Journal of Theoretical and Applied Information Technology
30th June 2023. Vol.101. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CUCKOO HASH BASED MULTI FACTOR AUTHENTICATION (CH-MFA) - IN SECURED COMMUNICATION WIRELESS SENSOR NETWORK

C. VENKATACHALAM[1], Dr.A.SURESH[2]

[1]Ph.D Research Scholar, Dept of Computer Science, Periyar University, Salem-636011.

[2]Assistant Professor, Dept of Computer Science, Sona College of Arts and Science, Salem – 636005.

E-mail: [1]venkatachalam2@gmail.com

ABSTRACT

As a vital component of the information sensing and aggregating for big data, cloud computing and information security in wireless sensor networks (WSN) is critical. Due to constrained sensor node resources, WSN is becoming a vulnerable target to many security attacks. Cuckoo Hash based Multi Factor Authentication (CH-MFA) Mechanism for secured communication in wireless sensor organization. CH-MFA instrument includes two phases, the registration and authentication phases. At first, in the registration phase, the node must register the node's ID and password to the base station. Then, at that point, the registered information is put away in the base station utilizing Cuckoo Hash work. The prototype model for cloud computing's cloud server uses open source technology. The proposed framework shows a close agreement with the standard criteria for security.

Keywords: *Cuckoo, Hash, Multi, Factor, Authentication, communication, Wireless, Sensor, Network;*

1. INTRODUCTION

A distributed and self-organized WSN is a progression of autonomous small-scale sensor nodes cooperating towards a typical objective. WSN has small sensor nodes comprising sensor modules, data processing, and a network. WSN is a progression of a few sensor nodes thickly distributed for military and standard applications in harsh environments. WSN regularly comprises a base station for radio communication with various wireless sensors. Wireless sensor node data is prepared, compacted, and sent straightforwardly to the base station. WSN has numerous limitations, for example, a low processing power, helpless memory, deficient energy assets, the utilization of unstable wireless organizations and the utilization of sensor nodes in an unattended climate and Particular transport attacks, Wormholes attacks, Sinkhole attacks, Sybil attacks, HELLO flood attacks, Acknowledgment spoofing, sniffing attacks, energy dump, dark opening attacks, service attacks deviation, smugglers research assault, privacy breach attacks, and clone attacks are various conceivable outcomes of assault on the WSN. An adversary can catch and concentrate the center materials from a sensor node. The interloper will reinvent the node to reproduce the node caught after a node has been caught. These duplicates should be found in all organization zones (or replicas). These node imitation attacks are hazardous for sensor network activities. However, the gatecrasher will make many reproduction nodes, as he needs a solitary caught sensor node. The enemy restricts the reproduction nodes, yet it has essential materials which cause them to give off an impression of being endorsed network members. A clone assault is likewise undeniably challenging to identify. WSN can be mobile or static. Arbitrarily, nodes are utilized in static WSN sensors and do not change their areas after sending. In mobile WSN, after establishment, the sensor nodes will move. There are centralized and distributed two types of recognition technologies in static WSN.

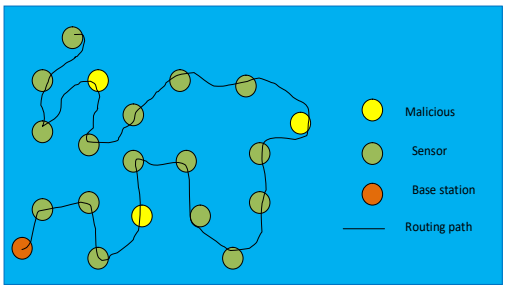

Figure 1: Malicious node identification in WSN

When another node joins the organization, it communicates a location requested containing its location and name to its neighbors in a center method for node replication. One of his neighbors sent this contention to the base station. The base station can rapidly recognize any pair of nodes with a similar identity yet at numerous locations with location information for all nodes in the organization. By compromising with the base station or deterring the course to the base station, rivals can consolidate several replicas in the organization, which is the critical downside.

There is extraordinary interest in setting up a multi-factor authentication (MFA) framework, which requires at least two authentication factors (i.e., knowledge, possession, identity) to approve clients when they log into a cloud service. A few well known services utilize MFA, as a rule, as a discretionary component that is deactivated, of course, like Amazon, Google, and Microsoft. Each method requires an out-of-band channel (e.g., an App, instant message) and an additional stage for a client, which diminishes convenience. Likewise, there have been many scholarly propositions for MFA systems.

Static password is the most notable and widespread authentication method. Nonetheless, an ordinarily static password is a constant person string. Convoluted passwords are hard to be recalled; then again, specific passwords are not difficult to recollect, and generally, a similar password is utilized in multiple accounts. Subsequently, static passwords are handily broken. Programmers can utilize numerous strategies to take passwords, such as shoulder surfing, sneaking around, sniffing, speculating, etc. Static password authentication is confronted with these security dangers. Numerous information security occasions that occurred as of late are identified with the insecurity of static passwords. Essential and straightforward passwords are becoming weak organization security problems.

Many methodologies of multi-factor authentication Weber and Frank proposed authentication by utilizing PIN shipped off email as the knowledge factor of the two factor authentication. Bhargav and Abhilasha proposed a method that utilizes biometrics as the extra authentication factor. Sabzevar, Alireza, and Angelos expressed a method of utilizing a graphical picture to address a password; the password is entered by pointing to suitable marks of the picture, which the client has acquired through his mobile device (eminently handheld device) from the service providers.[1–20]

2. PROBLEM STATEMENT

Many industries, including healthcare, agriculture, transportation, and industrial control systems, now depend on wireless sensor networks (WSNs). These networks are made up of tiny sensor nodes that wirelessly communicate with one another to carry out different activities. WSNs are also susceptible to a number of security risks because of their distributed and unattended nature, including eavesdropping, manipulation, and denial-of-service attacks. Therefore, secure communication is a vital necessity for wireless sensor networks.

3. RESEARCH OBJECTIVES

The development of efficient and effective security methods that can guarantee communication and defend against a variety of security threats is the main goal of research on secured communication in wireless sensor networks. The following are a few of the specific research goals in this field:

- Creating reliable and effective authentication systems that can stop network unauthorized access.
- Developing encryption methods that can protect the integrity and secrecy of data sent over a network.
- Investigating how wireless sensor node performance and energy use are affected by security measures.

Analyzing the compromises between security and other performance indicators like delay time, space complexity, accuracy and authentication time.

4. LITERATURE SURVEY

1. J. Gowthami and N. Shanthi (2018) et.al proposed Multi-factor Based User Authentication Scheme for Lightweight IOT Devices. Authentication of legitimate clients is a prominent part of the security of the Internet of Things (IoT) applications like smart homes, smart cities, wearable devices, etc. The services presented by IoT devices should be feasible from wherever and whenever, simply by legitimate clients. The current authentication schemes for IoT applications are not secure, helpless against many attacks, and any unlawful client might get to the data of intelligent devices. This research presents an authentication scheme that utilizes just basic algebraic tasks like hash and XOR, making the proposed authentication scheme more appropriate for the lightweight climate of IoT. The protocol is displayed utilizing the Security Protocol Description Language, and the verification is finished utilizing one of the conventional verification tools–Scyther. The Scyther tool's outcome affirms the protocol's security and clarifies that it is vigorous against different known attacks and is more reasonable for functional applications. The keys can be separated, notwithstanding the changes in the Biometrics, with the assistance of the Fuzzy extractor method. The execution of the protocol in the Scyther tool demonstrates that it is secure against different realized attacks like Impersonation attacks, Eavesdropping attacks, Man-in-the-middle attacks, Password change attacks, and so forth. Also, the protocol concurs on mutual authentication and key agreement scheme.

2. I. A. Althamary and E. M. El-Alfy (2017) A more secure scheme for CAPTCHA-based authentication in the cloud environment. Cloud computing is a striking model allowing on-demand network access to various configurable, versatile assets and components, including storage, software, infrastructure, and platform. Be that as it may, there are significant worries about security-related issues. An actual security work is client authentication utilizing passwords. Although many defects have been found in password-based authentication, it is the most helpful methodology individuals use. A few schemes have been proposed to strengthen its viability, like salted hashes, one-time password (OTP), single-sign-on (SSO), and multi-factor authentication (MFA). This review proposes another authentication component by consolidating the client's password and altered characters of CAPTCHA to create a passkey. The change of the CAPTCHA relies upon a mystery settled between the cloud supplier and the client to utilize various characters for certain characters in the CAPTCHA. Moreover, This illustrated another scheme that utilizes CAPTCHA with replaceable characters to be added to the password. Replaceable characters rely upon an agreement between the cloud workers and customers. Besides, this scheme allows the client to utilize another password for each meeting.

3. P. C. Mondal, R. Deb and M. N. Huda, (2016) Know your customer (KYC) based authentication method for financial services through the internet. Financial services through the internet are running under different dangers like phishing, pharming (digital attack planned to divert a website's traffic to another phony webpage), malware, and developing complexity of give and take procedures. Multi-factor authentication (MFA) financial service framework mitigates the danger and makes it secure. Different methods of MFA run in inconveniences like the authentication device is lost or taken, misguided feeling that all is well and good (whenever utilized on login device), a compromised reply of the nonexclusive inquiry, higher execution costs, detailed development profile, determined hash esteem taken (there is no possibility to supplant it), and so on Consistence with Anti-Money Laundering (AML), Know Your Customer (KYC), and assent prerequisites keep on being a critical center region for Financial Institutions (FIs) management; firms should guarantee that they are following suitable consistency methods to fulfill the expanding administrative needs. Analysis and reproduction results show that the proposed method gives control equivalent to existing MFA/2FA. The proposed method is costless and does not bring any obstacle to conveying extra equipment. This method can be utilized on any private or public device as key burglary is impossible.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. R. Amin and G. P. Biswas (2015) Anonymity preserving secure hash function based authentication scheme for consumer USB mass storage device. A USB (Universal Serial Bus) mass storage device, which makes a (USB) device accessible to a host computing device and enables file transfers after finishing mutual authentication between the authentication worker and the user. It is a highly famous device due to its portability, huge storage limit, and high transmission speed. A few security protocols have been proposed to ensure the privacy of a file moved to a storage device; however, none of them is liberated from security shortcomings. As of late, He et al. proposed a productive multi-factor-based security protocol; however, the protocol is not relevant for commonsense execution, as they do not give a password change method, which is a fundamental stage in any password based user authentication and key agreement protocol. As the calculation and execution of the cryptographic one-way hash work are more complicated and accessible than other existing cryptographic algorithms. Lightweight and anonymity preserving user authentication and key agreement protocol for giving security of the file put away in the USB mass storage device utilizing cryptographic one-way hash work. After rigorous security analysis and conversation utilizing BAN rationale and casual security analysis affirms that the proposed protocol withstands applicable security parts, including user anonymity, user-worker impersonation attack, meeting critical discloser attack, and encrypted vital discloser attack.

5. Ajeena A, Muneera Hashim (2015) Two Factor users Authentication in Wireless Sensor Networks. Wireless sensor networks (WSN) are ordinarily sent in an unattended climate, where legitimate users can log in to the organization and access data as and when required. Thus, user authentication is essential in this asset compelled climate before accessing data from the sensor/gateway nodes. User authentication is fundamental for modified services and restricted admittance control in wireless sensor network two-factor user authentication protocol for WSN utilizing just hash work. The proposed protocol dodges many logged in users with the equivalent login and takes verifier attacks, which are conspicuous dangers for a password-based framework if it keeps up with the verifier table at the GW-node or sensor node. This has shown the productivity of the proposed protocol in correlation with the connected ones. Be that as it may, a reproduced/exploratory outcome would have been a superior picture to show the feasibility of the proposed protocol. The work can also be reached with a trial result alongside the countermeasure against the node compromise security dangers.

5. PROPOSED METHODOLOGY

Cuckoo Hash based Multi Factor Authentication

To foster a Cuckoo Hash based Multi Factor Authentication (CH-MFA) Mechanism for secured communication in wireless sensor organization. CH-MFA instrument includes two phases in particular, registration phase and authentication phase. At first, in the registration phase, the node must register the node's ID and password to the base station. Then, at that point, the registered information is put away in the base station utilizing Cuckoo Hash work. Only authenticated nodes are permitted to access the information by utilizing CH-MFA instrument. To check whether the node is authenticated node or not, the base station requests that the node send the node's id and password. After sending the essential information, Multi Factor Authentication is completed where the base station confirms that the user id and password are coordinated from the registered information. At the point when the node information gets coordinated, the node is supposed to be authenticated node. The base station permits the node to access the information. Exploratory assessment is completed on factors like authentication accuracy, authentication time, and space intricacy for data size.

Cuckoo Hashing is a technique for settling collisions in hash tables that creates a dictionary with constant-time most pessimistic scenario lookup and erasure tasks just as amortized constant-time addition activities. First presented by Pagh in 2001 as an expansion of a past static dictionary data structure, Cuckoo Hashing was the primary such hash table with basically small constant factors.

Journal of Theoretical and Applied Information Technology
30th June 2023. Vol.101. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Cuckoo hashing is a hash Table scheme utilizing two hash tables T_1 and T_2 each with r cans with free hash work h_1 and h_2 each planning universe U to can locations $\{0, \dots, r-1\}$. A key x can be stored in exactly one of the location $T_1[h_1(x)] and\ T_2[h_2(x)]$. A lookup operation, cuckoo Hashing inspects the two locations $T_1[h_1(x)] and\ T_2[h_2(x)]$ Moreover, it succeeds if the key x is stored in either location. Formally, then, the following algorithm describes the lookup behavior of cuckoo hashing.

Function lookup(x)

Return $T_1[h_1(x)] = x \vee T_2[h_2(x)] = x$

End

Indeed, this calculation acts in composed case constant time. Erasure is also a primary activity since it may eliminate a key from containing a pail. The primary inquiry for cuckoo hashing and the subject of the central part of analysis then, at that point, given hash functions $h_1 and\ h_2$ and a bunch of keys to put away in hash tables; how probably is it that these keys be set into a pail so that for each key x, either $T_1[h_1(x)] = x\ or\ T_2[h_2(x)] = x$; further, would you be able to show a scheme to embed keys to arrive at such a design productively? Pagh already shows that for arbitrarily chosen $h_1\ and\ h_2$. If $r \geq (1+\epsilon)n, i.e.$ Each of the two hash tables has a size at least $1+\epsilon$ times the total no. of keys for some ϵ; the probability that the keys cannot be put in such a configuration is $O\left(\frac{1}{n}\right)$ All the more usefully, it exhibits a basic inclusion produce that considers amortized constant-time addition activity.

Cuckoo hashing is a simple hash table where

- Lookups are worst-case O (1).
- Deletions are worst-case O (1).
- Insertions are amortized, expected O (1).
- Insertions are amortized O (1) with reasonably high probability.

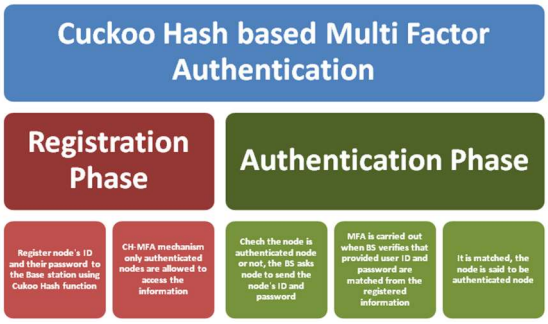

Figure 2: CH-MFA Overall Processes

Figure 2 shows an example of a Cuckoo channel that utilizes two hashes for everything and contains 8 buckets, each with 4 entries. A Cuckoo Filter has fundamentally two capacities: An Insert work that stores things in the channel and a Lookup work that checks whether a thing exists. These portray the tasks of these two functions in Algorithms

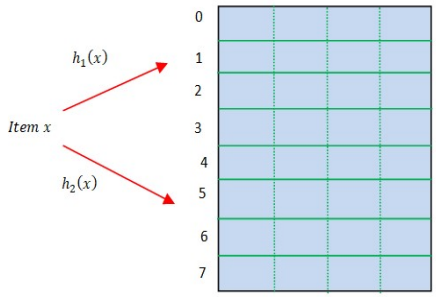

Figure 3: Cuckoo Filter: 2 hashes per item, 8 buckets each containing 4 entries

For the Insert operation, clarified in Algorithm 1, Cuckoo Filters store a fingerprint f of everything x rather than putting away the thing itself. For this, everything is first hashed into a constant-sized fingerprint (stage 1).

Algorithm 2 Lookup(x)
1: $fingerprint(x)$;
2: $i_1 = hash(x)$;
3: $i_2 = i_1 \oplus hash(f)$:
4: **if** $bucket[i_1] or\ bucket[i_2] has\ f$ **then**
Return True
Return False;

A fingerprint is then put away in the filler as follows. The algorithm checks in case there is a vacant entry in one of the two bucket

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

record i_1 *and* i_2 (steps 2 to 5) If an unfilled entry is discovered, f is added to the bucket. Else one of two buckets is picked haphazardly (step 6), and f is traded with one of the things in the bucket while the casualty thing (being traded with f) is moved to its substitute location, as displayed in algorithm 1. The space cost, in bits, of putting away one thing in the cuckoo channel utilizing the Insert function relies upon the target false positive rate ϵ and is given by$(\log_2\left(\frac{1}{\epsilon}\right)+2)/\alpha$ where α is the heap factor of the channel, which characterizes the most extreme channel limit. When the greatest feasible, α, is reached, insertions are (non-inconsequentially and progressively) prone to fall flat, and henceforth, the channel should extend to store more things.

The lookup operations are highlighted in algorithm 2. In order to check whether item x belongs to the filter, this only needs to compute its potential location $i_1 and\ i_2$ and then check whether $bucket[i_1] or\ bucket[i_2]$ contains the fingerprint of x.

As clarified, Cuckoo Filter is utilized in this phase to build a portrayal of the range geo-location database. What persuaded the utilization of the Cuckoo Filter is that it offers the most elevated space proficiency among every single existing methodology and is substantially more productive than Bloom Filters, particularly for exceptionally enormous sets, which is the situation of geolocation databases that contain sections compared to range accessibility with a location goal that can go up to 50 meters. Cuckoo Filter appreciates speedy Look up and Insert operations, hence diminishing the calculation overhead of the proposed scheme substantially, as will be seen later. Cuckoo Filter is the structure square of this scheme.

Multi Factor Authentication

This phase proposes an improved scheme that keeps the benefits of the first protocol and can withstand the security shortcomings portrayed in past areas. Likewise, another downside of the first scheme is that a user cannot safely and unreservedly change their password. To cure this design disadvantage, the password refreshing phase has been added. There are three phases in this upgraded scheme: the registration phase, the authentication phase, and the password refreshing phase.

Algorithm 1 Insert(x)
1: $f = fingerprint(x)$;
2: $i_1 = hash(x)$:
3: $i_2 = i_1 \oplus hash(f)$:
4: **if** $bucket\,[i_1] or\ bucket\,[i_2]$ has an empty entry, **Then**
5: add$f$ to that bucket:
Return Done
// must relocate existing item if no empty entries;
6: i= randomly pick $i_1\ or\ i_2$:
7: **for** $n = 0; n < MaxNumkicks: n++$ **do**
8: randomly select an entry e from $bucket[i]$
9: Swap f and the fingerprint stored in entry e:
10: $i = i \oplus hash(f)$;
11. **If** $Bucket[i]$ has an empty entry **then**
12: add f to $bucket[i]$
Return done
// hash able is consider full;
Return Failure;

3.1 Registration phase

This phase is summoned at whatever point the user, sayU_i, needs to register with the WSN. It chooses a subjective number b and figures $h(b \oplus PW_i)$.. Here the length of b is enormous, for example, 512 pieces. That is, b is a high-entropy varying number. Here h (.) utilized through the proposed protocol is sans impact one-way hash function, for example, SHA-1. Hence, the piece length of the yield of the hash function is 160. Then, at that point, it presents its identity ID_i and $h(b \oplus PW_i)$ to the GW-node for registration in a secure channel. Upon receiving the registration request, the GW-node computes $T_i = h(ID||J), V_i = T_i \oplus h(ID_i||h(b \oplus PW_i), H_i = h(T_i), N_i = h(ID_i) \oplus h(k)$. Here, K and J are two secret parameters held only by GW-Node, and ||is a bitwise concatenation operator. Then the GW-Node personalizes the intelligent card with parameters $V_i, H_i, h(.), N_i\ and\ x_a, where\ h(.)$ is a cryptographically secure hash function. Here, x_a are personal boundaries generated safely by the GW-node and put away in some designated sensor node before conveying the nodes in the field, which can exchange data with users. Note that the length of *b, k, j* and x_a are adequately enormous, e.g., 512 pieces. That is, they are high-entropy odd numbers. The GW-node presently sends the customized smart card to the U_i in a secure channel. Subsequently U_i enters b into its smart card. U_i 's smart card contains

$V_i, H_i, h(.), N_i, b\ and\ x_a$. Note that x_a is not known to the user, as it is generated and stored in the user's smart card securely by The GW-node.

3.2 Authentication Phase

This phase is invoked when U_i needs to play out specific questions to or access data from the organization. The phase is additionally separated into login and verification phases.

Login Phase

U_i Inserts its smart card into a terminal, and keys ID_i and PW_i. Then the intelligent card computes $T_i = V_i \oplus h(ID_i||h(b \oplus PW_i), and\ H_i^* = h(T_i)$. Subsequently, the smartcard checkswhether H_i^* and H_i is equivalent. If the user's authenticity can be guaranteed and continues to the subsequent stage, In any case, dismisses the login request.

Step-L1) Generate nonce M_i and compute $DID_i = ID_i \oplus h(x_a||T||M_i)$. Here T is the current timestamp of the $U_i's$ system.

Step-L2) Compute $C_i = h9N_i||x_a||T)$. Then send$<DID_i, C_i, T, M_i>$ to the GW-node

Verification Phase

Upon receiving the login request$<DID_i, C_i, T. M_i > at\ time\ T^*$, the GW-node authenticates the $U_i$by the following steps:

Step-V1) Validate T. If $T^* - T \leq \triangle T$, then the GW-node proceeds to the next step, else abort, where $\triangle T$ means the typical time stretch for the transmission delay.

Step-V2) Obtain the user's real identity ID_I by computing $ID_i = DID \oplus h(x_a||T||M_i)$. Verify the format of ID_i on the off chance that the organization is not substantial, the GW-node ends the association. Something else, compute $C_i^* = h\left((h(ID_i) \oplus h(K))||x_a||T\right)$.

Step-V3) If $C_i^* = C_i, the\ GW - node$ t;

Step-V4) GW-node now sends a message $< DID_i, A_i, T' > to$ some nearest sensor node, say S_n over an open channel to react to questions/data what U_i is looking for, where $A_i = h(DID_i||S_n||x_a||T')$, and T' is the current timestamp of GW-Node. Here, A_i is used to ensure the sensor node that the message $< DID_i, A_i, T' >$, has come from a legitimate GW-node, because A_i is generated with a secret parameter x_a Which is known to both sensors and GW-Sensor.

Step-V5) S_n first validate T' as Step-V1. Then S_n computes $h(DID_i||S_n||x_a||T')$ and checks whether it is equalA_i.If these two checks pass correctly, then S_n responds to $U_i's$query.

3.3 Password Updating Phase

This phase is invoked whenever U_i requests to change its password, and is described**3.3 Password Updating Phase**

This phase is invoked whenever U_i requests to change its password, and is described in the following:

1. Insert its smart card into a terminal, and keys ID_i and PW_i. Then the intelligent card computes $T_i = V_i \oplus h(ID_i||h(b \oplus PW_i))$ and $H_i^* = h(T_i)$. Subsequently, the intelligent card checks whether $H_i^*\ and\ H_i$ are equal to or not. If yes, the legitimate user can be assured and proceed to the next step. Otherwise, the intelligent card rejects the password update request by displaying a "Password update failure" to U_i.
2. Enter its new password $PW_{new}. U_i$ selects a random number b_{new} and computes $(h(b_{new} \oplus PW_{new})$. then U_i calculates $V_i^* = V_i \oplus h(ID_i||h(b \oplus PW_i)) \oplus h(ID_i||h(b_{new} \oplus PW_{new}))$ and replaces V_i^* with V_i . Finally, $U_i'ssmart\ card\ contains\ \{V_i^*, H_i, h(.), n_i, b_{new}\}$.

Security and Performance Analyses

The proposed scheme keeps the benefits of the first scheme. For instance, the proposed protocol is free from the password/verifier table. In this manner, it can oppose the taken verifier attack. Also, it can oppose speculating attacks since the user password is not communicated essentially as the hash of the password. Moreover, the timestamp is likewise used to keep from the replay attack. The proposed scheme can beat the security shortcomings that the existing falls

for. The benefits of the proposed scheme are clarified as follows.

Proposition: This approach can withstand insider and impersonation attacks.

Proof: By this proposed scheme $U_i$registers to the GW-node by presenting $h(b \oplus PW_i)$ instead of PW_i, the insider of the GW-node cannot help directly obtain PW_i. Moreover, as b is high entropy random number and not revealed to the GW-node, the insider of GW-node cannot get PW_i by performing an offline guessing attack on $h(b \oplus PW_i)$. Additionally, this proposition does not keep up with any password/verifier table by the same token. In this manner, this scheme can oppose insider attacks. The impersonation attack can't be dispatched because the insider cannot get PWi. For insider attacks, a few researchers have created different protection draws near. For instance, the creators propose that in the registration phase, a user presents the hashed worth of its password to the worker rather than its password.

Nonetheless, with the hashed worth of a user's password, a unique insider of the worker can accurately acquire the user's password by playing out a disconnected password speculating attack. Moreover, the creators recommend that, albeit a user presents his/her password to the far off framework during the registration interaction, he/she can change his/her password by conjuring the password change phase after registration. When a favored insider of the distant framework has acquired a user's password, he/she can change the user's password by summoning the password change phase before the legitimate user does that. In the present circumstance, their methodology is not adequate. As the above investigations indicate, handling insider attacks is safer and more proficient.

Proposition: This scheme can obtain a user's real identity

Proof: As described above, the GW-node obtains the user's real identity ID_i by computing $ID_I = DID_i \oplus h(x_a||T||M_i)$. Note that extracting boundaries from the smart card and the nodes is expected to be very troublesome. As depicted in, despite the fact that it occurs by the side channel attacks. Some intelligent card manufacturers take into the attack of hazards of these attacks and give countermeasures to concede the picking apart endeavors. Thus, the adversary, including users, cannot get $\{x_a, h(.)\}$ sorted in smartcards and nodes.

4. EXPERIMENTAL RESULT

This phase uses the computation cost (the computation time of different cryptographic operations, denoted by T) and

T_h: the time of performing a one – way hash function $h(.)$.

T_{pu}: the time of performing a public key computation.

T_{pr}: the time of performing a private key computation.

C_{ug}: the delay time of the communication between a user and the GW – node.

C_{gs}: the delay time of the communication between a GW – node and the sensor node.

C_{su}: the delay time of the communication between a user and the sensor node.

communication cost (denoted by C) as the metrics to evaluate the performance of the proposed protocol. Some notations are further defined as follows:

Note: XOR operation requires very few computations. Thus, its computation cost is neglected here.

4.1 Accuracy

Table 1: Comparison table of Accuracy

Threshold distance	SSO-MFA	CAPCHA BASED MFA	Proposed CH-MFA
250	45	53	89
500	65	59	92
750	68	65	95
1000	71	77	97
1500	76	81	99

Comparison table 1 of Accuracy Values explained the different values of existing algorithms (SSO-MFA, CAPCHA BASED MFA) and proposed CH-MFA. While

Journal of Theoretical and Applied Information Technology
30th June 2023. Vol.101. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

comparing the Existing algorithm (SSO-MFA, CAPCHA BASED MFA) and the proposed CH-MFA provides better results. The existing algorithm values start from 45 to 76, 53 to 81, and the proposed CH-MFA values start from 89 to 99. The proposed method provides excellent results.

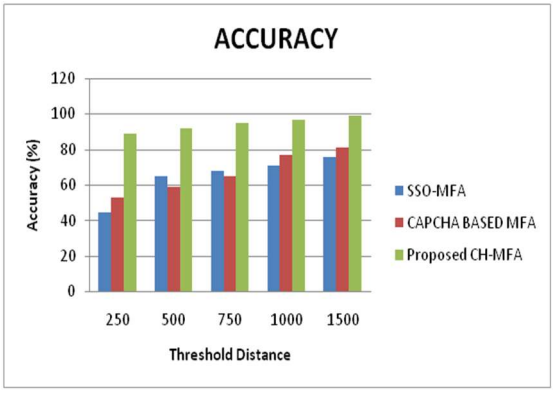

Figure 4: Comparison chart of Accuracy

Figure 4 shows the comparison chart of Accuracy demonstrating the existing 1, existing 2 (SSO-MFA, CAPCHA BASED MFA), and proposed CH-MFA. X axis denotes the Threshold distance, and y axis denotes the Accuracy in percentage. The proposed CH-MFA values are better than the existing algorithm. The existing algorithm values start from 45 to 76, 53 to 81, and the proposed CH-MFA values start from 89 to 99. The proposed method provides excellent results.

4.2 Authentication Time

Table 2: Comparison table of Authentication Time

Key generation interval (sec)	SSO-MFA	CAPCHA BASED MFA	Proposed CH-MFA
0.2	2.5	2.8	1.1
0.5	2.2	2.5	0.9
0.8	1.9	1.6	0.6
1.0	1.5	1.4	0.5
1.4	1.3	1.7	0.3

Comparison table 2 of Authentication Time Values explained the different values of existing algorithms (SSO-MFA, CAPCHA BASED MFA) and proposed CH-MFA. While comparing the Existing algorithm (SSO-MFA, CAPCHA BASED MFA) and the proposed CH-MFA provides better results. The existing algorithm values start from 1.3 to 2.5, 1.7 to 2.8, and the proposed CH-MFA values start from 0.3 to 1.1. The proposed method provides excellent results.

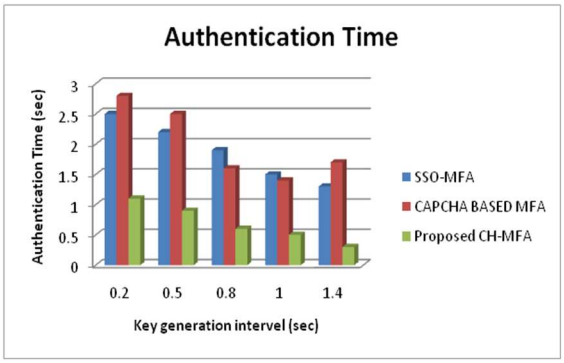

Figure 5: Comparison chart of Authentication Time

Figure 5 shows the comparison chart of Authentication Time demonstrates the existing 1, existing 2 (SSO-MFA, CAPCHA BASED MFA), and proposed CH-MFA. X axis denotes the key generation interval, and y axis denotes the Authentication time in percentage. The proposed CH-MFA values are better than the existing algorithm. The existing algorithm values start from 1.3 to 2.5, 1.7 to 2.8, and proposed CH-MFA values start from 0.3 to 1.1. The proposed method provides excellent results.

4.3 Space complexity

Table 3 Comparison table ofSpace complexity

No. of items	SSO-MFA	CAPCHA BASED MFA	Proposed CH-MFA
10	250	200	85
30	280	240	110

50	320	260	125
70	350	310	140
90	370	325	160

Comparison table 3 of Space complexity Values explains the different values of existing algorithms (SSO-MFA, CAPCHA BASED MFA) and proposed CH-MFA. While comparing the Existing algorithm (SSO-MFA, CAPCHA BASED MFA) and the proposed CH-MFA provides better results. The existing algorithm values start from 250 to 370, 200 to 325, and the proposed CH-MFA values start from 85 to 160. The proposed CH-MFA gives excellent results.

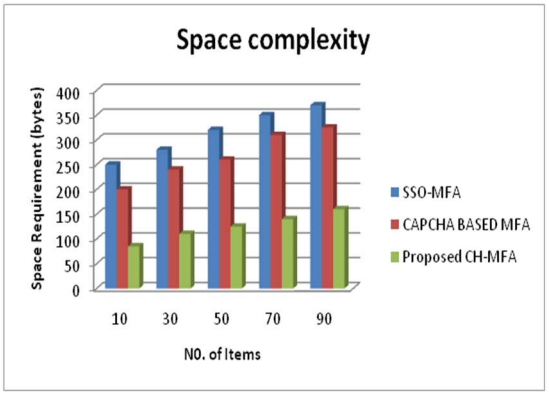

Figure 6 Comparison chart of Space complexity

Figure 6 shows the comparison chart of Space complexity demonstrates the existing 1, existing 2 (SSO-MFA, CAPCHA BASED MFA), and proposed CH-MFA. X axis denotes the No. of items, and y axis denotes the space requirement in bytes. The proposed CH-MFA values are better than the existing algorithm. The existing algorithm values start from 1.3 to 2.5, 1.7 to 2.8, and the proposed CH-MFA values start from 0.3 to 1.1. The proposed method provides excellent results.

4.4 Delay Time

Table 4 Comparison table of Delay Time

No. of BEACOMS	SSO-MFA	CAPCHA BASED MFA	Proposed CH-MFA
20	0.7	0.9	0.2
40	1.5	1.8	0.8
60	2.7	2.3	1.1
80	3.2	2.9	1.4
100	3.5	3.7	1.6

Comparison table 4 of Delay Time Values explained the different values of existing algorithms (SSO-MFA, CAPCHA BASED MFA) and proposed CH-MFA. While comparing the Existing algorithm (SSO-MFA, CAPCHA BASED MFA) and the proposed CH-MFA provides better results. The existing algorithm values start from 0.5 to 3.5, 0.9 to 3.7, and proposed CH-MFA values start from 0.2 to 1.6. The proposed CH-MFA gives excellent results.

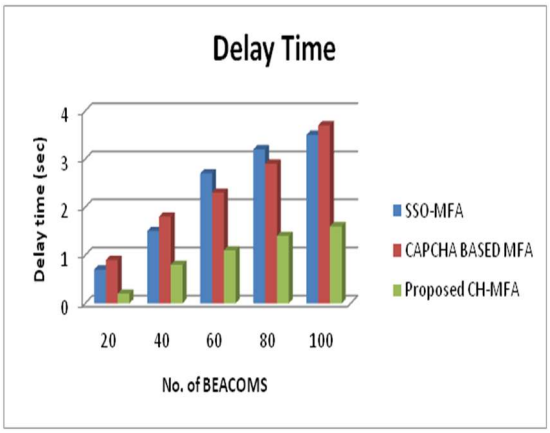

Figure 7 Comparison chart of Delay Time

Figure 7 shows the comparison chart of Delay Time demonstrates the existing 1, existing 2 (SSO-MFA, CAPCHA BASED MFA), and proposed CH-MFA. X axis denotes the No. of BECOMES, and y axis denotes the Delay time in seconds. The proposed CH-MFA values are better than the existing algorithm. The existing algorithm values start from 0.5 to

3.5, 0.9 to 3.7, and proposed CH-MFA values start from 0.2 to 1.6. The proposed CH-MFA gives excellent results.

5. CONCLUSION

When it comes to ensuring secure communication in wireless sensor networks, the Cuckoo Hash based Multi Factor Authentication (CH-MFA) Mechanism is a viable option. The technique offers a strong and effective method for authenticating wireless sensor nodes and prevents unauthorized access by combining the benefits of Cuckoo Hashing and Multi-Factor Authentication. The CH-MFA mechanism uses a two-factor authentication strategy that verifies a wireless sensor node's identification using both a password and a tangible token. The process is resistant to hash collision attacks because to the use of Cuckoo Hashing, and the multi-factor authentication strategy assures that even if an attacker succeeds in stealing a password, they will be unable to access the system without the physical token.

The CH-MFA method provides a high level of security overall and is suitable for usage in wireless sensor organizations where secure communication is essential. The CH-MFA mechanism provides a strong and efficient approach for tackling the security difficulties presented by wireless sensor organizations, even though no security system is totally impenetrable.

6. LIMITATIONS AND ASSUMPTIONS

Some limitations and assumptions are likely to have been made during the development and evaluation of the Cuckoo Hash based Multi Factor Authentication (CH-MFA) Mechanism for secured communication in wireless sensor networks. Some of these restrictions and presumptions could be:

Hardware restrictions: The testing hardware may have placed restrictions on the evaluation of the CH-MFA mechanism. The results might not be transferable to bigger or more potent nodes, for instance, if the testing was carried out on a small number of wireless sensor nodes or on nodes with low computational or memory capacities.

Network topology: A particular network topology or communication pattern that may not be typical of all wireless sensor networks was perhaps used in the evaluation of the CH-MFA mechanism.

User behavior assumptions: It's probable that the CH-MFA mechanism evaluation included user behavior or use scenario assumptions that weren't necessarily accurate representations of all potential outcomes. Further research is needed to evaluate the performance and scalability of the CH-MFA mechanism in larger and more complex wireless sensor networks.

REFERENCES

[1]. Benenson Z., Gartner F., and Kesdogan D. (2004). User authentication in sensor networks. In Proc. Workshop Sensor Networks, Lecture Notes Informatics Proceedings Informatics.

[2]. Chen, C.T.; Lee, C.C. A two-factor authentication scheme with anonymity for multi-server environments. Secure. Commun. Netw. 2013, 8, 1608–1625.

[3]. Choi, Y.; Lee, D.; Kim, J.; Nam, J.; Won, D. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors 2014, 14, 10081–10106.

[4]. Das M. (2009). Two-factor user authentication in wireless sensor networks. IEEE Transactions on Wireless Communications, vol. 8, no. 3, pp. 1086–1090.

[5]. Gnawali O., Jang K., Paek J., Vieira M., Govindan R., Greenstein B., Joki A., Estrin D., and Kohler E. (2006). The tenet architecture for tiered sensor networks. in ACM SenSys'06, pp. 153–166.

[6]. He, D.; Kumar, N.; Chilamkurti, N. A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. Inf. Sci. Int. J. 2015, 321, 263–277

[7]. He, D.; Zeadally, S.; Kummar, N.; Wu, W. Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. IEEE Trans. Inf. Forensics Secur. 2016, 11, 2052–2064.

[8]. Huang, X.; Xiang, Y.; Bertino, E.; Zhou, J.; Xu, L. Robust multi-factor authentication for fragile communications. IEEE Trans. Depend. Secure. Comput. 2013, 11, 568–581.
[9]. Jiang, Q.; Chen, Z.; Li, B.; Shen, J.; Yang, L.; Ma, J. Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems. J. Ambient Intell. Humaniz. Comput. 2017.
[10]. Jung, J.; Moon, J.; Lee, D.; Won, D. Efficient and security enhanced anonymous authentication with key agreement scheme in wireless sensor networks. Sensors 2017, 17.
[11]. Karlof C., Sastry N., and Wagner D. (2004). TinySec: link layer security architecture for wireless sensor networks. In Proc. International Conf. Embedded Networked Sensor Syst., pp. 162–175.
[12]. Kumari, S.; Khan, M.K.; Li, X.; Wu, F. Design of a user anonymous password authentication scheme without smart card. Int. J. Commun. Syst. 2016, 29, 441–458.
[13]. Lee, C.C.; Chen, C.T.; Wu, P.H.; Chen, T.Y. Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices. IET Comput. Digit. Tech. 2013, 7, 48–56.
[14]. Li, C.T.; Lee, C.C.; Lee, C.W. An improved tTwo-factor user authentication protocol for wireless sensor networks using elliptic curve cryptography. Sens. Lett. 2013, 11, 958–965.
[15]. Li, X.; Xiong, Y.; Ma, J.; Wang, W. An enhanced and security dynamic identity based authentication protocol for multi-server architecture using smart cards. J. Netw. Comput. Appl. 2012, 35, 763–769.
[16]. Ling, C.; Lee, C.; Yang, C.; Hwang, M. A secure and efficient one-time password authentication scheme for WSN. Int. J. Netw. Secure. 2017, 19, 177–181.
[17]. Ma1, C.; Wang, D.; Zhao, S. Security flaws in two improved remote user authentication schemes using smart cards. Int. J. Commun. Syst. 2012, 27, 2215–2227.
[18]. Park, Y.; Park, Y. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. Sensors 2016, 16, 2123.
[19]. Pecori, R.; Veltri, L. 3AKEP: Triple-authenticated key exchange protocol for peer-to-peer VoIP applications. Comput. Commun. 2016, 85, 28–40.
[20]. Sastry N., and Wagner D. (2004). Security considerations for IEEE 802.15.4 networks. in Proc. ACM Workshop Wireless Security, pp. 32–42.