GROUP SECURITY AUTHENTICATION AND KEY AGREEMENT PROTOCOL FOR LTE MILITARY GRADE COMMUNICATION

Padmapriya. B[1], Praison J[2], Rithick M[3], Sarankirthic C[4], Vasanthakumar M[5],
Assistant Professor[1], Department of Computer Science and Engineering,
UG Scholar[2,3,4,5], Department of Computer Science and Engineering,
Tirupur, Tamil Nadu, India.

Abstract—4G mobile communication is a global technology. Therefore, it is essential to enforce between mobile users and the networks. This represents a Group Security Authentication and Key Agreement Protocol Built by Elliptic Curve Diffie Hellman Key Exchange (GSAKA-ECDHKE) to overcome and address the LTE networks Evolved Packet System Authentication and Key Agreement Protocol (EPS-AKA) protocol flaws and vulnerabilities. GSAKA-ECDHKE is presented for 4G mobile military group communications to provide security, confidentiality, and privacy while the users and networks authenticate. By embedding the Group Commander (GC) role in the EPS-AKA protocol to control the member authentication in the group. GSAKA-ECDHKE protocol is based on Elliptic Curve Diffie Hellman Key Exchange (ECDHKE) and hash function to generate and share secret Elliptic Curve (EC) key to encrypt and protect the routing authentication parameters. The Eclipse programming IDE is used for security analysis and formal verification. It has been demonstrated that GSAKA-ECDHKE had overcome various known security attacks such as Man In The Middle (MITM), replay attacks, and Denial of Services (DoS) attacks, satisfying the evaluated security requirements. Additionally, the suggested protocol provides the lowest communication overheads compared to the existing protocols.

Keywords—Key Agreement protocol, Encryption, Decryption, Access control, Network Security, Threat detection.

I. INTRODUCTION

Mobile technology has been the most significant global event during the past thirty years, dominating all parts of life. Therefore, it has become vital for all individuals and businesses to save effort and fulfilling their duties. Each generation of mobile technology has added an appointed feature to its previous generation, for example, increasing the number of users, raising the data rates, and preserving privacy for each user.

Security is the base requirement for protecting the subscribers' privacy of any mobile telecommunications system. User privacy provides the network's users with convenient service connectivity, preventing the network from being exploited and securing their information. Authentication of the users for network access and ensuring a bidirectional trust between users and their network are key elements of building such secured systems. Both secure connectivity and user authentication are related to the authentication and access control mechanisms that provide secure network services for all the network subscribers.

Security among networks and users became a significant issue for any wireless network to save the daily subscribers' private data, financial transactions, and personal conversations. The authentication of users and networks in 4G mobile is the first step toward establishing security trust.

There are different attack types such as Man In The Middle (MITM), replay attacks, and Denial of Services (DoS) attacks, the Third Generation Partnership Project (3GPP) established the Evolved Packet System Authentication and Key Agreement Protocol (EPS-AKA) authentication protocol to serve as the authentication protocol for Long Term Evolution (LTE) networks that were presented for the 4G mobile communication, which demands adaptive bandwidth, low contact latency, greater data rates, and increased capacity and coverage of the networks.

To gain access and use the benefits of the network, the user verifies his identity using the EPS-AKA authentication mechanism. By attaching the user's authentication request, which includes pre-authentication information, to the Mobility Management Entity (MME), the MME temporarily transfers the user's identity information to the Home Subscriber Server (HSS), which is responsible for generating the authentication vector and returning it to the MME.

Any group authentication protocol's primary aims are to ensure secrecy and security among communicative entities, especially for crucial group military communications. Several group AKA methods have been presented to achieve the aims of the mutual AKA between the group members and their network, whether used for group mobile authentication or machine-type devices (MTDs) authentication. Privacy preservation and network overhead reduction are the most important aims of this type of communication.

II. RELATED WORK

Chen *et al.* presented the G-AKA protocol as the first group AKA protocol [13]. However, the protocol has been criticized for its high computational complexity and the need for frequent updates of group keys, which can lead to increased overhead and latency in communication. Additionally, the protocol does not provide perfect forward secrecy, which means that if a group key is compromised, all previous communications using that key could also be compromised. These limitations highlight the need for ongoing research and development of more efficient and secure group AKA protocols for military-grade communication.

Lai et al. [15] offered SE-AKA for 3GPP networks, while Jiang et al. [16] did. Both protocols have also been criticized for their high computational complexity and the need for frequent updates of group keys. Additionally, SE-AKA and EG-AKA do not provide perfect forward secrecy, leaving previous communications vulnerable if a group key is compromised. These limitations suggest that further research is necessary to develop more efficient and secure group AKA protocols for both 3GPP and non-3GPP networks.

Furthermore, Choi et al. [19] introduced the GROUP-AKA protocol. This protocol aims to address some of the limitations of SE-AKA and EG-AKA, such as reducing computational complexity and improving key management. However, GROUP-AKA has also been criticized for its reliance on a trusted third party and potential vulnerability to attacks from malicious insiders. Therefore, more research is needed to develop group AKA protocols that can provide both security and efficiency without relying on a single point of failure.

Cao et al. enhanced the group-based AKA protocols security by presenting a group signature-based GBAAM-AKA protocol [20]. This protocol uses group signatures to authenticate group members and reduce reliance on a trusted third party. However, it has been criticized for its high computational complexity and potential vulnerability to attacks from malicious insiders. Additionally, the use of group signatures may raise privacy concerns, as it allows for non-repudiation and traceability of group members. Therefore, further research is needed to develop group AKA protocols that can provide both security and privacy without sacrificing efficiency.

III. NETWORK SECURITY

Network security refers to the protection of computer networks and their resources from unauthorized access, theft, damage, or disruption. It involves implementing various measures, such as firewalls, encryption, access controls, and intrusion detection systems, to prevent attacks and maintain the confidentiality, integrity, and availability of network data and services. Network security is essential for ensuring the smooth operation of businesses, governments, and other organizations that rely on computer networks to store and transmit sensitive information.

The Diffie-Hellman key exchange is a cryptographic protocol that allows two parties to establish a shared secret key over an insecure communication channel. It was invented by Whitfield Diffie and Martin Hellman in 1976. The protocol is based on the discrete logarithm problem and involves the use of public and private keys. The two parties agree on a prime number and a generator, which are used to generate their own public and private keys. They then exchange their public keys and use them to compute a shared secret key. The shared secret key can be used for symmetric encryption of data transmitted between the two parties. The security of the protocol relies on the difficulty of solving the discrete logarithm problem.

The Diffie-Hellman key exchange protocol involves the following steps:

1. The two parties, Alice and Bob, agree on a prime number p and a generator g, where g is a primitive root modulo p.

2. Alice chooses a random secret integer a and computes A = g^a mod p. She sends A to Bob.

3. Bob chooses a random secret integer b and computes B = g^b mod p. He sends B to Alice.

4. Alice computes the shared secret key as S = B^a mod p.

5. Bob computes the shared secret key as S = A^b mod p.

6. Alice and Bob now have a shared secret key that they can use for symmetric encryption of their communication.

The security of the protocol relies on the difficulty of computing the discrete logarithm of A or B concerning g modulo p, which is believed to be a hard problem in computational number theory.

The Diffie-Hellman key exchange protocol is widely used in network security to establish a secure communication channel between two parties. It is used in various protocols such as SSL/TLS, SSH, and VPNs.

In SSL/TLS, the protocol is used to establish a secure session between a web server and a client. The server and client agree on a shared secret key using the Diffie-Hellman key exchange protocol, which is then used for symmetric encryption of the communication.

In SSH, the protocol is used to establish a secure connection between a client and a server for remote access. The client and server agree on a shared secret key using the Diffie-Hellman key exchange protocol, which is then used for encryption and authentication of the communication.

In VPNs, the protocol is used to establish a secure tunnel between two networks. The two endpoints of the tunnel agree on a shared secret key using the Diffie-Hellman key exchange protocol, which is then used for the encryption and decryption of the communication.

Overall, the Diffie-Hellman key exchange protocol is an essential tool for secure communication in network security.

All the headings in the main body of your paper are numbered (automatically).

Another type of heading is the “component heading”, which is used for other components that aren’t part of the main text. These are usually your acknowledgments and your references, which you can see examples of below. These headings are not numbered. The correct styling for them can be applied using the

"Heading 5" style, which is the same as the "Heading 1" style but without numbering.

IV. PROPOSED SYSTEM

Our proposed protocol has been successful in decreasing the length of the used variables and neglecting the required time to recalculate and update the group ID and the secret keys of the group when a member leaves or joins the group by authenticating each group member individually with the Group Commander (GC) when the Military User Equipment (MUEi) needs to access the network. That leads to reducing network congestion. The GC is a device with special capabilities for military aspects to guarantee the security requirements for MUEi. The GC's role is to transfer and receive the effective variables used in calculating and generating the authentication routing parameters. By controlling the mutual authentication between the protocol entities MME, HSS, and MUEs.

During protocol mechanism execution, the proposed protocol exploits hash functions and an Elliptic Curve (EC) secret key to authenticate entities with considerable security shape mutually. Also, in the proposed GSAKA-ECDHKE, the authentication mechanism will stop if the GC becomes unavailable. Additionally, the proposed protocol achieves the lowest network overheads compared to other protocols. Finally, the proposed protocol addresses the single key critical issue that earlier group-based AKA protocols could not address.

The algorithm can substantially reduce the computation and communication workload in a highly dynamic environment. The group key is used for future communication among the members of the group.

V. TESTING

A. Unit Testing

In unit testing the testing is performed on each module and this module is known as module testing. This testing was carried out during the programming state itself. In this testing, all the modules work satisfactorily as regards the expected output from the module. Unit testing is a method by which individual units of source code are tested to determine if they are fit for use. A unit is the smallest testable part of an application. In procedural programming, a unit may be an individual function or procedure. Unit tests are created by programmers or occasionally by white box testers.

Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviors that are to be trapped by the unit. A unit test case, in and of itself, documents these critical characteristics, although many software development environments do not rely solely upon code to document the product in development. Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit and how to use it can look at the unit tests to gain a basic understanding of the unit API

B.Acceptance Testing

Acceptance testing is black-box testing performed on a system (e.g. software, lots of manufactured mechanical parts, or batches of chemical products) before its delivery. It is also known as functional testing, black-box testing, release acceptance, QA testing, application testing, confidence testing, final testing, validation testing, or factory acceptance testing.

Acceptance testing generally involves running a suite of tests on the completed system. Each test, known as a case, exercises a particular operating condition of the user's environment or feature of the system and will result in a pass or fail, or Boolean, outcome. There is generally no degree of success or failure. The test environment is usually designed to be identical, or as close as possible, to the anticipated user's environment, including extremes of such. These test cases must each be accompanied by the test case input data or a formal description of the operational activities (or both) to be performed—intended to thoroughly exercise the specific case—and a formal description of the expected results.

VI. IMPLEMENTATION

The implementation phase focuses on how the engineer attempts to develop the system. It also deals with how data are to be structured, how procedural details are to be implemented, how interfaces are characterized, how the design will be translated into programming, and how the testing will be performed. The methods applied during the development phase will vary but three specific technical tasks should always occur.

A. Feasibility Assessment

In Feasibility, this stage problem was defined. Criteria for choosing a solution were developed, proposed possible solutions, estimated costs and benefits of the system and recommended the course of action to be taken.

B. Requirement Analysis

During requirement analysis, high-level requirements like the system's capabilities must be provided to solve a problem. Function requirements and performance requirements for the hardware specified during the initial planning were elaborated and made more specific to characterize features that the proposed system will incorporate.

C. External Design

The external design of any software development involves conceiving, planning out, and specifying the externally observable characteristic of the software product. These characteristics include user displays, report formats,

external data sources and data links, and functional characteristics.

D. Internal Design Architectural and Detailed Design

The internal design involved conceiving, planning out, and specifying the internal structure and processing details to record the design decisions and to be able to indicate why certain alternations were chosen in preference to others. These phases also include elaborating the test plans and providing blueprints of implementation, testing, and maintenance activities. The product of internal design is architectural structure specification.

The work products of internal design are architectural structure specification, the details of the algorithm, data structure, and test plan. In architectural design, the conceptual view is refined.

E. Detailed Design

The detailed design involved specifying the algorithmic details concerned with data representation, interconnections among data structures, and packaging of the software product. This phase emphasizes more on semantic issues and fewer synthetic details.

F. Coding

This phase involves actual programming, i.e., transacting detailed design into source code using an appropriate programming language.

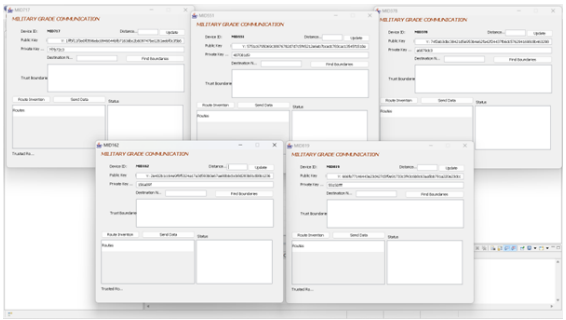
G. Debugging

This stage was related to removing and making errors from programs completely error-free.

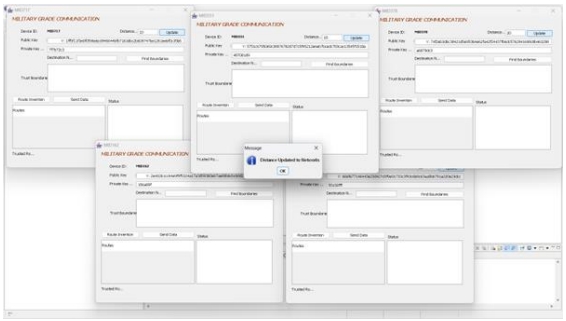
H. Maintenance

During this stage, the systems are loaded and put into use. They also get modified accordingly to the requirements of the user. These modifications included making enhancements to the system and removing problems.

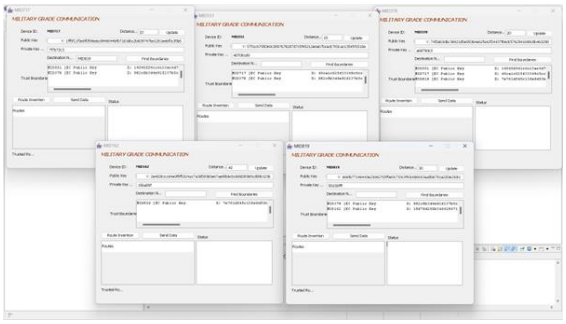
RESULTS:



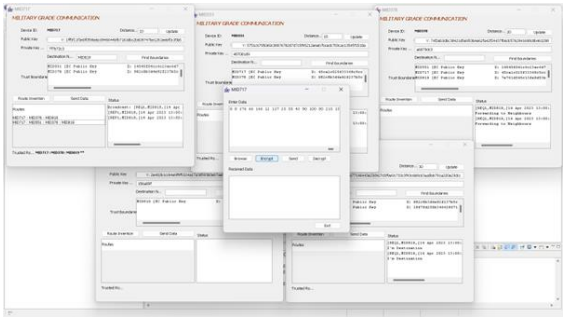
To create an individual ID for the Team Members.



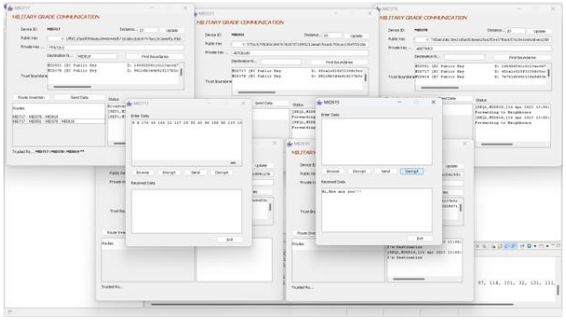
Set the individual distance for Team Members.



Finding the Trusted Boundaries.



Send the data by Encrypt.



Received the data by Decrypt.

CONCLUSION

To balance the requirements of user privacy protection and government-authorized monitoring in instant messaging systems, this paper proposed a session key escrow scheme based on threshold cryptography and a new tripartite authenticated key agreement protocol. The proposed session key escrow scheme takes into account the security of both key agreement and key escrow, and, unlike other solutions, does not

focus on the security of just one of them. To achieve authorization monitoring, the new scheme adopts a method that escrows the ephemeral private key of GSAKA-ECDHKE instead of the secure master key sKGM of KGM, and can therefore achieve fine-grained control in every session on avoiding the ``once monitor, monitor forever" scenario and reduce the management difficulty and risk of being corrupted of sKGM. In addition, GSAKA-ECDHKE will generate and use only one pair over the whole life of the IM system, which allows the proposed scheme to have low storage overhead for each KEA.

REFERENCES

[1] Karim H. Moussa, Ahmed H. El-Sakka, Shawky Shaaban, and Hassan Nadir Kheirallah, “Group Security Authentication and Key Agreement Protocol Built by Elliptic Curve Diffie Hellman Key Exchange”, June 2022.

[2] Y. Chen, J. Wang, K. Chi, and C. Tseng, ``Group-based authentication and key agreement," Wireless Pers. Commun., vol. 62, no. 4, pp. 965-979, Feb. 2012.

[3] P. K. Panda and S. Chattopadhyay, ``An enhanced mutual authentication and security protocol for IoT and cloud server," Inf. Secure. J., A Global Perspective, vol. 31, no. 2, pp. 144-156, Mar. 2022.

[4] C. Lai, H. Li, R. Lu, and X. S. Shen, ``SE-AKA: A secure and efficient group authentication and key agreement protocol for LTE networks," Comput. Netw., vol. 57, no. 17, pp. 3492-3510, 2013.

[5] R. Jiang, C. Lai, J. Luo, X.Wang, and H.Wang, ``EAP-based group authentication and key agreement protocol for machine-type communications," Int. J. Distrib. Sensor Netw., vol. 9, no. 11, Nov. 2013, Art. no. 304601.

[6] S. Chandra, S. Paira, S. S. Alam, and G. Sanyal, ``A comparative survey of symmetric and asymmetric key cryptography," in Proc. Int. Conf. Electron., Commun. Comput. Eng. (ICECCE), Nov. 2014, pp. 83-93.

[7] C. Lai, H. Li, X. Li, and J. Cao, ``A novel group access authentication and key agreement protocol for machine-type communication," Trans. Emerg. Telecommun. Technol., vol. 26, no. 3, pp. 414431, Mar. 2015.., vol. 24, no. 2, pp. 379–393, Jun. 2015.

[8] D. Choi, H.-K. Choi, and S.-Y. Lee, ``A group-based security protocol for machine-type communications in LTE-advanced," Wireless Netw., vol. 21, no. 2, pp. 405-419, 2015.

[9] J. Cao, M. Ma, and H. Li, ``GBAAM: Group-based access authentication for MTC in LTE networks," Secur. Commun. Netw., vol. 8, no. 17, pp. 3282-3299, 2015.

[10] A. Fu, J. Song, S. Li, G. Zhang, and Y. Zhang, ``A privacy-preserving group authentication protocol for machine-type communication in LTE/LTEA networks," Secur. Commun. Netw., vol. 9, no. 13, pp. 2002-2014, 2016.

[11] C. Lai, R. Lu, D. Zheng, H. Li, and X. Shen, ``GLARM: Group-based lightweight authentication scheme for a resource-constrained machine to machine communications," Comput. Netw., vol. 99, pp. 66-81, Apr. 2016.

[12] J. Li, M. Wen, and T. Zhang, ``Group-based authentication and key agreement with dynamic policy updating for MTC in LTEA networks," IEEE Internet Things J., vol. 3, no. 3, pp. 408-417, Jun. 2016.

[13] J. Yao, T. Wang, M. Chen, L. Wang, and G. Chen, ``GBS-AKA: Group-based secure authentication and key agreement for M2M in 4G network," in Proc. Int. Conf. Cloud Comput. Res. Innov. (ICCCRI), May 2016, pp. 42-48.

[14] N. S. Chaudhari, ``SEGB: Security enhanced group based AKA protocol for M2M communication in an IoT enabled LTE/LTEA network," IEEE Access, vol. 6, pp. 3668-3684, 2018.

[15] D. Abbasinezhad-Mood and M. Nikooghadam, ``An anonymous ECC-based self-certified key distribution scheme for the smart grid," IEEE Trans. Ind. Electron., vol. 65, no. 10, pp. 7996-8004, Oct. 2018.