Journal of science faculty at Sirte University
JSFSU
2021
Peer-review Periodical Journal

جامعة سرت
Sirte University
1989

ISSN INTERNATIONAL STANDARD SERIAL NUMBER

eISSN: 2789-858X

SCIENTIFIC JOURNAL FOR THE FACULTY OF SCIENCE – SIRTE UNIVERSITY

SJFSSJ

DOI: 10.37375/issn.2789-858X - Indexed by Crossref, USA

VOLUME 3 ISSUE 2 OCTOBER 2023

Bi-annual, Peer- Reviewed, Indexed, and Open Accessed e-Journal

Legal Deposit Number@NationaL Library (Benghazi): 990/2021

1.02/2022

jsfsu@su.edu.ly
journal.su.edu.ly/index.php/JSFSU

Scientific Journal for the Faculty of Science-Sirte University Vol. 3, No. 2 (2023) 115-125

Scientific Journal for the Faculty of Science-Sirte University

Journal home page: http://journal.su.edu.ly/index.php/JSFSU/index

DOI: 10.37375/issn.2789-858X

Secure Key Exchange Using Boolean algebra: A New Method Based on NP-Hard Problem

Mohammed M. A. Albrkoli[1], Khamiss M. S. Ahmed[2], Aisha M. Alfitouri[3], Mahmmoud H. Alawan[4]

[1,4]*Network and communication Department, Information Technology Faculty, Sebha University-Libya.*

[2,3]*Computer Science Department, Information Technology Faculty, Sebha University -Libya.*

DOI: https://doi.org/10.37375/sjfssu.v3i2.1663

ARTICLE INFO:

Received: 03 September 2023

Accepted: 20 September 2023

Published: 26 October 2023

***Keywords*:** Key exchange, Boolean algebra, NP-hard problem, Cryptography, Security, Public-key cryptography; secure communication, Man-in-the-middle attack, Private Key, Shared key

A B S T R A C T

Secure key exchange is essential for maintaining the confidentiality and integrity of transmitted data in contemporary communication systems. To restrict unwanted access to the transferred keys, traditional key exchange techniques relied on computational complexity. Traditional approaches could be attacked, though, if modern computing resources become more powerful. This article suggests a novel method for secure key exchange based on NP-hardness and Boolean algebra. The method creates a public value from the private keys of two participants and other information, and each person then uses their own private key and the other public value to obtain the shared key. The fact that the private keys are not disclosed and that both users compute the secret key using their respective sets of private keys and values received from the other side makes the system resistant to man-in-the-middle attacks. The major goal of the suggested solution is to safely retrieve the same value as a shared key for both participants, even if others already know the public values. Boolean algebra and an NP-hard issue offer higher security assurances than conventional techniques that only consider computational complexity. The study uses a key size of 128 bits, which produces good results and offers higher security guarantees than conventional techniques. The study has also created a brand-new key exchange strategy that enhances current methods. Overall, this method marks a substantial advancement in the use of Boolean algebra and NP-hard issues to achieve secure key exchange.

1 Introduction

The development and extension of the network require a system to secure the transmitted information through the network. Cryptography can offer this service by making communication secure over the network (Stallings, W. 2017). So the main goal of implementing cryptography is to keep the network information communicating through a secure channel, so the stored information may be accessed without proper authorization, but it is still secure, and the secret data cannot be obtained even if the transmitted messages are able to be read.

Most of the methods use the echo of the complexity of that method, and that complexity is used to generate a coded text or message to hide the original text or message (Paar & Pelzl,. 2010). The way to use that method to generate a coded message is called a key.

There are many kinds of keys, and each one has a different technique to implement it, and the difference depends on how to use or achieve that key. One type of key is called the public key, which has many methods to make or generate that public key (Katz & Lindell. 2014); each method depends on the way it is used and

Corresponding author: **E-mail:** moh.albarkoli@sebhau.edu.ly

the properties of that method to encode the information and get secure communication over the network.

The Boolean algorithm is one of the methods used to generate a public key (Wu & Chen. 2017). The Boolean operations have properties that make it possible to get different values by implementing them.

The idea is to implement the Boolean operations on the data and get another new date; the new date is supposed to be different than the original data. That property of Boolean operations forces many people to study and implement it to generate a public key (Li, & Zhang. 2019).

Even though there are some ways to achieve key sharing, it is possible to find a new method to achieve the same goal, maybe with better performance (Diffie & Hellman. 1976). By using the Boolean properties, it is possible to make a new method to distribute the keys.

There is a problem called the NP-hard problem, and depending on that problem, it is possible to share keys in a secure way (Rivest, et. al. 1978).

This study proposes a new secure approach for key exchange using Boolean algebra based on an NP-hard problem. The proposed approach generates a shared key that is computationally difficult to determine, even if an attacker has access to the public values exchanged during the key exchange process (Singh & Babu . 2018). By leveraging the properties of Boolean algebra and an NP-hard problem, this approach provides stronger security guarantees compared to traditional methods that rely solely on computational complexity. The proposed study presents a novel approach to key exchange that improves upon existing techniques (Yizhi H, et. al, 2023). This approach represents a significant step forward in achieving secure key exchange in modern communication systems. However, this study can be used as an alternative to other key-sharing methods, such as Diffie-Hellman, in a faster way (Vetrivelan & Padmavathi, 2019).

2 Previous Studies

A technique known as cryptography is used to ensure the security and protection of data or information while it is transferred and exchanged over a network, or to make it invisible to others, possibly because the owner of the data or information does not want it to be seen or because it contains secret information. Emails, credit card numbers, websites, and a variety of other items are protected using cryptography (K. Ahmed, et. al, 2023).

Information can be concealed via cryptography, and when concealed information is communicated through a network, nobody can decipher what it implies. (Stallings, W. 2017).

By encrypting the data to "make it non visible or non-understandable" or to transform the contents of a message into a format that cannot be read," As stated by the sender, and decrypting it to "make the data understandable" or to transform the message back into a readable format," as stated by the receiver, cryptography provides this service. When a communication is encrypted, it is known as cipher text, and when it is decrypted, it is known as plaintext (Paa & Pelzl . 2010).

The original text communication is known as the plaintext. Cipher text is the encrypted message created by employing the algorithm and secret key on the plaintext message. (Diffie & Hellman.1976).

There are many different types of keys that can be used for both encryption and decryption. A variety of cryptography systems are categorized. Depending on how the key(s) are used.

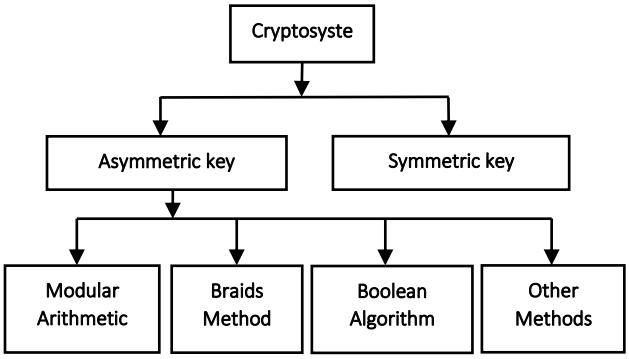

Figure (1). Types of keys

Certainly! The study of cryptography focuses on the methods and formulas used to protect data transfer and communication. The purpose of cryptography is to protect data's secrecy, integrity, and authenticity while it is being stored in a system or transferred over a network (Paar, et. al, 2010).

The original message, known as plaintext, is converted into an unintelligible format, known as cipher text, using a variety of encryption techniques. Symmetric-key encryption and public-key encryption are the two

encryption methods that are most frequently employed (Katz & Lindell. 2014)

The same secret key is employed in symmetric-key encryption for both encryption and decryption. This implies that secure communication requires a shared key between the sender and the recipient. Advanced Encryption Standard (AES) and Data Encryption Standard (DES) are the two most widely used symmetric-key encryption methods (Schneier, B. 1996).

Two separate keys are used for encryption and decryption in public-key encryption, sometimes referred to as asymmetric encryption. The private key is used to decode data, whereas the public key is used to encrypt data. RSA is the most widely used public-key encryption algorithm (Menezes, et. Al. 2010)

Numerous applications, such as secure online transactions, email encryption, secure messaging services, and secure communication networks, utilise cryptography. In order to guarantee the integrity and authenticity of digital documents, it is also utilized in digital signatures (Ferguson, et. al, 2010)

Overall, cryptography is essential for protecting communication and data transmission in the current digital era, and it is ever-evolving as new dangers and weaknesses are discovered.

2.1. Encryption and Decryption

2.1.1 Symmetric key

The sender encrypts the information using a certain key, and the receiver decrypts the encrypted information using the same key in a symmetric key system. Both "sender" and "receiver" must be aware of this key.

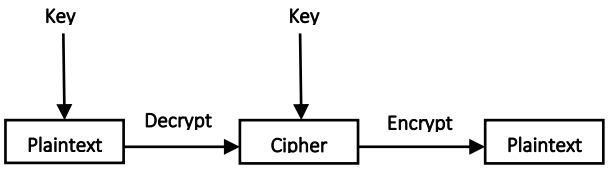

Figure (2). Symmetric key

Let's say sender A wishes to send a message to receiver B. Sender A would use a key to encrypt the message and convert it to cipher text before sending it to recipient B via the network. This message is delivered as a cipher text to the receiver B side.

To obtain the original message or plain text, recipient B should decrypt it using the same key that the sender used, (Buchmann, J. 2001).

2.1.2 Public Key or Asymmetric Key

This it uses two keys, a public key and a private key. Data encryption using public key cryptography is effective. Anyone sending a message can use the recipient's public key to encrypt the message and make the recipient's public key available. If a recipient has a private key, can share it with other people so can send those messages. Everyone should be aware of this key in order to be able to send messages to the owner of the public key by using it to encrypt the message and convert it to cipher text. Only the owner of the public key can then decrypt the message. The issue with public key systems is that in order to encrypt a message using the recipient's public key. As shown in the figure (3), (Schneier, B. 1996).

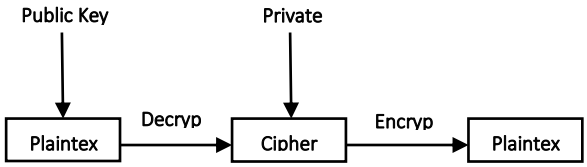

Figure (3). Public key or asymmetric key

Only the owner of the private key may determine the meaning of the message that has been given, making the private key a secret key. To recover the original communication's "plaintext," the owner of this key can decrypt a message that has been encrypted with a public key (Mao, W. 2003).

Sender A wants to communicate with recipient B. By utilizing the recipient's public key, A should encrypt the message and convert it to cipher text. Then deliver it through the network to receiver B. This message is delivered as a cipher text to the receiving side. In order to decrypt it and obtain the message's plaintext, recipient B should use his or her owner private key (Menezes, et. al, 1996).

It is impossible to know the private key if you know the public key since the public key can only be used to encrypt messages and only the associated private key can be used to decrypt those messages.

2.2 The Distribution of the Keys

The distribution of keys is one of the encryption's key functions. The use of public key cryptography has two components. The secret keys are distributed via the public key distribution, not the other way around. Numerous methods have been suggested for the keys distribution in order to accomplish this.

A. The Secret Key Exchange

There are numerous issues with sharing and exchanging secret keys, which is one of the two causes for the invention of public key cryptography.

To help us examine conventional methods of distributing secret keys as well as usual issues with secret key distribution, let's reintroduce Alice and Bob and their children, Casey and Dawn. Currently retired, Alice and Bob are residing in the upper Midwest. Casey, their son, has taken over the California Company. (Stallings, W. 2013)

The year DES became a standard was 1977. To describe common issues with secret key distribution as well as the conventional methods of doing so. Let's say Alice and Bob need to communicate each other some info. In this situation, Alice and Bob must create their own DES secret key. Bob communicates with Alice via DES.

Casey wants to send something to Bob on behalf of another individual. Bob is asked for his private key as a result. Bob will receive the messages between them if he uses the same DES that he did with Alice. To provide Casey, Bob creates another DES secret key (Ferguson, et. al, 2015).

B. The Problem and the Traditional Solution

How are Casey and Alice able to give him the secret key? In other words, how can it be sent to him? They are unable to guarantee the secret key's security during transportation; hence they are unable to send it through mail. The secret key might be given to Casey by Alice and Bob in person or by a reliable courier. This approach is outdated. If only the three users Casey, Bob, and Alice know the secret key. So that they can pass messages back and forth (Bellare, et. al, 2015)

What if Dawn also requests a private key? She requests that Alice and Bob create a new secret key for her because she does not want to use the one that the others share with Casey. Therefore, Alice (or Bob) must give Dawn a secret key.

The three secret keys that Alice and Bob now possess are one that they use exclusively, one that they share with Casey, and one that they also share with Dawn (Paar, et. al, 2016).

2.3 Boolean Algorithm

George Boole (1815–1864) created the Boolean logic, which is typically used to fine-tune the determination of system status or to set or clear certain bits. A technique to compare individual bits is Boolean logic. It can choose the 'operators' to specify how the bits are compared and the outcome. Boxes with several inputs and a single output are preferred by operators. The result will either be 0 or 1.The operations that are used the most frequently are AND operation", OR operation", and NOT operation" (Boneh, & Shoup, 2017).

The main concept is to incorporate Boolean operation expressions into the cryptosystem for encryption and decryption. The security of this algorithm depends on how challenging it is to find a task that satisfies the provided set of Boolean expressions (Li, & Zhang, 2020). This algorithm also involves security concerns and the use of public key cryptosystems.

The Boolean permutation is used in this work to create the public key cryptosystem. The Boolean permutation's inversion forms the foundation of this method's security (Singh, & Babu. 2021). The Boolean permutations have a variety of characteristics that can be employed in cryptography to ensure security.

If the following conditions are met, the Boolean permutation can be used to generate the public key:
With some expertise, finding the inverse of the Boolean permutation is simple (Katz, & Lindell. 2014).

It is computationally impossible to obtain the inverse of the permutation without prior knowledge.

The trapdoor is a special piece of information that must be understood in order to determine the inverse; one method for creating a Boolean permutation entails using a composition of Boolean permutations.

3 Materials and Methods

3.1 NP-Problem

A problem is called a NP (nondeterministic polynomial time) class if it is can be solved in polynomial time by a nondeterministic Turing machine.

The NP-complete problems are the most difficult problems in NP, that is because if one could find such way to solve any NP-complete problem quickly (in polynomial time), then it will be possible to use that algorithm to solve all NP problems quickly (Goldreich, O. 2017).

One example of an NP-complete problem is the Boolean satisfiability problem.

3.2 Boolean Satisfiability Problem

The Boolean satisfiability problem (SAT) is a decision problem considered in complexity theory. A formula of the problem is a Boolean expression written using only AND, OR, NOT, variables. The purpose was given the expression, is there some assignment of TRUE and FALSE values to the variables that will make the entire expression true (Zeng, & Yu. 2020)

Mathematically, a formula of propositional logic is said to be satisfiable if logical values can be assigned to its variables in a way that makes the formula true. The class of satisfiable propositional formulas is NP-complete (Boneh, & Shoup. 2015).

The problem can be significantly restricted while still remaining NP-complete.
By applying De Morgan's laws, assume that NOT operators are only applied directly to variables, not expressions; we refer to either a variable or its negation as a literal. For example, both x1 and not(x2) are literals, the first a positive literal and the second a negative literal. Together a group of literals, get a clause, such as (x1 or not(x2)). Finally, let us consider formulas that are a conjunction (AND) of clauses. This called conjunctive normal form. Determining whether a formula in this form is satisfiable is still NP-complete, even if each clause is limited to at most three literals. This last problem is called 3SAT, 3CNFSAT, or 3-satisfiability (Katz, & Lindell. 2019).

On the other hand, restrict each clause to at most two literals, the resulting problem, and 2SAT, is NL-complete. Alternately, if every clause must be a Horn clause, containing at most one positive literal, the resulting problem, Horn-satisfiability, is P-complete,

3.2.1 Complexity

SAT is considered a NP-complete. As proved by Stephen Cook in, the issue of an NP-complete problem did not even exist (Rogaway, P. 2015). The problem remains NP-complete even if all expressions are written in conjunctive normal form with 3 variables per clause (3-CNF), yielding the 3SAT problem. In this case the expression has the form:

(x11 OR x12 OR x13) AND (x21 OR x22 OR x23) AND (x31 OR x32 OR x33) AND...

When each x is a variable and each variable can appear many times in the expression.
A good useful property is that it preserves the number of accepting answers.

3.2.2 Satisfiability

It is a special case of k-satisfiability (k-SAT). When each clause contains at 3 variables.

E = (x1 or ~x2 or ~x3) and (x1 or x2 or x4) and (x1 or x3 or~ x4)
E has three clauses, four literals (x1, x2, x3, x4), and k=3 (three variables per clause).

Here to get a solution to this equation of the decision problem must determine whether there is a truth values (TRUE or FALSE) that can assign to the variables (x1 through x4) such that the entire expression is TRUE. For example, the next values to the x's variables (x1 = TRUE, x2 = TRUE, x3=TRUE, x4=TRUE), so the answer in this case is YES. This is one of many possible assignments (Canetti, R. 2016). Any set includes at least one TRUE value, it is enough to get YES answer. If there were no such assignment(s), the answer would be NO.

3.3 Key Sharing Using Boolean Satisfiability Problem

Depending on the Boolean satisfiability problem NP hard problem proprieties, and the main issue of the key exchanging, it is possible to achieve new way for key exchange the keys using the Boolean satisfiability problem, In this case it is possible to use 3- satisfiability to achieve key exchanging, Because of the properties

'as it shown prior ', so it allows to exchange the values is secret way.

Let to assume Alice and Bob attempt to exchange a secret key (Boneh, & Shoup. 2019). First of all, both of the users have to choose an integer random privet value. Alice chooses a, and Bob chooses b. and both of them agree on the same public value z. Then they compute their public values using their private values and the public value z. And these values should be 128 bit to make it secure enough against the attackers Lindell, (Y., & Katz. 2019).

The following steps explain how to get the shared key between Alice and Bob so first of all:

Alice generates his public value as:	Bob generate his public value as:
$A_1= (a_1 \vee a_2' \wedge z_1)$	$B_1 = (b_1 \vee b_2' \wedge z_1)$
$A_2= (a_2 \vee a_3' \wedge z_2)$	$B_2 = (b_2 \vee b_3' \wedge z_2)$
.	.
$A_{128}= (a_{128} \vee a_1' \wedge z_{128})$	$B_{128}= (b_{128} \vee b_1' \wedge z_{128})$

Then they exchange the generated public values.
Finally, Alice and Bob compute the same secrete key k. as shown it the following:

Alice receives the Bob's public key and computes k:	Bob receives the Alice's public key and computes k:
$K_1 = (b_1 \vee b_2' \wedge z_1) \wedge (a_1 \vee a_2')$	$K_1= (a_1 \vee a_2' \wedge z_1) \wedge (b_1 \vee b_2')$
$K_2 = (b_2 \vee b_3' \wedge z_2) \wedge (a_2 \vee a_3')$	$K_2= (a_2 \vee a_3' \wedge z_2) \wedge (b_2 \vee b_3')$
.	.
.	.
$K_{128}= (b_{128} \vee b_1' \wedge z_{128}) \wedge (a_{128} \vee a_1')$	$K_{128}= (a_{128} \vee a_1' \wedge z_{128}) \wedge (a_{128} \vee b_1')$

In this case both of Alice and Bob can get the same secret key. For example:

Alice gets

$$K_1 = (b_1 \vee b_2' \wedge z_1) \wedge a_1 \vee a_2' = (b_1 \vee b_2') \wedge z_1 \wedge (a_1 \vee a_2')$$

Bob gets

$$K_1= (a_1 \vee a_2' \wedge z_1) \wedge b_1 \vee b_2' = (a_1 \vee a_2') \wedge z_1 \wedge (b_1 \vee b_2')$$

This algorithm is secure against man in the middle attack. Both of Alice and Bob operate some Boolean operation on privet keys and send to each other's, so in this case the private keys are not shown. Also both of them compute the secret key from the privet keys and the received values from the other side, and the attackers cannot drive the secret key using the public values. That means the only way to get the secret key is getting the privet keys, which are not known.

3.4 The Proposed Framework

The goal of the proposed approach is to build a shared key from private keys and other information. Therefore, the suggested system only takes such values into account. To make the system easier to describe, the suggested technique employed the names Bob for the second participant and Alice for the first participant, as shown in figure (4).

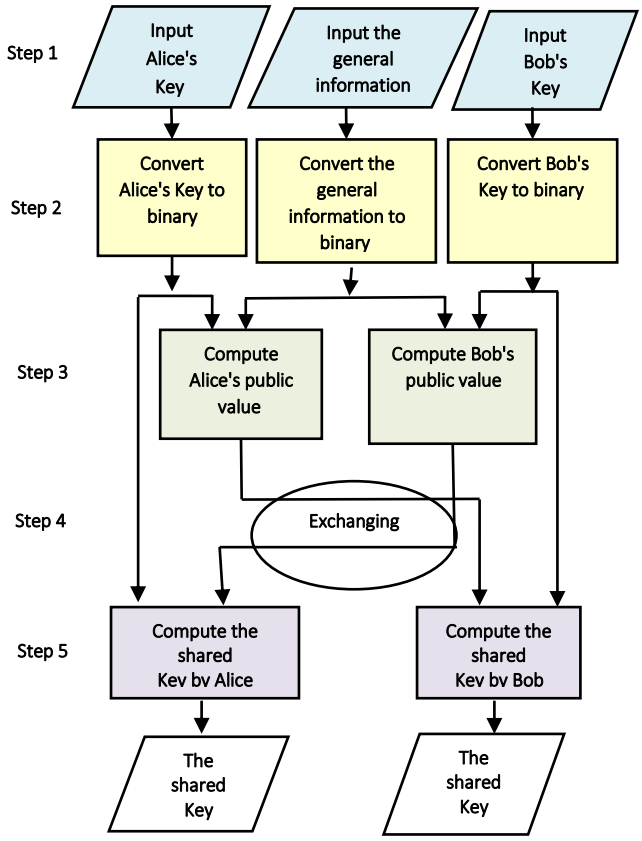

Figure (4). Flowchart to explain the proposed framework

The following steps explain clearly the processes that are followed in the system.

Step 1: First, entries such as Alice's private key, Bob's private key, and general information are sent to the system. Enter numbers, letters, or other characters. All activities are carried out using Boolean operations because the system is built on Boolean algebra, and as a result, all entries must be in binary format.

Step 2: All entries should be in binary format because the system's Boolean algebraic design requires it. The system transforms the entries into binary arrays in this stage to enable the Boolean operations. By submitting the entries to a conversion function, the system transforms them into binary format. The binary arrays are sent to the main programmer by the conversion function once it has converted the entries to binary format.

Step 3: The public values of both Alice and Bob are generated by the programmer in this stage using their respective private keys and the general data. In the beginning, the system determines Alice's public value by submitting his private key and other data to a function that determines public values. In this instance, it calculates Alice's public value before sending it to the primary programmer. The system next computes Bob's public value by providing his private key and other data to the same function that computed the public values in step two. In this instance, it calculates Bob's public value before sending it to the primary programmer.

3.5 The Project Functions

To accomplish the system's goals, some functions are created. To be specific, AND, OR, and NOT are the Boolean operations employed in this project. When the program receives the data—"the first participant private key, the second participant private key, and the general information"—it transforms the data to binary format and begins to carry out the functions. The first function computes the function for public values.

3.5.1 Computing the Public Values Function

This function, which is used in step 3, takes the data as binary arrays and computes the public values for each participant independently. In order to compute the first participant's public key, it must first receive both the general information and the first participant's private key, both of which are in binary format. The second participant's public key is then calculated after the second participant's private key and general information (both in binary format) are received.
Furthermore, the public value of each participant is determined independently. The main programmer then receives those public values and exchanges them in order to determine the shared key.

4 Results and discussion

The system design and the techniques employed to achieve the desired outcomes. In order to help readers have a deeper knowledge of the methods, the system's execution, and its outcomes, two issues with the proposed system are also highlighted: security and key size. There has also been a quick comparison between the offered methods and Daffier-Hellman.

4.1 Security of the Proposed Method

4.1.1 Key Size: Greater security is associated with larger keys. Key sizes of 128 bits or more are now the norm, and those of 64 bits or less are no longer regarded as sufficient. The key size in the suggested approach is 128 bits, which is deemed secure enough to withstand a brute force assault.

A cryptographic technique can be discovered through a brute force assault, which involves testing numerous options. In this instance, the attacker must attempt each potential key in order to find the shared key. Therefore, there is a 212 percent chance of a brute-force attack, which is currently seen to be secure enough.
In order to obtain the shared key, the attacker must attempt every key that might be used with one of the computed public values. The key size utilized is 128 bits, which means 2128 potential combinations.

4.1.2 The Design of the Equation: The Boolean satisfiability problem (SAT), a decision issue taken into consideration by complexity theory, was used in the construction of the functions. An illustration of an NP-complete problem is the Boolean satisfiability problem. The most challenging NP problems are those that are NP-complete because, if an algorithm could be developed to solve any NP-complete problem rapidly, it would then be easy to apply it to all NP problems. As a result, the equation is thought to be secure because it was built on an NP-complete problem.

Furthermore, even though the attacker has access to the general information, the first participant's private key, and the second participant's private key in the proposed method, they cannot compute the shared key because it is necessary to know the other participant's private key in order to obtain the shared key using one participant's public value. Which nobody else than the key holder is aware of.

4.1.3 The Randomness Checking: The equation should be designed with a few criteria in mind. Attempt to construct an equation with a reasonable avalanche probability, which states that a change in one bit of the input should result in a change in many bits of the output; a stricter version of this is the severe avalanche criteria (SAC). This specifies that any modification to any bit of the input should modify the probability of the output by a factor of two.
This requirement was attempted to be kept in our method, and the proposed approach has been evaluated to determine whether or not it is.

First, the second participant's private key was fixed and the general information. Next, the first participant's private key by flipping a bit in the first participant's private, and implemented the algorithm 128 times "the key size" while flipping a different bit in the first participant's private key each time. The number in use is:
10101010101010101010101010101010101010101010101010
10101010101010101010101010101010101010101010101010
10101010101010101010101010

Following the implementation of the method, the first result was compared to the other results, and the following graph was created as shown in Figure (5), depending on the quantity of altered bits.

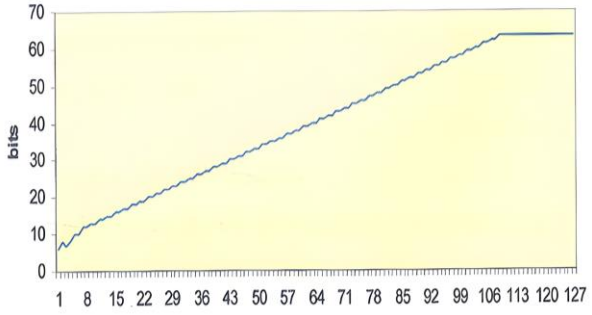

Figure (5). Distribution of the changing between the first result and the other results

Secondly, the first participant's and the second participant's private keys, assigned the same previous number to all three entries, and changed the general information by flipping a bit in it. Finally, implemented the algorithm 128 times "the key size" while flipping a different bit in the general information each time. Following the implementation of the method, the first result was compared to the other outcomes, and the following graph was created as shown in Figure (6), depending on the quantity of altered bits.

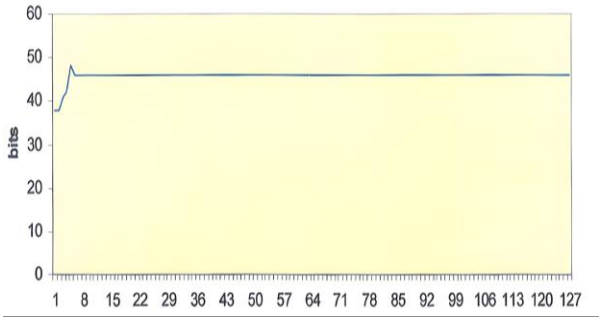

Figure (6). Distribution of the changing between the first result and the other results

The results for the two previous comparisons should be roughly 64, which is in the middle of the key size. But as evidenced, some adjustments need to be made to improve the outcomes.

4.2 Comparison with Diffie-Hellman

In some cases, a comparison between the proposed approach and Diffie-Hellman has been conducted.

4.2.1 Key Size: The key size in the Diffie-Hellman protocol is 256 bits, however in our suggested solution it is only 128 bits, which has become a standard size.

4.2.2 Running time: A variety of key sizes, including 46 bits, 128 bits, 256 bits, and 1024 bits, have been evaluated for both approaches' running times in the comparison. The test has been run numerous times for each key, the results average has been calculated, and the graph displayed in Figure (7) is then generated based on the results.

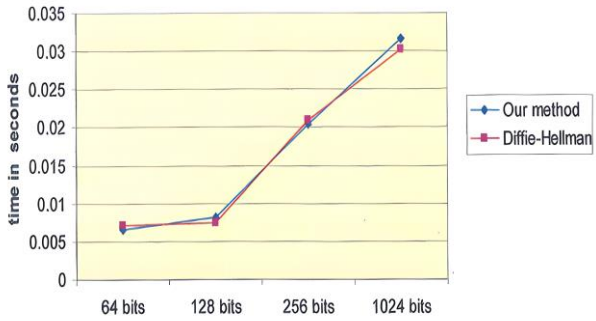

Figure (7). Comparing the running time with Diffie-Hellman

It is evident from the graph that there is not a significant difference between the running times of the two approaches. Both this approach and Diffie-Hellman can be faster on occasion.

However, so far, the key size for our suggested approach is 128 bits, whereas the key size for Diffie-Hellman is 256 bits. As can be seen in the graph, our method takes less time to run with a key size of 128 bits than Diffie-Hellman does with a key size of 256 bits. Therefore, the suggested method is thought to be faster than Diffie-Hellman.

4.2.3 Security: If p and g are properly selected, the Diffie-Hellman protocol is thought to be safe against attackers. To obtain gab, the assailants must be able to resolve the Diffie-Hellman puzzle. And that is regarded as challenging. The secret integers a and b are eliminated at the end of the session. Therefore, Diffie-Hellman key exchange alone provides security and effectively conceals the private keys.

The proposed method is thought to be secure because, in order to obtain the shared key, one must first know the private keys, which are impossible to obtain due to the Boolean satisfiability problem (SAT) concept. Even with knowledge of the computed public values, however, it is still impossible to obtain the shared key because no private keying material is available to be revealed.

The major objective is to introduce a novel Boolean algebraic technique for key exchange. This project meets the main conditions, which are: getting the same value for a shared key, by both the two participants, and obtaining that securely, i.e., even if the others acquired the public values, it is impossible to compute the shared key. According to the NP-hard problem, the security of this solution is attained.

Through communication channels, the cryptosystems are utilized to secure network information. Most cryptographic techniques use an echo of the technique's complexity, which is then used to produce a coded word or message that conceals the data. The major applications of a cryptosystem are key exchange, digital signatures, and encryption decryption.

These protocols make use of the NP-Hard problem difficulty to guarantee the security of the key exchange. In a Li and Zhang (2019) investigation, based on the Boolean satisfiability problem (SAT), the protocol for safe key exchange is suggested in this study. The two parties first create a set of random variables, which is how the protocol operates. They then create a SAT problem using these variables. Along with another Li and Zhang paper (2020), Based on the SAT problem, this study suggests an enhanced, secure key exchange system. The technique has been improved, using fewer random variables overall.

And another study by Singh and Babu (2018), Based on the knapsack problem (KP), a secure key exchange technique is suggested. Based on the KP, this study suggests an improved secure key exchange technique. The protocol has been improved in that it uses a KP problem-solving algorithm that is more effective.

Also the research by Vetrivel and Padmavathi (2019), Based on the graph coloring problem (GCP), a secure key exchange technique is suggested in this study. They create a GCP problem using these figures. The shared key is the answer to this GCP issue.

However, the complexity of the underlying NP-hard problem determines how secure the protocols will be.The effectiveness of the protocols depend on both numbers of random variables that used and the algorithm used to solve the underlying NP-Hard issue. For instance, the protocol proposed by Singh and Babu (2021) is a viable choice if the application requires a modest key size. If the application requires a high level of security, the protocol proposed by Li and Zhang (2019) is a possible choice.

The main objective of the proposed study is to safely get the same value for both participants as a shared key, even if others already know the public values. Higher security guarantees are provided by Boolean algebra and an NP-hard problem than by traditional methods that merely take into account computational complexity. The study used a 128-bit key size, which yields good results and provides greater security guarantees than traditional methods. The work has also produced a novel key exchange scheme that improves on existing techniques. Overall, this approach represents a significant improvement in the field of secure key exchange using Boolean algebra and NP-hard problems.

5 Conclusions and Recommendations

5.1 conclusions

The use of Boolean operations in cryptosystems, such as Boolean algebra to produce the public key and Boolean permutation, has been studied in considerable detail. A NP-hard problem, of which there are several different

varieties, has been employed in suggested methodology to create key exchange. One of these problems, the Boolean satisfiability problem, has been used. A Boolean expression that simply uses the variables AND, OR, and NOT serves as the solution to this problem. It is intended that when the expression is presented, the variables will have had their TRUE and FALSE values assigned, making the entire expression true. An approach has been suggested based on this notion, and the system has been created using that approach.

The suggested system utilizes the general knowledge and private keys of two participants to produce a public value. Following that, each participant uses his or her own private key along with the other public value to drive the shared key. Because the private keys are not displayed, this technique is regarded as being secure against man-in-the-middle attacks. Both of them compute the secret key using the private keys and the values they have received from the other side, making it impossible for attackers to drive the secret key using the public values. Therefore, obtaining the private keys, which are unknown, is the only method to obtain the secret key. The project is implemented, and positive outcomes are attained. Using a key with 128 bits makes the key secure; in fact, it is a size that is widespread at the moment.

5.2 Recommendations

The method can be enhanced in a few ways, such as getting a better equation to achieve better results and making it so that it has a good avalanche probability, which means that any change to the input should change the probability of the output by "half" if it occurs.

According to the NP-hard problem, it is conceivable to implement key exchange for more than only the two portions of this study. This paper focuses on key exchange based on the NP-hard issue, which is intriguing due to its difficulty and lack of usage in cryptosystems. As a result, this issue can be used to future work on digital signatures and encryption decryption.

The major goal of the suggested study is to use a shared key to accomplish the same value for both participants while remaining safe, even if others are aware of the public values. Better security guarantees are offered by NP-hard issues and Boolean algebra than by more conventional approaches that only consider computational complexity. The study's key size was 128 bits, which yields good results and provides greater security assurances than traditional methods. The work also led to the creation of a brand-new key exchange strategy that improves upon existing techniques. This approach represents a significant advancement in terms of secure key exchange using NP-hard problems and Boolean algebra.

Conflict of Interest: The authors declare that there are no conflicts of interest.

References

Bellare, M., Rogaway, P., & Stepanovs, I. (2015). The universal composability (UC) security framework. Foundations and Trends® in Theoretical Computer Science, 10(1-2), 1-239.

Boneh, D., & Shoup, V. (2019). A graduate course in applied cryptography (updated version). https://toc.cryptobook.us

Boneh, D., & Shoup, V. (2015). A graduate course in applied cryptography. https://toc.cryptobook.us

Boneh, D., & Shoup, V. (2017). A graduate course in applied cryptography. https://toc.cryptobook.us

Buchmann, J. (2001). Introduction to cryptography (2nd Ed.). Springer.

Canetti, R. (2016). Universally composable security: A new paradigm for cryptographic protocols. Foundations and Trends® in Theoretical Computer Science, 10(1-2), 1-239.

Diffie, W., & Hellman, M. (1976). New directions in cryptography. IEEE Transactions on Information Theory,

Rivest, R. L., Shamir, A., & Adleman, L. M. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

Ferguson, N., Schneier, B., & Kohno, T. (2010). Cryptography engineering: design principles and practical applications. John Wiley & Sons.

Ferguson, N., Schneier, B., & Kohno, T. (2015). Cryptography engineering: Design principles and practical applications. Wiley.

Goldreich, O. (2017). Foundations of cryptography: Volume 2, basic applications. Cambridge University Press.

K. Ahmed, S. Pal & R. Mohan (2023) A review of the tropical approach in cryptography, Cryptologia, 47:1, 63-87, DOI:10.1080/01611194.2021.1994486.

Katz, J., & Lindell, Y. (2019). Introduction to modern cryptography (3rd Ed.). CRC Press.

Katz, J., & Lindell, Y. (2014). Introduction to modern cryptography (2nd Ed.). Chapman and Hall/CRC.

Li, R., & Zhang, K. (2019). Secure Key Exchange Protocol Based on Boolean algebra and NP-Hard Problem. In 2019 IEEE 9th Annual Computing and

Communication Workshop and Conference (CCWC) (pp. 0575-0580).

Li, R., & Zhang, K. (2020). An Efficient Secure Key Exchange Scheme Based on Boolean Algebra and NP-Hard Problem. In International Conference on Computing, Networking and Communications (ICNC) (pp. 269-273).

Lindell, Y., & Katz, J. (2019). Modern cryptography, probabilistic proofs and pseudo randomness (2nd Ed.). Cambridge University Press.

Mao, W. (2003). Modern cryptography: Theory and practice. Prentice Hall.

Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of applied cryptography. CRC press.

Paar, C., Pelzl, J., & Preneel, B. (2016). Understanding cryptography: A textbook for students and practitioners. Springer.

Paar, C., & Pelzl, J. (2010). Understanding cryptography: A textbook for students and practitioners. Springer.

Ristenpart, T., Shrimpton, T., & Shrimpton, E. (2016). Foundations of cryptography: Volume 1, basic tools. Cambridge University Press.

Rogaway, P. (2015). The science of cryptography. Cryptology ePrint Archive, Report 2015/067. http://eprint.iacr.org/2015/067.

Schneier, B. (1996). Applied cryptography: Protocols, algorithms, and source code in C (2nd Ed.). Wiley.

Singh, T., & Babu, M. S. P. (2018). Secure Key Exchange Using Boolean algebra and NP-Hard Problem. In 2018 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-5).

Singh, T., & Babu, M. S. P. (2021). Enhanced Secure Key Exchange Using Boolean algebra and NP-Hard Problem. In 2021 International Conference on Communication and Signal Processing (ICCSP) (pp. 1055-1059).

Stallings, W. (2017). Cryptography and network security: Principles and practice (7th Ed.). Pearson Education.

Stallings, W. (2013). Cryptography and network security: Principles and practice (6th Ed.). Pearson Education.

Vetrivelan, S., & Padmavathi, G. (2019). Secure Key Exchange by Using Boolean algebra and NP-Hard Problem. International Journal of Advanced Research in Computer Science, 10(1), 13-18.

Wu, T., & Chen, X. (2017). A novel public-key cryptosystem based on Boolean function. Journal of Communications, 12(3), 166-171.

Yoshi H., Rahul I., and Hanlin R. (2023), NP-Hardness of Approximating Meta-Complexity: A Cryptographic Approach, 55th Annual ACM Symposium on Theory of Computing, DOI: 10.1145/3564246.3585154.

Zeng, X., & Yu, C. (2020). Secure Key Exchange Protocol Based on Boolean algebra and NP-Hard Problem. In 2020 International Conference on Cybersecurity and Protection (ICCS&P) (pp. 1-5).

SCIENTIFIC JOURNAL FOR THE FACULTY OF SCIENCE – SIRTE UNIVERSITY

Scientific Journal Impact Factor

TOGETHER WE REACH THE GOAL

jsfsu@su.edu.ly

journal.su.edu.ly/index.php/JSFSU