Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

DATA PROTECTION IN D2D COMMUNICATION USING LIGHTWEIGHT CRYPTOGRAPHIC PROTOCOL

AJITH KUMAR V [1] and Dr. K SATYANARAYAN REDDY[2]

[1]Research Scholar, Research Resource Centre, Visvesvaraya Technological University (VTU), Belagavi, Karnataka State, India, ajith.it@gmail.com
[2]Department of Information Science and Engineering, Cambridge Institute of Technology, Bangalore, Karnataka State, India, ksatyanreddy@yahoo.com

Abstract
In today's world of paradigm device computing, device communication plays an important role in the secure transmission of data. Data security is directly linked between devices without an intermediate node. Since the devices are battery-driven and are located in a different geographical area, it is very important to ensure secure communication. Considering 5G IOT based environment as one of the use cases for D2D Communication, to address security challenges in this kind of environment, it is essential to develop a lightweight security system to ensure confidentiality and data origin authentication. In this paper, we proposed a lightweight cryptographic technique capable of providing a solution for enhanced data security along with data origin authentication. The proposed method uses a standard Blowfish algorithm to derive a 128-bit secret key, a secure hash algorithm (SHA1) to generate two separate secret keys in which one specific derived key is used for data confidentiality and another for data origin authentication. The hash-based message authentication code (HMAC) algorithm is used to generate an integrity tag. Derived keys are generated using randomly selected secret key characters and random numbers. Experimental results show that the proposed scheme consumes low computational power while drifting keys, enhancing data security and resistance against key guessing attacks.

Index Terms — Confidentiality, Device to Device communication, Data Integrity, Data origin authentication, Key derivation, Key exchange, Lightweight protocol.

1 INTRODUCTION

The presence of a vast number of devices around the globe requires heterogeneous networks to establish data communication between them. Recently, device-to-device communication (D2D) has gained interest in data transmission but is vulnerable to different attacks during the key exchange process between end-user equipment (UE). In D2D communication, direct data exchange between the UE is carried out without the help of cellular or other network infrastructures. This open communication creates opportunities for malicious intruders to gain access to data and pose major security threats. It is therefore necessary to devise a secure key generation and exchange

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

mechanism capable of mitigating any network attacks aimed at data manipulation. Even though various cryptographic algorithms have been developed, maintaining end-to-end security is a difficult task. In many heterogeneous networks for data communication, millions of different types of devices or end-user equipment (UE) form the network entities. A significant advantage of D2D is the increased anonymity and privacy of the content since the information is not stored at the central location.

The derived benefits of D2D communication include spectrum sharing and higher throughput. Device-to-Device (D2D) communication often requires pairing of participating devices, synchronization, and data transfer between these devices. D2D communication facilitates the direct sharing of data between closely positioned devices. Secure communication involves key factors such as confidentiality, non-repudiation, mutual authentication, and robustness to withstand different cryptographic attacks. Devices can communicate in two modes, namely peer-to-peer and group modes. In peer-to-peer communication, one device communicates with a specific device while many devices participate and communicate in group mode using a common service to achieve a single goal. Key Management (KM) is a major challenge for achieving secure device-to-device communication. The Key Management scheme for D2D communication is referred to as Peer-to-Peer Key Management (PKM) scheme. This work focuses on the security layer of PKM schemes especially the encryption and decryption portion. The major objective of the proposed work is to design a lightweight and robust encryption technique to mitigate key guessing attacks. The symmetric encryption technique, which is time-efficient and energy-efficient compared to the asymmetric method, is used to generate a secure key.

2 Related Works

Various strategic design approaches reported in literature for secure data encryption have been reviewed in this section. Heterogeneous networks such as Wireless Sensor Network (WSN), Internet of Things (IoT), and Wireless Body Area Network (WBAN) have tiny devices with multiple resource limitations. These limited resource devices known as nodes have limited battery, memory, buffer, and computing capabilities. Energy will also be rapidly exhausted due to the execution of complex cryptographic algorithms and the device will soon become non-functional. In this context, some of the important works have been discussed below. Initially, various research issues related to D2D communication such as user mobility, D2D synchronization, device discovery, interference management, resource allocation, and security [1] have been discussed in detail. D2D communication is highly susceptible and vulnerable to several cryptographic, and network attacks [1].

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

KaziMasumSadique et al. [2] pointed out that it is highly desirable to develop and implement lightweight, faster security algorithms to ensure the continued presence of devices, and hence this open issue needs to be pursued to ensure data protection for both device and network entities.

Ankur Gupta et al. [3] proposed a novel lightweight and efficient mutual authentication and key agreement protocol by applying simple EX-OR and SHA-256 hashing functions in WBAN to overcome the problem of intermediate node attacks. Results show that it is resilient to most common attacks. The storage requirement is relatively mid-level while for the authentication and key agreement phase, the computation and communication costs are still higher than some of the counterpart algorithms.

Rama Krishna et al. [4] presented a new hybrid crypto- graphic system based on Huffman and binary coding techniques to achieve higher efficiency in block encryption and covert key agreement protocols that could improve the security of the color image transmitted over the D2D multi-hop communication channel. The proposed approach has been robust, lossless and can manage network traffic either by operating independently or using the existing communication infrastructure. Speed for algorithm encryption and decryption is very fast and accurate due to computing based on binary format.

Design of a secure certificate-less authentication scheme for D2D communication [5] was proposed based on a symmetric key method where partial private keys were generated separately at base station and UE. This study assumes that only small changes need to be made on the sender side during the key upgrade process, while the UE decryption information remains the same after the validation of the devices.

Proposed Shamir's (t, n) secret sharing schemes improved D2D-based secure data transmission protocol (SDTP) to im- prove the availability and efficiency of M-Health [6] with a sender-side delay of 184.02 milliseconds, access point of 25.52 milliseconds, and n-message delay of 12.76 (n+1) with a gross communication delay of 102 milliseconds.

Payal P. Tayade and PerumalVijayakumar [7] have proposed an additional security system in the SeDS protocol system to establish secure communication between end node and gateway for D2D communication in cellular LTE-A network. ByoungjinSeok et al. [8] proposed secure D2D communication based on lightweight cryptography for the 5G IoT network. The lightweight ciphers used are cryptosystem ECC- based public key (256-bit key) and the lightweight AEAD cipher. Because 5G IoT networks have limited resources, good performance of cryptographic algorithms may not cover all 5G IoT devices.

Aiqing Zhang et al. [9] proposed a lightweight and robust security-aware (LRSA) D2D-assist data transmission protocol for M-Health systems using a less generalized

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

certificate-use lower-public key cryptography (CLPKC) certificate, which is time-consuming and has low computational efficiency. They also proposed a new, more efficient CLGSC (Certificate less generalized Signcryption) scheme that can adapt as one of the three cryptographic primitives: Signcryption, signature, or encryption, but all within a single algorithm. But here relay selection strategies for security-aware D2D-assist data transmission for m-Health systems have not been considered. Based on uncertified cryptography, this paper proposes a remote authentication protocol that includes non-repudiation, client anonymity, key-resistance, and revocability of extra-body communication in WBANs. Both theoretic analysis and experimental simulations show that the proposed authentication protocol is proven to be safe in a random oracle model and highly practical [10].

Aiqing Zhang et al. [11] introduced a secure data sharing protocol (SeDS) through a cryptographic approach in which both the public key-based signature and symmetric encryption are used to achieve security objectives by jointly considering the characteristics of the cellular network and the digital signature, ensuring traceability with a simple, secure one-way hash function. The bilinear pairing and the Diffie-Hellman Key Exchange (DHKE) are the basic schemes. The overhead of the transmission, network, and link layer increased in the SeDS. Since the overhead of the application message is dominated by communication overhead in most cases, only application layer overhead has been considered.

Muhammad F Mushtaq et al. [12] [13] [14] [15] [16] conducted a survey on cryptographic encryption and decryption algorithms. A detailed description of symmetrical algorithms such as DES, 3DES, Blowfish, AES, and Hisea with various design considerations has been presented. The experimental results of these algorithms concentrate on evaluation parameters such as encryption and decryption execution time, avalanche effect, memory, entropy, throughput, and correlation assessment. Based on the availability of resources, Blowfish, AES, and Hisea algorithms provide better security compared to existing techniques. They say that there is a need to establish hybrid encryption techniques that are used to enhance the reliability of encryption algorithms.

Mohamed Ali Kandi et al. [17] presented a new, versatile key management protocol for the IoT, which secures both group and device-to-device communications. Assignment and re-order algorithms help to secure both modes of communication. The demerit of the above proposal is that the central version of the protocol is susceptible to attack and failure. Thus, it is inevitable to decentralize the encryption technique to more than one device in order to avoid a single point of failure and make data highly inaccessible.

WeitaoXu et al. [18] addressed the various key generation schemes and security vulnerabilities associated with the mitigation techniques for IoT devices. Data transfer is

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

susceptible to a variety of threats that may be posed by an intruder or opponent. Commonly man-in-the-middle (MITM) attacks, replay attacks, impersonations, and eavesdropping attacks [18]. Randomness in the key generation scheme plays a vital role in avoiding possible attacks that would compromise the integrity of the device. The randomness aspect needs to be assessed in accordance with NIST standards. Limited efforts have been made in the area of key generation for emerging connectivity technologies and remain an open issue. Secure Key sharing be- tween communication devices can be completed using Public Key Cryptography techniques (PKC). Device-level constraints pose many technical challenges and conventional KM schemes cannot be adopted directly and therefore appropriate key generation and sharing mechanisms need to be developed.

AkashGarg et al. [19] develop a secure shared key agreement protocol for multiple devices within the IoT home environment. The proposed MKA protocol is analyzed in light of the issue of energy consumption design. SHA1 generates a 160-bit hash for a data size of 1024 bits. The parameterized cost of communication and the approximate time of computation is lower compared to other works [19]. Computation time is significantly shorter as network devices increase compared to other works. This protocol facilitates the agreement of multiple devices on a single key for communication, but the compromise of one device will lead to easy susceptibility and the cloud-based functionality needs to be pushed on individual devices.

TABLE I
NOTATIONS AND MEANINGS

Notations	Meanings
$SecKey^1$	Secret key^1
$DerKey^1$	Derived key^1
$DerKey^2$	Derived key^2
X	Original data
Y	Encrypted data
E	Encryption
D	Decryption
HMAC	Hash based message authentication code
SHA_1	Secure hash $algorithm^1$
$strvar^1$ to $strvar^4$	String variables

3 DESIGN CONSIDERATIONS

The proposed scheme is designed to ensure confidentiality, authentication of data origin through the verification of signatures between participating devices. The scheme of notations and their meanings is illustrated in Table I.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

3.1 DesignGoals

1. The system is designed for the secure transmission of data between the sender and the receiver, using a lightweight cryptographic technique.
2. Encryption, decryption, and signature verification are carried out by deriving two separate keys from a single secret key. This reduces the generation of two different secret keys, which incur more computational costs.
3. The system is designed to derive the keys from the secret key using randomly selected characters by incorporating random numbers.
4. The derived key size is increased from 128 bits to 160 bits to enhance data security.
5. The system is designed in such a way that even the adversary can try to capture the secret key, it is impossible to crack the derived keys because the keys are derived using randomly selected characters and numbers.

3.2 Design Methodology

Following points are considered at the sender device point of view:

1. Sender device generates the secret key1 using the standard Blowfish algorithm.
2. Randomly extract part1 and part2 from the secret key1.
3. Derives the derived key1 by concatenating part1 of the secret key1 and Random number1.
4. Derives the derived key2 by concatenating part2 of the secret key1 and Random number2.
5. Encrypt the data using derived key1 and send it to the receiver device.
6. Generate the signature using derived key2 and send it to the receiver device.

The following points are considered at the receiver device:

1. Receiver revives the signature from the sender device and validates the signature using derived key2. The derived key2 is received from the sender device.
2. If the signature is validated successfully then the receiver receives the derived key1 from the sender device and decrypts the data.
3. If the signature validation is unsuccessful then the receiver may refuse to decrypt the data.
4. In D2D communication, the sender intends to send data to the receiver system but must maintain security during the communication session. One method is to prevent key guessing while preserving the secrecy of the generated key.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

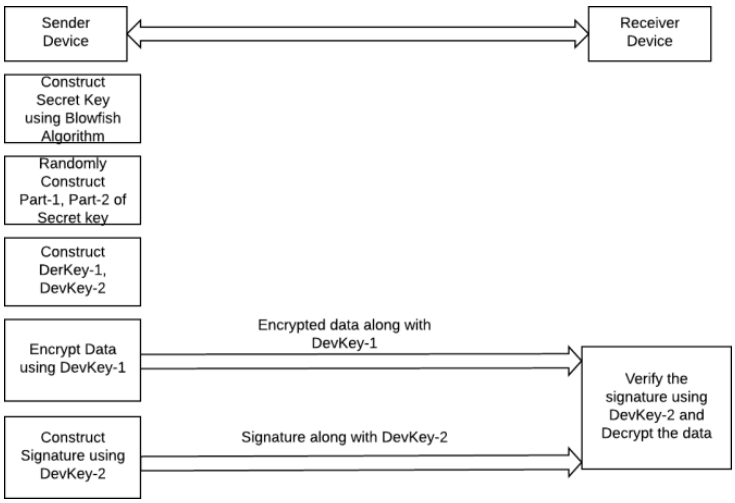


Fig.1. SystemArchitecture

Figure 1 depicts the system architecture as well as the various computations involved in the generation of the secret key. Generate secret key by applying Blowfish algorithm followed by which two sub-keys namely part1 and part2 are created from the single master key. Then derived keys namely DerKey1 and DerKey2 are constructed using Algorithm 3 and 4 respectively. The end user data is encrypted using DerKey1 at the sender side, while DerKey2 is used for signature generation. At the receiver end, the authenticity of signature is verified using DerKey2 and finally, data is decrypted to complete the reception process.

Algorithm 1 is designed to generate the secret key1 (SecKey1) on the sender device using the standard Blowfish algorithm. Here, the algorithm generates a secret key with a size of 128 bits, which is essential for generating the derived keys for confidentiality and data origin authentication. The important step of the proposed method is to convert the SecKey1 into two equal parts.

Algorithm 1	Generating secret key^1 ($SecKey^1$) at sender
Input:	Key generator algorithm (Blowfish)
Output:	secret key^1
1:	generate key using Blowfish algorithm
2:	initialize key with 128 bits
3:	initialize secret key^1 ($SecKey^1$) ← key
4:	convert secret key^1 into a string
5:	record secret key^1 for further processing

Algorithm 2 is designed to extract half the number of characters from the SecKey1 randomly on the sender device, and these characters are considered to be part1 of the *SecKey1.* The next step is to extract another half number of characters from the same secret key, and these characters are considered part2 of the *SecKey1.*

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

Algorithm 2	Convert $SecKey^1$ into two parts at sender
Input:	$SecKey^1$
Output:	$part_1$ and $part_2$
1:	receives $SecKey^1$ from **Algorithm 1**
2:	if $SecKey^1$ is string then compute length of the $SecKey^1$
3:	randomly extract half amount of characters from $SecKey^1$ store extracted characters into $part_1$
4:	randomly extract half amount of characters from $SecKey^1$ store extracted characters into $part_2$

Algorithm 3 is designed for generating derived key1 (DerKey1) from the sender device. First, the sender con- catenates part1 of the *SecKey1* to the Random number1 (RandNum1) and the result is sent to the secure hash algorithm (SHA1). The SHA1 intern generates 160 bits output which is considered to be *DerKey1* for confidentiality purposes.

Algorithm 3	Generate derived key^1 ($DerKey^1$) at sender
Input:	$part_1$, Random $number_1$ ($RandNum^1$)
Output:	$DerKey^1$
1:	receives $part_1$ of $SecKey^1$ from **Algorithm 2**
2:	initialize $strvar^1 \leftarrow part_1$
3:	compute $strvar^2 \leftarrow (strvar^1 \parallel RandNum^1)$
4:	compute $SHA_1(strvar^2)$ and produces 160 bit hash tag
5:	initialize $DerKey^1 \leftarrow$ 160 bits
6:	record $DerKey^1$ at sender device

Algorithm 4 is designed for generating key2 (DerKey2) from the sender device. First, the sender concatenates part2 of SecKey1 to Random number2 (RandNum2) and the result is sent to the secure hash algorithm (SHA1). The SHA1 intern generates a 160-bit output that is considered to be DerKey2 to perform data origin authentication. Previously, the data had been transmitted to the receiver device that the data had to be encrypted.

Algorithm 4	Generate derived key^2 ($DerKey^2$) at sender
Input:	$part_2$, Random $number_2$ ($RandNum^2$)
Output:	$DerKey^2$
1:	receives $part_2$ of $SecKey^1$ from **Algorithm 2**
2:	initialize $strvar^3 \leftarrow part_2$
3:	compute $strvar^4 \leftarrow (strvar^3 \parallel RandNum^2)$
4:	compute $SHA_1(strvar^4)$ and produce 160 bit hash tag
5:	initialize $DerKey^2 \leftarrow$ 160 bits
6:	kept $DerKey^2$ at the sender device

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

Algorithm 5 is designed to encrypt data using $DerKey^1$ using input as the original data (X). After that, the data is encrypted (Y) sent to the receiver device.

Algorithm 5	Encryption (E) at the sender device
Input :	Original data (X)
Output :	Encrypted data (Y)
1:	receives $DerKey^1$ from **Algorithm 3**
2:	initialize $DerKey^1$ for encryption/decryption
3:	compute Y ← E ($DerKey^1$ (X))
4:	transfer encrypted data (Y) to the receiver

Algorithm 6 is designed to generate a signature on the sender device to validate the data origin authentication. The hash-based message authentication code (HMAC) is a combination of MAC and hash functions that are essential for checking the data origin authentication. The HMAC algorithm takes input as encrypted data (Y), $DerKey^2$, and generates output as $HashTag^1$. The $HashTag^1$ is shared with the receiver to check whether or not the data is sent by a specific party.

Algorithm 6	Generate data integrity hash at the sender
Input :	Encrypted data (Y), $DerKey^2$
Output :	integrity $hash^1$ ($HashTag^1$) of Y
1:	receives $DerKey^2$ from **Algorithm 3**
2:	compute $HashTag^1$ ← HMAC (Y, $DerKey^2$)
3:	transfer $HashTag^1$ to the receiver device

Algorithm 7 is designed for the validation of data origin authentication by the receiver device. The receiver receives encrypted data, HashTag1, and receives DerKey2 from the sender via a secure channel. The receiver calculates hash2 (hashTag2) integrity using the HMAC algorithm and compares hashTag2 to hashTag1. If both signature tags are successfully compared, the receiver may decrypt the data otherwise there is no data decryption point. After the signature has been validated successfully, the receiver is willing to decrypt the data.

Algorithm 7	Verify signature at the receiver device
Input:	Encrypted data (Y), $DerKey^2$, $HashTag^1$
Output:	True or False
1:	receives Y from the sender device
2:	receives $DerKey^2$ from the sender device
3:	integrity $hash^2$ ($HashTag^2$) such that $HashTag^2$ ← HMAC(Y, $DerKey^2$)
4:	receives $HashTag^1$ from the sender device
5:	**if** $HashTag^2$ == $HashTag^1$ **then** return true i.e. signature successful
6:	**else** return false i.e. signature unsuccessful **end if**

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

Algorithm 8 is designed for data decryption on the receiver device. The receiver may request DerKey1 from the sender to decrypt the data. Since the data is encrypted by the sender using DerKey1, the original data cannot be extracted without DerKey1.

Algorithm 8	Decryption (D) at the receiver device
Input :	Encrypted data (Y), $DerKey^1$
Output :	Original data (X)
1:	**if** step 5 of **Algorithm 7** is successful **then**
	receives $DerKey^1$ from the sender device
2:	compute X ← D ($DerKey^1$, (Y))
3:	perform operations on X

4 Experimental Results

The proposed method is designed and implemented for a secure device for device communication using a lightweight cryptographic technique.

Figure 2 shows that the results obtained from the proposed method are compared to the existing Blowfish algorithm used to derive the secret key1. The graphic representation shows that the proposed scheme consumes less computational time compared to the existing solution.

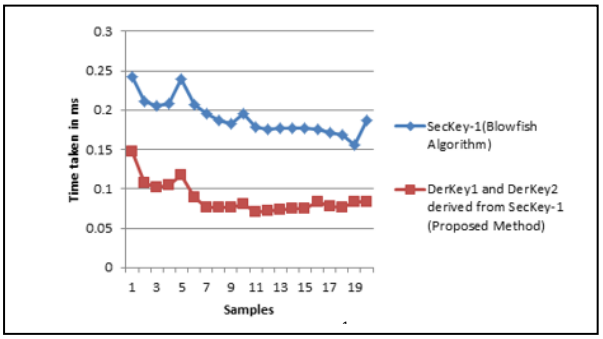


Figure 3 shows that the results obtained from the proposed method are compared to the existing method (Blowfish algorithm) used to derive the secret key2. The graphic representation shows that the proposed scheme consumes less computational time compared to the existing solution.

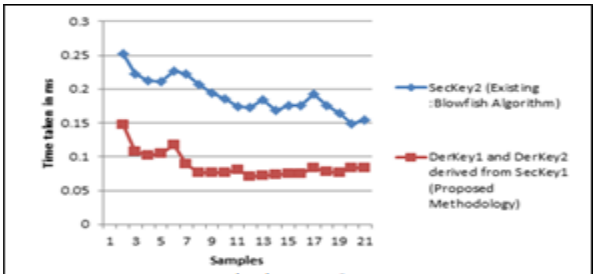
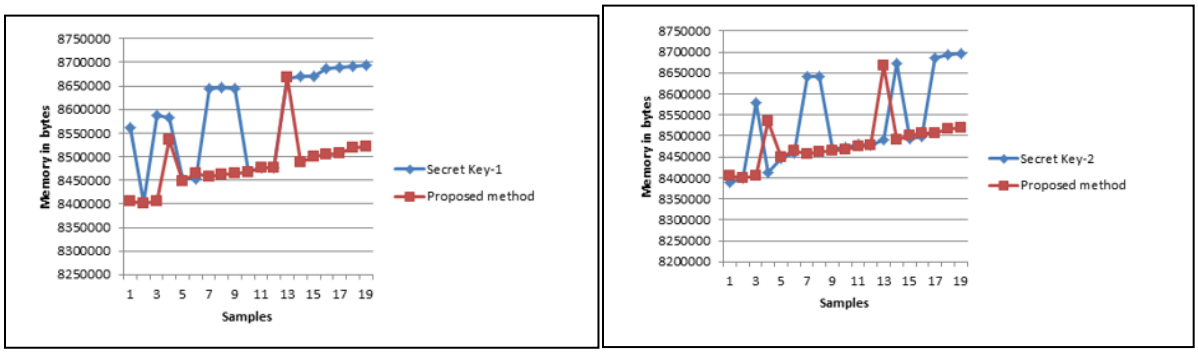


Fig. 3. Time taken (ms) $SecKey^2$ vs proposed method.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

The proposed method memory utilization is compared to existing solutions in Figure 4 and Figure 5. The graphical representation demonstrates that the proposed method consumes slightly less or same as the existing solution.



5 EXPERIMENTAL ANALYSES

The proposed method is designed and implemented for tackling following issues:

5.1 Efficiency

The efficiency of the system is strictly related to the cost of the calculation of the key derivation. More computational cost always leads to a reduction in the efficiency of the system. Here, the proposed scheme is designed to reduce the cost of computing while drifting keys to perform encryption/ decryption and data origin authentication. The lightweight cryptographic technique is incorporated in this paper, which reduces the computational costs that have an impact on the efficiency of the system.

5.2 Improving the data security

Cryptography plays a key role in network data security. Data security depends on the size of the key used in the encryption, decryption, or authentication of the data source. There is a lot of difference between 128 and 160-bit key sizes. For example, if the key size is 128 bits, then we can make 2128 keys called size space. The proposed scheme is designed to increase the key size from 128 bits to 160 bits, which will have an impact on improving data security.

5.3 Resistance to Key Guessing Attack

The proposed scheme is designed in such a way that it is impossible to predict the key at any cost while establishing the keys for the intended party. Initially, the 128-bit key is generated using the standard Blowfish algorithm because, whenever key generation is required, we must use standard certified cryptographic algorithms. Without a secure process for key generation and establishment throughout their life cycle, the benefits of

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

using strong cryptographic keys are potentially lost. In this paper, while the keys are drawn, the contents of the secret key are selected randomly, and the contents are concatenated with random numbers. Even if the opponent captures a copy of the secret key, it is very difficult for him to generate the derived keys because the derived keys are generated using randomly selected characters and random numbers.

5.4 Resistance against Non-Repudiation

The proposed scheme is designed to enable the receiver to easily validate the signature of the sender device. The receiver receives the encrypted data, the signature, and the key2 derived from the sender via a secure channel. The receiver re-computes the encrypted data signature using the key2 derivative. If both signatures are validated successfully, the receiver will know the data sent by the authenticated party. After validation of the signature, the receiver trying to encrypt the data cannot do otherwise.

Table II shows the time taken to draw the secret key1, the secret key2, and to generate the derived keys. The experimental results are compared to the existing Blowfish algorithm. From experimental results, we can pretend that the proposed method can reduce the computational time that leads to an improvement in the overall performance of the system.

TABLEII

TIME TAKEN(MS)FOR KEYDERIVATION(EXISTING VSPROPOSED METHODOLOGY)

Sample	***SecKey*^{1}(Blowfish Algorithm)**	***SecKey*^{2}(Existing: Blowfish Algorithm)**	***DerKey*^{1}and *DerKey*^{2}derived from *SecKey*^{1}(Proposed methodology)**
1	0.242	0.252	0.147
2	0.211	0.223	0.107
3	0.205	0.212	0.102
4	0.208	0.211	0.105
5	0.239	0.227	0.118
6	0.207	0.223	0.089
7	0.195	0.207	0.076
8	0.187	0.194	0.077
9	0.183	0.185	0.077
10	0.195	0.174	0.080
11	0.178	0.173	0.070
12	0.175	0.184	0.072
13	0.177	0.169	0.074
14	0.177	0.176	0.075

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

15	0.177	0.176	0.075
16	0.176	0.193	0.084
17	0.172	0.175	0.078
18	0.169	0.164	0.077
19	0.156	0.148	0.083
20	0.187	0.155	0.084
Avg.	**0.190**	**0.191**	**0.087**

Table III compares the proposed method to existing solutions in terms of functionality. In terms of computational cost for key derivation, Mingjun Wang et al. [20] have used the Diffie-Hellman Key Exchange algorithm for secure sharing of session keys in D2D communication. The Diffie-Hellman algorithm is used in this approach for secure key exchange, which results in a high computation cost. Payal P. et al. [7] used a digital signature and symmetric encryption to implement a secure D2D communication. As a result, the digital signature employs public-key cryptography to sign data, incurring a higher computational cost. Ya-Nan Zhang et al. [6] designed a secure D2D communication system based on elliptic curve cryptography (ECC) and lightweight authenticated encryption. Because elliptic curve cryptography generates public and private key pairs, absolutes have a higher computational cost. ByoungjinSeok et al. [8] incorporated a master private key and public key for secure D2D communication, which also incurs significant processing overhead to generate cryptographic keys.

TABLE III

FUNCTIONAL COMPARISON

Parameters	**Mingjun Wnag et al.[20]**	**Payal P et al. [7]**	**Ya-Nan Zhang et al. [6]**	**ByoungjinSeok et al. [8]**	**Proposed Method**
Data Confidentiality	Yes	Yes	Yes	Yes	Yes
Non-Repudiation	No	No	No	Yes	Yes
Enhancement of data security	No	No	No	No	Yes
Computation cost for key derivation	High	High	High	High	Low
Key robustness	Medium	Medium	Medium	Medium	High
Mutual authentication	Yes	No	No	No	No

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

To reduce computational costs, the proposed method generates two distinct keys from a single session key or master key. The system's design also improves key robustness because the content of the secret keys is extracted randomly and concatenated with the random numbers while deriving the keys.

6 Conclusions and Future work

From the literature survey, the development of a lightweight cryptographic technique for secure communication between devices is very important. In this paper, we designed and implemented a cryptographic technique to ensure data security and integrity with derived keys computed from a single secret key. The proposed method can reduce the cost of computing while deriving the keys that reduce the need to generate another secret key. The system is designed to generate the key1 derived and the key2 derived from the secret key using the SHA1 algorithm. Derived key1 is responsible for the confidentiality of data and derived key2 for authentication of data origin. This approach can provide better performance while drifting keys, improve data security and make key attacks unfeasible. Future work involves ensuring the secure delivery of keys between the devices and estimating the cost of communication.

References

[1] U. N. Kar and D. K. Sanyal, "A critical review of 3gpp standardization of device-to-device communication in cellular networks," SN Computer Science, vol. 1, no. 1, p. 37, 2020.
[2] K. M. Sadique, R. Rahmani, and P. Johannesson, "Towards security on internet of things: applications and challenges in technology," Procedia Computer Science, vol. 141, pp. 199–206, 2018.
[3] A. Gupta, M. Tripathi, and A. Sharma, "A provably secure and efficient anonymous mutual authentication and key agreement protocol for wearable devices in WBAN," Computer Communications, vol. 160, pp. 311–325, 2020.
[4] A. R. Krishna, A. Chakravarthy, and A. Sastry, "A hybrid cryptographic system for secured device to device communication." International Journal of Electrical & Computer Engineering (2088-8708), vol. 6, no. 6, 2016.
[5] H. Tan, Y. Song, S. Xuan, S. Pan, and I. Chung, "Secure d2d group authentication employing smartphone sensor behavior analysis," Symmetry, vol. 11, no. 8, p. 969, 2019.
[6] Y.-N. Zhang and H.-B. Mu, "An improved secure data transmission protocol based on d2d for mobile health system [j]," Journal of Computers, vol. 1, no. 30, pp. 52–63, 2019.
[7] P. P. Tayade and P. Vijayakumar, "Enhancement of Security and Confidentiality for D2D Communication in LTE-Advanced Network Using Optimised Protocol," in Wireless Communication Networks and Internet of Things. Springer, 2019, pp. 131–139.
[8] B. Seok, J. C. S. Sicato, T. Erzhena, C. Xuan, Y. Pan, and J. H. Park, "Secure d2d communication for 5g iot network based on lightweight cryptography," Applied Sciences, vol. 10, no. 1, p. 217, 2020.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/SW59M

[9] A. Zhang, L. Wang, X. Ye, and X. Lin, “Light-weight and robust security-aware d2d-assist data transmission protocol for mobile-health systems,” IEEE Transactions on Information Forensics and Security, vol. 12, no. 3, pp. 662–675, 2016.
[10] H. Xiong and Z. Qin, “Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks,” IEEE transactions on information forensics and security, vol. 10, no. 7, pp. 1442–1455, 2015.
[11] A. Zhang, J. Chen, R. Q. Hu, and Y. Qian, “SeDS: Secure Data Sharing Strategy for D2D Communication in LTE-Advanced Networks,” IEEE Transactions on Vehicular Technology, vol. 65, no. 4, pp. 2659–2672, 2015.
[12] M. F. Mushtaq, S. Jamel, A. H. Disina, Z. A. Pindar, N. S. A. Shakir, and M. M. Deris, “A survey on the cryptographic encryption algorithms,” International Journal of Advanced Computer Science and Applications, vol. 8, no. 11, pp. 333–344, 2017.
[13] M. Ebrahim, S. Khan, and U. B. Khalid, “Symmetric algorithm survey: a comparative analysis,” arXiv preprint arXiv:1405.0398, 2014.
[14] T. Nie and T. Zhang, “A study of des and blowfish encryption algorithm,” in Tencon 2009-2009 IEEE Region 10 Conference. IEEE, 2009, pp. 1–4.
[15] P. Patil, P. Narayankar, D. Narayan, and S. M. Meena, “A comprehensive evaluation of cryptographic algorithms: Des, 3des, aes, rsa and blowfish,” Procedia Computer Science, vol. 78, pp. 617–624, 2016.
[16] D. PalanivelRajan, and Dr. S. John Alexis, "Comparative Study on Data Encryption Algorithms in Cloud Platform", International Journal of Engineering Research & Technology (IJERT), vol. 6, pp. 126-129, 2017.
[17] M. A. Kandi, H. Lakhlef, A. Bouabdallah, and Y. Challal, “A versatile key management protocol for secure group and device-to-device communication in the internet of things,” Journal of Network and Computer Applications, vol. 150, p. 102480, 2020.
[18] W. Xu, J. Zhang, S. Huang, C. Luo, and W. Li, “Key generation for internet of things: A contemporary survey,” ACM Computing Surveys (CSUR), vol. 54, no. 1, pp. 1–37, 2021.
[19] A. Garg and T. Lee, “Secure key agreement for multi-device home iot environment,” Internet of Things, p. 100249, 2020.
[20] M. Wang, Z. Yan, and V. Niemi, “Uaka-d2d: Universal authentication and key agreement protocol in d2d communications,” Mobile Networks and Applications, vol. 22, no. 3, pp. 510–525, 2017.