we elaborate our proposed DMK scheme with privacy preservation for the smart devices in IoT scenario. Our scheme is applicable in group based scenarios where multiple member nodes are registered with group heads like healthcare, transportation, agriculture and industrial monitoring. By considering the healthcare scenario, a head node can register multiple health parameter collector device or cell phones of patients to setup a group. In this case, GH can take security credentials from multi parties to setup a key between any sender and receiver. Due to the advancement of the chaotic map based cryptosystem, we have utilized the robust and lightweight chaotic cryptographic operations. Chaotic maps are based on chaos theory that explores the study of measuring the impact of initial conditions over the dynamical systems. In the proposed system, chaotic maps are used to provide one-way hash functions. We have used the chebyshev polynomials that are used during secret and session keys establishment to provide confidentiality for exchanging data between users. In multiparty key establishment, a key distributor who must be a trusted server coordinates to set up secure key for intra and intergroup scenario. Our scheme consists of two phases where GH are authenticated from the trusted server along with session key establishment in the first phase. After that, group members perform intragroup key establishment and intergroup session key establishment in phase II as illustrated in figure 2. A list of notations is providing in Table-1 for proposed DMK scheme.

TABLE 1. List of notations for DMK.

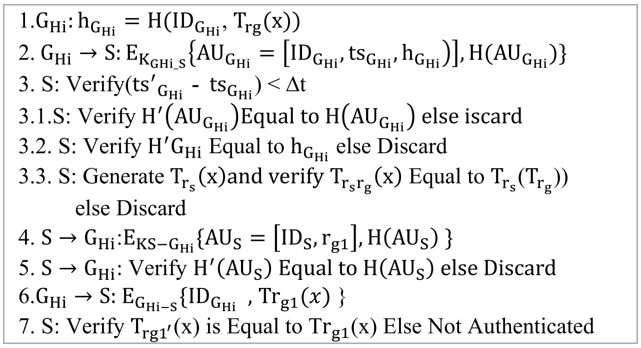
Notation	Description
$ID_{G_{Hi}}$	Identity for Group Head GHi
S	Trusted Server
G_{Hi}, G_{Hj}	Group Heads
MAC	Message Authentication Code for Data Integrity
r_g, r_s	Random number generated by GH and Server
A_{UG}, A_{US}	Authentication message by G_H and Server
ACK	Acknowledgment message
U_i	Member Node in the group
$C_1 - C_4$	Cipher Text between Ui and GH
T_{gs} (x)	CP range for random numbers of GH and server
H (.)	One way chaotic hash function
h_{PW}	Shared Password value
$\oplus$	Exclusive XOR
T_g, T_g	CP of degree g
K_{GS}	Common session key between G_H and Server
M_R	Registration Message by Group Member
$ts_{G_{Hi}}$	Timestamp assigned by GHi
$G_{Hi\ldots n}$	Set of multiple Gh from i=1 to n
ts_s, ts_g	Timestamp at S and $G_{Hi\ldots n}$
$H(T_r(ra))$	Hash value of CP and random number
$M_1 - M_3$	Messages between GH and server
CP	Chebyshev Polynomial
$K_{G_{Hi}-S}$	Key between GHi and server

A. PHASE-I: AUTHENTICATION AND SESSION KEYING BY GROUP HEADS

In this section, we have first explored the procedure for authenticating GHs at server and then establishing session key by utilizing distributed multiparty keying where multiple GH contribute by sharing security credentials to server as discussed in following section.

1) GROUP HEADS AUTHENTICATION PROTOCOL

The authentication process begins when a GH transmits a chaotic map based one-way hash to server that verifies the security credentials and then to generates CP. Server also gives a challenge to GH for authenticating the GH as presented in figure 3. Semi-group property of CP is utilized to generate a common polynomial by nesting the polynomials. We have considered a group head G_{Hi} and stepwise description for the protocol is explored as follows.



```
1.$G_{Hi}$: $h_{G_{Hi}} = H(ID_{G_{Hi}}, T_{rg}(x))$
2. $G_{Hi} \rightarrow S: E_{K_{GHi\_S}}\{AU_{G_{Hi}} = [ID_{G_{Hi}}, ts_{G_{Hi}}, h_{G_{Hi}})], H(AU_{G_{Hi}})\}$
3. S: Verify($ts'_{G_{Hi}}$ - $ts_{G_{Hi}}$) < $\Delta t$
3.1.S: Verify $H'(AU_{G_{Hi}})$Equal to $H(AU_{G_{Hi}})$ else iscard
3.2. S: Verify $H'G_{Hi}$ Equal to $h_{G_{Hi}}$ else Discard
3.3. S: Generate $T_{r_s}(x)$and verify $T_{r_s r_g}(x)$ Equal to $T_{r_s}(T_{r_g}))$
        else Discard
4. $S \rightarrow G_{Hi}$:$E_{KS-G_{Hi}}\{AU_S = [ID_S, r_{g1}], H(AU_S)\}$
5. $S \rightarrow G_{Hi}$: Verify $H'(AU_S)$ Equal to $H(AU_S)$ else Discard
6.$G_{Hi} \rightarrow S$: $E_{G_{Hi-S}}\{ID_{G_{Hi}}$ , $Tr_{g1}(x)$ }
7. S: Verify $T_{rg1'}(x)$ is Equal to $Tr_{g1}(x)$ Else Not Authenticated
```

FIGURE 3. GH authentication protocol.

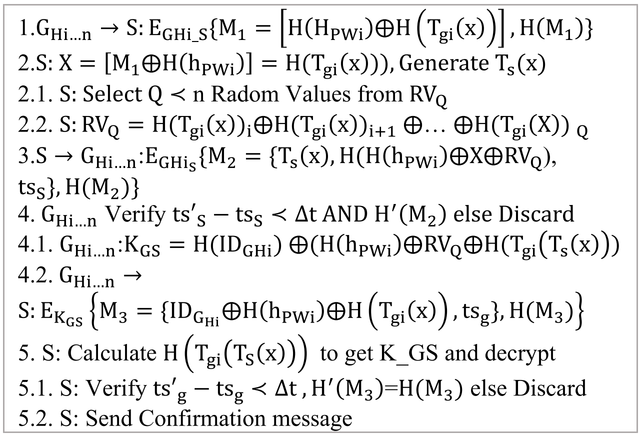
Steps (1) – (3):- Initially, G_{Hi} chooses an arbitrary large random integer $\in r_g[-1, 1]$ and then computes $T_{r_g}(x)$. After that, G_{Hi}computes a hash value $h_{G_{Hi}} = H(ID_{G_{Hi}}, T_{r_g}(x))$ where $ID_{G_{Hi}}$ is the identity of GH and H(.) represents chaotic based one-way hash function. Secondly, G_{Hi} prepares an authentication message AU_{GHi} that contains $ts_{G_{Hi}}$, $h_{G_{Hi}}$ and $H(AU_{G_{Hi}})$ represented as time stamp, hash of CP along with ID and hash of message respectively. It transmits the encrypted message to server S by using the symmetric key $K_{G_{Hi}-S}$ and transmits to trusted server S.

Upon receiving the message, server S verifies the message freshness by comparing the difference of sending and receiving time stamps as per threshold value Δt. Message integrity is also ensured by matching the similarity of received hash with the calculated $H'(AU_{G_{Hi}})$ using chaotic map based one-way hash function. Server also verifies the $h'_{G_{HA}}$ to verify the existence of $T_{rg}(x)$ which is CP at G_{Hi}. After verification process, S generates $T_{rs}(x)$ and calculates $T_{r_s r_g}(x)$ by using the semi-group property of CP as $T_{r_s r_g}(x) = T_{rs}(T_{rg}(x))$.

Step (4) – (7): Server S transmits an authentication reply $AU_S = (ID_S, r_{g1}), H(AU_S))$ to G_{Hi}. The message also contains a challenge for G_{Hi} to prepare CP using provided random number r_{g1} and send back to server. Upon receiving reply message, G_{Hi} verifies message integrity and then replies to S with $ID_{G_{Hi}}$ and newly calculated CP as $Tr_{g1}(x)$ for final authentication. Upon receiving verified information, the S verifies the $Tr_{g1}(x)$ for successful authentication of G_{Hi}, otherwise, discard message.

2) SESSION KEY ESTABLISHMENT AT GH

In this section, the session key establishment protocol is presented for GH and server to secure the messaging for a particular session. It needs to be renewed for each new session. By applying CP and chaotic maps in a multiparty manner, it becomes more secure and reliable to set up these keys. In this scenario, all GHs contribute by sharing their security credentials parameters with server that generates a new secret to share with all GHs to prepare the session key. Session key establishment protocol is presented in figure 4 and its stepwise description is presented as follows.



```
1.$G_{Hi...n} \rightarrow S: E_{GHi\_S}\{M_1 = [H(H_{PWi}) \oplus H(T_{gi}(x))], H(M_1)\}$
2.S: $X = [M_1 \oplus H(h_{PWi})] = H(T_{gi}(x)))$, Generate $T_s(x)$
2.1. S: Select $Q \prec n$ Radom Values from $RV_Q$
2.2. S: $RV_Q = H(T_{gi}(x))_i \oplus H(T_{gi}(x))_{i+1} \oplus \ldots \oplus H(T_{gi}(X))_Q$
3.S $\rightarrow G_{Hi...n}$:$E_{GHi_S}\{M_2 = \{T_s(x), H(H(h_{PWi}) \oplus X \oplus RV_Q)$,
$ts_S\}, H(M_2)\}$
4. $G_{Hi...n}$ Verify $ts'_S - ts_S \prec \Delta t$ AND $H'(M_2)$ else Discard
4.1. $G_{Hi...n}$:$K_{GS} = H(ID_{GHi}) \oplus (H(h_{PWi}) \oplus RV_Q \oplus H(T_{gi}(T_s(x)))$
4.2. $G_{Hi...n} \rightarrow$
S: $E_{K_{GS}}\{M_3 = \{ID_{G_{Hi}} \oplus H(h_{PWi}) \oplus H(T_{gi}(x)), ts_g\}, H(M_3)\}$
5. S: Calculate $H(T_{gi}(T_S(x)))$ to get K_GS and decrypt
5.1. S: Verify $ts'_g - ts_g \prec \Delta t$ , $H'(M_3)$=$H(M_3)$ else Discard
5.2. S: Send Confirmation message
```

FIGURE 4. Session key generation protocol at GH.

Steps (1) – (3): Initially, all *n* group heads $G_{Hi...n}$ individually prepare a message M_1 and transmit to server separately. The message contains $[H(H_{PWi}) \oplus H(T_{gi}(x))]$ which is XOR of hashed value of secret password with the hash value of CP generator $H(T_{gi}(x))$. Upon receiving the messages from all group heads $G_{Hi...n}$, server computes X separately by taking XOR with hashed password $H(h_{PWi})$ of each GH to extract all $H(T_{gi}(x))$ values from the messages of all GH. From these extracted *n* hash values, server S randomly selects Q values and takes XOR of these hash values to represent as RV_Q. Server S sends messages to *n* GHs by encrypting the message with appropriate key with GH. Message $M_2 = \{T_s(x), H(H(h_{PWi}) \oplus X \oplus RV_Q), ts_S\}$ also contains new timestamp ts_S and generated polynomial $T_s(x)$.

Steps (4) – (5): Group heads $G_{Hi...n}$ verify the difference between timestamp and hash values to ensure message freshness and message integrity respectively. After that, $G_{Hi...n}$ compute CP $H(T_g(T_s(x)))$ and then obtains a new secret key $K_{G-S} = H(ID_{GHi}) \oplus (H(h_{PWi}) \oplus RV_Q \oplus H(T_{gi}(T_S(x)))$ by taking XOR of hashed parameters. Now each $G_{Hi...n}$ transmits message M_3 to server S by encrypting using new key K_{GS}. The trusted server S also calculates $H(T_{gi}(T_S(x)))$ to get K_{GS} and decrypt the message M_3 and then verify the difference of timestamps as $ts'_g - ts_g \prec \Delta t$ and equivalence of hash values as $H'(M_3) = H(M_3)$.

Finally, server transmits the confirmation message to complete the session key establishment process.

B. PHASE II - AUTHENTICATION BETWEEN GROUP HEADS AND MEMBER NODES

In this section, we discuss multiparty key computing scenario between G_{Hi} and the member nodes as illustrated in figure 5. Member nodes first register to particular G_{Hi} and then G_{Hi} issues chaotic based token consisting of a random number, ids of member nodes and time stamps. These values are sent securely using the one-way chaotic map based hash function. Using these distributed parameters; member nodes generate a common session key for intra-cluster secure communication. The proposed technique can establish secure group communication between participant nodes within the cluster as well inter-cluster. For inter-cluster secure data exchange, the trusted server verify double authentication before inter-cluster network establishment to overcome malicious node penetration. The detail description of the intra-group secure session key in presented as a blow in which all computed steps described in detail.

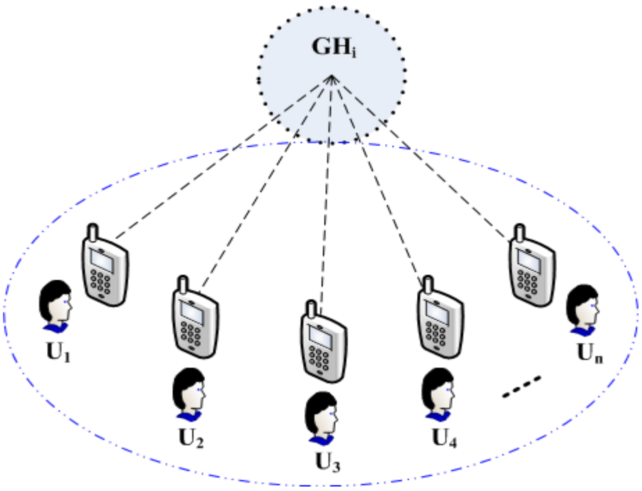


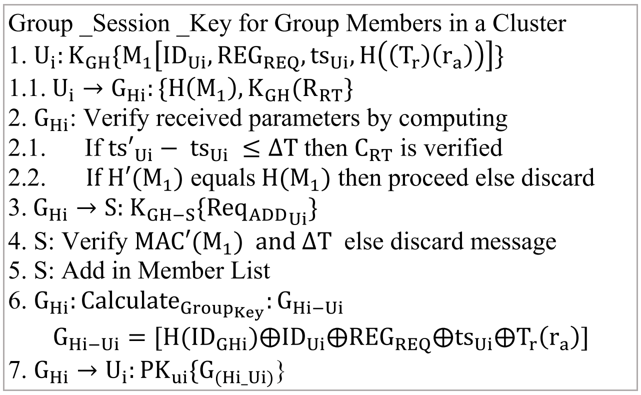
FIGURE 5. Intra-group multi-party key computing.

1) REGISTRATION PHASE

During the registration phase, the user Ui computes a large random number ra to prepare a registration message MR $= (ra_{Ui}||T_{ra_{Ui}}(x)|PW_{Ui}), MCA(ra_{Ui}||T_{ra_{Ui}}(x)||PW_{Ui})$ where $T_{ra_{Ui}}(x)$ is CP parameter pre-shared securely by GH to member nodes. Participant share this message in case of new membership or re-joining to cluster. The G_A verifies message integrity by computing MAC and comparing with received MAC. For verification, G_H compares $T_{ra_{Ui}}(x)$ and also computes the hash of some randomly selected parameter and stores it for further common session key establishment. The scheme uses chaotic maps in distributed way. It is assumed that during chaotic map parameter sharing, both CH and group member adopt secure communication and rely on each other to show their valid domain.

2) INTRA - GROUP KEY COMPUTATION PHASE

Group key establishment scheme can assemble members to arrange a typical common key to ensure secure communication within group. Initially, G_{Hi} establishes key with members by sharing a common CP with all group members. All the participants register with GH by sending registration message (MR). GH computes MAC and compares to verify otherwise reject. The detailed stepwise flow has been shown in figure 6. After cluster/group formation, member node U_i computes a request message M_1 for authentication for secure data transmission using secret and pre-shared credential. Upon receiving participants join request message, the GH firstly verifies credentials by computing time stamp and comparing MAC. After completing verification and data integrity, GH adds valid member nodes information in the routing table and removes additional primary security credential to overcome dictionary and static attacks. By using received parameters, GH computes a secure session key to communicate with members for a specific time slot or required session. Finally, the session key is distributed among members using the secure channel and validate for authorized user having pre-distributed secret parameters and authenticated identities.



```
Group _Session _Key for Group Members in a Cluster
1. $U_i$: $K_{GH}\{M_1[ID_{Ui}, REG_{REQ}, ts_{Ui}, H((T_r)(r_a))]\}$
1.1. $U_i \rightarrow G_{Hi}$: $\{H(M_1), K_{GH}(R_{RT}\}$
2. $G_{Hi}$: Verify received parameters by computing
2.1.     If $ts'_{Ui} - ts_{Ui} \leq \Delta T$ then $C_{RT}$ is verified
2.2.     If $H'(M_1)$ equals $H(M_1)$ then proceed else discard
3. $G_{Hi} \rightarrow S$: $K_{GH-S}\{Req_{ADD_{Ui}}\}$
4. S: Verify $MAC'(M_1)$ and $\Delta T$ else discard message
5. S: Add in Member List
6. $G_{Hi}$: $Calculate_{Group_{Key}}$: $G_{Hi-Ui}$
      $G_{Hi-Ui} = [H(ID_{GHi}) \oplus ID_{Ui} \oplus REG_{REQ} \oplus ts_{Ui} \oplus T_r(r_a)]$
7. $G_{Hi} \rightarrow U_i$: $PK_{ui}\{G_{(Hi\_Ui)}\}$
```

FIGURE 6. Intra-group secure key establishment protocol.

The key distribution is initiated by the GH where the security credentials are preloaded in U_i to transmit message M_1 to G_{Hi}. After receiving the message, G_{Hi} gets the node ids of participant device and secret credentials. Upon receiving registration request message for node authentication and addition in the group, G_{Hi} verifies message freshness by computing Δt and ensure message integrity by calculating MAC. After verification, G_{Hi} verifies node authenticity from trusted server TS. For this purpose, G_{Hi} computes request message $K_{GH-S}\{Req_{ADD_{Ui}}\}$ in cipher text to add this node in global member list. Next, server generates group key G_{Hi-Ui} as $[H(ID_{GHi}) \oplus ID_{Ui} \oplus REG_{REQ} \oplus ts_{Ui} \oplus T_r\,(r_a)]$ that is used to securely exchange data within the group.

3) SECURE INTER-GROUP KEY ESTABLISHMENT

In this section, we have considered the clustered scenario where every GH has different smart devices as member nodes within its transmission range. For secure communication between two members of different clusters, end users need to establish a secret key. We have assumed that GH, server, and their member nodes are working as a fair member. By using a server, each GH can establish a session key after the re-authentication process. In this case, we have assumed that Group heads GH_{i} and GH_{j} are already authenticated by the server and having session keys. Whenever any member of GH_{i} wants to establish a session key for any other member of GH_{j} then request is forwarded by its head via the server to other user group head. Detailed steps of this protocol are illustrated visually in figure 7.

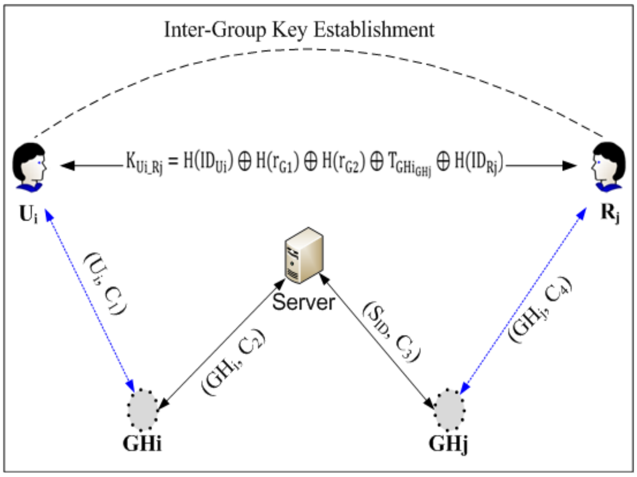

Inter-Group Key Establishment
$K_{Ui\_Rj} = H(ID_{Ui}) \oplus H(r_{G1}) \oplus H(r_{G2}) \oplus T_{GHi_{GHj}} \oplus H(ID_{Rj})$
U_i
R_j
Server
(U_i, C_1)
(GH_i, C_2)
(S_{ID}, C_3)
(GH_j, C_4)
GHi
GHj

FIGURE 7. Inter group session key scenario DMK architecture.

Inter-group key establishment process begins when sender node U_i transmits a message to GH_{i} by encrypting it using key $K_{U_i-GH_i}$ which is pre-established between sender S_i and GH_i as given in (2). In the message, ID_{Ui} is user ID for sender, r_{G1} is the random number pre-shared by GH_i, h_{PW} is pre-shared password. Moreover, ID_{Rj} is an identity of receiver which is the member of neighboring GH_j and T_i is time stamp from U_i to guard against replay attack by checking message freshness.

$$C_1 = E_{K_{Ui-GHi}}(ID_{Ui}, h_{PW}, r_{G1}, T_i, ID_{Rj}, MAC(ID_{Ui}||h_{PW}||r_{G1}||T_i||ID_{RJ})) \quad (2)$$

Sender U_i transmits the message (U_i, C_1) to GH_{i} that decrypts the message using key $K_{U_i-GH_i}$ to extract the values and the MAC of values. After that, GH_{i} subtracts system's timestamp with T_i and ensures that the time difference is less than threshold value Δt for freshness, otherwise packet is discarded. In case of successful result, GH_{i} validates the integrity of message taking MAC' of concatenated values to compare it with MAC in the message. In case of inequality, the message is discarded because message is altered due to some attack or noise in transmission. Finally GH_{i} prepares a message after taking timestamp T_{GH_i} for transmitting towards S by encrypting using secret key K_{GH_i-S} pre-established between sender GH_i and server S as given in (3).

$$C_2 = E_{K_{GH_{i}-S}}(GH_i, ID_{Ui}, h_{PW}, r_{G1}, ID_{Rj}, T_{GHi}, MAC(G_{Hi}||h_{PW}||r_{G1}||ID_{RJ}||T_{GHi}) \quad (3)$$

Group head GH_{i} transmits the message (GH_i, C_2) to server S that decrypts the C_2 using key K_{GH_i-S} to extract the message parameters. It also checks the message freshness and integrity to accept it and load the parameters in volatile memory. In another case, message is discarded. Server takes the time stamp T_S and prepares a message $(S_{ID}, ID_{GH_i}, T_S, MAC\ (S_{ID}, ID_{GHi}, T_S))$ for sending towards GH_{j} by encrypting using pre-established secret key $K_{S\text{-}GH_j}$ as given in (4).

$$C_3 = E_{K_{S\text{-}GHj}}(ID_S, ID_{GH_i}, T_S, MAC\ (ID_S, ID_{GHi}, T_S)) \quad (4)$$

Server S transmits message (S, C_3) to GH_{j} that decrypts the C_3 using key K_{S-GH_j} to get the message parameters and MAC. After that, GH_{j} verifies the freshness and integrity of message and further proceeds if the conditions are true. GH_{j} generates a random number r_{G2} and prepares message $(GH_j, r_{G2}, T_{GHj}, MAC(GH_j||r_{G2}||T_{GHj})$ and then encrypts it using K_{GH_j-S} which is pre-established secret shared key between GH_j and server S as given in (5).

$$C_4 = E_{K_{GHj-S}}(GH_j, r_{G2}, T_{GHj}, MAC(GH_j||r_{G2}||T_{GHj}) \quad (5)$$

Group head GH_{j} transmits the message (GH_j, C_4) to server S that decrypts the cipher text C_4 using key K_{GH_j-S} to extract the message along with parameters and MAC. After that, message freshness and integrity is also checked on the basis of time stamp and MAC respectively. In case of success, GH_{j} obtains the new secret key between U_i and R_j using (6) and then server S transmits to sender U_i and receiver R_j directly for future secure communication. Server also transmits the "Key_Success" message to both GH.

$$K_{Ui-Rj} = H\ (ID_{Ui}) \oplus H(r_{G1}) \oplus H(r_{G2}) \oplus T_{GHi_{GHj}} \oplus H(ID_{Rj}) \quad (6)$$

V. FORMAL MODELING AND ANALYSIS OF DMK

In this section, we have performed formal modeling for our DMK scheme to verify and analyze it using Non-monotonic Cryptographic Protocol (NCP) which is also known as Rubin Logic [42]. It is a standardized formal mechanism to benchmark and verify the essential requirements of security protocols along with common cryptographic operations. It can separately ensure the mandatory steps required to perform a certain security function at sender and receiver as well. Formal modeling assists to figure out the deviating steps in the proposed protocol scenario. It also help to identify the potential outcomes concerning security attack scenarios by comparing with intrinsically standardized steps. It is near to the actual implementation in a programming language. A global set is defined that contains entities, their roles along with globally accessible variables of the modeled protocol. It keep the information in sets and refreshes the states on the clients after each updateable operation. A local set is maintained at every element that also contains subsets including; possession set POSS(), belief set BEL(), seen and behavior list BL(). A structure of local Set for proposed DMK scheme is presented in Table 2.

It includes the detailed stepwise description during message exchange between Sender U_i, GH_i and neighboring GH_j. The fundamental implementation level operations including concatenation of parameters, hash values, encryption and decryption are performed before sending a message. Update operations are performed after sending the message to memorize the newly calculated values. On receiving side, the basic operations are decryption of cipher text, check message freshness using timestamp, compare and verify hash MAC for message integrity. A possession set like POSS (U_i) maintains the list of commonly used variables associated with message encryption, decryption and related operations at $\mathrm{U_i}$. Similarly, sets are maintained at other entities including $\mathrm{POSS(GH_i)}$, $\mathrm{POSS(GH_j)}$ and POSS(S). A Behavior List $\mathrm{BL(U_i)}$ holds details regarding cryptographic operations and information exchanging operations that are performed in close to execution schemes at $\mathrm{U_i}$. Similarly these sets are maintained including $\mathrm{BL(GH_i)}$, $\mathrm{BL(GH_j)}$ and BL(S) at participating entities $\mathrm{GH_i}$, $\mathrm{GH_j}$ and Server S respectively.

The proposed DMK scheme is analyzed for iner group key establishment scenario where member Node$\mathrm{U_i}$ authenticates itself from GHi by sending join request message. The monitoring authority in this scenario is trusted sever TS that distributes the participants' information to network members in their possession. During the message processing at $\mathrm{GH_i}$ by receiving from Ui and transmitting to server S, the state of possession set is given as follows;

$$\mathrm{POSS(GH_i)} = \{\mathrm{ID_{GHi}}, \mathrm{K_{Ui-GHi}}, \mathrm{K_{GHi-S}}, \mathrm{ID_{Ui}}, \mathrm{C_1}, \mathrm{ID_{Ui}}, \mathrm{h_{PW}}, \mathrm{r_{G1}}, \mathrm{T_i}, \mathrm{ID_{Rj}}, \mathrm{H_{Ui}}, \mathrm{T_i'},\ H^*, \mathrm{P_{GHi}}, \mathrm{H_{GHi}}, \mathrm{C_2}, \mathrm{M_2}\}.$$

After the completion of the process, the forget operation identifies and removes the out of scope variables including $\mathrm{C_1}$, $\mathrm{ID_{Ui}}$, $\mathrm{h_{PW}}$, $\mathrm{r_{G1}}$, $\mathrm{T_i}$, $\mathrm{ID_{Rj}}$, $\mathrm{H_{Ui}}$, $\mathrm{T_i}$‘, H^*, $\mathrm{P_{GHi}}$, $\mathrm{H_{GHi}}$ and $\mathrm{C_2}$. New state of possession set is $\mathrm{POSS(GH_i)} = \{\mathrm{ID_{GHi}}$, $\mathrm{K_{Ui-\,GHi}}$, $\mathrm{K_{GHi-S}}\ \}$ after removing the temporary values and $\mathrm{M_2}$ as well. Similarly, the other entities maintain their sets. It follows secure communication mechanisms for secure key generation between $\mathrm{S-CH_i}$, $\mathrm{S-CH_j}$ and $\mathrm{CH_i}$ − $\mathrm{CH_j}$. After transmitting the message, the Update ($\mathrm{M_{ID}}$) operation saves the identity of message in memory for future use when the other parties like receiver replies back. Finally, all the sets at participating entities are refreshed after the completion of inter-group key establishment between member nodes of GHi and GHj.

TABLE 2. Local set for DMK at sender, GHi server and GHj.

1. Sender (Ui)
$\mathrm{POSS(U_i)} = \{ \mathrm{ID_{Ui}}, \mathrm{K_{Ui\text{ - }GHi}}\}$ $\mathrm{BEL(Ui)} = \{ \#(\mathrm{ID_{Ui}}), \#(\mathrm{K_{Ui\text{ - }GHi)}}\ \}$ BL(Ui) = $\mathrm{Concat(ID_{Ui}, h_{PW}, r_{G1}, T_i, ID_{Rj})} \rightarrow \mathrm{P_{Ui}}$ $\mathrm{Hash(\ h(.); P_{Ui})} \rightarrow \mathrm{H_{Ui}}$ $\mathrm{Encrypt(\ \{ ID_{Ui}, h_{PW}, r_{G1}, T_i, ID_{Rj}, H_{Ui}\ \}_{K\ Ui\text{-}GHi}\ )} \rightarrow \mathrm{C_1}$ $\mathrm{Send(GHi,\ \{ ID_{Ui}, C_1\})} \rightarrow \mathrm{M_1}$, Update $(\mathrm{ID_{Rj}})$
2. Group Head (GHi)
$\mathrm{POSS(GHi)} = \{ \mathrm{ID_{GHi}}, \mathrm{K_{Ui\text{ - }GHi}}, \mathrm{K_{GHi\text{-}S}}\ \}$ $\mathrm{BEL(GHi)} = \{\#(\mathrm{ID_{GHi}}), \#(\mathrm{K_{Ui\text{ - }GHi}}), {}_{\#}(\mathrm{K_{GHi\text{-}S}})\ \}$ BL(GHi) = $\mathrm{Receive(GHi,\ \{ ID_{Ui}, C_1\})}$ and $\mathrm{Split(\ \{ ID_{Ui}, C_1\})}$ $\mathrm{Decrypt(\{ C_1 \}_{K\ GHi\text{-}Ui}\ )}$ to acquire $[\mathrm{ID_{Ui}, h_{PW}, r_{G1}, T_i, ID_{Rj}, H_{Ui}}]$ $\mathrm{MsgFreshness(T_i‘\text{-}T_i)} \geq \Delta t$ if true then Msg is aborted $\mathrm{MAC(\{Concat(ID_{Ui}, h_{PW}, r_{G1}, T_i, ID_{Rj})\})} \rightarrow H^*$ $\mathrm{Verify(H_{Ui}},\ H^*)$ if mismatch, then abort $\mathrm{Concat(ID_{GHi}, ID_{Ui}\ , h_{PW}, r_{G1}, ID_{Rj}, T_{GHi})} \rightarrow \mathrm{P_{GHi}}$ $\mathrm{Hash(\ h(.); P_{GHi})} \rightarrow \mathrm{H_{GHi}}$ $\mathrm{Encrypt(\ \{ ID_{GHi}, ID_{Ui}\ , h_{PW}, r_{G1}, ID_{Rj}, T_{GHi}, H_{GHi}\ \}_{K\ GHi\text{-}S}\ )} \rightarrow$ $\mathrm{C_2}$ $\mathrm{Send(\ ID_S,\ \{ ID_{GHi}, C_2\})} \rightarrow \mathrm{M_2}$ $\mathrm{Update(M_{ID})}$
3. Server (S)
$\mathrm{POSS(S)} = \{ \mathrm{ID_S}, \mathrm{K_{S–GHi}}, \mathrm{K_{S–GHj}}\ \}$ $\mathrm{BEL(S)} = \{\#(\mathrm{ID_S}),\#(\mathrm{K_{S–GHi}}),\#(\mathrm{K_{S–GHj}})\}$ BL(S) = $\mathrm{Receive(\ S,\ \{ ID_{GHi}, C_2\})}$ and $\mathrm{Split(\ \{ ID_{GHi}, C_2\})}$ $\mathrm{Decrypt(\{ C_2 \}_{K\ GHi\text{-}S}\ )}$ to obtain $[\mathrm{ID_{GHi}, ID_{Ui}\ , h_{PW}, r_{G1}, ID_{Rj}}$, $\mathrm{T_{GHi}, H_{GHi}}]$ $\mathrm{MsgFreshness(T_{GHi}‘\text{-}T_{GHi})} \geq \Delta t$ if true, then Msg is aborted $\mathrm{MAC(\{Concat(ID_{GHi}, ID_{Ui}\ , h_{PW}, r_{G1}, ID_{Rj}, T_{GHi})\})} \rightarrow H^{\sim}$ $\mathrm{Verify(H_{GHi}},\ H^{\sim})$ if mismatch, then abort $\mathrm{Hash(\ h(.); Concat(S_{ID}, ID_{GHi}, T_S, ID_{Rj}))} \rightarrow \mathrm{H_S}$ $\mathrm{Encrypt(\ \{ S_{ID}, ID_{GHi}, T_S, ID_{Rj}, H_S\}_{K\ GHj\text{-}S}\ )} \rightarrow \mathrm{C_3}$ $\mathrm{Send(GHj,\ \{ ID_S, C_3\})} \rightarrow \mathrm{M_3}$, $\mathrm{Update(M_{ID})}$ $\mathrm{Receive(\ S,\ \{ ID_{GHj}, C_4\}), Split(\ \{ ID_{GHj}, C_4\})}$ $\mathrm{Decrypt(\{ C_4 \}_{K\ GHj\text{-}S}\ )}$ to get $[\mathrm{ID_{GHj}, r_{G2}, T_{GHj}, H_{GHj}}]$ $\mathrm{MsgFreshness(T_{GHj}‘\text{-}T_{GHj})} \geq \Delta t$ if yes, then abort $\mathrm{MAC(\{Concat(ID_{GHj}, r_{G2}, T_{GHj})\})} \rightarrow H^{\wedge}$ $\mathrm{Verify(H_{GHj}},\ H^{\wedge})$ if mismatch, then abort
4. Neighboring Group Head (GHj)
$\mathrm{POSS(GHj)} = \{ \mathrm{ID_{GHj}}, \mathrm{K_{Ui\text{ - }GHj}}, \mathrm{K_{GHj\text{-}S}}\ \}$ $\mathrm{BEL(GHj)} = \{\#(\mathrm{ID_{GHj}}), \#(\mathrm{K_{Ui\text{ - }GHj}}), {}_{\#}(\mathrm{K_{GHj\text{-}S}})\ \}$ BL(GHj) = $\mathrm{Receive(GHj,\ \{ ID_S, C_3\})}$ and $\mathrm{Split(\ \{ IDs, C_3\})}$ $\mathrm{Decrypt(\{ C_3 \}_{K\ GHj\text{-}S}\ )}$ to obtain $[\mathrm{s_{ID}, ID_{GHi}, T_S, ID_{Rj}, H_S}]$ $\mathrm{MsgFreshness(T_S‘\text{-}T_S)} \geq \Delta t$ if condition is true then abort Msg $\mathrm{MAC(\{Concat(S_{ID}, ID_{GHi}, T_S, ID_{Rj})\})} \rightarrow H^{+}$ $\mathrm{Verify(H_S},\ H^{+})$ if mismatch, then abort $\mathrm{Hash(\ h(.); Concat(ID_{GHj}, r_{G2}, T_{GHj}))} \rightarrow \mathrm{H_{GHj}}$ $\mathrm{Encrypt(\ \{ ID_{GHj}, r_{G2}, T_{GHj}, H_{GHj}\ \}_{K\ GHj\text{-}S}\ )} \rightarrow \mathrm{C_4}$ $\mathrm{Send(\ S,\ \{ ID_{GHj}, C_4\})} \rightarrow \mathrm{M_4}$, $\mathrm{Update(M_{ID})}$

VI. SECURITY ANALYSIS

The proposed work utilizes the chaotic maps with collision resistant one-way hash functions. The two chaotic map problems that can be compromised in CMDLP and computational chaotic maps Deffie-Hellman (CCMDHP) are defined as follows; i) In CMDLP, if x, y belonging to R [−1, 1] then finding out the solution of integer "a" is not feasible for y $=\mathrm{T_a}$(x) to get its output; ii) For CCMDHP, if there are more than two parameters like x,$T_a(x)$ and $T_b(x)$ then calculating the value of $T_{ab}(x)$ is not possible.

By exploiting these two strong assumptions, we show security investigation of the proposed keying mechanism. The theoretical investigation indicates that the proposed method defeats the security issue of M-PAKE convention and could successfully oppose well known- attacks. Also, it is productive and viable. Moreover, we have analyzed some performance and security issues of the existing schemes by opposing to the conventional secret password cryptosystems in [16] and [43]-[45]. In these schemes, the authors presented efficient mathematical methods to reduce the computational expenses. The security of the proposed protocol does not depend on the degree of polynomials to be high. Therefore, it is pointless to choose substantial values rg and a random number *s* chosen by GH and server TS respectively. By applying possible security concerns, the security strength of proposed scheme is satisfied. Replay attack is prevented by including the time stamp and ensuring the message freshness at each step as per threshold value.

A. MUTUAL AUTHENTICATION

Mutual authentication means that each participating entity is authenticated by neighboring entity. In our scheme, the trusted server and GH are trustworthy yet they do not know session key where chaotic opeartions are used to produce a random number and one-way hash values. In case if server gets crashed then can't affect the established keys as session keys are saved in a pair-wise manner. As proposed scheme is dynamic in which instead of single server authentication, both server and GH are responsible to authenticate participants and verify their legitimacy.

B. MESSAGE-INTEGRITY

Message authentication code is used to provide integrity of a message when transferring over a network. Chaotic maps ensure to generate the one-way hash values that have very low probability to have collusion where two different text can generate the same hash value. To establish group key, GH and server play a vital role by ensuring the integrity of entire message using MAC. In existing M-PAKE scenario, server, and participant share their information on the assumption of the secure channel and does not ensure message integrity. The integrity of security credentials inside the message is assured by using chaotic maps. In other similar maps like confusion and disorderly maps, the concept of chaos theory is used to ensure one-way hash operations. In image based confusion-diffusion maps, more resources extensive operations are needed.

C. OFFLINE DICTIONARY ATTACKS

An intruder can intercept the messages to check for plain text password or analyze the traffic to extract the passwords. In the beginning, all participants generate a random number using chebyshev polynomials and send the chaotic map based hash value of chosen password to guard against the offline guessing attacks after traffic capturing. However, an intruder will not be able to verify the exact password without getting the R_g or r_s or r_{G1}& r_{G2} in all scenario which is a random number generated by group head and trusted server.

VII. RESULTS AND ANALYSIS

In this section, we have evaluated different metrics to present the performance of DMK and compared it with M-PAKE [24], P3-PAKE [23], ECMA [36], CMPA [37], CGKA [37] and GKCH [40] schemes. For testing the functionality for GH to server authentication and keying, a PC with Windows-7 operating system, 2 GHz processor and 2 GB RAM is used as a server. The smartphones with Android 5.0 OS, 32 GB memory, and 64-bit processor are used as GHs. A number of mobile users are considered as members varied from 50 to 250 members divided into 4 clusters. Moreover, the member to GH intra and inter-key establishment scenario is also simulated using NS-2.35 where separate classes are maintained for GH and member node to handle send, receive, encrypt, decrypt and hash operations. In TCL file, node deployment, messaging patterns and traffic flows are included. It also contains the node configurations settings for member and GH node creation by calling constructors of appropriate classes. Results are extracted from trace files by using AWK script files.

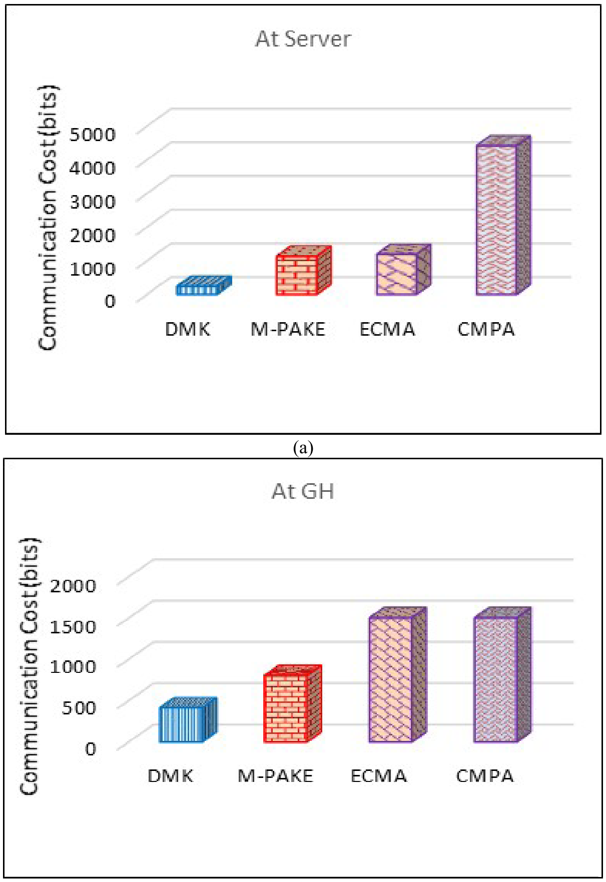
A. COMMUNICATION OVERHEAD

For **c**ommunication cost calculation, we have considered the cost between GH to server and a member as well. figure 8(a) elucidates communication overhead at server and 8(b) represents the cost for GH for the intra key establishment scenario.

At GH, for the proposed DMK scheme 240 bits are consumed where GID = 16 bits, R_g = 64 bits, ID and Hash=160 bits in the first message. In the second message which is acknowledged message, 160 bits for hash transmission and hence total communication cost for authentication process during Phase I equals 400 bits. At the server side, the first message takes 224 bits ad the second message is ACK confirmation of 16 bit only which results in 240 bits of messages. Proposed scheme is compared with schemes M-PAKE [24], ECMA [36] and CMPA [37] where the cost of DMK scheme in the first message is 256 bits where GID = 16 bits, R_g = 64 bits, Hash=160 bits. Figure 9 illustrates the size of a message from participant node. A message contains Participant Id = 32-bits, one-way hash function = 160-bits, random number = 210-bit, AES=128 bits, and ACK = 160 bits in DMK.

B. COMPUTATION COST

The total computation time of all parameters involved in the whole procedure is obtained. The computation time for one-way hash function is 0.20ms. For computing CP, 21.02ms are consumed. A symmetric encryption/decryption operation consumes 8.7ms, and 63.05ms are required to compute ECC. By this calculation, DMK is much better than existing M-PAKE [24] and P3-PAKE [23] which used



(a)

(b)

FIGURE 8. Communication costs in bits at (a) server and (b) group head for intra group keying scenario.

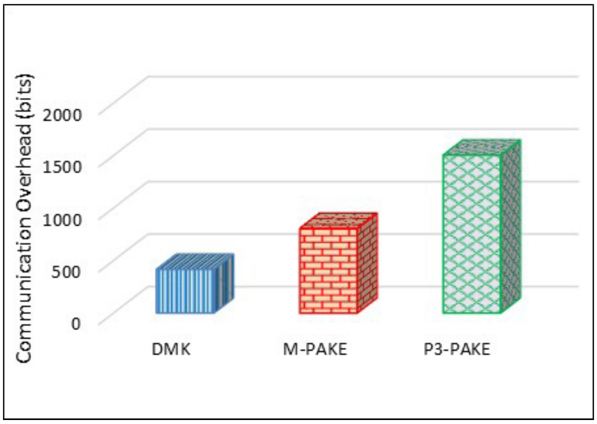


FIGURE 9. Communication overhead by member nodes.

a heavy algorithm like ECC and RSA. The CP calculation offers smaller key size, speedier calculation, less memory utilization and transmission capacity. Figure 10 elucidates the computation costs where DMK dominates over preliminaries. In DMK, the CP computations are much less extensive than exponentiation calculations for ECC.

Figure 11 elucidates the cost of chaotic and hash operations during key establishment. In proposed DMK, the trusted server and GH needs $(\delta + 1)$ and δ chaotic operation respectively where δ denotes the span of access in the network. The trusted server computes two chaotic operations for itself and GH. It also utilizes four one-way hash functions. GH computes five hash operations and one chaotic operation during participant's authentication process. CGKA [37] and GKCH [40] require 13 and 15 hash operations respectively whereas our proposed DMK needs only 10 operations.

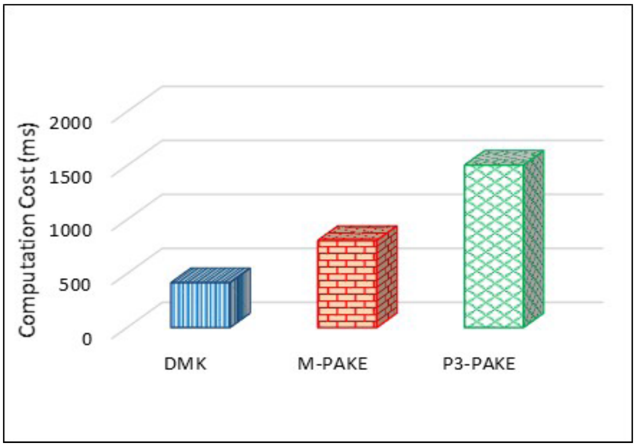


FIGURE 10. Computation cost at member node.

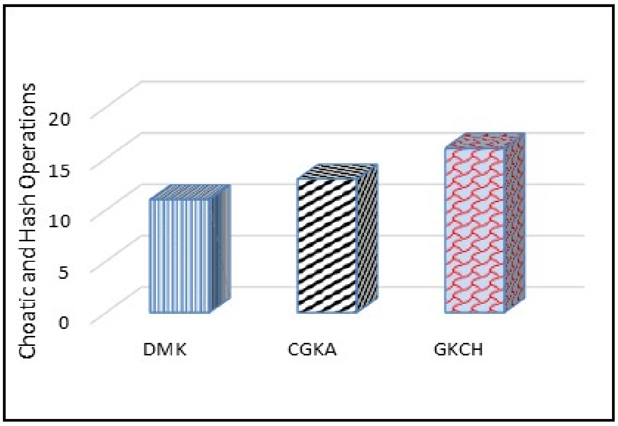


FIGURE 11. Chaotic and hash operations count.

Figure 12 illustrates the message preparation cost during the member authentication phase. Total message size computation in proposed MDK scheme is 480-bits and 256-bits during GH authentication phase by trusted server and end-nodes by group head respectively. So, the total message size is 796-bits for the whole authentication. In counterparts, the computation for message sizes are 1132 bits, 1192 bits, 4416 bits and 3712 bits for M-PAKE, P3-PAKE, ECMA, and CMPA respectively.

C. RESILIENCE

During the participant's authentication and initialization phase, intruders can attack to breach the security and grab some portion of data. In this section, we have deduced

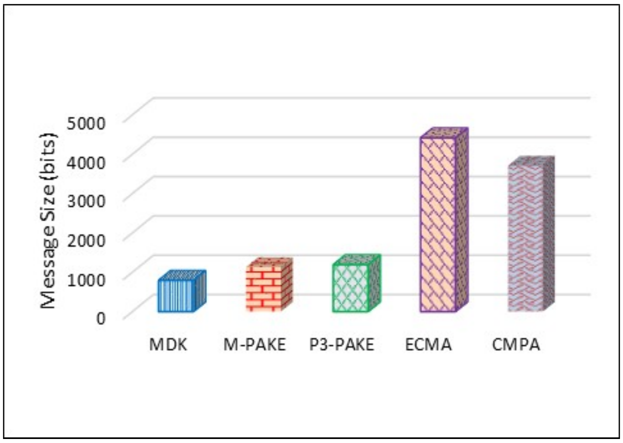


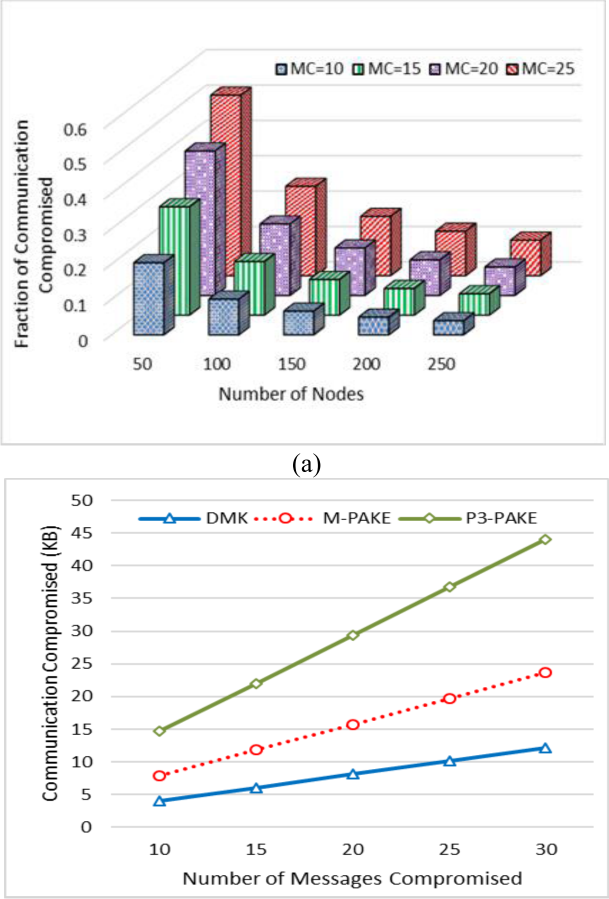
FIGURE 12. Message size in bits.

the chances of compromised data in the network during initialization phase when participants send an authentication request to the GH. We have presented the probability for compromising the communication when a few nodes are compromised out of total participant nodes varying from 50 to 250. In this scenario, the probability P_σ as given in (7) predicts the chances that a participant is compromised. In this equation, N represents the total number of participants whereas σ represents number of participants compromised. In this case, N-2 shows that sender and receivers are excluded from set of compromised nodes. The term N-1 means to exclude the sender from total nodes which is supposed to be uncompromised.

$$P_\sigma = 1-\left(\frac{N-2}{\sigma}\right) \Big/ \left(\frac{N-1}{\sigma}\right) \tag{7}$$

Figure 13(a) elucidates fraction of communication compromised which is 0.067114, 0.100671, 0.134228 and 0.167785 for $\sigma = 10$, 15, 20, and 25 respectively for a total of 150 participants. In case of compromising an intermediate device, the intruder can grab the data and security credentials stored in that device. In proposed scheme, the distributed scenario guard against secrecy exposed during setup phase, as the credentials are stored distributed instead of central point. Moreover, the number of compromised stored security credentials on individual nodes is for a single particular session key.

Figure 13(b) elucidates the amount of communication compromised in KB when a number of messages are compromised. We have considered 10 to 30 numbers of compromised messages for a particular session. The fraction of compromised communication is analyzed and compared with existing techniques. In presented scenario, if there are 20 compromised messages then the amount of compromised communication is 8.125 KB, 15.742 KB and 29.375 KB for DMK, M-PAKE and P3-PAKE respectively. Our proposed DMK scheme dominates in terms of resilience.



(a)

(b)

FIGURE 13. Fraction of communication compromised for number of nodes compromised is presented in (a) and amount of communication compromised by compromising messages is presented in (b).

VIII. CONCLUSION

A novel distributed key management scheme is presented by utilizing Chebyshev polynomials and chaotic maps for cryptographic operations. Our work includes the session key establishment between the group heads and server in phase -I. In phase-II, we have presented the intra and inter group secret key establishment schemes between the GH and smart devices. In our proposed DMK scheme, GH uses chaotic map based one-way hash functions to ensure integrity of message. Chebyshev polynomials are used to for key establishment and cipher text generation. It utilizes the semi-group property to generate same value by using two different polynomial generators at two different devices. Rubin Logic is applied to formally model and analyze the DMK. For the validation of DMK, we have performed simulation for phase II using NS-2.35. Moreover, a and testbed is setup for validating schemes of phase I. Security analysis is conducted to present the strength of our proposed scheme against different security attacks. Results demonstrate the dominance of propose DMK over preliminaries. Communication and computation cost is reduced by 62% and 58% respectively as compared to M-PAKE and P3-PAKE schemes. Communication cost at group heads and server is reduced by 20% for M-PAKE and almost 87% percent as compared to ECMA and CMPA.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] D. Pishva, "Internet of Things: Security and privacy issues and possible solution," in *Proc. ICACT*, Bongpyeong, South Korea, Feb. 2017, pp. 797–808.

[3] H. El-Sayed *et al.*, "Edge of Things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment," *IEEE Access*, vol. 6, pp. 1706–1717, 2018.

[4] Z. Mahmood, H. Ning, and A. Ghafoor, "A polynomial subset-based efficient multi-party key management system for lightweight device networks," *Sensors*, vol. 17, no. 4, p. 670, 2017.

[5] H. Lin, Z. Yan, Y. Chen, and L. Zhang, "A survey on network security-related data collection technologies," *IEEE Access*, vol. 6, pp. 18345–18365, 2018.

[6] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1294–1312, 3rd Quart., 2015.

[7] L. Kocarev, "Chaos-based cryptography: A brief overview," *IEEE Circuits Syst. Mag.*, vol. 1, no. 3, pp. 6–21, Mar. 2001.

[8] M. S. Baptista, "Cryptography with chaos," *Phys. Lett. A*, vol. 240, nos. 1–2, pp. 50–54, 1998.

[9] H. Ning, H. Liu, and L. T. Yang, "Aggregated-proof based hierarchical authentication scheme for the Internet of Things," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 3, pp. 657–667, Mar. 2015.

[10] H. Zhou and X.-T. Ling, "Problems with the chaotic inverse system encryption approach," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 44, no. 3, pp. 268–271, Mar. 1997.

[11] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Netw.*, vol. 20, pp. 96–112, Sep. 2014.

[12] Q. Xie, J. Zhang, and N. Dong, "Robust anonymous authentication scheme for telecare medical information systems," *J. Med. Syst.*, vol. 37, p. 9911, Apr. 2013.

[13] M. S. Farash, M. A. Attari, and S. Kumari, "Cryptanalysis and improvement of a three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps," *Int. J. Commun. Syst.*, vol. 30, no. 1, p. e2912, Jan. 2017.

[14] L. Zhang, S. Zhu, and S. Tang, "Privacy protection for telecare medicine information systems using a chaotic map-based three-factor authenticated key agreement scheme," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 2, pp. 465–475, Mar. 2017.

[15] Y. Sun, H. Zhu, and X. Feng, "A novel and concise multi-receiver protocol based on chaotic maps with privacy protection," *Int. J. Netw. Secur.*, vol. 19, no. 3, pp. 371–382, 2017.

[16] Q. Xie, J. Zhao, and X. Yu, "Chaotic maps-based three-party password-authenticated key agreement scheme," *Nonlinear Dyn.*, vol. 74, no. 4, pp. 1021–1027, 2013.

[17] D. Xiao, X. Liao, and S. Deng, "One-way Hash function construction based on the chaotic map with changeable-parameter," *Chaos, Solitons Fractals*, vol. 24, no. 1, pp. 65–71, 2005.

[18] Y. Li, X. Li, and X. Liu, "A fast and efficient hash function based on generalized chaotic mapping with variable parameters," *Neural Comput. Appl.*, vol. 28, no. 6, pp. 1405–1415, 2017.

[19] K. Zhang, X. Liang, R. Lu, and X. Shen, "PIF: A personalized fine-grained spam filtering scheme with privacy preservation in mobile social networks," *IEEE Trans. Comput. Social Syst.*, vol. 2, no. 3, pp. 41–52, Sep. 2015.

[20] T. Liu, Q. Pu, Y. Zhao, and S. Wu, "ECC-based password-authenticated key exchange in the three-party setting," *Arabian J. Sci. Eng.*, vol. 38, no. 8, pp. 2069–2077, 2013.

[21] Q. Pu, J. Wang, S. Wu, and J. Fu, "Secure verifier-based three-party password-authenticated key exchange," *Peer-to-Peer Netw. Appl.*, vol. 6, no. 1, pp. 15–25, 2013.

[22] S.-Y. Chiou and C.-H. Lin, "An efficient three-party authentication scheme for data exchange in medical environment," *Secur. Commun. Netw.*, vol. 2018, Jan. 2018, Art. no. 9146297. [Online]. Available: https://www.hindawi.com/journals/scn/2018/9146297/cta/

[23] J. O. Kwon, I. R. Jeong, and D. H. Lee, "Practical password-authenticated three-party key exchange," *KSII Trans. Internet Inf. Syst.*, vol. 2, no. 6, pp. 312–332, 2008.

[24] C.-F. Lu, "Multi-party password-authenticated key exchange scheme with privacy preservation for mobile environment," *KSII Trans. Internet Inf. Syst.*, vol. 9, no. 12, pp. 5135–5149, 2015.

[25] H. T. T. Nguyen, M. Guizani, M. Jo, and E.-N. Huh, "An efficient signal-range-based probabilistic key predistribution scheme in a wireless sensor network," *IEEE Trans. Veh. Technol.*, vol. 58, no. 5, pp. 2482–2497, Jun. 2009.

[26] R. Amin and G. Biswas, "A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks," *Ad Hoc Netw.*, vol. 36, pp. 58–80, Jan. 2016.

[27] M. Garcia-Bosque, C. Sánchez-Azqueta, A. Pérez, A. Martínez, and S. Celma, "Fast and secure chaotic stream cipher with a MEMS-based seed generator," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, May 2017, pp. 1–6.

[28] W. Liu, H. Liu, Y. Wan, H. Kong, and H. Ning, "The yoking-proof-based authentication protocol for cloud-assisted wearable devices," *Pers. Ubiquitous Comput.*, vol. 20, no. 3, pp. 469–479, 2016.

[29] R. Hasimoto-Beltran, F. Al-Masalha, and A. Khokhar, "Performance evaluation of chaotic and conventional encryption on portable and mobile platforms," in *Chaos-Based Cryptography. Studies in Computational Intelligence*, vol. 354, L. Kocarev and S. Lian, Eds. Berlin, Germany: Springer, 2011. https://link.springer.com/chapter/10.1007%2F978-3-642-20542-2_11#citeas

[30] T. Qiu, X. Liu, M. Han, H. Ning, and D. O. Wu, "A secure time synchronization protocol against fake timestamps for large-scale Internet of Things," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1879–1889, Dec. 2017.

[31] Y. Li, W. K. S. Tang, and G. Chen, "Generating hyperchaos via state feedback control," *Int. J. Bifurcation Chaos*, vol. 15, no. 10, pp. 3367–3375, 2005.

[32] K. S. Halle, C. W. Wu, M. Itoh, and L. O. Chua, "Spread spectrum communication through modulation of chaos," *Int. J. Bifurcation Chaos*, vol. 3, no. 2, pp. 469–477, 1993.

[33] S. Mukhopadhyay, M. Mitra, and S. Banerjee, "Chaos synchronization with genetic engineering algorithm for secure communications," in *Chaos Synchronization and Cryptography for Secure Communications: Applications for Encryption*. Hershey, PA, USA: IGI Global, 2011, pp. 476–509. [Online]. Available: https://www.igi-global.com/about/

[34] H. Dedieu, M. P. Kennedy, and M. Hasler, "Chaos shift keying: Modulation and demodulation of a chaotic carrier using self-synchronizing Chua's circuits," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 40, no. 10, pp. 634–642, Oct. 1993.

[35] U. Feldmann, M. Hasler, and W. Schwarz, "Communication by chaotic signals: The inverse system approach," in *Proc. IEEE Int. Symp. Circuits Syst.*, Apr./May 1995, pp. 680–683.

[36] X. Hao, J. Wang, Q. Yang, X. Yan, and P. Li, "A chaotic map-based authentication scheme for telecare medicine information systems," *J. Med. Syst.*, vol. 37, p. 9919, Apr. 2013.

[37] C.-T. Li, C.-C. Lee, and C.-Y. Weng, "A secure chaotic maps and smart cards based password authentication and key agreement scheme with user anonymity for telecare medicine information systems," *J. Med. Syst.*, vol. 38, p. 77, Sep. 2014.

[38] X. Li, F. Wu, M. K. Khan, L. Xu, J. Shen, and M. Jo, "A secure chaotic map-based remote authentication scheme for telecare medicine information systems," *Future Gener. Comput. Syst.*, vol. 84, pp. 149–159, Jul. 2018. [Online]. Available: https://doi.org/10.1016/j.future.2017.08.029.

[39] Z.-Y. Cheng, Y. Liu, C.-C. Chang, and S.-C. Chang, "A practical secure chaos-based group key agreement protocol suitable for distributed network environment," *Int. J. Innov. Comput. Inf. Control*, vol. 9, pp. 1935–1949, May 2013.

[40] X. Guo, J. Zhang, M. K. Khan, and K. Alghathbar, "Secure chaotic map based block cryptosystem with application to camera sensor networks," *Sensors*, vol. 11, no. 2, pp. 1607–1619, 2011.

[41] X. Guo and J. Zhang, "Secure group key agreement protocol based on chaotic Hash," *Inf. Sci.*, vol. 180, no. 20, pp. 4069–4074, 2010.

[42] A. D. Rubin and P. Honeyman, "Nonmonotonic cryptographic protocols," in *Proc. Comput. Secur. Found. Workshop VII (CSFW)*, Jun. 1994, pp. 100–116.

[43] R. Song, "Advanced smart card based password authentication protocol," *Comput. Standards Interfaces*, vol. 32, nos. 5–6, pp. 321–325, 2010.

[44] D.-Z. Sun, J.-P. Huai, J.-Z. Sun, J.-X. Li, J.-W. Zhang, and Z.-Y. Feng, ''Improvements of Juang's password-authenticated key agreement scheme using smart cards,'' *IEEE Trans. Ind. Electron.*, vol. 56, no. 6, pp. 2284–2291, Jun. 2009.

[45] X. Li, W. Qiu, D. Zheng, K. Chen, and J. Li, ''Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards,'' *IEEE Trans. Ind. Electron.*, vol. 57, no. 2, pp. 793–800, Feb. 2010.

ZAHID MAHMOOD born in Jammu & Kashmir, Pakistan. He received the B.S. degree in computer science from the University of Baluchistan, Quetta, the M.S. degree in computer sciences from International Islamic University, Islamabad, Pakistan, and the Ph.D. degree in computer science from the University of Science and Technology Beijing, Beijing, China. He has been with International Islamic University, Islamabad, after that he join the Mohi-Ud-Din Islamic University Nerian Sharif, as a Lecturer. His major research area is key management techniques in wireless sensor network and lightweight cryptography techniques for Internet of Things, and authentication, privacy, and secure communication for wearable devices.

ATA ULLAH received the B.S. and M.S. degrees in computer science from COMSATS, Islamabad, Pakistan, in 2005 and 2007, respectively, the Ph.D. degree in computer science from IIUI, Pakistan, in 2016, in the area of wireless network security. From 2007 to 2008, he was a Software Engineer with Streaming Networks, Islamabad. He joined NUML, Islamabad, Pakistan, in 2008, and was an Assistant Professor/Head Project Committee with the Department of Computer Science, since 2017. He is currently a Research Fellow with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China. He has supervised 110 projects at under graduate level. He has published several papers in ISI indexed impact factor journals and international conferences. His areas of interests are WSN, IoT, cyber physical social thinking space, health-services, NGN, VoIP, and their security solutions. He was a recipient at one International and 45 National Level Software Competitions. He was a recipient of ICT funding for the development of projects. He is also a reviewer and a guest editor for conference and Journal Publications. He remained faculty partner for industrial collaboration in software development. He has programming expertise in C, C#, Java, PHP, and NS2.

HUANSHENG NING (M'10–SM'13) received the B.S. degree from Anhui University in 1996 and the Ph.D. degree in Beihang University in 2001. From 2002 to 2003, he was with Aisino Co. From 2004 to 2013, he was an Associate Professor with the School of Electronic and Information Engineering, Beihang University. In 2013, he was a Professor and a Vice Dean with the School of Computer & Communication Engineering, University of Science & Technology Beijing. His research interests include cybermatics, Internet of Things, and cyber-physical social systems. He is a Founder and the Chair of the Cybermatics and Cyberspace International Science and Technology Cooperation Base, the Co-Founder and the co-chair of the IEEE Systems, Man, and Cybernetics Society Technical Committee on Cybermatics, and the Co-Founder and the Vice Chair of the IEEE Computational Intelligence Society Emergent Technologies Technical Committee Task Force on Smart Word. He has presided over many research projects, including the Natural Science Foundation of China and the National High Technology Research and Development Program of China. He has served as Associate Editor of many well reputed Journals. He has published over 70 journal/conference papers.