QRSMS: A SMS client with end-to-end encryption via Quick Response (QR) code

John Mel R. Ramos and Jaderick P. Pabico

I. Introduction

Despite the emergence of instant messaging (IM) and Over-the-top (OTT) messaging applications, short message service (SMS) remains as a popular option for mobile communication. With the first text message sent in 1992 through the Global System for Mobile communication (GSM) network, the technology has seen a lot of growth and development [1]. While mostly used for person-to-person communication [2], the medium has seen other applications such as alert notification [3], [4], empowering businesses and online services [5]–[7], mobile health (m-health) interventions [8], [9], mobile government applications [10], [11], education [12], Two-Factor Authentication (2FA) [13] and more. During the year 2022, Globe Telecom gained Php 8.8 billion revenue from SMS with 86.7 million mobile customers. Furthermore, the company has invested Php 1.1 billion to improve spam and fraudulent SMS detection and prevention [14]. Unreliable internet service as well as the lack of Internet infrastructure in some area remains a challenge in the adoption of online services for communication such as Facebook Messenger, Viber, Telegram, etc. SMS on the other hand is ubiquitous. Anyone with a mobile device and a registered subscriber identity module (SIM) card can send and receive text messages regardless of their platform and cellular provider [15].

Regardless of its wide range of applications and sustained use, however, SMS is not ideal for private communication such as sending sensitive and confidential information as early specifications for SMS does not have security in mind [16]. The European Telecommunications Standards Institute (ETSI) specified the use of A5 algorithm for encryption over the GSM network together with the A3 and A8 algorithm for authentication and cipher key generation to provide security [17] but attacks on the A5/1 and the weaker A5/2 algorithms has been demonstrated even with minimal resources [18]–[20] and even the A5/3 used for Third Generation (3G) network has been compromised [21]. Mobile network operators may also opt not to use any encryption during the transit of message wirelessly [22]. This means that a Man-in-the-Middle (MITM) can sit between the Mobile Equipment (ME) and Base Station Subsystem (BSS) using an International Mobile Subscriber Identity (IMSI) catcher to actively intercept traffic and even request the victim ME to use a weaker or no encryption at all [23]. Additionally, encryption is only performed between the Mobile Equipment (ME) and the Base Station Subsystem (BSS) [17]. Once the message reaches the SMS Center (SMSC), messages are stored and kept in plain text for a while allowing malicious entities to target the SMSC to view the contents of these messages. This makes an end-to-end encryption a huge improvement over what is available in the GSM specifications.

To provide a better security system for SMS, independent researchers have proposed frameworks and protocols that offers end-to-end encryption. PK-SIM [24] provides end-to-end security between the mobile user and service provider through Public Key Infrastructure (PKI). SMSSec [25] was developed using the Java Wireless Messaging API (WAP) and uses a combination of asymmetric and symmetric cryptography to provide security. EasySMS [26] and SecureSMS [7] are completely based on symmetric key cryptography. Similarly, SmartSMS [27] uses symmetric key encryption for m-health application. These protocols however, requires additional resources and infrastructure on the side of the service provider in order to be implemented.

This paper propose an SMS client that uses quick-response (QR) codes for key exchange. Initially developed by Toyota's subsidiary Denso Wave for inventory of vehicle parts manufacturing, QR code has seen many application in today's era [28]. Able to store up to a maximum of 7,089 characters in one code [29] it can be used to encode text, URL, and other data and can be decoded using a mobile device equipped with a camera and the appropriate software which can display a text, connect to a wireless network, open a webpage, or open a linked application [30]. The proposed client will encode the public key and parameters needed to generate a shared secret in a QR code. The code can be shared to the intended recipient through offline (in person) or other secure online platform that allows sending of images.

End-to-end security will be achieved through the combined use of symmetric and asymmetric cryptography as well a key-agreement protocol. Advaned Encryption Standard (AES) will be used for encrypting the message as it provides the most security while being performant [31]. Elliptic Curve Cryptography (ECC) will be used for generating key pairs and Elliptic Curve Diffie-Hellman (ECDH) for generating a shared secret. The use of ECC allows for a smaller key length while providing an comparable security as the Rivest–Shamir–Adleman (RSA) public key cryptosystem [32].

The proposed method will use the android platform to develop an SMS client that can generate keypair, exchange public keys through QR code, and establish an end-to-end encrypted communication.

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

II. Literature Review

Regardless of the insecurity present in SMS, due to its wide spread application, it remains to be of use to many users around the world. To provide better end-to-end security, protocols/frameworks has been proposed.

PK-SIM [24] card is a framework developed based on Public Key Infrastructure (PKI) providing end-to-end encryption between the Service Provider (SP) and the mobile user which contains the PK-SIM card. Their framework is composed of the mobile client, a Certification Authority and Registration Authority (CA/RA), a Secure Access gateway (SAG) found in the service provider's infrastructure and the Mobile Operator (MO) that provides the network for SMS communication. Before the user can send secure messages, the SAG together, with the CA/RA must first validate the identity of the mobile client. A primary key shared between the client and the SAG is generated during the authentication phase. This key is used to establish a session key which is used to symmetrically encrypt/decrypt the message. However, in their approach, end-to-end security is only available between the mobile client and the SAG on the Service Provider's side. Additionally, it is stated that the generation of primary key in the authentication phase takes up to a day or longer. Every time this key expires, the protocol must be performed again from the beginning. Similarly, the SMSSec [25] protocol is designed such that it uses a combination of asymmetric cryptography for authenticating the user and uses symmetric cryptography for the actual exchange of messages. It also falls short such that end-to-end encryption is between the mobile device and the authentication source. Unlike PK-SIM however, SMSSec is a software solution built with the Java Wireless Messaging API (WAP).

In contrast, the EasySMS [26] protocol is designed to provide end-to-end encryption using only symmetric algorithm. In their approach, an Authentication Server (AS) is responsible for the storage of all secret key shared between the AS and the respective Mobile Station (MS) or the user. It is also assumed that there is a shared secret key between the AS and a CA/RA which contains information about every mobile subscriber. Before a SIM card could be used, a subscriber must first register within the CA/RA. With these requirements met, the AS, with the help of CA/RA can now authenticate any MS that wants to send an SMS securely. However, this approach falls under the same disadvantage as the previous protocols. When user A wants to send a message to user B, the AS validates the identity of both users and generate different key for each respective user. This means that the message from user A is encrypted with the key shared between the AS and user A, decrypted in the AS then re-encrypt the message for user B using the key shared between the AS and user B. This approach is also used for the development of SecueSMS [7] but is proposed to be specific for value added services (VAS) and commercial applications. SmartSMS [27] follows a similar concept but instead of the AS being connected to a CA/RA, a trusted third party hosted in a public cloud is responsible for the functions of CA/RA.

In these protocol, apart from being not truly end-to-end encrypted (encryption only occurs between the mobile station and the service provider), they also require a third party for authenticating the user. This means that in order to achieve secure communication, changes within the service provider's and the network itself must be consider. This paper will introduce a different approach that does not require additional infrastructure by having the ciphertext sent through the SMS network while encryption and decryption is performed by the communicating devices.

III. Objectives

To create a secure platform for the exchange of encrypted SMS, the main objective of this research is to develop an SMS client application that can encrypt and decrypt messages and utilize QR codes as means of exchanging keys for generating a shared secret. To be specific, this research aims to do the following:

1. Design and implement a key-agreement protocol using QR code as means for key exchange.
2. Design and implement an SMS client for mobile android device with end-to-end encryption.
3. Conduct a usability study of the SMS client application.

IV. Materials and Methods

A. Key-Agreement Protocol

To achieve end-to-end security, participants in the exchange should be able to generate a shared key known only to them. This process can be further divided into two steps. Keypair generation and shared secret key generation. This two step process is shown in Figure 1. The javax.crypto library contains the KeyAgreement class needed to establish a shared secret while the java.security library will enable us to use the KeyGenerator class which allows the generation of public and private key pair using EC. Since there are no built-in libraries for encoding and decoding of QR code, ZXing [33] and Google's ML kit barcode scanning [34] will be bundled with the application to provide the missing functionalities.

1) Keypair Generation: To be able to generate a shared secret with another user, each party involved must first generated their public and private key pair. For each contact the user wants to exchange keys with, a unique key pair will be generated using Elliptic Curve (EC) algorithm. The generated public key will be encoded to a QR code while the complementing private key will be kept securely in the system and will be used to finalize the key exchange.

2) Shared Secret key Generation: Once each party have their generated QR code, they will give their QR code to the each other and scan the code to retrieve the key. Using ECDH, their private key for that particular user will be fed along with the scanned public key to generate a shared secret. This process is illustrated in fig. 1. The shared secret will be stored inside Android's built in KeyStore to protect the keys from unauthorized use [35].

It should be noted that in order to keep the integrity of the system, an assumption is made that the user will not feed a QR code whose origin is unknown. To ensure that the code being decoded is from the intended contact, the QR code

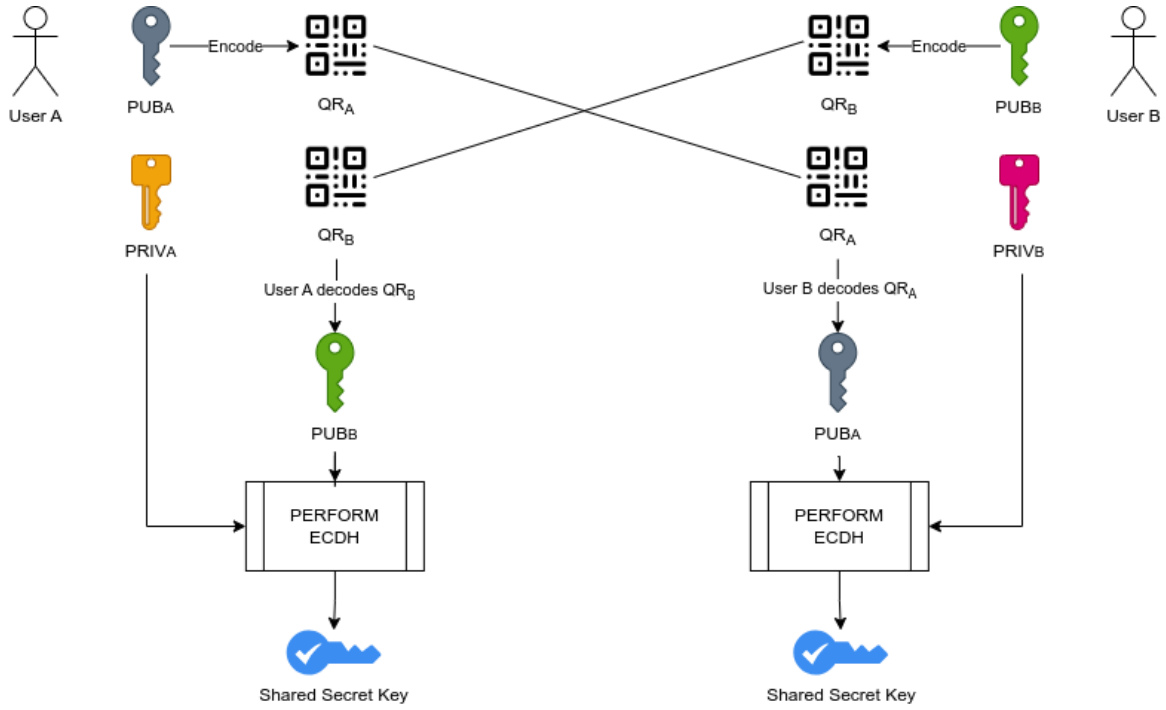

Fig. 1. Overview of key generation and key agreement in QRSMS

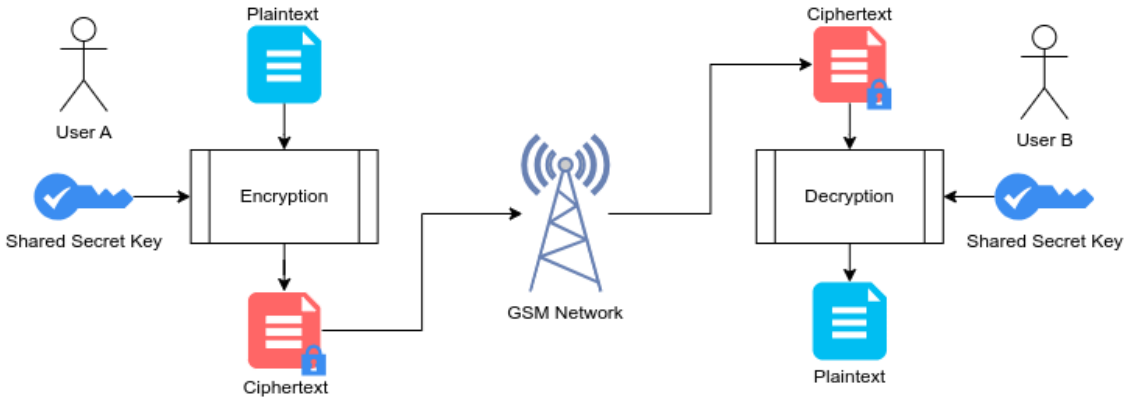

Fig. 2. Overview of sending and decryption of messages in QRSMS

should only come from the other party involved in the secure communication through other means such as offline (in-person exchange) or secure online application. This eliminates the need for authentication methods as the users themselves are the one responsible for ensuring that the participant in the exchange is the intended receipient.

B. Encrypted Messaging

To enable end-to-end encryption, both the sending and receiving party must have the application installed and have exchanged their QR code generated by the application through other means suggested below. In case only one party has the software installed, if they decide to send an encrypted message to a user without the application, or the receipient have not yet established the shared secret, the latter might receive the message in ciphertext, unable to read said text. If the opposite were to occur, the application should be able to determine if the received text was encrypted or not.

When each of the user involved in the key-agreement now having generated a shared secret with their intended contact, the exchange of encrypted SMS can now begin. AES encryption algorithm can be used generate the ciphertext with the shared secret as the key for symmetric cryptography shown in fig. 2. The android platform comes with the android . telephony package contains the classes needed for sending and receiving SMS messages while the javax . crypto contains classes that allows the use ofcryptographic libraries that will help achieve the goals of the application [36].

Equipped with features and support for developing android applications, Android Studio will be used as the integrated

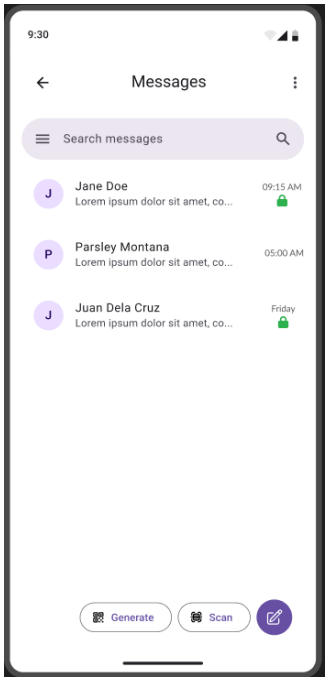

Fig. 3. QRSMS Inbox UI Prototype

development environment (IDE) to facilitate the development of the SMS client. The Kotlin programming language will also be used for the application as it fully supported for android development and it eliminates the verbosity of Java while allowing the use of Java libraries [37].

C. User Testing

Throughout the duration of the development process, the features and interface of the system will be thoroughly tested to ensure that the program can perform its intended functions. However, upon the completion of a minimum viable product, usability test will be conducted to further inspect the working condition off the application as well as analyze users' thoughts with regards to the system. Users will be asked to perform various tasks that involves the following capabilities of the application:

1. Generation of QR code containing the public key to be used for a contact
2. Scanning and decoding of QR code received from a contact
3. Sending of unencrypted and encrypted messages
4. Reading of unencrypted and encrypted messages
5. Management of stored keys

Along with the given tasks are questionnaire that aims to understand users' perspective throughout their usage of the application. Single Ease Question (SEQ) will be asked after completing each task. With measures of central tendency, a general sentiment can be derived on how well the users performed in each of the tasks. This can provide an insight on the ease-of-use of the system. This is followed by a System Usability Scale (SUS) Questionnare to assess the system's perceived usability.

V. Expected Outcome

A. Key-agreement with QR Code

Behind the SMS client's ability to conceal the contents of each messages, the key-agreement protocol's function is vital to generate the secret key that will be used to encrypt each messages. Figure 4 shows the sequence of views that the user can see when they tap the generate QR in the inbox user interface (UI). On the other hand, when scanning a QR code, the sequence of views during this action is shown in figure 5. When users A and B wants to send each other encrypted messages, User A must first generate a QR code intended for user B and provide them the code and vice-versa. Through this exchange, one of the following cases may occur.

1) Case 1: User A and B receives the correct code: In this case, when user A generated a QR code they correctly supplied user B's phone number and gave them the correct QR code. User B does the same and upon scanning each others given code, the key-agreement protocol is done succesfully and came up with the same secret key for both user. Encrypted SMS can now occur.

2) Case 2: User A or User B or Both receives the wrong code: In this case, during generation or sending of QR code, either user A or user B selected the incorrect phone number or supplied the incorrect QR code. If this happened, the application should throw a user friendly error message to allow them to troubleshoot the issue and take appropriate action.

B. SMS Client

Built for the android operating system, QRSMS is an SMS client that enables users to exchange end-to-end encrypted messages while allowing them to send regular SMS if they choose to. To ensure seamless operation by the user, the system should be designed with a simple and user-friendly interface such that users of any background can make operate the application without much difficulty. Figure 3 shows a prototype of the UI upon entering the application in which most of the significant functions of the system should be present and apparent. While users should be able to choose to send a regular or secure SMS, the system should be able to recognize if the message is encrypted or not and transform it if necessary to make it readable by the user.

In general, for the SMS client perform its intended function, the following requirements must be considered. The application should be able to:

1. Encrypt and send SMS message.
2. Decrypt and read SMS message.
3. Generate a private and public key pair then encode it into a QR Code.
4. Decode the QR code to generate the shared secret.
5. Allow users to manage saved keys.

C. Understanding System Usability and Ease of Use

In general, user testing allows for understanding how a system performs in real world applications. Having users test and evaluate the system enables the researchers to recognize

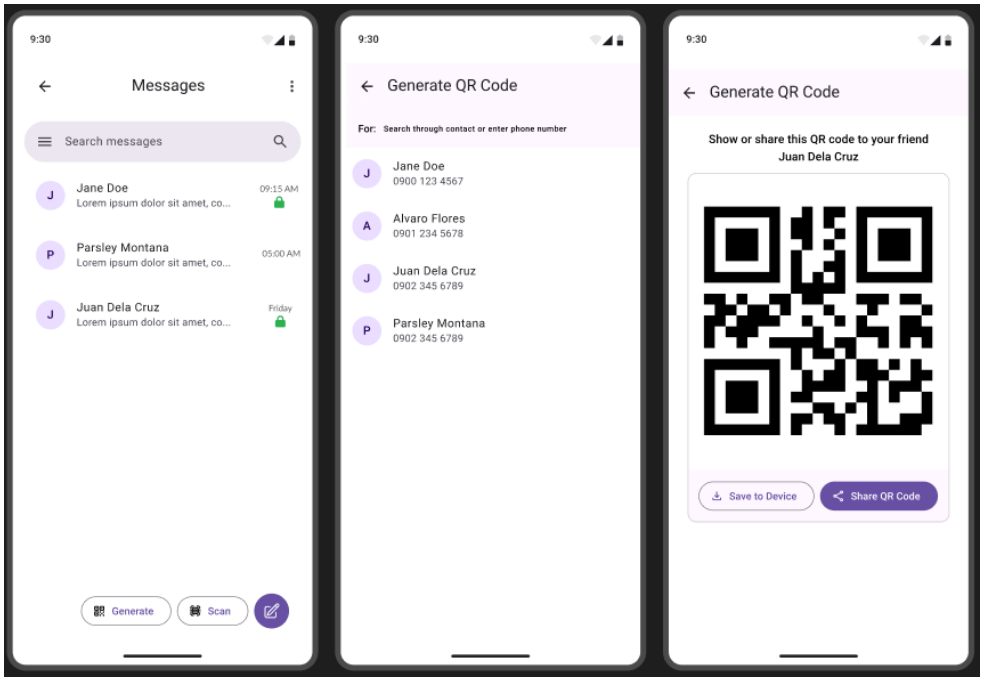

Fig. 4. QR Code generation view prototype

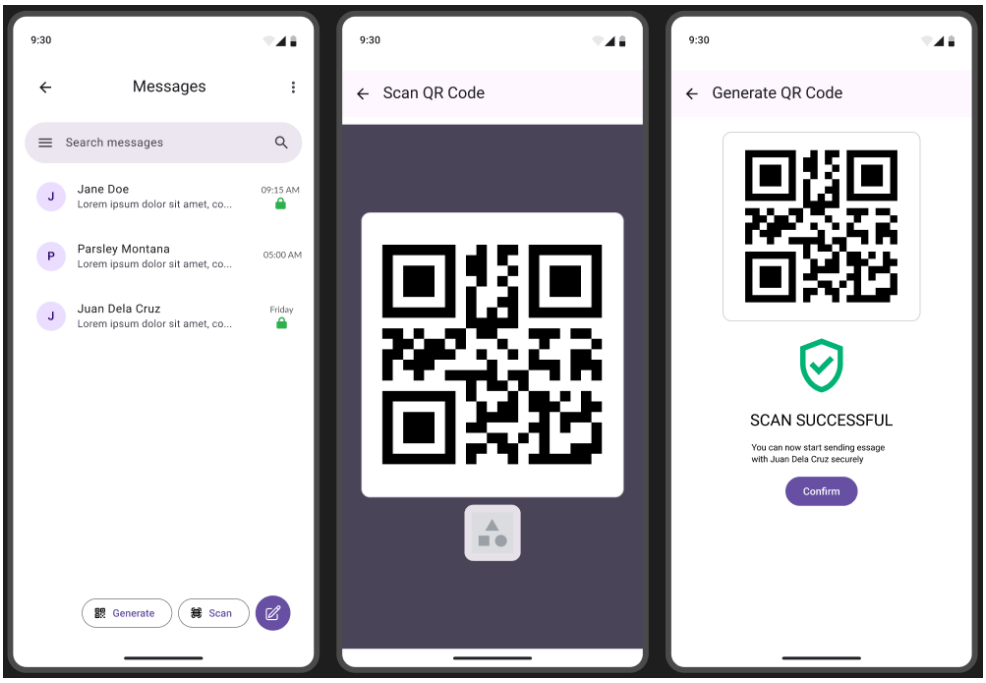

Fig. 5. QR Code scan view prototype

the potential of the proposed approach. With the way the key-agreement protocol is designed, users might initially find this method different from what they are used to. Users has to initiate the exchange themselves with the QR code to simplify the act of inputting of keys. But an intuitive and well designed interface should be able to guide the users to take their intended actions. Because of the popularity of QR code, users generally recognizes it and knows how to use it. However, knowing what aspect of the system users found the most useful and easy to operate can help identify further improvements to be made.

REFERENCES

[1] G. Le Bodic, *Mobile messaging technologies and services: SMS, EMS, and MMS*, 2nd ed. Chichester, West Sussex, England; Hoboken, NJ: J. Wiley & Sons, 2005, ch. 3, p. 47.

[2] S. Herring, D. Stein, and T. Virtanen, *Pragmatics of Computer-Mediated Communication*. Walter de Gruyter, Jan. 2013, ch. 7, p. 162, google-Books-ID: 9cTmBQAAQBAJ.

[3] C. W. Yoo, J. Lee, C. Yoo, and N. Xiao, "Coping behaviors in short message service (sms)-based disaster alert systems: From the lens of protection motivation theory as elaboration likelihood," *Information & Management*, vol. 58, no. 4, p. 103454, June 2021.

[4] S. Azid, B. Sharma, K. Raghuwaiya, A. Chand, S. Prasad, and A. Jacquier, "Sms based flood monitoring and early warning system," *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 15, 2015.

[5] D. Gargaro, "What is sms for business?" Sept. 2023. [Online]. Available: https://www.business.com/articles/sms-for-business/

[6] S. Okazaki and C. R. Taylor, "What is sms advertising and why do multinationals adopt it? answers from an empirical study in european markets," *Journal of Business Research*, vol. 61, no. 1, p. 4–12, Jan. 2008.

[7] N. Saxena and N. S. Chaudhari, "Securesms: A secure sms protocol for vas and other applications," *Journal of Systems and Software*, vol. 90, p. 138–150, Apr. 2014.

[8] J. Tuckerman, K. Harper, T. R. Sullivan, A. R. Cuthbert, J. Fereday, J. Couper, N. Smith, A. Tai, A. Kelly, R. Couper, M. Friswell, L. Flood, C. C. Blyth, M. Danchin, and H. S. Marshall, "Short message service reminder nudge for parents and influenza vaccination uptake in children and adolescents with special risk medical conditions: The flutext-4u randomized clinical trial," *JAMA Pediatrics*, vol. 177, no. 4, p. 337–344, Apr. 2023.

[9] S. L. King, J. Lebert, L. A. Karpisek, A. Phillips, T. Neal, and K. Kosyluk, "Characterizing user experiences with an sms text messaging–based mhealth intervention: Mixed methods study," *JMIR Formative Research*, vol. 6, no. 5, p. e35699, May 2022.

[10] A. Onashoga, A. Ogunjobi, T. Ibharalu, and O. Lawal, "A secure framework for sms-based service delivery in m-government using a multicast encryption scheme," *African Journal of Science, Technology, Innovation and Development*, vol. 8, no. 3, p. 247–255, June 2016.

[11] T. D. Susanto and R. Goodwin, "User acceptance of sms-based e-government services: Differences between adopters and non-adopters," *Government Information Quarterly*, vol. 30, no. 4, p. 486–497, Oct. 2013.

[12] J. B. Hill, C. M. Hill, and D. Sherman, "Text messaging in an academic library: Integrating sms into digital reference," *The Reference Librarian*, vol. 47, no. 1, p. 17–29, July 2007.

[13] A. Rhoda Iyanda and M. Ebenezer Fasasic, "Development of two-factor authentication login system using dynamic password with sms verification," *International Journal of Education and Management Engineering*, vol. 12, no. 3, p. 13–21, June 2022.

[14] "Empowering customers with everyday digital solutions," Globe Telecom, Inc., Apr. 2023. [Online]. Available: https://www.globe.com.ph/content/dam/globe/brie/AboutUs/investor-relations/integrated-report/Globe-2022-Integrated-Report.pdf

[15] E. F. Rosales, "Text messaging remains relevant in philippines," Nov. 2022. [Online]. Available: https://www.philstar.com/business/2022/11/25/2226194/text-messaging-remains-relevant-philippines

[16] *Short Message Service (SMS) for Wideband Spread Spectrum Systems*, 3GPP2 Technical Requirement C.S0015-C, Nov. 2012. [Online]. Available: https://www.arib.or.jp/english/html/overview/doc/STD-T64v6_20/Specification/ARIB_STD-T64-C.S0015-Cv1.0.pdf

[17] J. Schiller, *Mobile communications*, 2nd ed. London: Addison-Wesley, 2011, pp. 120–122.

[18] A. Biryukov, A. Shamir, and D. Wagner, "Real time cryptanalysis of a5/1 on a pc," in *Fast Software Encryption*, ser. Lecture Notes in Computer Science, G. Goos, J. Hartmanis, J. van Leeuwen, and B. Schneier, Eds. Berlin, Heidelberg: Springer, 2001, p. 1–18.

[19] A. Bogdanov, T. Eisenbarth, and A. Rupp, "A hardware-assisted realtime attack on a5/2 without precomputations," in *Cryptographic Hardware and Embedded Systems - CHES 2007*, ser. Lecture Notes in Computer Science, P. Paillier and I. Verbauwhede, Eds. Berlin, Heidelberg: Springer, 2007, p. 394–412.

[20] E. Barkan, E. Biham, and N. Keller, "Instant ciphertext-only cryptanalysis of gsm encrypted communication," in *Advances in Cryptology - CRYPTO 2003*, ser. Lecture Notes in Computer Science, D. Boneh, Ed. Berlin, Heidelberg: Springer, 2003, p. 600–616.

[21] O. Dunkelman, N. Keller, and A. Shamir, "A practical-time attack on the a5/3 cryptosystem used in third generation gsm telephony," Cryptology ePrint Archive, Paper 2010/013, 2010, publication info: Published elsewhere. Unknown where it was published. [Online]. Available: https://eprint.iacr.org/2010/013

[22] *Digital cellular telecommunications system (Phase 2+); Security mechanisms for SIM application toolkit; Stage 2*, European Telecommunications Standards Institute Technical Specification 03.48, Dec. 2001. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/101100_101199/101181/08.08.00_60/ts_101181v080800p.pdf

[23] G. Cattaneo, G. De Maio, P. Faruolo, and U. F. Petrillo, "A review of security attacks on the gsm standard," in *Information and Communication Technology*, ser. Lecture Notes in Computer Science, K. Mustofa, E. J. Neuhold, A. M. Tjoa, E. Weippl, and I. You, Eds. Berlin, Heidelberg: Springer, 2013, p. 507–512.

[24] H. Rongyu, Z. Guolei, C. Chaowen, X. Hui, Q. Xi, and Q. Zheng, "A pk-sim card based end-to-end security framework for sms," *Computer Standards & Interfaces*, vol. 31, pp. 629–641, June 2008.

[25] J. L.-C. Lo, J. Bishop, and J. Eloff, "Smssec: An end-to-end protocol for secure sms," *Computers & Security*, vol. 27, no. 5–6, p. 154–167, Oct. 2008. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167404808000151

[26] N. Saxena and N. S. Chaudhari, "Easysms: A protocol for end-to-end secure transmission of sms," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 7, p. 1157–1168, July 2014.

[27] P. Vijayakumar, S. M. Ganesh, L. J. Deborah, and B. S. Rawal, "A new smartsms protocol for secure sms communication in m-health environment," *Computers & Electrical Engineering*, vol. 65, p. 265–281, Jan. 2018.

[28] S. Tiwari, "An introduction to qr code technology," in *2016 International Conference on Information Technology (ICIT)*, Dec. 2016, p. 39–44. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7966807

[29] What is a qr code. [Online]. Available: https://www.qrcode.com/en/about/

[30] D.-H. Shin, J. Jung, and B.-H. Chang, "The psychology behind qr codes: User experience perspective," *Computers in Human Behavior*, vol. 28, no. 4, p. 1417–1426, July 2012.

[31] S. N. Karale, K. Pendke, and P. Dahiwale, "The survey of various techniques & algorithms for sms security," in *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Mar. 2015, p. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7192943

[32] M. Amara and A. Siad, "Elliptic curve cryptography and its applications," in *International Workshop on Systems, Signal Processing and their Applications, WOSSPA*, May 2011, p. 247–250. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5931464

[33] zxing. [Online]. Available: https://github.com/zxing/zxing

[34] Scan barcodes with ml kit on android. [Online]. Available: https://developers.google.com/ml-kit/vision/barcode-scanning/android

[35] Android keystore system. [Online]. Available: https://developer.android.com/privacy-and-security/keystore

[36] Package index. [Online]. Available: https://developer.android.com/reference/packages

[37] M. Moskala and I. Wojda, *Android Development with Kotlin*. Packt Publishing Ltd, Aug. 2017, pp. 8–9, google-Books-ID: PJZGDwAAQBAJ.