

An efficient lightweight key exchange algorithm for internet of things applications

Fasila Kasfa Ali, Sheena Mathew

Division of Computer Engineering, School of Engineering, Cochin University of Science and Technology, Cochin, Kerala, India

Article Info

Article history:

Received Dec 10, 2021

Revised May 21, 2022

Accepted Jun 18, 2022

Keywords:

Blockchain

Diffie-Hellman algorithm

Internet of things security

Key exchange

Key management

No-share key transfer

ABSTRACT

Internet of things (IoT) gained wide popularity in recent years, and this is proved by tremendous increase in use of IoT applications worldwide. Distributed IoT applications can be implemented securely with the support of blockchain. By default, blockchain will ensure authentication of involved entities as well as integrity of data. Due to storage restrictions, use of hybrid system is preferred, and this involves cloud server for storage and blockchain for other functionalities. Data kept in cloud has to be encrypted by a strong encryption algorithm. Even though core security objectives are achieved, it is necessary to provide a secure method to exchange the key. Since, the key is the backbone of a security algorithm, protection of the key has to be ensured. In this work, an algorithm is proposed to provide a no-share key exchange between two communicating parties in a resource constrained environment. The same was implemented and compared with conventional key sharing algorithms. Security analysis was formally conducted by using widely accepted automated validation of internet security protocols and applications (AVISPA) tool and the proposed method proved to be secure.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Fasila Kasfa Ali

School of Engineering, Cochin University of Science and Technology

Kerala, India

Email: fasilaka@cusat.ac.in

1. INTRODUCTION

For the past few decades, internet of things (IoT) has become very popular worldwide. At the same time, vulnerabilities are also increased. IoT comprises of heterogeneous ‘things’ that are uniquely identifiable [1]. These devices have interoperable communication capabilities. Security goals have to be achieved in communication of these devices with internet and other IoT nodes. IoT applications are used widely in areas such as smart homes, health care, smart grids, and vehicular networks. The outbreak of cyber-attacks to IoT applications has to be addressed carefully. It is always preferred to study the existing vulnerabilities and threats, and then develop efficient security algorithms. Security goals include features such as integrity, confidentiality, authentication, and access control.

According to the well accepted and commonly used suite [2] of security algorithms, advanced encryption standard (AES) is being widely used to achieve confidentiality. Rivest Shamir Adleman (RSA) algorithm and elliptic curve cryptography (ECC) are used for digital signatures and verification purposes. Diffie-Hellman algorithm is widely used for exchanging key securely. SHA-2 and SHA-3 algorithms are accepted widely for providing integrity checks with the help of hash values.

One of the key security objectives is confidentiality and this is accomplished by applying data encryption. Encryption algorithms are used to provide secure data transfer by converting data to a format that

is not recognizable by unintended users. If encryption algorithm is symmetric, only one key will be used for both encryption and decryption. Examples include AES and data encryption standard (DES). RSA is a public key encryption algorithm. Public key algorithms are also known as asymmetric algorithms. Each entity will possess a private key and a shared public key. Other algorithms such as ECC and Diffie-Hellman key exchange fall under asymmetric category.

Major strength behind an encryption algorithm is the key value used. Hence, key security becomes a severe point of concern and key management algorithms have to be designed efficiently. Conventional security algorithms were designed to apply on networks. When IoT based systems are considered, scenario becomes different. Devices in IoT systems are resource constrained. Similar to any other communicating entities, IoT devices are also vulnerable to various attacks. Several studies were conducted on different types of attacks [3]–[5] and they were broadly categorized as wireless reconnaissance and mapping, physical security attacks, security protocol attacks and application security attacks [6]. Security needs and possible attacks in IoT networks are discussed in [7], [8]. Fabrication, denial of service (DoS) attack, man in the middle (MITM) attack, and eavesdropping, are some common types of attacks. So, there must be properly designed security measures to restrict these attacks.

In a conventional IoT application, data collected from various sensors are uploaded to the cloud through a gateway node. Data processing and analysis are performed in the cloud, and this will be accessed by other users. Anyway, the data will be uploaded to a central storage in an encrypted form, knowingly or unknowingly, by all the stakeholders. Figure 1 shows the data flow in the network model of a centralized architecture based IoT network.

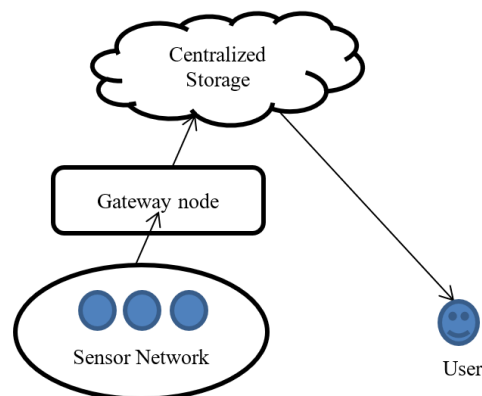


Figure 1. Network model of a conventional IoT application

IoT systems require lightweight algorithms. On the basis of this requirement, a lot of research works were conducted. Encryption algorithms are used to ensure data security and thus help to achieve confidentiality. These algorithms can be symmetric or asymmetric. A lot of researchers introduced lightweight algorithms [9]–[19] for IoT communication based on symmetric algorithm, AES. Major concern that arises in case of such encryption algorithms is that separate authentication algorithm should be applied to achieve authentication. In such a scenario, it is preferred to use authenticated encryption (AE) algorithms. Another suggestion to achieve multiple goals simultaneously is to use attribute based encryption (ABE) schemes [20], [21]. ABE based security methods for IoT applications are given in [22], [23]. In this, data can be accessed by only those who possess a specified set of attributes. If the encryption is based on DES, AES, or any other lightweight cipher, in other words, if it is a symmetric one, then an efficient key management technique must be used. Same key will be used to decipher if a symmetric method is followed and hence compromising the key will definitely collapse the system. Similarly, session keys may be used to protect communication sessions. So, session key agreement also becomes a point to be addressed.

Performance analysis of IoT oriented security algorithms is given in [24], in which the derivations are based on cryptographic libraries like Crypto++. A comparison study between Diffie-Hellman (DH) and elliptic curve Diffie-Hellman (ECDH) is given in [25], and in this, ECDH is concluded to be better in terms of power consumption and robustness. However, still, there are many challenges existing in this domain [26], [27]. Some of these include identity and access management, access control, and secret information exchange between the participating nodes. Keoh *et al.* [28] had given a study of IoT key management protocols in which centralized and decentralized approaches were discussed. In the study [29], it is mentioned that the

biggest issue in smart home energy management IoT systems is to establish a common session key initially. A novel key exchange algorithm is given in [2]. Koduru *et al.* [2] state that it is better than DH algorithm. However, in that method also, a common value has to be assumed initially. However, in the proposed method, it is exactly, a zero-share technique.

From the literature review, it can be concluded that there does not exist a unique solution that satisfies all requirements for an IoT environment, even though requirements vary depending on the applications. So, before applying appropriate security solution, requirements for the specific IoT application must be studied in detail and appropriate security algorithm must be chosen. As far as any security algorithm is considered, whether it is symmetric or asymmetric, key management is a crucial matter of concern. Hence, designing an efficient key management algorithm is an important task. In this paper, a secure key exchange method is proposed, and it is compared with existing algorithms including the commonly accepted DH algorithm and ECDH algorithm for key exchange.

The following sections are arranged. Next section explains the proposed method, and section 3 implementation details. Section 4 gives the performance analysis which contains both informal and formal analysis with the automated validation of internet security protocols and applications (AVISPA) tool.

2. PROPOSED METHOD

Hybrid distributed IoT applications involve distributed IoT nodes that are connected to a cloud through gateway node. Implementing such an application with blockchain will definitely ensure authentication of involved gateway nodes. In our system, several sensing devices will be connected to a gateway node, and this is a member of blockchain. When, a new participant enters blockchain, user credentials are assigned to provide authentication of users. When a new device is added under a gateway node, a master key is generated corresponding to that device. Uploading of data generated from that device has to be encrypted with the master key. All the communications between a gateway node and its connected devices will be protected by a group key. The group key used here is K_{AG} and this is a symmetric key that will be agreed upon by the connected devices to a particular gateway. The encryption with key K_{AG} provides a double layer of protection to the data. First level encryption is done by the master key, K and this provides the basic stronger protection for the data. The encryption algorithm used is AES with 128 bit key. Since this is a symmetric algorithm; same key has to be distributed with intended recipient. The proposed system provides more security with the assistance of ABE scheme. Key will be shared with the recipient only if the attribute for that entity matches with the specified set of policies for a particular data item. Now, an efficient secure key exchange algorithm should be selected for sharing the key between sending node and approved recipient node. Several conventional key transfer algorithms exist.

One-time pad is the only cryptosystem that provides perfect secrecy. However, the drawback is to have a means for sharing common key between the sender and recipient. This kind of information sharing is always a point of concern in any security algorithm. In order to deal with this, following idea can be used. The sender (S) and recipient (R) will have to choose their own private key values, say K_1 and K_2 , respectively. Then, the common information (here it is the master key, K), can be shared through a fixed number of handshaking steps, as mentioned in [30]. The steps involve matrix computations. In order to reduce the computation, size of the matrix is restricted to 4×4 , so that the effort is not tedious for constrained IoT devices. At the same time, it is not easy to retrieve the real value, for an eavesdropper since matrix operations are mostly non-commutative in nature. The proposed method is able to stand up against brute force attacks, as calculating inverse of matrices is impossible if it is a singular matrix.

In case of conventional DH key exchange algorithm, a common secret has to be shared between the two participants and this will be used to exchange actual key. In DH, both sender and receiver have to select and agree on a large prime number and a base value. The initial parameters have to be shared over a medium and these values may be received by an eavesdropper also. For this sharing, a secure channel or some kind of synchronization would be needed. This will create overhead and possibility of errors or attacks if the medium is prone to an attacker. DH algorithm does not provide authentication. That means any person can impersonate as the sender and exchange the key with a recipient. It is also possible to have a DoS attack since an attacker can establish a communication. ECDH provides authentication by using private and public key pairs. However, in ECDH also, some initial values have to be shared between the communicating entities. Both algorithms are having exponential computations. One major drawback in these two existing key exchange algorithms is that both are prone to active MITM attack. They provide protection from passive attackers. However, an unauthorized party can impersonate as the sender/recipient and gain access to the key. In ECDH, the attacker can intercept a communication and pass his own public key, by replacing actual sender's public key to recipient. Recipient, without knowing this, may execute further steps with the unauthorized person. Thus, it is a valid conclusion that initial parameter-exchanging steps are creating

loopholes to attacks. In proposed method, it is not needed to share anything between the sender and recipient. Also, all the sessions between a sensor node and its gateway node are encrypted by the group key.

Proposed method is given in Figure 2 and Table 1 gives the algorithm. First step is to randomly generate a master key for a particular device by the sender (A). Since the algorithm is based on square matrix computations, it is preferred to use a master key of convenient length. In this work, algorithm uses a 4×4 matrix. So, it is convenient if the master key generated contains 128 bits, to convert that to get a matrix with 16 elements. First eight bits of the key will form the first element in matrix, second eight bits form second matrix element and so on. All further communications from the device will be encrypted with this master key, say K.

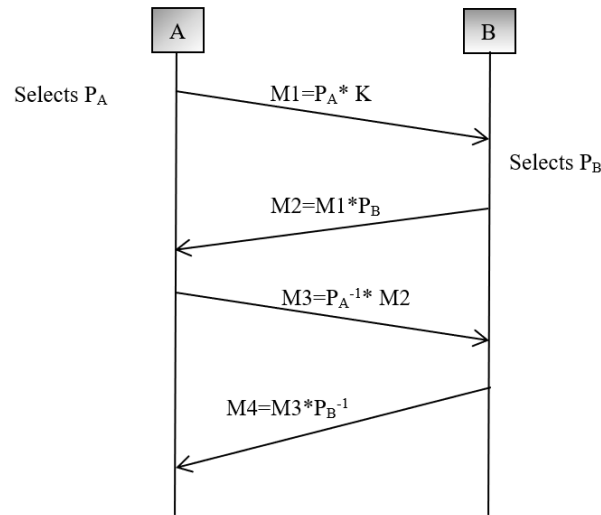


Figure 2. No-share key exchange method

Table 1. Proposed algorithm

Prerequisites: Sender, A and recipient, B select their private key matrices P_A and P_B , respectively. All the message transfers given below will be protected by the group key, K_{AG} , which is the symmetric key shared between the gateway node and the device.
Steps: 1. When a new device is added at a user, A, a master key, K is generated. The key size is 128 bits. 2. For sharing K between A and B, following steps are proposed. <ol style="list-style-type: none"> 2.1. K is converted initially to a 4×4 matrix. 2.2. Sender, A selects a private key square matrix P_A, which is of same dimension. 2.3. Find the product, $M1 = P_A \times K$ and send to B. 2.4. At the receiver side, B selects private key matrix, P_B of same dimension. 2.5. B computes $M2 = M1 \times P_B$ and sends to A. 2.6. At A, $M3 = P_A^{-1} \times M2$ and transfer to B. 2.7. B computes $M4 = M3 \times P_B^{-1}$ and this $M4$ is actually the key, K. 3. A encrypts the data by a strong encryption algorithm with encryption key as K. B can decrypt the data only if he/she possesses the key, K. 4. B gets the key value, K through the steps in 2, and decrypts the data

Value of K cannot be retrieved at any of these intermediate steps by an attacker if some conditions are imposed. This means, even if the channel is insecure, key cannot be compromised. It is mandatory that private key matrices of size 4×4 , selected by A and B are invertible. Communicating parties do not have to share it over the insecure channel. However, intermediate steps involve multiplication with inverse of these matrices. Another point to be taken care is that the intermediate matrices like M1 and M2 should be non-invertible.

If this matrix M1 is invertible, attacker can retrieve private key of B after step 2.5 because M2 is a product of M1 and B's private key, P_B . So, if M1 is invertible, an eavesdropper can catch it after step 2.3, take the inverse and multiply that with the matrix M2 after step 2.5, and easily gain P_B . So, it must be ensured that M1 is noninvertible. If the product matrix, M1 is invertible, there is a possibility for the attacker to retrieve the private key matrix by performing some inverse operations. For a square matrix to be

non-invertible it must be singular or, its determinant must be zero. Here, it is ensured that the master key matrix generated randomly, is having a determinant equal to zero. Hence, following are the conditions to be satisfied when the private matrices as well as key are generated: i) private matrices, P_A and P_B should be invertible and ii) $M1$ must be noninvertible.

3. RESULTS AND DISCUSSION

An application scenario based on patient health information was developed. The hardware components include Raspberry Pi machine, connected to a healthcare application in a distributed system. Blockchain was used for implementing the same. The platform selected was Hyperledger fabric supported with NodeJS, node package manager (NPM), Go and docker installations. As a test case, two organizations were created, and each organization had two peers. A channel is created to share the ledger and organizations are added to this channel. Chaincodes (smart contracts) are written in go programming language (Golang), and these are installed to channels. Then, the chaincode has to be instantiated by any member in the network. Required certificates and cryptographic materials are generated using CryptoGen tool in Hyperledger fabric. When the user node is enrolled to blockchain, a private and public key pair is created. Private key will provide authentication in all further communication from this user. Same key pair can form a basis for authenticating devices to this particular user. Implemented system consists of a patient connected to smart home, which is considered as Organization-1 and several IoT devices are connected to this member patient. Assume that patient named Johny is connected to a device, with ID "D1" When user authenticates and enrolls a new IoT device, a master key is generated. The sensed data from D1 will be encrypted with a strong encryption algorithm and uploaded to the storage. Here, for implementation, AES was used for encryption. The key used for this encryption is protected using ABE. Key will be shared with the second party only if he has the specified set of attributes. This helps to achieve access control also. To decipher the data retrieved, receiver has to obtain the key used for encryption. For this key exchange, the proposed algorithm is used. With the 128 bits key, this algorithm provides nonlinear level of security. This makes the efforts to be put by an eavesdropper to retrieve the key, tedious. When the device is added to the patient (Organization-1), a master key is generated. Here, the random 128 bits key generated is "#8+&8fNM+NBC&MCJ". Hence, the matrix obtained is,

$$K = \begin{bmatrix} 35 & 56 & 43 & 38 \\ 56 & 102 & 78 & 77 \\ 43 & 78 & 66 & 67 \\ 38 & 77 & 67 & 74 \end{bmatrix}$$

$$\text{Private matrix of the sender, } P_A = \begin{bmatrix} 218 & 188 & 131 & 32 \\ 40 & 42 & 22 & 5 \\ 27 & 28 & 17 & 4 \\ 6 & 5 & 4 & 1 \end{bmatrix}.$$

$$\text{Private matrix of receiver (doctor in this test case), } P_B = \begin{bmatrix} 1265 & 358 & 479 & 32 \\ 355 & 112 & 85 & 5 \\ 277 & 77 & 323 & 23 \\ 11 & 3 & 14 & 1 \end{bmatrix}$$

Data generated by the device is encrypted by using K and uploaded to the storage (here it is MongoDB). Along with this, patient will specify some set of attributes, and this is to facilitate ABE. This will help to achieve access control also. When a particular doctor would like to access the data item, blockchain will perform the access control checks. If the attributes of the doctor are matched with the specified attributes, the key can be shared by using the proposed no-share scheme.

4. PERFORMANCE ANALYSIS

Performance analysis of the proposed method is given in two subsections. First one discusses the informal analysis. Informal analysis section discusses the resistance of proposed method against various attacks. This section is followed by a formal study with the help of widely accepted AVISPA tool [31].

4.1. Informal analysis

This section explains the measures taken in the proposed method to resist various kinds of attacks. Possible attacks like eavesdropping attack, MITM attack and node tampering attack are being considered. The ways in which proposed method provides protection against these attacks are discussed.

4.1.1. Eavesdropping attack

IoT devices communicate with the gateway node through wireless networks and hence, all the message exchanges may be observed by an adversary. In the proposed method, once the device gets authenticated successfully with the gateway node, both parties agree upon a symmetric group key. Data is first encrypted by the master key and all steps involved in exchanging master key are protected by this group key. So, it is proved that the method is secured from eavesdropping attack and the attacker will never be able to retrieve the data.

4.1.2. MITM attack

In MITM attack, the adversary stands in between the communicating entities. The proposed method is resilient to this kind of attack since, the group key is known only to authenticated devices that are connected to the gateway node. Also, steps for exchanging the master key involve computation with private key matrices which is specific to the intended sender and recipient only. The gateway nodes involved are members of blockchain and hence, it is not required to recheck the authentication of those nodes. Blockchain provides strong authentication to the participants.

4.1.3. Node tampering

Physical device security cannot be assured. This means, the node may be captured by an attacker and its local memory may be accessed. Hence, the master key generated by the device will be kept in local storage in an encrypted form. For this encryption, the group key is used. The group key will be updated after each session. This shows our method is resilient to node tampering. Even though the node is tampered, and local memory is accessed by an attacker, he/she will not be able to retrieve the symmetric master key of the device.

In addition to the resilience to such common attacks, proposed method ensures authentication of involved entities. Gateway nodes are authenticated with the blockchain parameters. These gateway nodes will have to perform mutual authentication steps to verify the credentials of the device. A summary of the resilience of proposed method against various attacks and its comparison with the conventional and widely used key exchange methods such as DH algorithm and ECDH key exchange algorithm is given in Table 2. DH is based on modular arithmetic and its security is based on discrete logarithm problem while ECDH is adapting the concept of DH. In case of DH, large prime number has to be selected and for ECDH, curve parameters have to be selected. So, both these algorithms involve the exchange of some initial parameters. Steps involved in these two algorithms are irreversible since they are based on computational Diffie-Hellman Problem. Similarly, steps involved in proposed method are non-commutative. One major drawback of DH is that it does not ensure authentication. So, any person can involve in initial parameter exchange session and get the secret key. However, this drawback is not present in the next two methods (ECDH and proposed), since they involve private keys associated with each entity.

Table 2. Comparison between DH, ECDH and proposed method

Characteristics	Diffie-Hellman Key Exchange (DHKE)	Elliptic Curve Diffie-Hellman (ECDH) KE	No-Share Key Exchange (NSKE)
Based on Discrete logarithm	Yes	Yes	No
Initial parameters to be shared	Required, Large prime numbers and value to be exchanged	Required; Public keys and curves has to be generated/selected	Not required
Steps involved	Irreversible	Irreversible	Non-commutative
Authentication	Not present	Present	Present
Types of operation	Exponential steps involved	Elliptic curve arithmetic involved	Matrix multiplications and inverse calculations involved
Level of security	Linear	Linear	Nonlinear
Brute Force attack	Not possible	Not possible	Not possible
MITM (active)	Possible	Not possible	Not possible
DoS attack	Possible	Not possible	Not possible

Another advantage of proposed algorithm is that it provides nonlinear computations (using matrices). All these methods are resistant to Brute Force attacks and passive MITM attacks. In DH, computations are involving exponential steps, whereas in ECDH, elliptic curve arithmetic operations are involved. However, in proposed method, steps include only matrix computations that are not much complicated.

4.2. Formal Analysis

Here, a formal analysis of the proposed method is done with the help of AVISPA tool. AVISPA is an analysis tool that helps to model security protocols and perform the analysis using different in-built backend compilers. The tool consists of four different back-ends: on-the-fly model-checker (OFMC), constraint-logic-based attack searcher (CL-AtSe), SAT-based model-checker (SATMC) and tree automata based on automatic approximations for the analysis of security protocols (TA4SP).

Proposed method was first written in Alice-Bob notation and then, it was converted to high level protocol specification language (HLPSL). For illustration, the first 2 steps of Alice-Bob notation of encryption are given below. Assume that A is the sending device. G is the gateway node and B is the recipient node. K is the master key used for symmetric encryption of data by A, and K_{AG} is the group key shared by G with A.

$$A \rightarrow G: \{\{X\}_K\}_{K_{AG}} \text{ and } G \rightarrow B: \{\{X\}_K\}_{K_B}.$$

K_B is the public key of recipient, B. Gateway node, G decrypts the received message with the symmetric key, K_{AG} and encrypts it with public key of B and sends to B. B receives $\{\{X\}_K\}_{K_B}$ and it retrieves X by decrypting with its own private key first and then by the master key shared with NSKE method. As an example, HLPSL code snippet of the gateway role is given below. Similarly, role of the sender and role of the recipient node were also implemented and tested.

```
role role_G(
G, A, B      : agent,
K, Kag      : symmetric_key,
Kb          : public_key,
SND, RCV    : channel(dy)) played_by G
def=
local State : nat, S: text
init
State      :=0
transition
1. State=0/\RCV(A.\{\{S\}_K\} Kag)=|>State':=1/\SND(\{\{S\}_K\}_Kb)
end role
```

The protocol has been tested and the results obtained for OFMC and CL-AtSe compilers in AVISPA tool are given, in Figure 3. The protocol specification turned out to be SAFE under OFMC, SAFE under CL-AtSe, INCONCLUSIVE under SATMC and INCONCLUSIVE under TA4SP compilers. OFMC backend compiler tests the specified protocol against passive attacker and here, proposed method proved to be SAFE. Hence, it is protected from replay attack. The system is based on Dolev-Yao model [31]. According to this DY model, all the information exchange will be visible to the intruder. The knowledge shared with intruder was specified to be role_A, role_G, role_B, K_{AG} and K_B , in the HLPSL file specification. This means, the proposed method satisfies the security goals (secrecy and authentication) specified in the environment section, provided the master key K is not known to the intruder.

The exchange of master key steps involves 4 steps as already discussed. All the data exchanges are protected by encryption with the symmetric key K_{AG} . This key is available only to the authenticated devices which are connected to the gateway node, G. The first step of transferring master key, K is $M1=P_A*K$ and second step is $M2=M1*P_B$.

<pre>% OFMC % Version of 2006/02/13 SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/span/span/testsuite/results/nske2.if GOAL as_specified BACKEND OFMC</pre>	<pre>SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/results/nske2.if GOAL As Specified BACKEND CL-AtSe</pre>
---	---

Figure 3. Results of OFMC and ATSE compilers in AVISPA testing

All these steps are passed through the wireless communication medium and are protected with the group key encryption. Another possibility is that adversary can retrieve $M1$ and compute its inverse so that in second step, $M2 * M1^{-1}$ can be done to obtain the private key (P_B) of B. However, in our method, $M1$ is a non-invertible matrix which terminates the possibility of such an attack. This can be proved by method of proof by contradiction. Assume that the proposition, $X \rightarrow Y$:

If $M1$ is noninvertible, P_B is not computable.

So, according to Proof by contradiction method, initially, we assume that negation of this proposition is true. It can be written as (1).

$$\sim X \text{ is true} \quad (1)$$

This means, assume $M1$ is invertible. Inverse of $M1$ can be computed as (2).

$$M1^{-1} = 1/\det(M1) * (\text{adj}(M1)) \quad (2)$$

Where, $\det(M1)$ and $\text{adj}(M1)$ denote the determinant and adjoint of $M1$, respectively. However, we have put the constraint as $\det(M1)$ should be 0. Therefore, (2) becomes $M1^{-1} = (1/0) * (\text{adj}(M1))$, in which, division by zero is not defined. This means $M1$ does not have an inverse or $M1$ is non-invertible, which is a contradiction. So, our initial assumption, “ $\sim X$ is true” was wrong and hence, we conclude that X is true. This proves that even though an adversary can obtain the contents of $M1$ by breaking the encryption with group key, he/she will not be able to proceed to find out the actual private key matrix. The efficiency of the algorithm can be generally explained with the following points; i) as it is not required to share a common data between two parties, private key matrices can be generated randomly. Only thing to be taken care is that determinants of these matrices should not be zero and ii) handshaking steps involve noncommutative matrix multiplication steps. Thus, it is assured that any combinations of these intermediate steps will not help the attacker to retrieve the data. The costs involved in computation of matrix operations are.

- Steps 2.1, 2.2, 2.3: (Taking a number $+2x$ multiplication) cost of computation involves $2xN^2$ steps, for P_A and P_B .
- Step 2.4, 2.5: Cost of matrix multiplication= N^3 steps for matrices of order N
- Step 2.6, 2.7: Cost of inversion+Cost of multiplication= $2xN^3$ steps for matrices of order N .

Complexity increases as the matrix order, N increase. Since the system consists of resource constrained devices, it is better to choose a low value to form the square matrix. However, since security cannot be compromised, it should not be too low also. A 4×4 matrix is a sufficient value to provide a good level of security features. Since the sensed data is of moderate size, 128 bits key is enough to provide sufficient security. Compared to the well accepted DH algorithm and ECDH key exchange protocol, the major strength of this proposed method is that it works on the basis of zero-share scheme, since nothing needs to be shared at the beginning. Comparison between DH, ECDH and proposed NSKE method in terms of execution time is given in Figure 4. The algorithms were implemented in JavaScript, run in Chrome web browser and system specifications include Windows-64-bit operating system, 4 GB RAM.

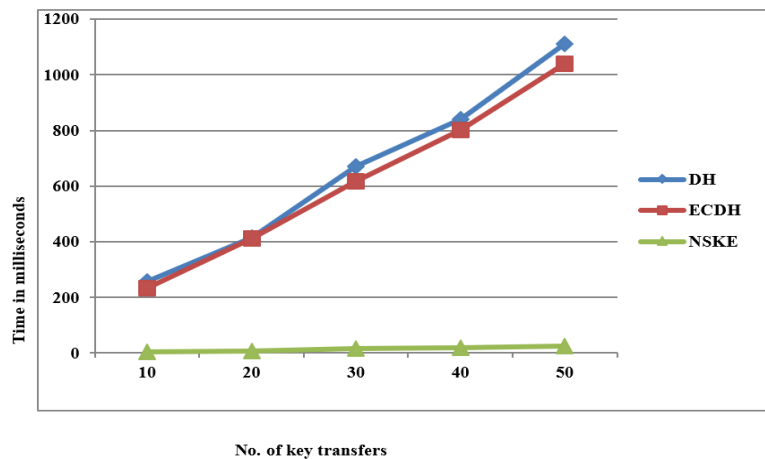


Figure 4. Execution time comparison between DH, ECDH and NSK (proposed method)

From this figure, it is clear that NSKE method takes very less time to exchange a key between 2 parties compared to the time taken by DH and ECDH key exchange methods. NSKE performs 45.74 (average of all trial runs) times faster than DHKE and ECDHKE. The proposed algorithm proved to be secure under various attacks. Now, it is also proved that computational cost and complexity involved are very less compared to other conventional algorithms. It can be concluded that proposed method is efficient in terms of complexity also. Hence, it is well suited for resource constrained devices.

5. CONCLUSION

Several challenges arise when IoT applications are implemented in a distributed manner. The upcoming blockchain paradigm will help to achieve the security goals like authentication and integrity. Confidentiality will be achieved if the data is encrypted with an efficient algorithm. Major security challenge still exists is the key management. Hence, a zero-share secret information exchange algorithm is proposed. The algorithm is able to provide better security than conventional key exchange algorithms, provided encryption is done. Since there is no need to share any information between the communicating nodes, unlike existing key exchange algorithms, this proposed method can be used to exchange initial session key. Also, the proposed method performs much faster compared to conventional key exchange algorithms. The transferred session key can be used for securing all further data exchanges between the parties.




REFERENCES

- [1] P. Raj and A. C. Raman, *The internet of things enabling technologies, platforms, and use cases*, 1st Edition. Routledge Taylor and Francis Group, 2017.
- [2] S. Koduru, P. Prasad Reddy, and P. Preethi, "A novel key exchange algorithm for security in internet of things," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 16, no. 3, pp. 1515–1520, Dec. 2019, doi: 10.11591/ijeecs.v16.i3.pp1515-1520.
- [3] D. Etter, *IOT (internet of things) programming: a simple and fast way of learning IOT*. Goodreads, 2016.
- [4] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for internet of things: a survey," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 70–95, Feb. 2016, doi: 10.1109/JIOT.2015.2498900.
- [5] A. Bahga and V. Madisetti, *Internet of things-a hands-on approach*, Universities Press, 2014.
- [6] Z. Ling, J. Luo, Y. Xu, C. Gao, K. Wu, and X. Fu, "Security vulnerabilities of internet of things: a case study of the smart plug system," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1899–1909, Dec. 2017, doi: 10.1109/JIOT.2017.2707465.
- [7] E. Kenneally, "The TTPs of privacy and security of the IoT," *IEEE Internet of Things Magazine*, vol. 1, no. 2, pp. 8–11, Dec. 2018, doi: 10.1109/MIOT.2018.8717595.
- [8] I. Bhardwaj, A. Kumar, and M. Bansal, "A review on lightweight cryptography algorithms for data security and authentication in IoTs," in *2017 4th International Conference on Signal Processing, Computing and Control (ISPPCC)*, Sep. 2017, pp. 504–509, doi: 10.1109/ISPPCC.2017.8269731.
- [9] S. Sahoo, S. S. Sahoo, B. Sahoo, and A. K. Turuk, "Design of an authentication scheme for cloud-based IoT applications*," *IETE Technical Review*, pp. 1–14, Feb. 2021, doi: 10.1080/02564602.2020.1854059.
- [10] W. Yu and S. Kose, "A lightweight masked AES implementation for securing IoT against CPA attacks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 11, pp. 2934–2944, Nov. 2017, doi: 10.1109/TCSI.2017.2702098.
- [11] P. Luo, L. Zhang, Y. Fei, and A. A. Ding, "Towards secure cryptographic software implementation against side-channel power analysis attacks," in *2015 IEEE 26th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, Jul. 2015, pp. 144–148, doi: 10.1109/ASAP.2015.7245722.
- [12] D. D. Hwang *et al.*, "AES-based security coprocessor IC in 0.18- μ m CMOS with resistance to differential power analysis side channel attacks," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 781–792, Apr. 2006, doi: 10.1109/JSSC.2006.870913.
- [13] H. Pahlevanzadeh, J. Dofe, and Q. Yu, "Assessing CPA resistance of AES with different fault tolerance mechanisms," in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2016, pp. 661–666, doi: 10.1109/ASPDAC.2016.7428087.
- [14] N. Benhadjoussef, H. Mestiri, M. Machhout, and R. Tourki, "Implementation of CPA analysis against AES design on FPGA," in *2012 International Conference on Communications and Information Technology (ICCIT)*, Jun. 2012, pp. 124–128, doi: 10.1109/ICCITechnol.2012.6285774.
- [15] C. Wang, M. Yu, J. Wang, P. Jiang, and X. Tang, "A more practical CPA attack against present hardware implementation," in *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, Oct. 2012, pp. 1248–1253, doi: 10.1109/CCIS.2012.6664584.
- [16] C. Tokunaga and D. Blaauw, "Securing encryption systems with a switched capacitor current equalizer," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 1, pp. 23–31, Jan. 2010, doi: 10.1109/JSSC.2009.2034081.
- [17] W. Yu, O. A. Uzun, and S. Köse, "Leveraging on-chip voltage regulators as a countermeasure against side-channel attacks," in *Proceedings of the 52nd Annual Design Automation Conference*, Jun. 2015, pp. 1–6, doi: 10.1145/2744769.2744866.
- [18] O. A. Uzun and S. Kose, "Converter-gating: a power efficient and secure on-chip power delivery system," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 4, no. 2, pp. 169–179, Jun. 2014, doi: 10.1109/JETCAS.2014.2315880.
- [19] W. Yu and S. Kose, "Time-delayed converter-reshuffling: an efficient and secure power delivery architecture," *IEEE Embedded Systems Letters*, vol. 7, no. 3, pp. 73–76, Sep. 2015, doi: 10.1109/LES.2015.2433175.
- [20] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security-CCS '06*, 2006, pp. 89–98, doi: 10.1145/1180405.1180418.
- [21] A. Sahai and B. Waters, "Fuzzy identity-based encryption cryptology." Accessed: Jul. 19, 2021. [Online]. Available: <https://eprint.iacr.org/2004/086.pdf>




- [22] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, May 2007, pp. 321–334, doi: 10.1109/SP.2007.11.
- [23] S. Atram and N. R. Borkar, "A review paper on attribute-based encryption scheme in cloud computing," *International Journal of Computer Science and Mobile Computing (IJCSMC)*, vol. 6, no. 5, pp. 260–266, 2017.
- [24] N. Khan, N. Sakib, I. Jerin, S. Quader, and A. Chakrabarty, "Performance analysis of security algorithms for IoT devices," in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, Dec. 2017, pp. 130–133, doi: 10.1109/R10-HTC.2017.8288923.
- [25] T. K. Goyal and V. Sahula, "Lightweight security algorithm for low power IoT devices," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2016, pp. 1725–1729, doi: 10.1109/ICACCI.2016.7732296.
- [26] A. Ukil, J. Sen, and S. Koilakonda, "Embedded security for internet of things," in *2011 2nd National Conference on Emerging Trends and Applications in Computer Science*, Mar. 2011, pp. 1–6, doi: 10.1109/NCETACS.2011.5751382.
- [27] A. Rghioui and A. Oumnad, "Challenges and opportunities of internet of things in healthcare," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 5, pp. 2753–2761, Oct. 2018, doi: 10.11591/ijece.v8i5.pp2753-2761.
- [28] S. L. Keoh, S. S. Kumar, and H. Tschofenig, "Securing the internet of things: a standardization perspective," *IEEE Internet of Things Journal*, vol. 1, no. 3, pp. 265–275, Jun. 2014, doi: 10.1109/JIOT.2014.2323395.
- [29] S. Naoui, M. E. Elhdhili, and L. A. Saidane, "Security analysis of existing IoT key management protocols," in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, Nov. 2016, pp. 1–7, doi: 10.1109/AICCSA.2016.7945806.
- [30] P. R. Mahalingam and K. A. Fasila, "Zero-share key management for secure communication across a channel," *Design and Analysis of Security Protocol for Communication*. Wiley, pp. 95–108, Jan. 2020, doi: 10.1002/9781119555759.ch4.
- [31] T. Genet, "A short SPAN-AVISPA tutorial," Research Report, IRISA, 2015.

BIOGRAPHIES OF AUTHORS



Fasila Kasfa Ali    is working as Assistant Professor in Department of CSE, Muthoot Institute of Technology and Science, Kochi, India. She is currently pursuing PhD from School of Engineering, Cochin University of Science and Technology. She has a teaching experience of more than 8 years. She took her undergraduate degree in Computer Science and Engineering from Cochin University of Science and Technology and postgraduate degree from Mahatma Gandhi University. She has authored 2 international journals, 9 conference papers and 2 book chapters. Her current research interests include cryptography, network security, Internet of Things and blockchain. She can be contacted at email: fasilaka@cusat.ac.in.



Sheena Mathew    is working as Professor in Division of Computer Engineering, School of Engineering, Cochin University of Science and Technology. She completed her Ph.D. from Cochin University of Science and Technology, undergraduate degree from Madurai Kamaraj University and M.Tech from Indian Institute of Science, Bangalore. She has a teaching experience of 27 years. She has authored more than 50 publications in international journals and conferences. Her areas of interest include cryptography and network security. She can be contacted at email: sheenamathew@cusat.ac.in.