SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

SAKTI©: Secured Chatting Tool Through Forward Secrecy

Azni Haslizan Ab Halim[1,2,*], Farida Ridzuan[1,2], Nur Hafiza Zakaria[1,2], Abdul Alif Zakaria[3], Najwa Hayaati Mohd Alwi[1,2], Sakinah Ali Pitchay[1,2], Ismail Az-Zuhar[2], Ahmed A AlSabhany[4]

[1] Faculty of Science and Technology, Universiti Sains Islam Malaysia, Bandar Baru Nilai, 71800 Nilai, Negeri Sembilan, Malaysia
[2] Cyber Security and Systems (CSS) Research Unit, Faculty of Science and Technology, Universiti Sains Islam Malaysia, Bandar Baru Nilai, 71800 Nilai, Negeri Sembilan, Malaysia
[3] Cybersecurity Malaysia, Menara Cyber Axis, 63000 Cyberjaya, Selangor, Malaysia
[4] Computer Science Department, Almaarif University College, Ramadi, Anbar, Iraq

ARTICLE INFO

Article history:
Received 27 December 2023
Received in revised form 29 May 2024
Accepted 24 June 2024
Available online 25 July 2024

Keywords:
Secured chatting; Forward secrecy; Elliptic curve Diffie Hellman ephemeral

ABSTRACT

The critical issue of academic misconduct is of utmost importance in the field of education and understanding whistleblowing behaviour can be a potential measure to effectively address this issue. This paper highlights the benefits of using the Tree-based Pipeline Optimization (TPOT) framework as a user-friendly tool for implementing machine learning techniques in studying whistleblowing behaviour among students in universities in Indonesia and Malaysia. The paper demonstrates the ease of implementing TPOT, making it accessible to inexpert computing scientists, and showcases highly promising results from the whistleblowing classification models trained with TPOT. Performance metrics such as Area Under Curve (AUC) are used to measure the reliability of the TPOT framework, with some models achieving AUC values above 90%, and the best AUC was 99% by TPOT with a Genetic Programming population size of 40. The paper's main contribution lies in the empirical demonstration and findings that resulted in achieving the optimal outcomes from the whistleblowing case study. This paper sheds light on the potential of TPOT as an easy and rapid implementation tool for AI in the field of education, addressing the challenges of academic misconduct and showcasing promising results in the context of whistleblowing classification.

1. Introduction

The current usage of smartphones has given people across the globe the ability and the opportunity to store important and confidential data within their handheld devices [1]. The term "confidential" describes the nature of the data being stored implies that the main objective of its safe keeping is to ensure that personal data is never leaked or stolen at all costs. With that being said, it is essential that every smartphone user's private information is always kept under the radar and as far away as possible from unauthorized or unwanted parties [2]. Other than storing data, it is also important to realize that smartphone users might also want to transport or transmit highly classified

* *Corresponding author.*
E-mail address: ahazni@usim.edu.my

https://doi.org/10.37934/araset.49.1.5462

information to other intended parties which could quite literally be anything subjectively crucial, depending on the different contexts and groups of the people involved.

With the intention to share data between trusted parties, smartphones offer the ability to instantaneously message someone using a specific application to facilitate the data transportation process such as WhatsApp and Telegram. According to NengTang and HuiLin [3], there has been a large-scale implementation of instant messaging practices within corporations where the employees are be able to convey messages through their smartphones by conversing within a group or on a one-to-one basis for collaboration purposes. Thus, the securities of a conversation involving two parties are alarming. SAKTI can be best to be used in the industry that need highly secured conversation tool such as military, finance and business.

This paper proposed a Secured App for Encrypting Text Information also known as SAKTI to secure conversation involving two parties using identical cryptographic keys for both the encryption and decryption processes [4]. The objective of this paper is to develop a mobile application with ECDHE key distribution protocols installed to secure the communication between the contacting parties. This mobile application is equipped with an asymmetric key exchange algorithm to be used in any type of symmetric encryption known as perfect forward secrecy (PFS). The best performing and most compatible set of schemes will be chosen to fulfil the security needs of the application that will only be available on Android based mobile phones.

2. Related Works

Perfect Forward secrecy (PFS) is defined as a situation where there is a guarantee for the protection of the previously generated session keys from being compromised as a result of the leakage of a long-term private key [5-8]. According to Avoine *et al.*, [6] they acknowledged that the public key cryptography has generally been regarded as the only key exchange scheme being able to provide PFS which they consider to be too resource hungry for resource constrained devices. Nevertheless, they proposed a key exchange protocol which would go against the norm that entirely relies on symmetric keys to provide PFS. They claimed that their processes involve the constant evolution of a master key which would redefine the previous master keys into the newer versions where it would not be possible for the predecessors to be reverse engineered using their successors.

Other than that, Wang *et al.*, [7] also recognized asymmetric cryptography as the go-to approach to be implemented in a wireless sensor network in order to gain forward secrecy. In their proposed approach, they affirmed that they would integrate an asymmetric key scheme in order to provide forward secrecy. However, they did note that some of the past researchers have solely relied upon symmetric ciphers which were more efficient than their asymmetric counterparts when it came to resource consumption as they had expected, but it was at the cost of forward secrecy. Furthermore, Li *et al.*, [5] proposed the utilization of ephemeral secret tickets by the servers as a means to transport the ephemeral public key. According to them, the generation of the common session key is derived from a long-term private key and random ephemeral secret which means that it would not be enough for an attacker to only steal the private keys as they would also need to calculate the value of the ephemeral secrets, thus enforcing forward secrecy.

Lu *et al.*, [8] proposed a forward secrecy scheme which utilizes the Elliptic Curve Cryptography point multiplication for the session key production. They claimed that a compromised private key would not matter as the adversaries must solve both the Elliptic Curve Discrete Logarithm Problem and the Elliptic Curve computational Diffie-Hellman Problem which are considered to be unsolvable in order to obtain the parameters that are related to the session keys themselves. Finally, Yang *et al.*, [9] stated that they would not be utilizing asymmetric cryptography to gain forward secrecy and

instead chose to adopt a Dynamic Authentication Credential (DAC) framework approach. It was emphasized by them that the DAC would continuously evolve for every unique session after the session key is successfully generated and will only be utilized once per session.

3. Research Methods and Design

This section describes the Secure App for Encrypting Text Information (SAKTI) design diagrams to give a clear picture of how the system works. The system comprised of three phases as shown in Figure 1 include: registration, key exchange, and transfer data phase. For each one of the phases is given to provide some context and meaning to the events that take place within the illustrations.

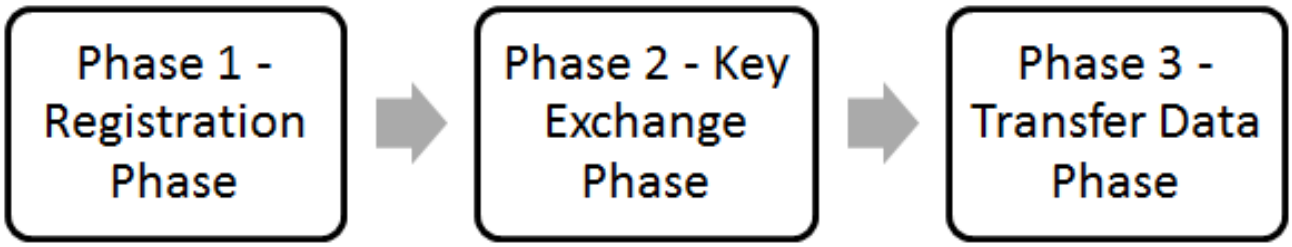

Fig. 1. SAKTI System Process

3.1 SAKTI Registration Phase

For initial use of the SAKTI, users must complete a registration process for authentication at the login page. Before the Transfer data phase, a pair of public keys must be generated and exchanged with the selected receiver. This operation may succeed or fail based on underlying verification process. In the event of success, the receiver's public keys are retrieved to generate a session key for data transmission. If unsuccessful, the process iterates until success is achieved. The Receive data phase mirrors the Transfer data phase, differing primarily in the timing of public key pair generation and sender's public key retrieval. After successful session key generation, the data retrieval process commences. Figure 2: shows registration and login interface for first time user.

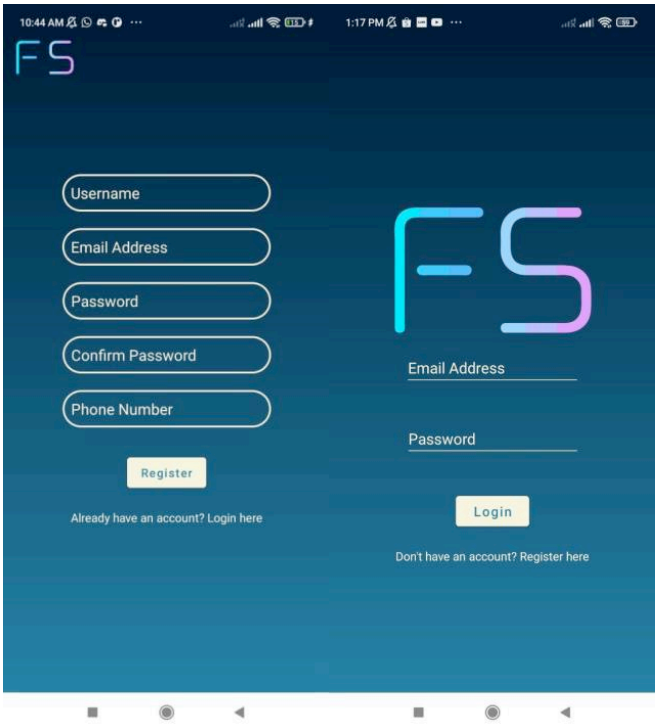

Fig. 2. SAKTI: Registration and Login Page

3.2 SAKTI Key Exchange Phase

SAKTI uses Elliptic Curve Diffie Hellman Ephemeral (ECDHE) as the key exchange algorithm to provide forward secrecy [10] and LAO 3D Algorithm for encryption [11]. Figure 3 shows key exchange process takes place before secure conversation started.

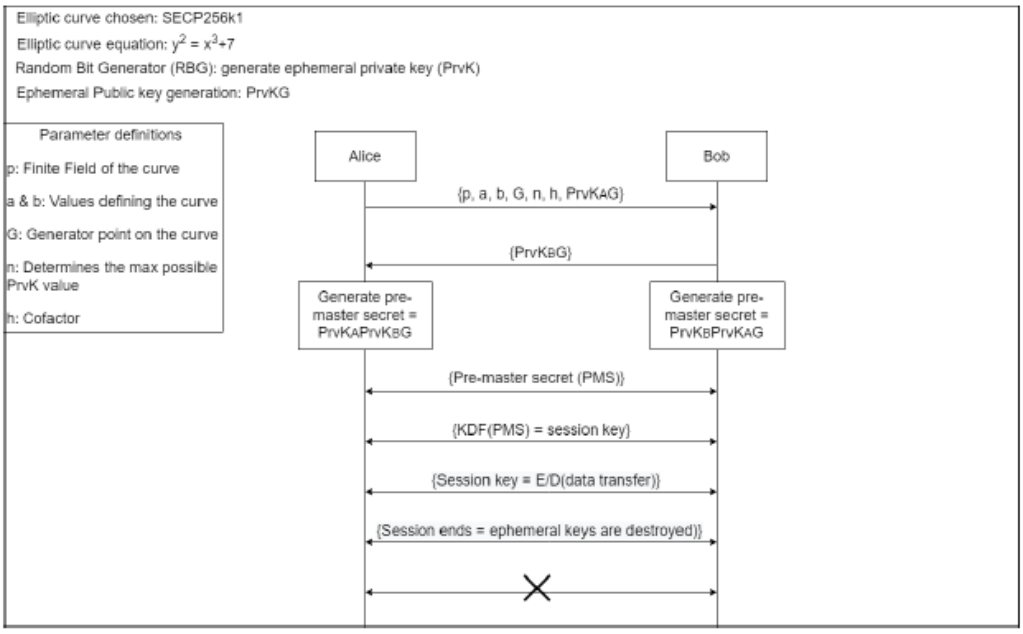

Fig. 3. Key Exchange Process

Before the key exchange process started, both sender (Alice) and Receiver (Bob) independently generate ephemeral private keys, denoted as PrvKA and PrvKB, employing a secure random bit

generator. Subsequently, they compute their respective public keys using ECC (Elliptic Curve Cryptography) multiplication algorithms, incorporating the parameters PrvK (derived from their private keys) and the generator point G. Once Alice, as the conversation initiator, acquires her public key, she dispatches the domain parameters to Bob, as the intended recipient. The most important part is the generated public key, which pivotal in the computation of the pre-master secret key [12].

In response, Bob reciprocates by sending his private key to Alice. Once both parties possess each other's public keys, they jointly calculate a shared pre-master secret key denoted as PrvKAPrvKBG. This shared secret subsequently undergoes processing via a Key Derivation Function (KDF) to derive the shared session key, also known as the symmetric encryption key. These session keys enable secure data transfer and retrieval—permitting one party to transmit data while the other party receives it.

At the end of the session, the ephemeral public and private key pairs are securely deleted. For every new session, fresh ephemeral key pairs and a unique session key are generated. This meticulous process ensures the attainment of perfect forward secrecy, as each session employs distinct key sets with no inter-session correlation, safeguarding past conversations from potential leaks [13]. Moreover, this security architecture employs the Ephemeral Diffie Hellman approach, wherein secret keys are created at the user's end and subsequently obliterated upon session termination.

3.3 SAKTI Transfer Data Phase

The transfer data feature allows the user to create a public key for the key exchange process which will be used as an input to create a secret session key. The session key will then be used to encrypt the data intended to be transferred to the device of a particular friend [14]. A friends list will appear after the transfer data feature is selected which will prompt the user to select a particular friend that they wish to send data to. After making a choice, the user is greeted with a key icon that they must press in order to generate a public key.

During the key transmission process, there will only be three possible outcomes that a user can expect to achieve. These three outcomes are the successful, unsuccessful or declined state of the public key transference which fully depends on the action of the person on the receiving end. If said person chose to decline the key transfer attempt, the original sender will be notified that the key transference request was declined. Other than that, If the receiver manages to fetch the key within a 30 second window, the application will notify the key’s sender that the transfer was successful and vice versa. A successful key transfer will reveal a button to the user to press in order to start the data transferring process.

4. Results and discussion

The algorithm used in SAKTI is the ECC key exchange algorithm which is utilized in the main application function for the key exchange process. Table 1 shows comparison between generation test results of ECC and RSA. The actual results for the key generation functional testing match with the expected results, it is safe to say that the key generation process takes place whenever it is supposed to. So, for the case where the key pair generation does occur, the time taken to generate said key pairs must also be taken into account as it plays a vital role in determining how well the application will be able to perform. As a point of reference, Table 1 and Table 2 are provided to give some context to the issue at hand while Table 6.10 provides the actual results of the ECC key generation times.

Table 1
Key Generation between ECC vs RSA

Key Size (bits)	ECC Algorithm Key Gen Time (ms)	RSA Algorithm Key Gen Time (ms)
160	252	-
224	262	-
256	270	-
384	282	-
512	312	654
1024	-	872
2048	-	1996
3072	-	16692

From the Table 1, it is clear differences in terms of the speed at which the key generation process takes place. To make things simpler, the same or similar key length should be compared such as the 256-bit key from Table 1, the 233 and 283-bit keys from Table 2, and the 256-bit keys from Table 3. From here, it can be seen that the times are 270 milliseconds for the 256-bit key in Table 6.8, 0.18 seconds and 0.27 seconds for the 233 and 283-bit keys in Table 6.9 and the average value of 606.5 milliseconds for the 256-bit keys in Table 3. It must be noted here that due to the difference in processing power and other relevant factors such as the ECC curve choice, it is not a surprise that the key generation results vary from one test to another [15].

Table 2
Key Generation Time Analysis

Key Length (bits)	ECC Algorithm Key Gen Time (s)	RSA Algorithm Key Gen Time (s)
162	0.08	-
233	0.18	-
283	0.27	-
409	0.64	-
571	1.4	-
1024	-	0.16
2240	-	7.47
3072	-	9.8
7680	-	133.9
15360	-	679.06

While it is true that the actual results in Table 3 are slower when compared to the results in Table 6.8 and 6.9, the main takeaway here is that ECC is still more viable than RSA in terms of the key generation times as it can be seen in Table 1 where a 3072-bit RSA key pair is generated in 16692 milliseconds while it takes 9.8 seconds to generate a 3072-bit RSA key pair in Table 2. Even though it does not seem fair to compare the key generation times of a 256-bit ECC keys against a 3072-bit RSA key, the fact that cannot be overlooked here is that a 256-bit ECC key has the ability to provide the same level of security as a 3072-bit RSA key. This notion is supported by Choi *et al.,* [16] who claimed that a 128-bit level of security can be achieved by utilizing a 128-bit AES key, 256-bit ECC keys or 3072-bit RSA keys. With that, it is clear that in terms of overall performance, the shorter length ECC keys will take less time to generate than the longer RSA keys.

Table 3
Actual ECC 256-bit key generation results

Test No	ECC Algorithm Key Gen Time (ms)
1	607
2	595
3	618
4	686
5	565
6	616
7	603
8	566
9	614
10	595

In the realm of contemporary communication, various chatting tools have emerged, each boasting its unique features and security protocols. WhatsApp, a widely-used platform, offers end-to-end encryption, ensuring the privacy of messages exchanged [17]. However, its association with Facebook raises privacy concerns due to the social media giant's history. Telegram on the other hand provides end-to-end encryption in secret chats and introduces a self-destructing message feature, yet its default chats lack the same level of security [18]. Signal, renowned for its commitment to privacy, stands out with universally applied end-to-end encryption, though its user base might be smaller compared to mainstream alternatives. Compare with SAKTI, it offers application layer security where users of this tool will be able to encrypt and send encrypted sentences to other users to convey messages securely. The third party could not be able to read the messages and only intended recipient can open the message with a secret key. This is only can be achieved through forward secrecy in which other chatting tools did not apply in their applications [19]. Since forward secrecy is achieved in this application, the most damage than an attacker can do is uncover one session worth of encrypted sentences which will not at all effect the previous sessions and session keys. The key distribution algorithm selection utilized in the application can be used as future reference for developers who are looking for a way to achieve perfect forward secrecy [20]. In selecting a chatting tool based on security, users must consider their specific needs and the nature of the information exchanged, staying vigilant for updates and improvements in the rapidly evolving landscape of communication platforms such as IoT [21].

4. Conclusions

Within this project, it is crucial to address specific limitations for future improvements, to ensure the research continuity and ease of maintenance. The primary limitation revolves around the research inability to verify user identities during communication which leaving room for potential man-in-the-middle attacks. This vulnerability poses a significant threat, as attackers can manipulate transmitted data, jeopardizing its integrity. Another constraint is the application's exclusivity to Android devices, limiting its user to use the chatting tool. Finally, the Forward Secure application operates solely in a one-way communication flow, allowing only one party to send data while the other can solely access previously transmitted data without mutual capabilities. For future enhancement, SAKTI will consider adding digital signatures and certificates using advanced algorithms like RSA or ECDSA to prevent man-in-the-middle attacks. This step ensures user identity verification and data integrity where it can improve overall security.

In conclusion, adopting the ECDHE algorithm as a key exchange method is a viable alternative for SAKTI. It has demonstrated effectiveness in handling forward secrecy, offering improved

performance with its shorter key length. While the research achieved its goals, it is important to note the acknowledged limitations mention above. Hence, special attention should be given to the suggested future work, aiming to propel the current research to the next level and foster continuous improvement.

Acknowledgement
This research was funded by the Malaysian Communications and Multimedia Commission (MCMC) under the Digital Society Research Grants 2023 Cycle 1 (USIM/MCMC/FST/LUAR-K/42623).

References

[1] Almohtasib, Subhi, and Alaa H. Al-Hamami. "Securing Data Communication for Data Driven Applications Using End to End Encryption." *Indonesian Journal of Electrical Engineering and Computer Science* 10, no. 2 (2018): 756-762. https://doi.org/10.11591/ijeecs.v10.i2.pp756-762

[2] Li, Zhenhua, Weiwei Wang, Christo Wilson, Jian Chen, Chen Qian, Taeho Jung, Lan Zhang, Kebin Liu, Xiangyang Li, and Yunhao Liu. "FBS-Radar: Uncovering Fake Base Stations at Scale in the Wild." In *NDSS*. 2017. https://doi.org/10.14722/ndss.2017.23098

[3] Neng-Tang, Huang, and Lee Hui-Lin. "Using instant messaging for collaboration: A study of the relationships among organizational trust, justice, and spirituality." In *Wireless Internet: 11th EAI International Conference, WiCON 2018, Taipei, Taiwan, October 15-16, 2018, Proceedings 11*, pp. 141-147. Springer International Publishing, 2019. https://doi.org/10.1007/978-3-030-06158-6_14

[4] Kılıç, Muhammed Burak. "Encryption methods and comparison of popular chat applications." *Advances in Artificial Intelligence Research* 1, no. 2 (2021): 52-59.

[5] Li, Xiong, Jieyao Peng, Mohammad S. Obaidat, Fan Wu, Muhammad Khurram Khan, and Chaoyang Chen. "A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems." *IEEE Systems Journal* 14, no. 1 (2019): 39-50. https://doi.org/10.1109/JSYST.2019.2899580

[6] Avoine, Gildas, Sébastien Canard, and Loïc Ferreira. "Symmetric-key authenticated key exchange (SAKE) with perfect forward secrecy." In *Topics in Cryptology–CT-RSA 2020: The Cryptographers' Track at the RSA Conference 2020, San Francisco, CA, USA, February 24–28, 2020, Proceedings*, pp. 199-224. Springer International Publishing, 2020. https://doi.org/10.1007/978-3-030-40186-3_10

[7] Wang, Ding, Ping Wang, and Chenyu Wang. "Efficient multi-factor user authentication protocol with forward secrecy for real-time data access in WSNs." *ACM Transactions on Cyber-Physical Systems* 4, no. 3 (2020): 1-26. https://doi.org/10.1145/3325130

[8] Lu, Rongxing, and Zhenfu Cao. "Simple three-party key exchange protocol." *Computers & security* 26, no. 1 (2007): 94-97. https://doi.org/10.1016/j.cose.2006.08.005

[9] Yang, Zheng, Jun He, Yangguang Tian, and Jianying Zhou. "Faster authenticated key agreement with perfect forward secrecy for industrial internet-of-things." *IEEE Transactions on Industrial Informatics* 16, no. 10 (2019): 6584-6596. https://doi.org/10.1109/TII.2019.2963328

[10] Adrian, David, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger *et al.,* "Imperfect forward secrecy: How Diffie-Hellman fails in practice." In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 5-17. 2015. https://doi.org/10.1145/2810103.2813707

[11] Zakaria, Abdul Alif, Azni Haslizan Ab Halim, Farida Ridzuan, Nur Hafiza Zakaria, and Maslina Daud. "LAO-3D: A symmetric lightweight block cipher based on 3D permutation for mobile encryption application." *Symmetry* 14, no. 10 (2022): 2042. https://doi.org/10.3390/sym14102042

[12] Romdhane, Rihem Ben, Hamza Hammami, Mohamed Hamdi, and Tai-Hoon Kim. "A novel approach for privacy-preserving data aggregation in smart grid." In *2019 15th international wireless communications & Mobile Computing Conference (IWCMC)*, pp. 1060-1066. IEEE, 2019. https://doi.org/10.1109/IWCMC.2019.8766472

[13] Banerjee, Soumya, Vanga Odelu, Ashok Kumar Das, Jangirala Srinivas, Neeraj Kumar, Samiran Chattopadhyay, and Kim-Kwang Raymond Choo. "A provably secure and lightweight anonymous user authenticated session key exchange scheme for Internet of Things deployment." *IEEE Internet of Things Journal* 6, no. 5 (2019): 8739-8752. https://doi.org/10.1109/JIOT.2019.2923373

[14] Ali, R. M., and S. N. Alsaad. "Instant messaging security and privacy secure instant messenger design." In *IOP Conference Series: Materials Science and Engineering*, vol. 881, no. 1, p. 012117. IOP Publishing, 2020. https://doi.org/10.1088/1757-899X/881/1/012117

[15] Alrowaithy, Majed Humaid. "Performance-efficient cryptographic primitives in constrained devices." PhD diss., Newcastle University, 2021.
[16] Choi, Soohyeon, Sangwon Shin, Xiaozhu Jin, and Sung Shin. "Secure and low computation authentication protocol for Wireless Body Area Network with ECC and 2D hash chain." In *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, pp. 130-135. 2020. https://doi.org/10.1145/3400286.3418256
[17] Akram, Raja Naeem, and Ryan KL Ko. "End-to-end secure and privacy preserving mobile chat application." In *Information Security Theory and Practice. Securing the Internet of Things: 8th IFIP WG 11.2 International Workshop, WISTP 2014, Heraklion, Crete, Greece, June 30–July 2, 2014. Proceedings 8*, pp. 124-139. Springer Berlin Heidelberg, 2014.
[18] Abu-Salma, Ruba, Kat Krol, Simon Parkin, Victoria Koh, Kevin Kwan, Jazib Mahboob, Zahra Traboulsi, and M. Angela Sasse. "The security blanket of the chat world: An analytic evaluation and a user study of telegram." Internet Society, 2017. https://doi.org/10.14722/eurousec.2017.23006
[19] Unger, Nik, Sergej Dechand, Joseph Bonneau, Sascha Fahl, Henning Perl, Ian Goldberg, and Matthew Smith. "SoK: secure messaging." In *2015 IEEE Symposium on Security and Privacy*, pp. 232-249. IEEE, 2015. https://doi.org/10.1109/SP.2015.22
[20] Gueron, Shay, and Vlad Krasnov. "Fast prime field elliptic-curve cryptography with 256-bit primes." *Journal of Cryptographic Engineering* 5, no. 2 (2015): 141-151. https://doi.org/10.1007/s13389-014-0090-x
[21] Salbiah Zainal, Rasimah Che Mohd Yusoff, Hafiza Abas, Suraya Yaacub, & Norziha Megat Zainuddin. (2021). Review of Design Thinking Approach in Learning IoT Programming. *International Journal of Advanced Research in Future Ready Learning and Education*, 24(1), 28–38.