JTH

https://jthkkss.com/
e-ISSN 2805-4431
DOI: https://doi.org/10.53797/jthkkss.v3i2.5.2022

The Development of Mobile Application Security Through Encryption

Aziz, Azrahayu Abdul & Bawamohiddin, Aminah Bibi[1*]

[1]Polytechnic Ungku Omar, Ipoh, 31000, Perak, MALAYSIA

*Corresponding author email: aazrahayu@puo.edu.my

Available online 02 December 2022

Abstract: This paper aims to propose a security solution for android users. Recently, there have been many cases of lost mobile devices since the advent of a new norm during the COVID-19 pandemic. This had impacted victims as it resulted in the loss of sensitive personal information. Therefore, the community must always take precautions to protect personal data from cases of data loss. With the increase in advanced technology, the ISY Mobile Crypt (IMC) application was developed. This mobile security application can store personal data, thus in the event of a device loss data can be compromised and restored. The IMC was developed by using the rapid application development (RAD) approach. The IMC can encrypt and decrypt data which can be obtained through Firebase. This application assists android users to increase their data security by converting plaintext to an alternative type of ciphertext. The IMC requires a key to store, encrypt and decrypt data. The application can benefit android users by keeping their data safe and accessible in the form of images and text. Without a set key no one can access their personal data if their phones are lost or stolen.

Keywords: Mobile application, text encryption, image encryption, ciphertext, text decryption, image decryption

1. Introduction

Today, mobile devices have become an inescapable necessity for carrying out daily tasks. They not only work to communicate, but also help to store and transfer data everywhere. Now that information technology is advancing, the chances of data loss and theft will increase if the device security is not prioritized. A myriad of security methods, techniques, or algorithms was used to secure mobile phone data. One of the techniques that can address the problem is encryption (Lisonek & Drahanský, 2008; Žitovský, 2007). Encryption is a method of securing digital data by combining one or more mathematical procedures with a password to decrypt the data. The encryption process translates information by using algorithms that render the original information unreadable (Chandrashekhar et al., 2022). For example, it can turn plaintext into an alternative kind of ciphertext. When users are given permission to read data, they will employ a binary key to decrypt it. This will return the ciphertext to plain text, allowing authorized users to view the original data (Ranjbar & Mahdi, 2012).

This encryption is an important way for individuals and companies to protect sensitive information from hacking. On the other hand, decryption is the process of returning encrypted data to its original format (Aljazaery et al., 2022). Users can talk privately utilizing encryption algorithms to encrypt and decrypt messages by using this method (Smriti et al., 2022). Protecting encrypted information that is vulnerable is extremely challenging (Abdelmaboud et al., 2022). It is likely to fall into the hands of adversaries who will exploit cryptographic tools to decrypt the data. In addition, security issues occur due to how users commonly take authorization and authentication for granted when designing their systems (Almaiah et al., 2022; Sinha et al., 2022). In particular, in terms of mobile devices, authentication can play a significant influence in reducing and mitigating several unauthorized access and data breaches (Ullah et al., 2019). Despite the employment of many mechanisms at the backend of the system to secure the personal and private data of users, malicious attackers may get access to the data through various methods. The most prevalent and significant attack is ransomware, which involves authorization and authentication on mobile devices (Wang, 2022). Though it is vital to keep data in storage and transit safe from malicious actors, establishing a safe and secure mechanism to access that data is also crucial (Seth et al., 2022). The purpose of authorization is to allow a specific user

*Corresponding author: aazrahayu@puo.edu.my

to use the system once his identity has been verified (Thakur et al., 2022).

To summarize, the privacy of mobile phone data security is not guaranteed and can lead to data leakage. The potential for mobile devices to be stolen or lost is very high. Data stored by users is easily compromised and may be misused by outsiders (Gupta et al., 2022). Most applications do not have to enter passwords to encrypt and decrypt their data, so the chances of their accounts being hacked are high (Wanpeng & Wei, 2014). To overcome the above issues, it is crucial to provide encryption and decryption functions for the data. So, the data will be more secured, and the data security can be improved. Also, mobile phone users can access accounts that have been registered on other android devices and can change their password to a new one (Tayde & Siledar, 2015). Most importantly, encryption and decryption should both require a passcode to eliminate the possibility of hacking.

Therefore, a mobile application security design by using an android studio called ISY Mobile Crypt (IMC) is proposed for smartphones or tablets with the android operating system. The main goal of this application is to enable secured and confidential text and image transmission by encrypting and decrypting it with a passcode via an authenticated receiver on the same account. To outline the interactivity of a user with an internal software system, the context diagram of IMC is illustrated in Fig. 1. The context diagram is primarily used to help businesses wrap their heads around the scope of a system. As a result, developers can figure out how best to design a new system and its requirements or how to improve an existing system.

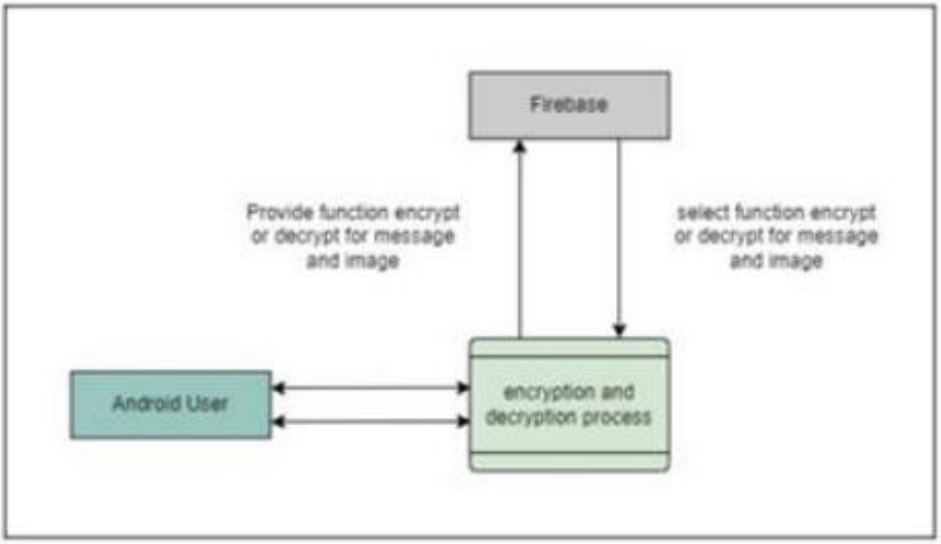


Fig. 1: Context diagram of the IMC

Next, Fig. 2 shows the flow of encryption and decryption process from plaintext to ciphertext and vice versa in IMC. Details of the IMC system development are described in the next section.

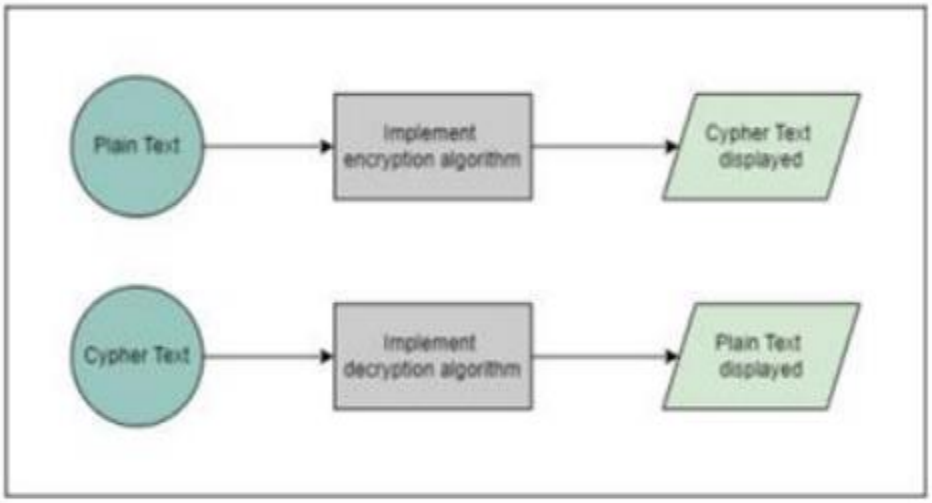


Fig. 2: The flow diagram of the IMC

2. Methodology

2.1 Rapid Application Development (RAD) Methodology

The rapid application development (RAD) is a development lifecycle that produces significantly faster and higher quality outputs than the standard lifecycle. It is built to take full advantage of the latest generation of sophisticated development software (Daud et al., 2010; Martin, 1991). Fig. 3 shows the methodology that was used in this mobile application development.

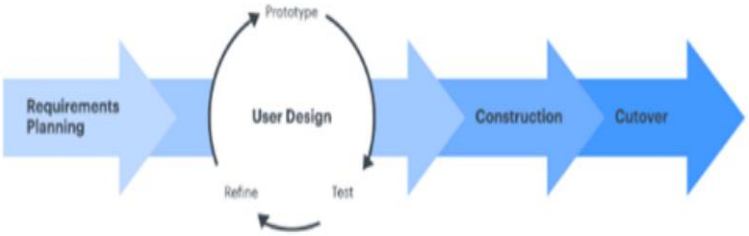


Fig. 3: RAD methodology diagram

This methodology minimizes the planning stages and will produce greater efficiency, faster and more effective communication. Each phase was described as follows:

2.1.1 Requirement Planning

In this process, the android users from Polytechnic Ungku Omar students and staff were probed about the features they would like to see in the application as well as what they would expect to see as output from it. Then, a general review of the project was carried out to evaluate the amount of time and resources needed for the development process. The requirements obtained were translated into functional requirements. Each demand will be prioritized based on the discussion with a client. Before the working subsystem is given over to the customer for review, the highest priority requirement will be defined, implemented, and tested. Any client feedback is taken into account when making changes to the present requirements. These procedures are repeated until the system is finished. Table 1 listed the software requirements.

Table 1: Software requirements

Software requirement	Specification
Android Studio (IDE)	Version: Android Studio version 4.1.2 Programming: JAVA
Microsoft Office	Windows 10
JAVA	JDK Version 8
Firebase	Open source
Android mobile	Operating system: Android 9 and higher

Table 2 presents the functional decomposition diagram (FDD) for the mobile application. Next, in the design phase, system design is formed, and certain aspects are evaluated, which are whether the application to be developed is easy to learn or not, whether the visuals and icons used are appropriate or not; and the extent to which the application is effective for users. During the development process for the application, the coding work will run and make sure that the system will work perfectly as planned and every database will fit perfectly in its place. In the testing phase, users will sequentially try out the application and give feedback on whether the application can be launched or needs to be improved. If the application needs to be improved or changed, this process will return to the application development and improvement phase. Table 2 listed the hardware requirements.

Table 2: Hardware requirements

Hardware	Requirement specification
Laptop	Operating system: 64-bit operating system, x64-based processor Processor: Intel® Core™ i5-8265U CPU @ 1.60GHz 1.80GHz Resolution: 1920 x 1080
Android mobile	Operating system: Android

Based on Table 3, the non-functional requirements were also figured out to support performance and reliability of the application. Meanwhile, Fig. 4 show the functional decomposition diagram.

Table 3: Functional requirements

Functional Requirement	Specification
Registration	Users must first register in the application before they can log in. Only then can they encrypt or decrypt an image or message For the registration, user is required to enter data as: 1) profile picture; 2) full name; 3) email; 4) phone number; and 5) password The registration process is applicable for any type of user All the data that the user has entered will be stored in the database
Login	Users log into the application to encrypt or decrypt their image or message For login, users are required to enter below data: 1) Phone number; and 2) password
Forgot password button	The user could reset their password if they forget the password, and get a link via email
Home	The user can reset their password if they forget the password, and the user will get a link via email. This application has a select function to choose a menu Users can choose what they want to encrypt or decrypt
About us	Users can learn about the app's main objective and what it will do by visiting the 'About Us' page
Key	The user will register a password that they will use for the encrypt and decrypt processes later, and each password registered is stored in the database. On this page, the user will enter: 1) message encryption key; 2) message decryption key; 3) image encryption key; and 4) image decryption key
Message	The user will select the function they want, and the user will enter the password they have registered Menu function available: 1) encryption; and 2) decryption
Image	Users will upload the image they want to encode, and user will click ‘Decrypt’ button to encode the image Encryption Decryption
Logout	Users will be redirected to the login page after pressing the logout button

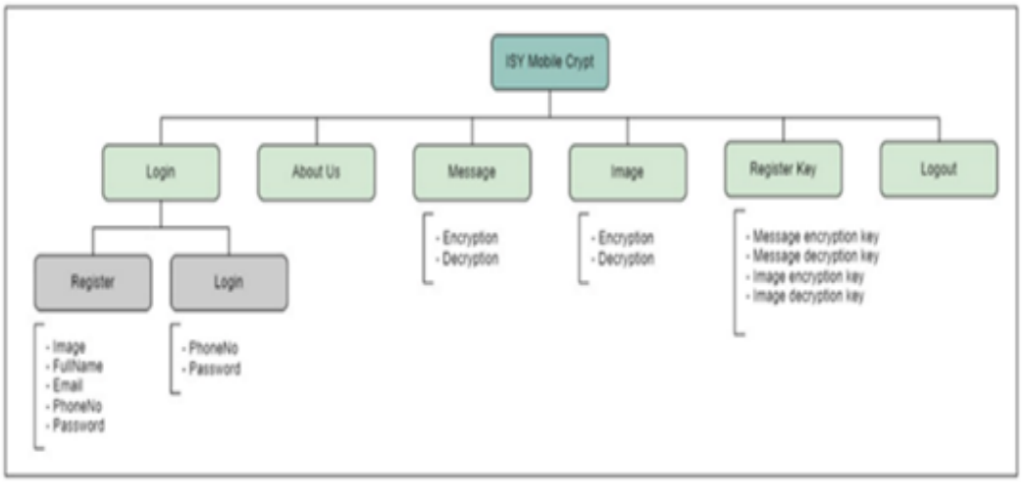


Fig. 4: Functional decomposition diagram

2.1.2 User Design

This is the core component of RAD methodology and distinguishes it from other project management approaches. Clients and developers collaborate closely throughout this stage to make sure that their needs are satisfied at every stage of the design process. The users can evaluate each product prototype at each level to make sure it matches their expectations, almost like custom software development.

Through an iterative process, all bugs and glitches are sorted out. A prototype is created by the developer, customer (user) tries it, and then both parties discuss what worked and what did not.

2.1.3 Construction

Developers may build the final functioning model more quickly than they can by using a conventional project management methodology because the majority of issues and adjustments are handled during the rigorous iterative design process. The phase is divided into many more concise steps: 1) getting ready for rapid construction; 2) development of program and applications; and 3) unit, integration, and system testing.

During this phase, the software development team, which consists of programmers, coders, testers, and developers, collaborates to ensure that everything is running flawlessly, and that the final product meets the goals and expectations of users.

During this third vital step, the clients can still be able to offer their input. To deal with problems as they develop, they can recommend enhancements, modifications, or even entirely new ideas.

2.1.4 Cut over

The finished product is launched during this phase of execution. While the coders and customers continue to scan the system for problems, all final adjustments are done. If there is a problem during testing, the teams will be required to return to construction stage to check and solve the identified problems.

2.2 System Configuration

The encryption and authorization configurations are detailed in this section. The computer hardware, methods, and other devices that make up the entire system and its constraints are referred to as system configuration in systems engineering. This term also refers to the hardware-software configuration and how each device, software, or process interacts with one another, as defined by a system settings file generated automatically or customized by the user. Because the project is totally software-based, each operation is carried out via ISY Mobile Crypt and the Firebase, Storage, and Firestore Real-Time Databases. A successful link between the two requires an Internet connection. Users must first register to use the application. The software delivers details once users sign up.

2.2.1 Keys to Access the Service

Services provided: Text - encryption and decryption and Image - encryption and decryption. This application will provide a key which consists of 16 characters for users of text and image encryption and decryption. Fig. 5 shows the firebase connection.

```
dependencies {

    implementation 'commons-io:commons-io:2.5'
    implementation 'androidx.appcompat:appcompat:1.3.1'
    implementation 'com.google.android.material:material:1.4.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.1'
    implementation 'org.jetbrains:annotations:15.0'
    implementation 'com.google.firebase:firebase-auth:21.0.1'
    implementation 'com.google.firebase:firebase-database:20.0.2'
    implementation 'com.google.firebase:firebase-storage:20.0.0'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
    implementation 'com.mikhaellopez:circularimageview:4.3.0'
    implementation 'com.squareup.picasso:picasso:2.71828'
}
```

Fig. 5: Firebase connection

2.2.2 Ask Permission to Upload an Image from Storage

The application also asks the user for permission to access the external storage, for example, an image. Users have to choose “allow” so that the database will be retrieved to be displayed in the file list page and can be opened by the user. Fig. 6 show the source code for login system.

```
public class MainActivity extends AppCompatActivity {

    EditText loginPhoneNum, password;
    Button login, signUp, btnForgotPassword;
    private FirebaseAuth mAuth;

    ProgressDialog progressDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        loginPhoneNum = findViewById(R.id.loginPhoneNum);
        password = findViewById(R.id.password);
        login = findViewById(R.id.login);
        signUp = findViewById(R.id.signUp);
        /*txtInLayoutUsername = findViewById(R.id.txtInLayoutUsername);
        txtInLayoutPassword = findViewById(R.id.txtInLayoutPassword);*/
        mAuth = FirebaseAuth.getInstance();
        btnForgotPassword = findViewById(R.id.btnForgotPass);

        progressDialog = new ProgressDialog( context: MainActivity.this);
        progressDialog.setTitle("Please Wait");
        progressDialog.setMessage("Log in account.....");

        ClickLogin();

        //SignUp's Button for showing registration page
        signUp.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { ClickSignUp(); }
        });
```

Fig. 6: Login

2.2.3 Text Encryption and Decryption

Users can change the text by using encryption and decryption styles by entering a key that has been set by the application. Fig. 7 shows source code for reset password via email.

```
private EditText edtTxtEmail;
private Button btnSendEmail;
FirebaseAuth auth;

@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.reset_password);

    edtTxtEmail = findViewById(R.id.reset_email);
    btnSendEmail = findViewById(R.id.btnSendEmail);
    auth = FirebaseAuth.getInstance();

    btnSendEmail.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            String email = edtTxtEmail.getText().toString().trim();

            if (email.isEmpty()) {
                edtTxtEmail.setError("Please fill out this field");
                edtTxtEmail.requestFocus();
                return;
            }
            if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
                edtTxtEmail.setError("Invalid email");
                edtTxtEmail.requestFocus();
                return;
            }
```

Fig. 7: Reset password via email

2.2.4 Image Encryption and Decryption

Users can change the image by using encryption and decryption styles by entering a key that has been set by the application. Fig. 8 show the generate key for encrypt system.

```
private static Key generateKey() throws Exception {
    Key key = new SecretKeySpec(keyValue, ALGORITHM);
    return key;
}

public static String encrypt(String valueToEnc, Key key) throws Exception {
    Cipher cipher = Cipher.getInstance(ALGORITHM);
    cipher.init(Cipher.ENCRYPT_MODE, key);

    byte[] encValue = cipher.doFinal(valueToEnc.getBytes());
    byte[] encryptedByteValue = Base64.encode(encValue, Base64.DEFAULT);

    return new String(encryptedByteValue);
}
```

Fig. 8: Generate key for encrypt

2.2.5 Source Code

The coding used to make a connection to Firebase.

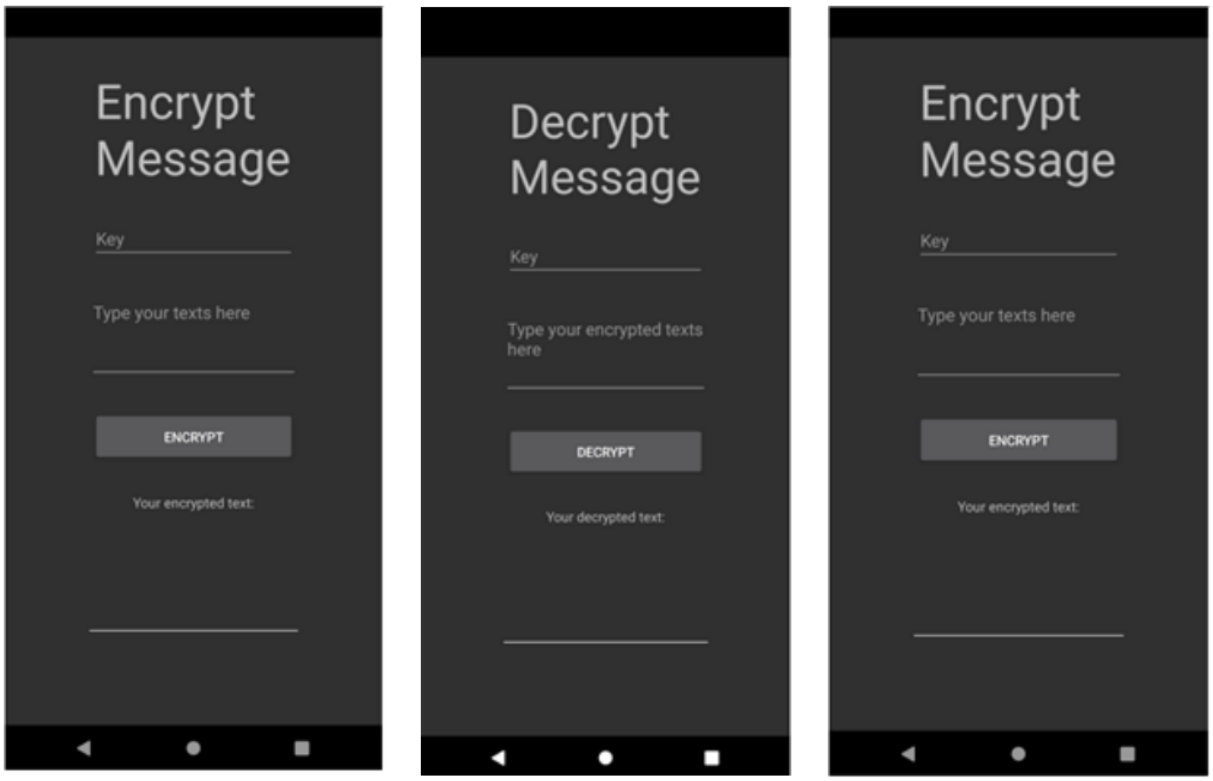
3. Results

This section describes the results of ISY Mobile Crypt (IMC) application development as shown in Fig. 9. IMC was created in general to coincide with encryption and decryption functions. The authorization aspect was also incorporated into this application to improve the security mechanisms. IMC will display an interface menu initially. Then, as a requirement of authorization, users must first register and login to the application, as shown in Fig. 10. The user must

next register a key crypt, as seen in Fig. 11 (a). Users can select whether to register a key for image or text encryption. Following that, the users can select an encryption key for text (message) or image. The functionality for message encryption. Next, the message decryption interface is shown in Fig. 11 (b). Finally, Fig. 10 (c) depicts the image encryption and decryption interface.

Fig. 9: Menu interface

Fig. 10: Login main page



(a) **(b)** **(c)**

Fig. 11: (a) Encrypt message interface; (b) decrypt message interface; (c) encrypt and decrypt image interface

4. Discussion

In this study a solution of privacy and security for smart phones users was proposed, which is the increasing risk of privacy and security risk threats for mobile phone users (Yang et al., 2021). The privacy and security of mobile applications must be emphasized on the significance of user authentication technology (Shirazi & Iqbal, 2017; Wei et al., 2012). Therefore, the development of IMC is an alternative solution to user data security to ensure that user data is kept in a secure state for smart phone users. The advantage of IMC is that we can easily secure data in one application. In the event that an android user loses their phone, it is not necessary to worry because the user can access another android phone by using the same phone number and password to recover the personal data. By using the IMC system, data leakage could be avoided. Since the encrypted data is stored in an encrypted format, other malware will not be able to access the data. If an attacker manages to access the data, they will only get the data that is in the form of ciphertext. For users who want privacy, this application is suitable because it can keep confidential data. Users can access this application anywhere as long as they have Internet access. Finally, the security of this system will be able to guarantee the data more firmly. In this application, when the user wants to access the data, he needs to enter the phone number and password, and to encrypt and decrypt the data, he only needs to enter the key that has been given. So, if anyone needs to access sensitive data, it should not be a problem. To protect user data, key-password entry is provided as a system security measure. If an android user does not have a key password, he or she will be unable to use our service. For the time being, this software can only be used by android users who have entered a key password into their device.

5. Conclusion

The IMC was developed by using the rapid application development (RAD) methodology due to its flexibility of implementation. As RAD methodology is employed, the activities are iterated based on system requirements. Client feedback is taken into account each time the implementation is completed, making this methodology useful to including clients in the system development process. It may be inferred that for a small-scale system, RAD and rapid implementation are appropriate because the client can see the outcomes. It may be concluded that RAD apt for a small-scale system and rapid implementation is appropriate because the client can view result in a quickly while the developer retains control over the development process. This paper presents a method for user data encryption in mobile phones, as well as three steps for implementing the method. In addition, to ensure effectiveness, this technology is safer than similar methods in terms of data decryption complications because the encryption process includes more unpredictability and dynamics. The suggested privacy-preserving techniques are well suited to mobile phone user data

encryption according to the detailed performance analysis and evaluation. Furthermore, as new encryption algorithms emerge on a regular basis, the encryption strength of the proposed encryption method will be enhanced by including them in the encryption algorithm library. With rapid improvement in mobile phone performance, the encryption effectiveness will also be increased.

References

Abdelmaboud, A., Ahmed, A. I. A., Abaker, M., Eisa, T. A. E., Albasheer, H., Ghorashi, S. A., & Karim, F. K. (2022). Blockchain for IoT applications: taxonomy, platforms, recent advances, challenges and future research directions. *Electronics*, *11*(4), 630. https://doi.org/10.3390/electronics11040630

Aljazaery, I. A., Salim ALRikabi, H. T., & Alaidi, A. H. M. (2022). Encryption of Color Image Based on DNA Strand and Exponential Factor. *International Journal of Online & Biomedical Engineering*, *18*(3), 101-113. https://doi.org/10.3991/ijoe.v18i03.28021

Almaiah, M. A., Hajjej, F., Lutfi, A., Al-Khasawneh, A., Alkhdour, T., Almomani, O., & Shehab, R. (2022). A conceptual framework for determining quality requirements for mobile learning applications using delphi method. *Electronics*, *11*(5), 788. https://doi.org/10.3390/electronics11050788

Chandrashekhar, R. V., Visumathi, J., & Anandaraj, A. P. (2022, January). Advanced lightweight encryption algorithm for android (IoT) devices. In *2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)* (pp. 1-5). IEEE. https://doi.org/10.1109/ACCAI53970.2022.9752555

Daud, N. M. N., Bakar, N. A. A. A., & Rusli, H. M. (2010, June). Implementing rapid application development (RAD) methodology in developing practical training application system. In *2010 International Symposium on Information Technology*, *3*, 1664-1667. IEEE. https://doi.org/10.1109/ITSIM.2010.5561634

Gupta, I., Mittal, S., Tiwari, A., Agarwal, P., & Singh, A. K. (2022). TIDF-DLPM: Term and inverse document frequency-based data leakage prevention model. *arXiv preprint arXiv:2203.05367*. https://doi.org/10.48550/arXiv.2203.05367

Lisonek, D., & Drahanský, M. (2008, December). SMS encryption for mobile communication. In *2008 International Conference on Security Technology* (pp. 198-201). IEEE. https://doi.org/10.1109/SecTech.2008.48

Martin, J. (1991). *Rapid application development*. Macmillan Publishing Co., Inc. *Scribbr*. https://doi.org/10.5555/103275

Ullah, A., Xiao, H., & Barker, T. (2019). A multi-factor authentication method for security of online examinations. In *Smart Grid and Internet of Things: Second EAI International Conference, SGIoT 2018, Niagara Falls, ON, Canada, July 11, 2018, Proceedings*, *256*, 131-138. Springer International Publishing. https://doi.org/10.1007/978-3-030-05928-6_13

Ranjbar, N., & Mahdi, A. (2012). *Authentication and authorization for mobile devices*, 2-15. *Scribbr*. https://gupea.ub.gu.se/handle/2077/30043

Seth, B., Dalal, S., Jaglan, V., Le, D. N., Mohan, S., & Srivastava, G. (2022). Integrating encryption techniques for secure data storage in the cloud. *Transactions on Emerging Telecommunications Technologies*, *33*(4), e4108. https://doi.org/10.1002/ett.4108

Shirazi, F., & Iqbal, A. (2017). Community clouds within M-commerce: a privacy by design perspective. *Journal of Cloud Computing*, *6*(1), 1-12. https://doi.org/10.1186/s13677-017-0093-0

Sinha, A., Shrivastava, G., Kumar, P., & Gupta, D. (2022). A community-based hierarchical user authentication scheme for Industry 4.0. *Software: Practice and Experience*, *52*(3), 729-743. https://doi.org/10.1002/spe.2832

Smriti, M., Venkatraman, S. V., Raj, A., Shukla, V. R., & Cherukuri, A. K. A. (2022). Secure File Storage in Cloud Computing Using a Modified Cryptography Algorithm. In *Advancing Smarter and More Secure Industrial Applications Using AI, IoT, and Blockchain Technology*, 200-224. IGI Global. https://doi.org/10.4018/978-1-7998-8367-8.ch011

Tayde, S., & Siledar, S. (2015). File Encryption Decryption using AES algorithm in android phone. *International Journal of Advanced Research in Computer Science and Software Engineering*, *5*(5), 550-554.

Thakur, V., Indra, G., Gupta, N., Chatterjee, P., Said, O., & Tolba, A. (2022). Cryptographically secure privacy-preserving authenticated key agreement protocol for an IoT network: A step towards critical infrastructure protection. *Peer-to-Peer Networking and Applications*, *15*, 206-220. https://doi.org/10.1007/s12083-021-01236-w

Wang, X. (2022). Security Threats and Protection Based on Android Platform. In *2021 International Conference on Big Data Analytics for Cyber-Physical System in Smart City: 2* (pp. 179-186). Springer Singapore.

https://doi.org/10.1007/978-981-16-7469-3_19

Wanpeng, C., & Wei, B. (2014). Adaptive and dynamic mobile phone data encryption method. *China Communications*, *11*(1), 103-109. https://doi.org/10.1109/CC.2014.6821312

Wei, C. H., Hwang, M. S., & Chin, A. Y. H. (2012). An improved authentication protocol for mobile agent device in RFID environment. *International Journal of Mobile Communications*, *10*(5), 508-520. https://doi.org/10.1504/IJMC.2012.048884

Yang, M., Jia, L., Gao, T., Zhang, T., & Xie, W. (2021). Research on privacy security steady StateEvaluation model of mobile application based on information entropy and Markov theory. *International Journal on Network Security*, *23*(5), 807-816. https://doi.org/10.6633/IJNS.202109

Žitovský, O. (2007). *Šifrování komunikace mobilních zařízení pomocí Java ME (Communication Encryption of Mobile Devices using Java ME)* (Doctoral dissertation, Master thesis, Masaryk University Faculty of Informatics, Brno).