This full-text paper was peer-reviewed and accepted to be presented at the IEEE WiSPNET 2017 conference.

Secure Key Exchange Using Enhanced Diffie-Hellman Protocol Based on String Comparison

Ankit Taparia,[1] Saroj Kumar Panigrahy[2] and Sanjay Kumar Jena[3]
Department of Computer Science and Engineering, National Institute of Technology Rourkela, Odisha, 769008, India
Email: [1]tapariaankit13@gmail.com [2]skp.nitrkl@gmail.com [3]skjenanitrkl@gmail.com

***Abstract*—Authenticated key exchange protocols play an important role in securing communications and are extensively deployed for use in various real-world network applications. Diffie-Hellman key exchange protocol has been widely used for establishing a secret key in two-party communications without using any third-party key management entity. But, Diffie-Hellman key exchange protocol does not authenticate the users in the communication and is vulnerable to man-in-the-middle attack when the environment is wireless. In this paper, a variant of Diffie-Hellman key exchange protocol based on string comparison has been presented that addresses the security vulnerabilities identified in the trivial Diffie-Hellman Protocol. It uses a combination of commitment scheme and authentication strings to withstand man-in-the-middle attack and can be used in both wired as well as wireless networks.**

***Index Terms*—Security, Key Exchange, Key Agreement, Diffie-Hellman, MITM, Commitment Scheme, String Comparison.**

I. Introduction

Due to advancement in mobile and computing technologies, people have a tendency to rely more on them in different situations [1]. In case of wireless communication such as Bluetooth, WiFi, wireless sensor network (WSN), wireless body area network (WBAN), vehicular ad hoc network (VANET), etc., the channels are deliberately vulnerable to multiple attacks due to its broadcast nature [2]. Hence, security and privacy are the main concerns in device to device (D2D) communications [3] and other type of wireless networks like VANET etc. [4], [5]. To ensure secure communication between two devices, the most important step is forming a shared secret key. However, due to absence of a trustworthy third party in peer devices, communication makes this a non-trivial task. In this paper, we focus on secured communication between two mobile/computing devices with less possible computation. One such ideal protocol for communication is trivial Diffie-Hellman key exchange protocol (DHKEP) [6]. In this protocol, two parties generate and exchange a secret key (shared) without sharing any other prior information about each other. But, in case of wireless environment, DHKEP is susceptible to the man-in-the-middle (MITM) attack [7], [8], by which an attacker can impersonate as one of the users, succeed in deceiving the other user and intercept and gain access to information that the two parties were trying to send to each other.

There are various proposed methods by researchers withstanding MITM attacks in DHKEP. Balfanz et al. [9] proposed a simple protocol, in which two devices A and B interchange the hash values of their public keys for mutual authentication, over a secure channel. It involves a more number of bits to be conjointly authenticated. The protocol proposed by Gehrmann et al. [10] decreases the message size for authentication to k bits, but it also requires a secure authentication channel. Cagalj et al. [11] proposed a protocol based on DHKEP and a commitment scheme. It requires a total of 4 rounds of communications to create a shared secret key between two users over the wireless channel. Shen et al. [12] proposed a secure key establishment scheme for D2D communications similar to [11], in which 3 rounds of message communications are involved. The authors claim that their protocol involves less overhead for computation and communication while attaining equal security level. The above protocols [11], [12] require user intervention to enter their IDs and also during the authenticated string verification phase. In this paper, a 3-round key exchange protocol based on commitment scheme similar to [12] has been presented, but with least possible user intervention, meanwhile providing the same level of security.

The rest of the paper is organized as follows. DHKEP has been briefed in Section II. In Section III, the proposed protocol for enhanced DHKEP with string comparison (EDH-SC) has been discussed. Implementation details have been described in Section IV. Section V discusses the results and analysis. Finally, Section VI gives the concluding remarks.

II. Diffie-Hellman Key Exchange Protocol

Diffie-Hellman (DH) Key Exchange protocol is the most commonly used public key system, which permits two parties to agree on a secret key which is shared between the two, while they can only interchange plaintext messages over insecure channels. The working of trivial DHKEP is as follows:

Alice (A) and Bob (B) are two users who want to communicate with each other. In the first step, A selects a random secret X_a and B selects X_b and both calculate the DH public parameters $a = g^{X_a} (\text{mod } p)$ and $b = g^{X_b} (\text{mod } p)$ respectively, where g is a generator of a group of large order and p is a large prime number. In the next step, A and B exchange the public parameters a and b and finally compute the shared secret key K which shall be used for future

978-1-5090-4442-9/17/$31.00 ©2017 IEEE

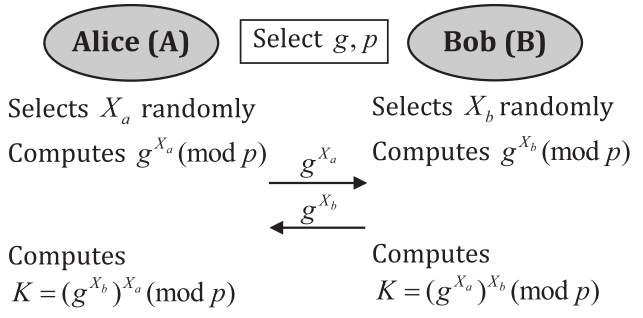

Alice (A)
Select g, p
Bob (B)
Selects X_a randomly
Computes $g^{X_a} (\text{mod } p)$
g^{X_a}
g^{X_b}
Selects X_b randomly
Computes $g^{X_b} (\text{mod } p)$
Computes
$K = (g^{X_b})^{X_a} (\text{mod } p)$
Computes
$K = (g^{X_a})^{X_b} (\text{mod } p)$

Fig. 1. Diffie-Hellman key exchange protocol.

communications. The secret key K can be calculated as $K = (g^{X_b})^{X_a} (\text{mod } p) = (g^{X_a})^{X_b} (\text{mod } p) = g^{X_a X_b} (\text{mod } p)$.

The communication in DHKEP has been depicted in Fig. 1. It is known that the attacker can modify certain DH-parameters and break into the communication without being noticed. This is known as MITM attack. So, there is a necessity for finding a way by which the attacker cannot be able modify the DH-parameters without being noticed.

III. Proposed Protocol

Before discussing the proposed protocol, the communication scenario and the assumptions have been discussed. A brief discussion of the commitment scheme used in the protocol has also been made.

A. Communication Scenario

The following communication scenario has been considered in our work. Two users A and B want to agree upon a shared secret key for one to one communications between them. Both of them have wireless devices such as smartphone or tablet which can communicate over wireless medium [13]. Both the devices can have computational capability to perform DHKEP, and have an input/output interface. There is no sharing of any other cryptographic information previously, by the two users and there is no availability of trusted third party. Users A and B know each other's IP addresses. They can recognize each other visually or verbally for authenticating a short message mutually.

B. Assumptions

It is assumed that users A and B agree upon $\langle G, g, p \rangle$ where, G is a finite cyclic group, g is a generator in G, and p is a large prime number. It is also assumed that G should be a subgroup of Z_p^* of prime order q. Here, Z_p^* is a multiplicative group which consists of non-zero integers $(\text{mod } p)$. Dolev-Yao adversary model [14] has been considered, in which the adversary has complete control of the wireless medium. It is capable of eavesdropping, intercepting, and modifying any message. The adversary can also initiate a communication with any user. It is also assumed that genuine users are not compromised and will follow the protocol. The protocol proposed in this section ensures the integrity of public parameters of DH protocol fairly than the integrity of the shared key.

C. Commitment Schemes

A commitment scheme is a significant cryptographic building block that have been used in our protocol. A commitment scheme permits an individual user to commit to a chosen value (or chosen message) while keeping it unseen to other users, with the capability to disclose the committed message or value later. A commitment scheme is defined by the following two functionalities—

Commit: $(c, d) \leftarrow m$ transforms a message m into a commitment/open pair (c, d). The commit value c reveals no information of m, but with d (decommit value) together (c, d) will disclose the message m.

Open: $m \leftarrow (c, d)$ gives original message m if (c, d) is the commitment/open pair generated by $Commit(m)$.

It has the following two main properties: (i) the commitment scheme is *binding* if a user cannot change the value or message after she has committed to it, and (ii) the commitment scheme is *hiding*, if the commitment value is hidden from its receiver until the sender opens it.

D. Enhanced DHKEP Based on String Comparison

The enhanced DH protocol based on string comparison (EDH-SC) is shown in Fig. 2. The protocol is divided into three phases: *initialization*, *exchange*, and *verification*. In the initialization phase, A and B randomly select their secret exponents X_a and X_b respectively, from large primes Z_q and calculate DH public parameters g^{X_a} and g^{X_b} respectively. Then, A and B generate k-bit random strings N_a and N_b, respectively. After that, both the users A and B prepare the messages $m_a = 0||ID_a||g^{X_a}||N_a$ and $m_b = 1||ID_b||g^{X_b}||N_b$, respectively where ID_a and ID_b are identifiers for users A and B that can be readable by any human. Here, 0 and 1 are used to prevent a reflection attack [12]. Then, A computes the commitment/opening pair (C_a, D_a) of her message m_a.

In the exchange phase, A sends the commitment C_a to B. B responds with her message m_b. In turn, A sends D_a, by which B opens the commitment C'_a and checks the correctness of commitment C_a sent by A. After verifying the commitment, both of them proceeds to the next phase, i.e., verification phase.

In the verification phase, both the users A and B generate verification strings $S_A = N_a \oplus N'_b$ and $S_B = N_b \oplus N'_a$ (where $\oplus$ represents XOR operation). If the strings S_A and S_B match, then both the users A and B accept each other's DH-parameters g^{X_a} and g^{X_b} as being authentic and unchanged. Then, they both generate shared key $K = g^{X_a X_b} (\text{mod } p)$.

E. Security Assessment

In this protocol (Fig. 2), any user has to commit on a value m'_a before actually seeing the value m_b; and any user has to send a value m'_b before seeing the value m_a. It follows the binding and hiding properties of the commitment scheme.

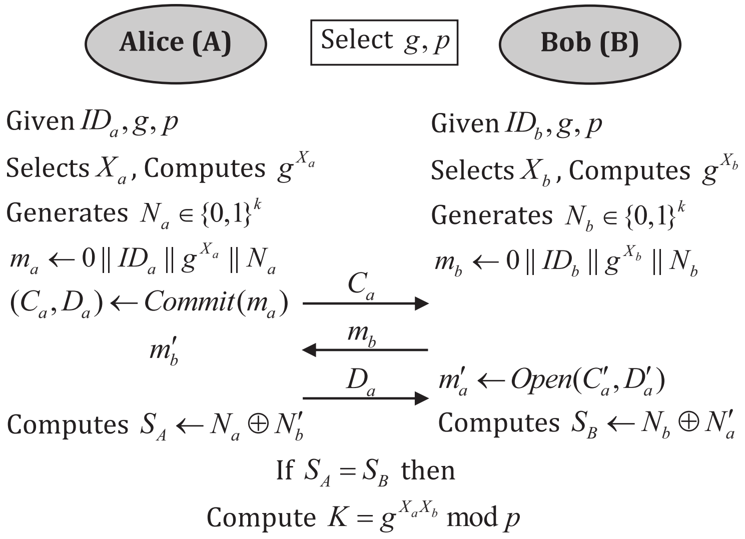

Alice (A)
Select g, p
Bob (B)
Given ID_a, g, p
Selects X_a, Computes g^{X_a}
Generates $N_a \in \{0,1\}^k$
$m_a \leftarrow 0 \| ID_a \| g^{X_a} \| N_a$
$(C_a, D_a) \leftarrow Commit(m_a)$
C_a
m'_b
m_b
D_a
Computes $S_A \leftarrow N_a \oplus N'_b$
Given ID_b, g, p
Selects X_b, Computes g^{X_b}
Generates $N_b \in \{0,1\}^k$
$m_b \leftarrow 0 \| ID_b \| g^{X_b} \| N_b$
$m'_a \leftarrow Open(C'_a, D'_a)$
Computes $S_B \leftarrow N_b \oplus N'_a$
If $S_A = S_B$ then
Compute $K = g^{X_a X_b} \bmod p$

Fig. 2. Enhanced Diffie-Hellman with string comparison protocol.

So either attacker has to first commit to or submit its own message. Supposing the attacker E initiates a communication with user B impersonating A, it will first commit to a value $m_e = ID_a||g^{X_e}||N_e$ and sends the commitment value C_e to B. After getting the reply message m_b, the attacker can manipulate m_b into $m'_b = ID_b||g^{X_e}||N_b$ and forward it to A. But at the end of the protocol, $S_B = N_b \oplus N_e$ with $S_A = N_a \oplus N_b$ will be compared by both the users A and B. Here, the user identity is the concatenation of random alphanumeric string (l bits) with the MAC-ID of the user, i.e, $USER\_ID = MAC\_ID||random\_string$ (l bits). To break into the protocol, the attacker E has to find a message m_e such that $Commitment(m_a) = Commitment(m_e)$. Earlier, when the $USER\_ID$ was static, the attacker has to find a string S such that $m_e = ID_a||S$ and $Commitment(m_a) = Commitment(m_e)$.

The process has been automated in such a way that the users do not have to enter their IDs manually, rather the MAC-IDs of their devices are extracted automatically and random alphanumeric strings are appended to the MAC-IDs at both the ends and sent (exchanged) to the other device. This provides an extra layer of security as the attacker impersonating as any of the users cannot generate the same appended ID, thus the attacker will be bound to send the wrong ID or commitment for the wrong ID. Hence, at the verification phase, either the commitments will not match or the verification strings at both the ends of the users will differ.

It follows in this way: The only way that A and B mutually agree on the authentication string is if $N_e = N_a$. The attacker E cannot modify N_e after it sends out the commitment as per the binding property of the commitment scheme. The probability of such successful attack is 2^{-k} at maximum. Since $USER\_ID$ of A changes with every session, the attacker is bound to compute a string m_e such that: $Commitment(m_a) = Commitment(m_e)$. Since $|m_e| > |S|$, the probability of successful attack decreases to $2^{-(k+l)}$ where l is length of random alphanumeric string appended to ID of A.

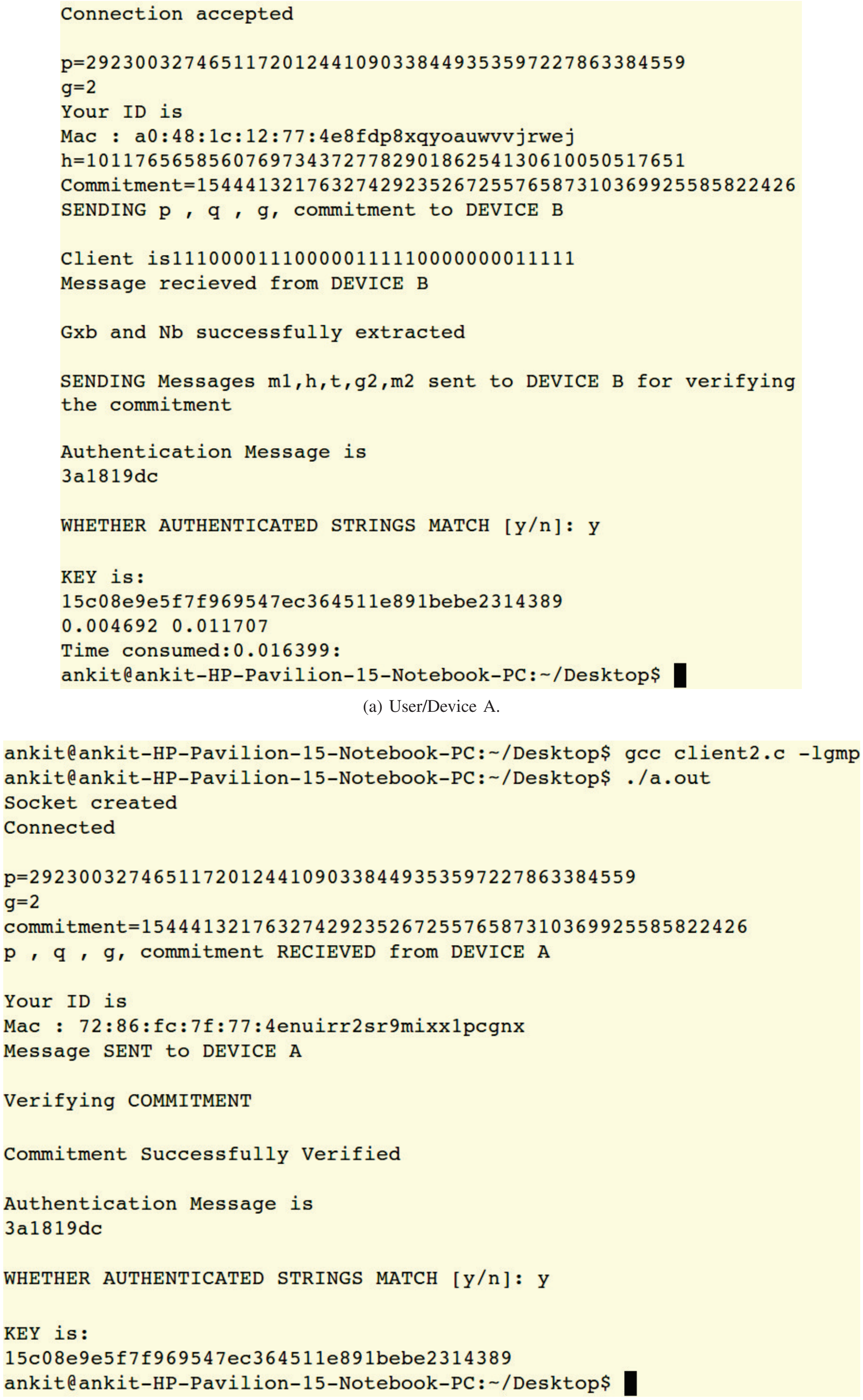
IV. Implementation Details

In order to simulate the real-time behavior of the proposed secured key exchange protocol, we implemented the EDH-SC protocol in C language using Linux socket programming which can be used for secure key exchange between two computers provided we know the IP addresses of both the computers. The configuration of the computer used for implementation is Intel i5 2.3 GHz processor, 4 GB RAM and the OS is Ubuntu 14.0.4. The GMP (GNU Multiple Precision Arithmetic Library) [15], an open source library, is used for handling the cryptographic operations on large numbers. Pedersen Commitment Scheme [16] has been used in the implementation. Fig. 3 shows the snapshots of the sample run of the proposed protocol.

V. Results and Analysis

An analysis of execution time of the proposed protocol against size of primes and length of random string (N_a and N_b) is performed to find the optimal size of parameters which can be used for faster execution without compromising security level of the protocol. Fig. 4(a) and (b) show the plots of length of random strings ($k = 10, 15, 30, \ldots, 80$ bits) versus execution time of the protocol for varying size of primes $(100, 140, \ldots, 512$ bits). The behavior of verification strings was also studied for each run of the protocol.

We observed that with the increase in the number of bits (k) and size of prime p, the execution time increases and it increases drastically at $|p| = 512$ bits. As the security of protocol improves on large value of $|p|$ and k, so there is a trade off between security and cost of computation. For length of $|N_a| \geq 80$, authentication message contains mostly 0 s and 1 s. Prime sizes of order 256 and 448 bits give weak



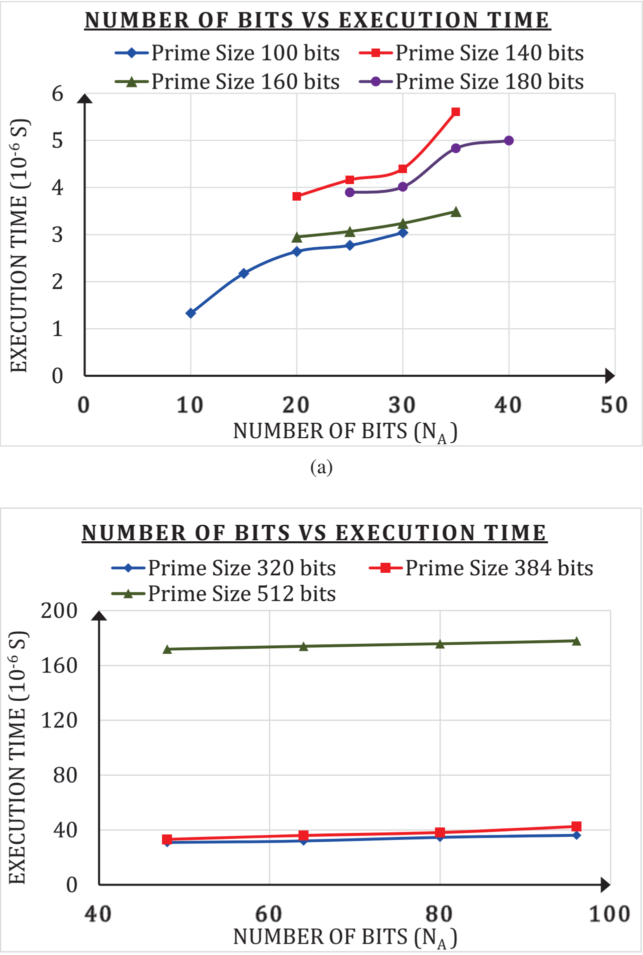
```
Connection accepted

p=29230032746511720124410903384493535972278633845 59
g=2
Your ID is
Mac : a0:48:1c:12:77:4e8fdp8xqyoauwvvjrwej
h=101176565856076973437277829018625413061005051765 1
Commitment=1544413217632742923526725576587310369925585822426
SENDING p , q , g, commitment to DEVICE B

Client is1110000111000001111110000000011111
Message recieved from DEVICE B

Gxb and Nb successfully extracted

SENDING Messages m1,h,t,g2,m2 sent to DEVICE B for verifying
the commitment

Authentication Message is
3a1819dc

WHETHER AUTHENTICATED STRINGS MATCH [y/n]: y

KEY is:
15c08e9e5f7f969547ec364511e891bebe2314389
0.004692 0.011707
Time consumed:0.016399:
ankit@ankit-HP-Pavilion-15-Notebook-PC:~/Desktop$ █
```

(a) User/Device A.

```
ankit@ankit-HP-Pavilion-15-Notebook-PC:~/Desktop$ gcc client2.c -lgmp
ankit@ankit-HP-Pavilion-15-Notebook-PC:~/Desktop$ ./a.out
Socket created
Connected

p=2923003274651172012441090338449353597227863384559
g=2
commitment=1544413217632742923526725576587310369925585822426
p , q , g, commitment RECIEVED from DEVICE A

Your ID is
Mac : 72:86:fc:7f:77:4enuirr2sr9mixx1pcgnx
Message SENT to DEVICE A

Verifying COMMITMENT

Commitment Successfully Verified

Authentication Message is
3a1819dc

WHETHER AUTHENTICATED STRINGS MATCH [y/n]: y

KEY is:
15c08e9e5f7f969547ec364511e891bebe2314389
ankit@ankit-HP-Pavilion-15-Notebook-PC:~/Desktop$ █
```

(b) User/Device B.

Fig. 3. Snapshot of sample run of EDH-SC protocol.



(a)

(b)

Fig. 4. Execution time analysis of EDH-SC protocol.

authentication messages. Thus, the optimal size of primes which should be used in the protocol is found to be in the range [140, 384] bits. The EDH-SC protocol can be modified in a such a way that it can be implemented for security applications in wireless sensor networks (WSN) and wireless body area networks (WBAN) [17]. The proposed protocol can also be used in case of communication between Android based smartphones in D2D communication environments.

VI. Conclusion

The analysis of the security necessities and experiments for establishment of secret key between two unknown devices has been done. The proposed key exchange protocol enables two users to securely agree on a secret key with a less computational cost and little overhead for mutual authentication. The security analysis of the proposed protocol shows that the probability that an attacker can launch an attack is $2^{-(k+l)}$ at maximum, where k and l are length of the USER_ID and the random string respectively, used for mutual authentication.

We have implemented the protocol to set up secured connection between two host computers and also analyzed the execution time for varying length of the parameters. We have also automated the protocol to make the user intervention minimum. The results show that the proposed protocol is competent, and attains usability at a great level. In our future work, we are planning to implement a variant of the proposed protocol for security in WSN, WBAN and also in Android based smartphones.

Acknowledgements: This research work is partially supported by Information Security Education and Awareness Project, Phase-II (ISEA-II) funded by Ministry of Electronics and Information Technology (MeitY), Government of India.

References

[1] "Mobile fact sheet." Pew Research Center, Jan. 2017. http://www.pewinternet.org/fact-sheet/mobile/.

[2] S. K. Panigrahy, S. K. Jena, and A. K. Turuk, "Security in bluetooth, rfid and wireless sensor networks," in *International Conference on Communication, Computing and Security (ICCCS-2011)*, (Rourkela), pp. 628–633, ACM, Feb. 2011.

[3] D. Zhu, A. L. Swindlehurst, S. A. A. Fakoorian, W. Xu, and C. Zhao, "Device-to-device communications: The physical layer security advantage," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, (Florence), pp. 1606–1610, IEEE, 2014.

[4] S. Mohanty, D. Jena, and S. K. Panigrahy, "A secure rsu-aided aggregation and batch verification scheme for vehicular networks," in *International Conference on Soft Computing and its Applications (ICSCA-2012)*, (Kuala Lumpur), pp. 174–178, Aug. 2012.

[5] B. Mishra, S. K. Panigrahy, T. C. Tripathy, D. Jena, and S. K. Jena, "A secure and efficient message authentication protocol for vanets with privacy preservation," in *World Congress on Information Communication Technologies (WICT-2011)*, (Mumbai), pp. 884–889, IEEE, Dec. 2011.

[6] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, pp. 644–654, Nov. 1976.

[7] Z. Chen, S. Guo, K. Zheng, and Y. Yang, "Modeling of man-in-the-middle attack in the wireless networks," in *International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007)*, pp. 2255–2258, IEEE, Sep. 2007.

[8] T. R. Muchukota, S. K. Panigrahy, and S. K. Jena, "Man-in-the-middle attack and its countermeasure in bluetooth secure simple pairing," in *International Conference on Information Processing (ICIP-2011)*, (Bangalore), pp. 367–376, Springer, Aug. 2011.

[9] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, "Talking to strangers: Authentication in ad-hoc wireless networks.," in *Network and Distributed System Security Symposium (NDSS 2002)*, (San Diego), Feb. 2002.

[10] C. Gehrmann, C. J. Mitchell, and K. Nyberg, "Manual authentication for wireless devices," *RSA Cryptobytes*, vol. 7, no. 1, pp. 29–37, 2004.

[11] M. Cagalj, S. Capkun, and J.-P. Hubaux, "Key agreement in peer-to-peer wireless networks," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 467–478, 2006.

[12] W. Shen, W. Hong, X. Cao, B. Yin, D. M. Shila, and Y. Cheng, "Secure key establishment for device-to-device communications," in *IEEE Global Communications Conference (GLOBECOM 2014)*, (Austin), pp. 336–340, IEEE, 2014.

[13] "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications." IEEE Standard 802.11, 1999.

[14] W. Mao, *Modern Cryptography: Theory and Practice*. USA: Prentice Hall Professional Technical Reference, 2004.

[15] "GMP: GNU multiple precision arithmetic library." https://gmplib.org/.

[16] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Advances in Cryptology (CRYPTO'91): Proceedings* (J. Feigenbaum, ed.), pp. 129–140, Berlin, Heidelberg: Springer Berlin Heidelberg, 1992.

[17] S. K. Panigrahy, S. K. Jena, and A. K. Turuk, "Study and analysis of human stress detection using galvanic skin response (GSR) sensor in wired and wireless environments," *Research Journal of Pharmacy and Technology*, vol. 10, no. 3, 2017. (In Press).