ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

Multi-Factor Group Authentication in the Virtual World using Block Chain Technology

[1]KAPA NAGA VENKATA SAI NIKHIL, [2]MUDUNURI THANISHKA SAI MANOJ VARMA, [3]KUNA PAVAN GANESH, [4]Dr. G. SATYANARAYANA

[123]Student, Department of CSE, DNR College of Engineering & Technology, Balusumudi, Bhimavaram, India.

[4]Professor, Department of CSE, DNR College of Engineering & Technology, Balusumudi, Bhimavaram, India.

Abstract—

In this hypothetical metaverse, users transfer sensitive information to the platform server via public wireless channels, making the data susceptible to security breaches caused by hostile actors. Furthermore, it is simple to generate network congestion and overburden the primary server when a significant number of users complete group authentication concurrently. This article introduces a metaverse-appropriate multi-factor group authentication technique that is based on blockchain technology. Specifically, our system employs multi-factor authentication data, which includes biological information, to contribute to the creation of an anchor key, alleviating the shortcomings of single-factor authentication. Our plan calls for forming a key-value pair out of the user's MFA data, which is subsequently saved on the blockchain and uniquely tied to their smart device. Additionally, the blockchain's immutability helps in identifying and tracking down hostile actors. To top it all off, we improve the signaling process to prevent network congestion while designing the group authentication method. We conclude that our approach may provide additional security features, such as the capacity to track malevolent adversaries, according to the security analysis. The performance study, meanwhile, shows that our technique can make authentication more efficient.

Index Terms—Metaverse, group authentication and key agree ment, multi-factor, blockchain.

INTRODUCTION

Metaverse generates visuals to facilitate a variety of social activities in the virtual world, therefore overcoming the constraints of conventional online communication and the real world [1]. Wearing smart equipment, for instance, allows patients and medical professionals from across the globe to create photos simultaneously, access a virtual hospital, and really have face-to-face consultations with physicians. Metaverse has the potential to provide many services, however there are still several security issues that need to be addressed [2]. An attacker may loiter on the wireless channel that users use to communicate with the metaverse platform server, potentially listening in on their conversations or even altering the data that is transferred [3], [4]. Attacks like this violate users' right to privacy and damage their property and rights [5, 6]. Performing Authentication and Key Agreement (AKA) prior to user access is a typical approach to address such issues [7]-[9]. Few studies have examined authentication and key agreement in the metaverse, whereas most have concentrated on future possibilities and applications (such education and healthcare) [10]-[12]. No one addressed group-based authentication protocols in their works; instead, Ryu et al. [13] and Yang et al. [14] offered mutual authentication schemes and an authentication framework based on chameleon signatures, respectively. “Han et al.” [15] said. It has been noted that the degree to which users are acquainted with a virtual reality scenario is closely correlated with their level of happiness with the service. A common occurrence in metaverse scenarios is the presence of groups of people who have same interests and knowledge. Group authentication makes them more likely to use the service simultaneously. Nevertheless, current methods need that every group member undergo a full AKA procedure when authenticating as a group, leading to increased data use and network latency. Additionally, the primary server is more likely to experience overload as a result of the frequent collection of authentication data. The main points of this article are these: • To begin, we create a metaverse-specific multi-factor group authentication mechanism that is based on the blockchain. To be more precise, in order to reduce network latency and main server burden, the first smart device in the

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

group completes the authentication procedure while the other smart devices just need to execute a simplified authentication. Additionally, in order to address the linkability threat, the smart device's identity is encrypted using ECIES while using the temporary shared key that was established via the ECIES-KEM process [16]. • Secondly, we use multi-factor authentication information, which includes biological information, to generate an anchor key [18], which helps us overcome the problems with single-factor authentication [17]. Using the immutability of the blockchain, it may securely tie smart device credentials and information about multiple-factor authentication into transparent key-value pairs, and then use these to track down a malevolent attacker. In this way, the paper is organized. The second section introduces the system model and the threat model. In Section III, we lay out our plan in great detail. An examination of safety and efficiency is given in Section IV. Section V concludes the paper and provides directions for further research.

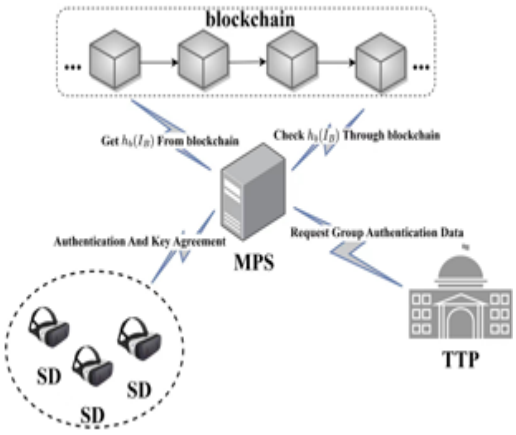

Fig. 1: System model

SYSTEM MODEL AND THREAT MODEL

Systems and threat models are introduced in this section. S. Model of the System Fig. 1 shows the components of the system concept, which include blockchain, trustworthy third parties (TTPs), metaverse platform servers (MPSs), and smart devices (SDs). • SD: To join the metaverse, users touch the SD sensor, which generates authentication data. To guarantee the security of future sessions, the SD initiates authentication by sending a request for mutual authentication to the MPS. The two parties then negotiate an anchor key KMPS. • Multi-Protocol Signature Service (MPS): MPS is in charge of maintaining group authentication data that is collected via TTP queries and making sure that all SDs in the group authenticate with each other. It operates under the metaverse. Using the SD anonymous identity, MPS checks the blockchain for user multi-factor authentication information throughout the authentication process. Adds a record with the details of the malevolent attacker to the blockchain in order to identify them if malicious authentication is found. In any other case, it removes the user's MFA credentials from the process of building the KMPS anchor key. • TTP: TTP is a very trustworthy organization that has exceptional storage capacities. In the TTP database, you can find all the details about the groups, as well as the security values and pre-shared keys for each SD and TTP. When a user registers, TTP is in charge of storing their multi-factor authentication details and SD anonymous identity on the blockchain. During authentication, it receives authentication requests from MPS and distributes group authentication data. • The blockchain: MPS and TTP work together in the blockchain's con sensus mechanism, and this study makes use of a public permission blockchain. Unique key-value pairs consisting of user multi-factor authentication details and SD anonymous identity are stored on the blockchain. The blockchain will be updated whenever MPS identifies a malicious authentication attempt. We can use the immutability of the blockchain to track down the malevolent attacker, unlike traditional storage techniques. Section B. Danger Analysis One popular model that the threat model takes into account is the Dolev-Yao model. • The encryption assumption: An adversary without knowledge of the key cannot decode the message. Also, the random value and key would be completely out of their reach. Channel assumption: A hostile actor may take full control of the SD-MPS wireless radio channel. For example, if an attacker wants to compromise the protocol, they may start a flood of malicious sessions and eventually get all the public keys. While active attackers may alter, replay, or mimic communications, we let passive attackers eavesdrop. We take it as read that the channel between MPS and TTP is safe and reliable. • Component assumption: Our approach forbids attackers from harming components in SD, MPS, and TTP, which means that secrets like security value SV and long-term key Ki−j cannot be stolen from these components. • Function assumptions: We take it as read that the attacker has complete control over the scheme's inputs and can

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

exploit any of its functional functions. Nevertheless, the ECIES mechanism's KEM and DEM must be secure, and the output created after going through the functional functions must be guaranteed to be both intact and confidential. Section III: The Plan The following sections provide the specifics of the three stages that make up the proposed concept. In Table I, we detail all of the notations that are needed for the proposed scheme.

TABLE I: Notations and descriptions

Notation	Description
K_{i-j}	The secret key pre-shared by the jth SD and TTP in the ith group
SV_{i-j}	The security value pre-shared by the jth SD and TTP in the ith group
$SD_{ID}^{i-j}, SD_{AID}^{i-j}$	The identity and anonymous identity of the jth SD in the ith group
(PK_{TTP}, SK_{TTP})	TTP's public and private keys
GK_{G_i}	The group key pre-shared by all the jth SDs and TTP in the ith group
GTK_{G_i}	Group ephemeral key for ith group
GID_{G_i}	ID of the ith group
SQN_{SD}^{i-j}	Sequence number of the jth SD in the ith group
SQN_{TTP}^{i-j}	Sequence number stored in TTP about the jth SD in the ith group
I_B	The biological information of the user
$h_b()$	The biological hash function
$f^1, f^2, f^3, f^4, f^5, f^6$	The functional function
K_{MPS}	The anchor key agreed by SD and MPS
AK	The anonymous key computed by TTP
k	The temporary shared key generated by ECIES mechanism
R_{SD}^{i-j}	The random challenge of jth SD in the ith group
R_{MPS}^{i-j}	The random challenge of MPS generate for jth SD in the ith group
R_{TTP}, R_{TTP}^{*}	Random challenge and random challenge encrypted with k generated by TTP
MAC_{SD}^{i-j}	The message authentication code for jth SD in the ith group
$MAC_{MPS}^{i-j}, MAC_{TTP}^{i-j}$	The message authentication code of MPS and TTP generate for jth SD in the ith group
$RES_{i-j}, XRES_{i-j}$	The response and expected response to jth SD in ith group
$Encap(), Decap(), SEnc()$	Algorithm of ECIES Mechanism

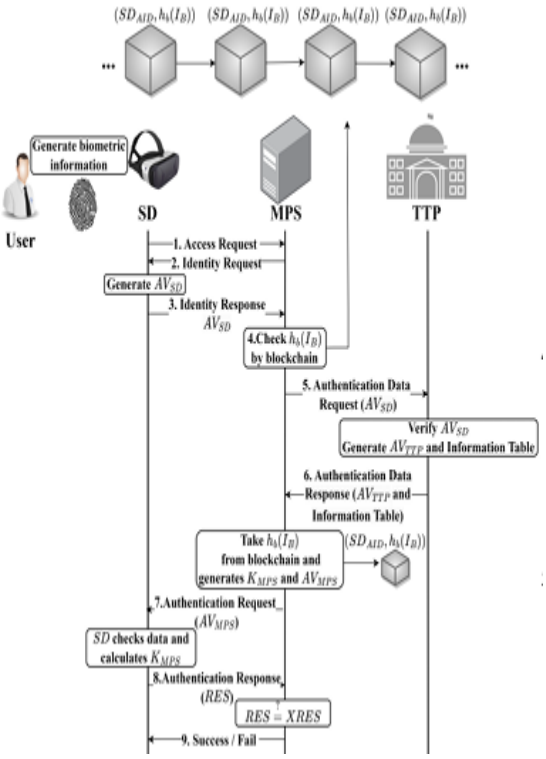

Fig.2:TheAKAprocessof thefirstSDinthegroup

Initialization To register for the first time, all users wear smart devices that collect user fingerprints using sensors and smart devices. Collect MFA data using biological hash functions, combine it with smart device data to create key-value pairs, then send them to TTP over trusted channels. Ensure its validity and create a block on the blockchain. A distinct identification, SDID, is stored in each SD and is shared with TTP. Those who are already acquainted with each other are encouraged to establish groups and engage in the metaverse in order to increase customer service satisfaction. For identity validation, TTP keeps information like SV and GK for each SD in the group. We expect GK to be updated whenever a new SD is added. if an old SD departs the organization. B. Finish the AKA as carried out by the first SD in the group SD1-1 becomes the first smart device in the group to use AKA in this phase. Figure 2 shows the procedure. The user initiates the generation of IB by touching the SD1−1 sensor, and simultaneously sends an access request to the MPS. 2) The MPS answers the SD1-1's identify request. 3) The following is the output of the AV1−1 SD while sending an identity response to MPS: 1. PKTTP Generation and Storage 2. Produce SD1−1 AID by use of SEnc(k,SD1−1 ID). Pick a random R1 minus 1 standard deviation. 4. Find the authentication code for the message sent on the first day: MAC1−1 SD=f1 K1−1 (R1−1 SD). G1 PublicInfo Information Table II This is the GIDG1 set of equations: SD1−x ID SD1−x, K1−x SV1−x, SQN1−x TTP, XRES1−x, MAC1−x TTP. GTKG1 TTP XRES1−2 MAC1−2 TTP AK, ID SD1−2, AID K1−2, SV1−2, SQN1−2, and TTP XRES1−2. This is a sentence that needs to be paraphrased. R∗ TTP XRES1−n MAC1−n TTP ID SD1−n AID K1−n SV1−n SQN1∂n TTP 5. Find the identity response of SD1−1 by comparing the following variables: AV1−1 SD = (SD1−1 AID||R1−1 SD||MAC1−1 SD||SQN1−1 SD||hb(IB)) 4) The authentication message including SD1−1 AID and hb(IB) is checked for a match according to the blockchain when it is received by MPS. When the two don't match, MPS will deny the authentication and add an error record to the blockchain to identify the bad actor. The MPS will send the authentication data request to TTP if the two match. 5) The transfer key decryption process begins when TT gets this request. Next, the appropriate authentication data is retrieved from the database based on the SD1−1 ID, GTKG1 is calculated, and an information table is generated for this group. The following are the particular steps: • 1. As temporarydata(C0,C), parse SD1−1. 2. Decap using Generate and Store Keys (SKTTP, C0) Decipher the togetSD1−1 ID using

SDec(k,C). 4. Locate GKG1, SQN1−1 TTP, GIDG1, and K1−1 in the database based on the SD1−1 ID. 6. Verify whether MAC1−1 SD equals Checkf1 K1−1 (R1−1), Unless both conditions are met, the verification is considered successful. 6. Select an RTTP at random and encrypt it using ENC(k,RTTP) to get R∗ TTP. 7. Find GTKG1 by summing f6 GKG1 with GIDG1 and RTTP. 8. Get data for all group SDs and create MAC1−x TTP = f3 K1−x (SQN1−x TTP||RTTP), XRES1−x =f4 K1−x (RTTP), and SD1−x AID for every team device. In this case, x is an integer in the range from 1 to n, and n is the total number of SDs in the group. 9. Enter AK=f5 k(RTTP) and see the result.

Produce the authentication data response for SD1−1 by using the formula AV1−1 TTP=(R1−1 SD||SQN1−1 SD) and an information table to hold the authentication data for all SDs, as illustrated in Table II. Last but not least, revise the data held in TTP 6). Through a secure connection, TTP sends the information table of G1 with AV1−1 TTP to MPS for storage. 7. The MPS gets hb(IB) from the blockchain so it may take part in creating KMPS, and it calculates and sends an authentication request message to SD1−1. Below is a thorough description of the procedure: 1. Pick an R1−1 MPS at random. 2. Find the verification code for the message sent on SD-1: MAC-1 MPS=f2 GTKG1 (SQN-1 SD||R-1 MPS||SV1-1)

The authentication request for SD1−1 should be generated as follows: AV1−1 MPS=(R∏ TTP||CON||MAC1−1 MPS||R1−1 SD||R1−1 MPS). where CON is equal to (AK⊩SQN1−1 TTP,MAC1−1 TTP). You may get the AK, MAC1−1 TTP, and SQN1−1 TTP from the G1 information table that is kept in MPS. 4. Take part in the creation of the anchor key and get the associated hb(IB) from the blockchain in accordance with SD1−1 AID. The equation KMPS = f6 GTKG1 (SQN1−1 SD||SV1−1||hb(IB)||R1−1 SD||R1−1 MPS) can be formalized as follows. 5. After getting the authentication request from MPS, SD1−1 checks the data and sends it to AV1−1 MPS. When it passes the test, SD1−1 sends a RES1−1 authentication response message to MPS and determines the anchor key KMPS. Here are the particular steps: 1. Use DEC(R* TTP,k) to decrypt RTTP. 2. Verify the message using RTT and CON. 3. Find GTKG1 by summing f6 GKG1 with GIDG1 and RTTP. Determine whether MAC1−1 MPS is equal to Checkf2 GTKG1 (SQN1−1 SD||R1−1 SD||SV1−1). Determine the anchor key if they are equal: The equation KMPS = f6 GTKG1 (SQN1−1 SD||SV1−1||hb(IB)||R1−1 SD||R1−1 MPS) can be formalized as follows. 5. Get RES1−1=f4 K1−1 (RTTP) and send it back to the MPSasan authentication response message in step 9. Before sending a successor failure message to SD1−1, MPS checks whether the receivedRES1−1 matches the XRES1−1 that corresponds to the SD stored in the information table. The identity identification of SD1-1 has been finished at this stage. Moreover, SD1−1 makes use of an anchor key KMPS with MPS, which is generated for session keys that come after it. C. Simplified AKA Performed by the Remaining Group SDs in Order for the Remaining Group SDs to Carry Out The AKA Process Using MPS. It is sufficient for them to finish the AKA procedure that includes the interaction between SD and MPS; the process in which MPS requests authentication data from TTP is absent. This is because all of the group's authentication data was previously saved in MPS during the AKA procedure of the initial SD. Here is the simplified AKA process: 1) The second user initiates generation of IB by touching the sensor of SD1−2, and simultaneously sends an access request to MPS. 2) The MPS grants the SD1−2 an identification request. 3) The following is the output of the AV1−2 SD while sending an identity response to MPS: 1. PKTTP Generation and Storage 2. Create SD1−2 AID using SEnc(k,SD1−2 ID). It's that simple! 3. Select R1-2 SD at random. 4. Create the identity response of SD1−2 by adding all the variables AID, R1−2 SD, GIDG1, SQN1−2 SD, and hb(IB) together. When these on-device sensors create an identification response message, AV1−2 SD is used instead of AV1−1 SD. Due to the fact that the second SD is not required to travel to TTP in order to get group authentication data, it discards the message authentication code. 4) When MPS gets authentication data that includes SD1−2 AID and hb(IB). The sentence checks whether SD1−2 AID and hb(IB) are same by doing a check over the blockchain. the production of AV1- 2 MPS and KMPS. Here is the full procedure: 1. Pick a random R1 minus 2 MPS. 2. Find the message authentication code of SD1−2 by summing of the following: MAC1−2 MPS = f2 GTKG1 (SQN1−2 SD||R1−2 MPS||SV1−2) where the group information table contained in the MPS may be queried to acquire both the GTKG1 and the SV1−2. 3. Make the authentication request for SD1−2 using the following formula: AV1−2 MPS=(R∏ TTP||CON||MAC1−2 MPS||R1−2 SD||R1−2 MPS). in which CON is equal to (AK⊕SQN1−2 TTP,MAC1−2 TTP) Determine whether the values of SD1−2 AID and hb(IB) are same on the blockchain. If the two do not match, MPS will deny the authentication and add a record to the blockchain indicating the issue. To take part in generating KMPS= f6 GTKG1 (SQN1−2 SD||SV1−2||hb(IB)||R1−2 SD||R1−2 MPS), remove hb(IB) from the blockchain if it is an exact match.

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

Step 5: Transfer AV1−2 MPS to SD1−2 Once SD1−2 receives the authentication request from MPS, it verifies the data. When it passes the test, SD1−2 sends a RES1−2 authentication response message to MPS and determines the anchor key KMPS.

Below is a thorough description of the procedure: 1. Use DEC(R* TTP,k) to decrypt RTTP. 2. Verify the message using RTT and CON. 3. Find GTKG1 by summing f6 GKG1 with GIDG1 and RTTP. 4. Verify whether GTKG1 is equal to MAC1−2 MPS given that SQN1−2 SD||R1−2 SD||SV1−2. Determine the anchor key if they are equal: The equation for KMPS is given by f6 GTKG1 multiplied by the following: (SQN1−2 SD||SV1−2||hb(IB)||R1−2 SD||R1−2 MPS) 5. Find the result of RES1−2=f4 K1−2 (RTTP) and send it back to MPS as an authentication response message 6) MPS checks whether the received result matches the XRES1−2 that corresponds to the SD stored in the information table and sends a successor failure message to SD1−2. So far, SD1−2's AKA procedure is finished, and the other SDs in the group (SD1−3,..., SD1−n) are run using the same approach as SD1−2.
Chapter IV: Systematic Review Here, we undertake the security analysis and the performance analysis of the suggested method, respectively. A. Analyzing Security Table III provides the results of our security investigation and functional comparison.

TABLE III: Functionality comparison

Functionality	Ryu et.al [13]	Panda et.al [19]	ha
Stolen Device Attack	✓	✗	✓
User Anonymity	✓	✓	✓
Forward and Backward Secrecy	✗	✗	✗
Malicious Adversary Traceability	✗	✗	✗

Theft of Device: Our system ensures that no one can steal an SD from an innocent user. Each SD is uniquely associated with the user, and the user's binding information is stored in the immutable blockchain. Therefore, even if an adversary were to attempt to authenticate, it would fail the blockchain check. We assume that a malevolent opponent may intercept the identify response message in 3 and access the information contained therein; hence, our suggested technique ensures user anonymity. But the enemy can't figure out who we really are because we encrypt SDID as SDAID using the ECIES mechanism and they don't have the key to decode it. For the purpose of forward and backward secrecy, we provide procedures for new SD entering or old SD departing the organization in the proposed system. It is necessary to update the GK and complete the AKA procedure whenever a new SD wants to join the group in order to keep forward secrecy. The goal of the aforementioned procedure is to prevent the newly installed SD from decrypting group data that has already been transferred. Update the GK and delete the relationship between the outgoing SD and group when the old SD wants to leave in order to retain backward secrecy. Doing so will ensure that the previous SD cannot decode any future group data transmissions. • Malicious Adversary Traceability: Our suggested approach stores hb(IB) and related SD information using blockchain technology. By checking the blockchain and adding the fraudulent record to a block, the MPS may identify when an attacker attempts to authenticate with an SD that does not match. Therefore, our plan takes use of the immutability of the blockchain to foil any attempts by an attacker to tamper with the data. Section B: Evaluating Results We evaluate our system in relation to other schemes in comparable situations in terms of computing cost, signaling cost, and communication cost. • computation cost: Considering the computation cost of SD and MPS during the authentication phase, we compared the proposed system with references [13], [19]-[21]. Using the MIRACL library, we measured 1000 cryptographic operations and obtained the average time on an Intel Core i5-7200U CPU with 16GB of RAM in a Windows 10 system environment. Below, you can find the results. Matrix resynthesis About 23.0195 milliseconds,

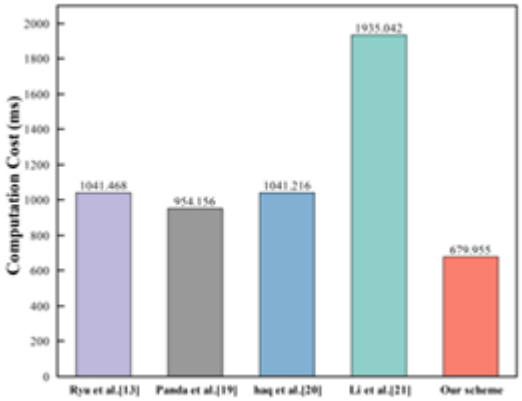

Fig. 3: Comparison of computation cost

ISSN 2454-9940

www.ijasem.org

Vol 19, Issue 2, 2025

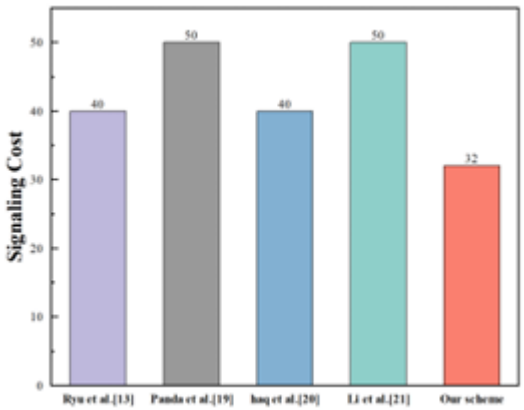

Fig. 4: Comparison of signaling cost

calculation of the area under the elliptic curve Addition at elliptic curve point with time m = 8.6732 ms Hash function Th ≈ 0.0008ms, biological hash, symmetric encryption and decryption Tsye ≈ 0.1426ms, and time Ta ~ 0.008ms. Approximately 0.01 milliseconds is the duration of the Tbh function. For counting functions like f1 in the scheme, we employ Th, and we disregard operations with very low execution durations like splicing and XOR. Taking into consideration that a group of 10 SDs is going through the authentication procedure. The computing cost for both the suggested and compared schemes is shown in Figure 3. The cost of signaling and the cost of communicating: In order to make calculations easier, we will first show the possible lengths of notations in the proposed scheme: 160 bits for SQN, 64 bits for MAC, 128 bits for R, 64 bits for RES, and 128 bits for SDAID. According to our plan, the first SD to join the group must send all five messages necessary for authentication. The cost of communication is 2304 bits, where i=1. The other SDs in the group, meanwhile, send three messages as part of a streamlined authentication procedure. The cost of communication is 1312 bits, which is equal to 3i=1. Assuming ten SDs are running the AKA process in parallel, we display the overall signaling cost and communication cost of each scheme in Figure 4 and Figure 5, respectively.

CONCLUSIONS AND FUTURE WORK

We provide a multi-factor group authentication technique that is built on the blockchain in this paper. Our technique streamlines the authentication process so that just the first SD is needed to finish it, while the other SDs in the group carry it out in a simplified manner. Avoiding the drawbacks of single-factor authentication and providing tamper-resistance, the blockchain is used to store the user's multi-factor authentication information. We then compared our system to others in the same situation and found that it was far more efficient at authentication and had additional security features. There may be excessive network latency during the handover process and additional issues will still be encountered by the SD in future work when it migrates from one MPS to another. The metaverse is an ideal setting for studies on handover authentication.

REFERENCES

[1]. R. Cheng, N. Wu, S. Chen, and B. Han, “Will Metaverse Be NextG Internet? Vision, Hype, and Reality,” IEEE Network, vol. 36, no. 5, pp. 197–204, 2022.

[2]. P. K¨urt¨ unl¨ uo˘ glu, B. Akdik, and E. Karaarslan, “Security of Virtual Reality Authentication Methods in Metaverse: An Overview,” 2022.

[3]. C.-C. Chang and H.-D. Le, “A Provably Secure, Efficient, and Flexible Authentication Scheme for Ad hoc Wireless Sensor Networks,” IEEE Transactions on Wireless Communications, vol. 15, no. 1, pp. 357–366, 2016.

[4]. M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, “Secure Remote User Authenticated Key Establishment Protocol for Smart Home Environment,” IEEE Transactions on Dependable and Secure Computing, vol. 17, no. 2, pp. 391–406, 2020.

[5]. A. Esfahani, G. Mantas, R. Matischek, F. B. Saghezchi, J. Rodriguez, A. Bicaku, S. Maksuti, M. G. Tauber, C. Schmittner, and J. Bastos, “A Lightweight Authentication Mechanism for M2M Communications in Industrial IoT Environment,” IEEE Internet of Things Journal, vol. 6, no. 1, pp. 288–296, 2019.

[6]. L. Wu, J. Wang, K.-K. R. Choo, and D. He, “Secure Key Agreement and Key Protection for Mobile Device User Authentication,” IEEE Transactions on Information Forensics and Security, vol. 14, no. 2, pp. 319–330, 2019.

[7]. R. Arul, G. Raja, A. K. Bashir, J. Chaudry, and A. Ali, “A Console GRID Leveraged Authentication and Key Agreement Mechanism for LTE/SAE,” IEEE

Transactions on Industrial Informatics, vol. 14, no. 6, pp. 2677–2689, 2018.

[8]. Z. Xu, W. Liang, K.-C. Li, J. Xu, A. Y. Zomaya, and J. Zhang, "A Time-Sensitive Token-Based Anonymous Authentication and Dynamic Group Key Agreement Scheme for Industry 5.0," IEEE Transactions on Industrial Informatics, vol. 18, no. 10, pp. 7118–7127, 2022.

[9]. C. Lai, H. Li, X. Li, and J. Cao, "A novel group access authentication and key agreement protocol for machine-type communication," Transactions on emerging telecommunications technologies, vol. 26, no. 3, pp. 414 431, 2015.

[10]. W. Zheng, L. Yan, W. Zhang, O. Liwei, and D. Wen, "D→K→I: Data Knowledge-Driven Group Intelligence Framework for Smart Service in Education Metaverse," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 53, no. 4, pp. 2056–2061, 2023.

[11]. R. Hare and Y. Tang, "Hierarchical Deep Reinforcement Learning With Experience Sharing for Metaverse in Education," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 53, no. 4, pp. 2047–2055, 2023.

[12]. Y. Han and S. Oh, "Investigation and Research on the Negotiation Space of Mental and Mental Illness Based on Metaverse," in 2021 In ternational Conference on Information and Communication Technology Convergence (ICTC), pp. 673–677, 2021.

[13]. J. Ryu, S. Son, J. Lee, Y. Park, and Y. Park, "Design of Secure Mutual Authentication Scheme for Metaverse Environments Using Blockchain," IEEE Access, vol. 10, pp. 98944–98958, 2022.

[14]. K. Yang, Z. Zhang, Y. Tian, and J. Ma, "A Secure Authentication Framework to Guarantee the Traceability of Avatars in Metaverse," 2022.

[15]. E. Han, M. R. Miller, N. Ram, K. L. Nowak, and J. N. Bailenson, "Un derstanding group behavior in virtual reality: A large-scale, longitudinal study in the metaverse," in 72nd Annual International Communication Association Conference, Paris, France, 2022.

[16]. Y. Wang, Z. Zhang, and Y. Xie, "Privacy-Preserving and Standard Compatible AKA Protocol for 5G," in USENIX Security Symposium, pp. 3595–3612, 2021.

[17]. X. Huang, Y. Xiang, E. Bertino, J. Zhou, and X. Li, "Robust Multi Factor Authentication for Fragile Communications," IEEE Transactions on Dependable and Secure Computing, vol. 11, no. 6, pp. 568–581, 2014.

[18]. A. T. B. Jin, D. N. C. Ling, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," Pattern recognition, vol. 37, no. 11, pp. 2245–2255, 2004.

[19]. P. K. Panda and S. Chattopadhyay, "A secure mutual authentication pro tocol for IoT environment," Journal of Reliable Intelligent Environments, vol. 6, pp. 79–94, 2020.

[20]. I. ul haq, J. Wang, and Y. Zhu, "Secure two-factor lightweight authen tication protocol using self-certified public key cryptography for multi server 5G networks," Journal of Network and Computer Applications, vol. 161, p. 102660, 2020.

[21]. Y. Li, M. Xu, and G. Xu, "Blockchain-based mutual authentication protocol without CA," The Journal of Supercomputing, vol. 78, no. 15, pp. 17261–17283, 2022.