IMPROVING THE PRIVACY-PRESERVING OF COVID-19 BLUETOOTH-BASED CONTACT TRACING APPLICATIONS AGAINST TRACKING ATTACKS

Ali M. Allam

Department of Communications and Electronics Engineering,
Helwan University, Cairo, Egypt

ABSTRACT

Bluetooth is an essential wireless standard for short-distance and low-power wireless networks. Health departments' contact-tracing applications depended on Bluetooth technology to prevent infectious diseases from spreading, especially COVID-19. The security threats of the Bluetooth-based contact-tracing applications increased because an adversary can use them as surveillance tools that violate the user's privacy and revealpersonal information. The Bluetooth standard mainly depends on the device address in its authenticated pairing mechanism (Secure Simple Pairing), which can collect with off-the-shelf hardware and software and leads to a tracking attack. To avoid the risk of tracking based on this security vulnerability in the Bluetooth protocol, we suggest a novel authentication protocol based on a non-interactive zero-knowledge scheme to substitute the authentication protocol used in the Bluetooth standard. The new protocol can replace the authentication protocol in the Bluetooth stack without any modification in the device pairing flow. Finally, we prove the security of our proposed scheme against the man-in-the-middle attack and tracking attack. A performance comparison with the authentication algorithm in the BLE standard shows that our method mitigates the tracking attack with low communication messages. Our results help enhance the contact-tracing application's security in which Bluetooth access is available.

KEYWORDS

Bluetooth low energy, Bluetooth threat, Authentication protocol, Non-interactive Zero-Knowledge Proof, Contact tracing, Tracking Attacks, COVID-19.

1. INTRODUCTION

Contact tracing is a technology used to reduce the spreading of infectious diseases, such as coronavirus disease 2019 (COVID-19). However, without contact tracing applications and social distance, the virus can still unfold. That leads to more people becoming sick and infecting the most vulnerable. Contact tracing applications are essentially using Bluetooth technology beaconing to alert proximity contact [1-3].

Bluetooth low energy (BLE) [4] is a lightweight subset of classic Bluetooth developed by the Bluetooth Special Interest Group (SIG). Most mobile devices have an integrated BLE module used in proximity applications and connectivity to peripheral devices. Like other wireless systems, BLE has a detection method called advertisement [5]. For BLE devices' advertisement, devices regularly broadcast advertisement packets filled with various information that promotes the available features, services, and BLE device address (BD_ADDR). These packages

communicate as plaintext, which opens up new possibilities for passive tracking of users during their daily activities [6, 7].

Unfortunately, a security vulnerability in the BLE protocol stack [5] leads to many attacks. For example, the Blue Printing Attack is implemented by linking the information disclosed about the device to obtain extra information such as developer, device model, and firmware version [8, 9]. This attack can only be carried out when the device's BD_ADDR is broadcasted in the advertisement stage [9]. An attack vector called BlueBorn [9] spreads quickly between the proximity devices over the air via Bluetooth, and a hacker doesn't have to be compelled to combine with the target device. The hacker needs only the broadcasted device's addresses (BD_ADDR) to penetrate and control mobile phones. That attack vector affects almost 5.3 billion devices across Android, Windows, Linux, and iOS [10]. Other attacks, such as the Backdoor attack, also exploited the same vulnerability in the BLE protocol stack. As a consequence of the attacks, the collected mobility information can be exploited for user profiling, which threatens the user's privacy and affects the user usages of contact tracing applications.

This paper considers the tracking problem of a user while enabling his Bluetooth module in his smart device while using the contact tracing application. This tracking attack exploits a security vulnerability in the Bluetooth protocol stack [5]. This vulnerability is the broadcasting of the device's address (BD_ADDR) during the advertisement stage of the protocol [5]. Therefore, a random address must be used instead of BD_ADDR to mitigate these attacks. The challenge in stopping sending (BD_ADDR) affects the authentication process, which depends on this address as one of its inputs.

We suggest a novel authentication protocol based on non-interactive zero-knowledge proof to overcome that challenge and mitigate tracking attacks. The proposed authentication protocol can replace the one in the current BLE protocol stack [5], which depends on BD_ADDR, and so a random address can be broadcast instead of BD_ADDR in the advertisement packages and avoid the leakage of information used in the tracking attack.

Our suggested authentication protocol is based on the zero-knowledge proof (ZKP) technique [11], which verifies the knowledge of specific information without disclosure anything about it. However, authentication protocols based on the ZKP technique use several challenge-response messages between parties, leading to communication overhead. The authentication protocol in the BLE stack [5] depends on the challenge-response technique. Therefore, the concept of non-interactive ZKP (NIZKP) [12, 13] will be used to reduce the communication overhead; only one message is used for authentication between two parties, which satisfies our purpose to replace the authentication protocol in the BLE protocol stack [5] with a minimum number of exchanged messages between parties to enhance communication efficiency.

The following are the paper's contributions:

A NIZKP-based authentication protocol for Bluetooth–based contact tracing applications is designed and proposed to increase user's privacy and avoid tracking attacks, along with strong security robustness and adequate system performance.

A secure NIZKP-based authentication mechanism is being developed to improve the privacy of COVID-19 contact tracking apps that use Bluetooth to detect proximity. The proposed authentication protocol used only one message for authentication without the other party's BD_ADDR to mitigate tracking attacks.

A tracking attack is performed by Kali Linux essential tools to exploit the security vulnerability of the BLE protocol stack to show the importance of our suggested protocol to eliminate this vulnerability.

The proposed authentication protocol is subjected to security analysis. The proposed protocol allows for mutual authentication while still preventing tracking attacks.

Comparing the proposed protocol's efficiency to that of the existing protocol [5] reveals satisfactory.

We hope that the results of our systematic review and specific recommendations will lead to the creation and implementation of applications against COVID-19 and help governments and the application development industry create safe and privacy-conserving contact tracing applications.

The remainder of the paper is organized in the following manner. Some preliminaries are mentioned in section 2. Section 3 details the complete structure of our suggestedprotocol. Section 4 looks at the security analysis of our proposed scheme. Section 5 demonstrates a functional comparison with the candidate scheme. Finally, Section 6 brings this paper to a conclusion.

2. PRELIMINARIES

In ZKPs, we have a Prover$\boldsymbol{P}$, Verifier $\boldsymbol{V}$, and a statement that can be true or false. $\boldsymbol{P}$ ensues to know that this statement is true and wants to convince $\boldsymbol{V}$. They interact to send some messages back and forth, and in the end, $\boldsymbol{V}$ is convinced and knows that the statement is true without knowing any information about it. Thus, there are both interactive and non-interactive zero-knowledge proofs (NIZKP).

Our suggested protocol is based on a NIZKP, which can be formalized as follows.

Let $\{0,1\}^*$indicates the set of all strings and $\boldsymbol{W}$ represents a witness, for a language $L \subseteq \{0,1\}^*$, a pair of probabilistic turningmachines (P,V), in which P has probabilistic polynomial-time power and V has deterministic polynomial time power, is said to be a non-interactive zero-knowledge proof system of the language L if it fulfills the following conditions related to correctness and security against malicious provers and verifiers:

Completeness that it should be possible to convince the verifier that the statement is true. With the Prover has a witness, it is soundness that if this statement is false, it should not be possible to convince the verifier, and finally, the zero-knowledge that the proof should reveal nothing.

3. SUGGESTED AUTHENTICATION PROTOCOL

This section of the paper will explain the suggested authentication protocol that we assume will replace the authentication protocol in the Bluetooth protocol stack [5]. The reason for the proposal, as previously explained, is to reduce the security vulnerability in the Bluetooth protocol stack [5]. The exploit of that vulnerability leads to the success of all tracking attacks on the user and violate his privacy due to the advertisement of BD_ADDR, so we suggest broadcasting a random address as an alternative to BD_ADDR, and this random address is changed every communication session to increase the privacy of users and reduce the success of these attacks. Unfortunately, stopping exchanging BD_ADDR between users leads to the malfunction of the current authentication protocol in the Bluetooth protocol stack [5] because it is one of its inputs. So, we suggest a NIZKP-based authentication protocol that does not depend on BD_ADDR of

other devices and uses only one message for authenticating one device to the other to reduce the communication overhead of this stage in the Bluetooth protocol stack.

The suggested scheme is as follows:

Step1. Public parameters of the system.

1. Let $\mathbb{G}$ is a finite cyclic group of prime order p defined by the NIST p-256 curve.
2. Let us define two hash functions:

$$\mathbb{H}_1: \{0,1\}^* \rightarrow \mathbb{Z}_p$$
$$\mathbb{H}_2: \{0,1\}^* \rightarrow \mathbb{G}$$

Step 2. The Prover generates an authentication message $\mathcal{M}$.

1. The inputs used by Prover to generate $\mathcal{M}$ are
a. BD_ADDR of the Prover's device.
b. RND_ADDR generated by Prover's device for each session.
c. k_{Link}: Key link shared between verifier and Prover from pairing stage in the Bluetooth protocol stack [5].
2. Set$x = \mathbb{H}_1(BD_ADDR), A = \mathbb{H}_2(RND_ADDR), B = \mathbb{H}_2(k_{Link})$
3. Compute xA, xB
4. Select randomly $v \in_R \mathbb{Z}_p$
5. Compute $c = \mathbb{H}_1(xA \parallel xB \parallel vA \parallel vB)$
6. Compute $r = v - cx$
7. Send$\mathcal{M} = (r \parallel xA \parallel xB \parallel vA \parallel vB)$

Step3. Verifier tries to check the correctness of Prover's device authentication $\mathcal{M}$
1. Compute $c = \mathbb{H}_1(xA \parallel xB \parallel vA \parallel vB)$
2. Accept the identity of Prover's device if

$$vA = rA + c(xA)$$
$$vB = rB + c(xB)$$

4. SECURITY ANALYSIS

This section will analyze the security strength of our suggested protocol based on some formal attacks.

4.1. Attacks on the Cryptographic Algorithms

As a start for the security analysis of our protocol, the scheme's security is determined by the hash algorithm selected. A collision attack on a cryptographic hash attempts to find two inputs that produce the same hash value, i.e., a hash collision. In contrast to a preimage attack, which specifies a unique target hash value. Preimage and collision attacks should be avoided when using a cryptographic hash function. This paper suggests utilizing the FIPS-approved hash algorithm SHA-256, which is also used in the BLE standard. Three characteristics make SHA-256 so safe. It is almost impossible to recover the original data from the hash value. A brute-force attack would need to make 2^{256} attempts to produce the original data. Second, two inputs with the same hash value are highly improbable. With 2^{256} possible hash values, the chances of two being the same are infinitesimally, unimaginably small. Finally, the avalanche effect occurs when a slight change to the original data changes the hash value so significantly that it is not apparentthat the new hash value is obtained from comparable data.

Our suggested scheme's security, on the other hand, is dependent on the elliptic curve discrete logarithm problem (ECDLP) used to conceal the commitment process. The elliptic curvesuggested to be used in the implementation is p-256, which is also used in the BLE standard. The P-256 curve was approved by NIST and is used in the encryption process of the BLE standard.

As a result of the standards used: SHA-256 and p-256, the security of cryptographic operations is assured.

4.2. Man in the Middle Attack

This attack occurs when an opponent secretly relays and even interrupts communication between two parties who assume they communicate directly. That is one of the most common wireless network attack schemes.

A MitM attack will fail in our protocol because there is no retrieval of any information during the protocol execution. In particular, the protocol relies only on one message from Prover to the verifier. Therefore, there is no interaction between them, so other parties cannot intercept transmissions, and communications cannot be impersonated.

Therefore, it would not be successful in attacking the suggested system by MitM.

4.3. Tracking Attack

The attacker has to capture the BD_ADDR of the victim's device for the tracking attack, as shown in the next section, to track the user using his device.

Therefore, it would not be successful in attacking the suggested system by tracking the attack. The attacker might intercept beacons, but it cannot obtain information about BD_ADDRwithout breaking the hash function.

5. PERFORMANCE ANALYSIS

This section will show the significance of our contribution by performing a tracking attack using essential tools in the Kali Linux machine. The objective of the attack is to use basic tools in the Kali Linux machine to get the BD_ADDR of the nearby Bluetooth devices and show how the success of our attack can violate the privacy of people by tracking them due to the BD_ADDR exposure, which broadcast from every Bluetooth enabled device. Therefore, to avoid this security vulnerability, we have to use a random address instead of BD_ADDR. Furthermore, to achieve thatgoal, we have to replace the authentication protocol in the Bluetooth stack with our suggested protocol to mitigate this vulnerability.

5.1. Exploiting The Bluetooth Vulnerability

We'll use two built-in tools in kali Linux to explore and do some reconnaissance on the nearby Bluetooth devices. We can track nearby Bluetooth devices if we succeed in capture their BD_ ADDR. This reconnaissance step is helpful if we can either take control of the device, identify a vulnerability, or track the device.

To perform this attack, we need a functional Bluetooth interface and a Kali Linux machine.

Step 1. We use the HCI (Host Controller Interface) tool by typing "*hcitool scan,"* this will use the Bluetooth interface to scan for nearby Bluetooth devices and present their BD_ADDR and devices' names for us to do additional scans or inquiries.

Step 2. We use SDP (Software-Defined Perimeter) tool by typing *sdptool browse* "BD_Address," of course, the BD_ADDR in the command will be the one capture from step 1. SDP tool provides the interface for performing SDP queries on Bluetooth devices and administering a local SDP database. Browse all available services on the devices specified by Bluetooth address as a parameter.

We can do more with this information because most Bluetooth devices do not bother to randomize their broadcasted BD_ADDR, meaning it will be the same all the time. That can be used to track a person from place to place and breach his privacy.

To avoid that, we recommend each device to adverse a random address and replace the authentication protocol of the BLE standard with our suggested protocol for the device's successin communicating without tracking.

5.2. Comparison of Security Features and Performance

This section compares the proposed authentication protocol against the authentication protocol in the BLE protocol stack [5]. The reason for choosing only one protocol to be compared with ours is because, to our knowledge, there is no protocol suggested in the literature for the purpose discussed in this paper. Table 1 shows the comparison of various characteristics between the proposed authentication protocol and the existing protocol.

Table 1. The comparison of various features among our proposed protocol and the existing protocol.

Feature	**BLE Authentication Protocol [5]**	**Proposed Protocol**
Symmetric cryptosystem	Yes	No
Public key cryptosystem	No	Yes
Mutual Authentication	Yes	Yes
Defense against tracking attacks	No	Yes

Based on Table 1, our proposed protocol can provide defense against tracking attacks because there is no exchange of BD_ADDR between devices. Our suggested algorithm is based on theelliptic curve algorithm, while the one in the protocol passed on SAFER+ symmetric algorithm. Both protocols support mutual authentication by applying the same procedures to change the role of the devices.

Table 2. Performance comparison in terms of computation cost among the proposed authentication protocol and existing protocol.

Protocol	**Computation Cost**
BLE Authentication Protocol [5]	$2T_E + 2T_D + 4T_K + 2T_H$
Proposed Protocol	$5T_H + 10T_{SM}$

Table 2 compares the performance of our proposed authentication protocol to the current protocol described in Bluetooth technology specification [5].

Table 3. Notations are used for time consumption on differing computing operations.

Symbol	Description
T_H	The time required to perform the one-way hash function
T_K	The time required to create a key for symmetric cryptosystem
T_E	The time required to perform encryption operation in symmetric cryptosystem.
T_D	The time required to perform decryption operation in symmetric cryptosystem.
T_{SM}	The time required to perform scalar multiplication over Elliptic curve cryptosystem

Table 3 illustrates the mathematical symbols that represent individual time consumption for various computing operations.

The following assumptions are used to compare the computing costs of the proposed authentication protocol and the other current protocol:

1. SHA-3 is the one-way hash function.
2. Elliptic Curve Cryptography (ECC) over the NIST p-192 curve is used in the scalar multiplication operation.
3. AES 256 is the symmetric encryption/decryption cryptosystem.

Table 4. The execution time of several cryptographic operations based on [14-16]

Cryptographic Operation	Execution Time
SHA3 512-bits hash function with 288 bits input sequence	0.2 ms
Symmetric cryptosystem key generation (AES 256)	0.52 ms
Scalar Multiplication over ECC 196-bits curve	1.064 sec
Encryption/Decryption operation of symmetric cryptosystem (AES 256)	0.87 ms

The execution time for various cryptographic operations using Arduino Uno is shown in Table 4 based on measurement results in [14-16]. According to Table 4, the T_H for a 768-bit input, the sequence is approximately 0.5 ms. As the largest input sequence in the proposed protocol for a hash operation is 768-bit, the T_H is not exceeding 0.5 ms. The time T_K to generate the symmetrical key for AES 256 with KDF is 0.52 ms. The time T_{SM} it takes to perform an ECC scalar multiplication over the NIST p-192 curve is 1.064 seconds. With a 256-bit key size, performing AES encryption T_E or decryption T_D takes 0.87 milliseconds.

Based on table 4, our proposed authentication protocol takes 10.641 sec to complete the authentication process. Given that our proposed authentication protocol is intended to protect users' privacy and avoid tracking attacks, its performance is adequate as a trade-off between privacy and using the other protocol, which has 10 sec less with a non-privacy grantee. This 10-sec computation cost came from the scalar multiplication over the ECC 196-bits curve, which can

be reduced by 14% according to [17] or even using the technique suggested in [18] to achieve the operation computation cost few microseconds.

Since our suggested authentication protocol is built for BLE-based applications, our suggested authentication protocol's communication cost is:

$$T_c = \frac{L}{BW}$$

$T_c$denotes the total transmission time required to send the authentication message from Prover to Verifier, which equals the division of the length of sending message L by the BLE communication channel bandwidth BW. According to the above assumption, the authentication message consists of 5 parts with 960 bits. According to the BLE 4.2 specification [4], the theoretical bandwidth of a BLE channel is 236.7 Kbit/s. However, using the Bluetooth sniffer module CC2540, the actual measurement of BLE 4.2 throughput is 57.8Kbit/s [19]. Therefore, our suggested authentication protocol will take 16.6 ms to send the authentication message in one authentication session based on the actual BLE 4.2 throughput.

6. CONCLUSIONS

As most contact tracing applications feature BLE communication technology, users should be encouraged to use this type of application by increasing the privacy-preserving. A secure NIZKP-based authentication protocol for BLE communication is proposed in this paper to mitigate the usage of a security vulnerability in the BLE protocol stack. Furthermore, to mitigate the tracking of nearby BLE devices, we designed our protocol to avoid the exchange of BD_ADDR between devices that were used for tracking nearby devices, as we have shown in our attack, which depended on essential tools of the KALI Linux machine. The proposed authentication protocol requires about 10 sec for the computing operations within one authentication session based on the performance analysis. The proposed protocol's communication cost requires 16.6 ms for the proposed protocol to send the only communicating message over the BLE channel ultimately. In brief, the proposed authentication protocol for contact tracing applications is adequate over the current authentication protocol used in the BLE protocol stack due to the privacy introduced and avoiding tracking.

REFERENCES

[1] Blog.google, 2020. [Online]. Available:https://www.blog.google/documents/58/Contact_Tracing_-_Bluetooth_Specification_v1.1_RYGZbKW.pdf.

[2] B. Sowmiya, V. Abhijith, S. Sudersan, R. Sakthi Jaya Sundar, M. Thangavel and P. Varalakshmi, "A Survey on Security and Privacy Issues in Contact Tracing Application of Covid-19", SN Computer Science, vol. 2, no. 3, 2021. DOI: 10.1007/s42979-021-00520-z.

[3] L. Ferretti et al., "Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing," Science, vol. 368, no. 6491, p. eabb6936, 2020. DOI: 10.1126/science.abb6936.

[4] Bluetooth SIG. 2019. Bluetooth Core Specification Supplement v8.0. https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457081 Accessed: 2021-08-30.

[5] Bluetooth SIG. 2019. Bluetooth Core Specification v5.1. https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457080 Accessed: 2021-08-30.

[6] T. Issoufaly and P. U. Tournoux, "BLEB: Bluetooth Low Energy Botnet for large scale individual tracking," 2017 1st International Conference on Next Generation Computing Applications (NextComp), 2017, pp. 115-120, DOI: 10.1109/NEXTCOMP.2017.8016185.

[7] D. Oosterlinck, D. Benoit, P. Baecke and N. Van de Weghe, "Bluetooth tracking of humans in an indoor environment: An application to shopping mall visits," Applied Geography, vol. 78, pp. 55-65, 2017. DOI: 10.1016/j.apgeog.2016.11.005.

[8] N. Ibn Minar, "Bluetooth Security Threats and Solutions: A Survey," International Journal of Distributed and Parallel Systems, vol. 3, no. 1, pp. 127-148, 2012. DOI: 10.5121/ijdps.2012.3110.

[9] A. Lonzetta, P. Cope, J. Campbell, B. Mohd, and T. Hayajneh, "Security Vulnerabilities in Bluetooth Technology as Used in IoT," Journal of Sensor and Actuator Networks, vol. 7, no. 3, p. 28, 2018. DOI: 10.3390/jsan7030028.

[10] M. Almiani et al., "Bluetooth Application-Layer Packet-Filtering For Blueborne Attack Defending," 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC), 2019, pp. 142-148, DOI: 10.1109/FMEC.2019.8795354.

[11] E. Morais, T. Koens, C. van Wijk and A. Koren, "A survey on zero-knowledge range proofs and applications," SN Applied Sciences, vol. 1, no. 8, 2019. DOI: 10.1007/s42452-019-0989-z.

[12] H. Wu and F. Wang, "A Survey of Noninteractive Zero-Knowledge Proof System and Its Applications," The Scientific World Journal, vol. 2014, pp. 1-7, 2014. DOI:10.1155/2014/560484.

[13] A. De Santis and G. Persiano, "Zero-knowledge proofs of knowledge without interaction," Proceedings., 33rd Annual Symposium on Foundations of Computer Science, 1992, pp. 427-436, DOI: 10.1109/SFCS.1992.267809.

[14] G. Pereira, R. Alves, F. Silva, R. Azevedo, B. Albertini, and C. Margi, "Performance Evaluation of Cryptographic Algorithms over IoT Platforms and Operating Systems," Security and Communication Networks, vol. 2017, pp. 1-16, 2017. DOI: 10.1155/2017/2046735.

[15] G. Singh Tanwar, G. Singh, and V. Gaur, "Secured encryption—concept and challenge," InternationalJournal of Computer Applications, vol. 2, pp.89–94, 2010.

[16] K. Yeh, C. Su, K. Choo, and W. Chiu, "A Novel Certificateless Signature Scheme for Smart Objects in the Internet-of-Things," Sensors, vol. 17, no. 5, p. 1001, 2017. DOI: 10.3390/s17051001.

[17] C. Guo and B. Gong, "Efficient scalar multiplication of ECC using SMBR and fast septuple formula for IoT," EURASIP Journal on Wireless Communications and Networking, vol. 2021, no. 1, 2021. DOI: 10.1186/s13638-021-01967-7.

[18] M. Anagreh, E. Vainikko, and P. Laud, "Accelerate Performance for Elliptic Curve Scalar Multiplication based on NAF by Parallel Computing," Proceedings of the 5th International Conference on Information Systems Security and Privacy, pp. 238-245, 2019. DOI: 10.5220/0007312702380245.

[19] A. Yohan, N. Lo and D. Winata, "An Indoor Positioning-Based Mobile Payment System Using Bluetooth Low Energy Technology," Sensors, vol. 18, no. 4, p. 974, 2018. DOI: 10.3390/s18040974.

AUTHORS

Ali M. Allam received his Ph.D. degree in Communication Engineering from Helwan University in 2008. From 2016 to current works as an associated professor in the communication department at Helwan University. His research interests include wireless communication, network security, and cryptography.