Journal of Engineering, Electrical and Informatics
Vol.3, No.3 oktober 2023
e-ISSN: 2809-8706; p-ISSN : 2810-0557, page 19-33
DOI: https://doi.org/10.55606/jeei.v3i3.2252

Notary Deed Security Application Using RC4 Method on Android-Based

Reza Saputra
Faculty of Science & Technology, Yogyakarta Technology University

Yuli Asriningtias
Faculty of Science & Technology, Yogyakarta Technology University

Corresponding author: rezakosambi5@gmail.com

Abstract: *In the digital era, security concerns related to notary deeds can arise due to several factors, particularly in safeguarding important documents. Without a secure application, these important documents, such as notary deeds, can be threatened by unauthorized access. This could result in material or reputational loss for both the notary and the client. This application provides facilities for users to encrypt and decrypt notarial deeds in several file types, including PDF, Excel, Microsoft Word, and images. The encryption and decryption process in this application utilize the RC4 method. In this application, users can process notarial deeds to be encrypted according to the type of document file they possess in order to obtain an encrypted files ready to be sent or exchanged notary deed data. The application result will display the history of the encrypted document, the date when the file was encrypted, as well as the token in the file for the decryption process.*

Keywords: *Notarial Deed, RC4, Encryption, Decryption, Token*

INTRODUCTION

A notary is a person authorized by the government to ratify and witness various agreements, wills, deeds and so on (Alincia & Sitabuana, 2021; Xiong et al., 2022; Yildiz & Cakmak, 2021). Notaries are appointed and dismissed by the State, represented in this case by the government through the Minister of Law and Human Rights *(Menkumham).* The difference between a Notary and a State official is that the Notary does not receive a salary, but only gets compensation for his or her service to the community. This amount has been determined within the INI association or organization. Different from others, the Notary profession is bound by government regulations and is not independent when a Notary has to work, then is bound by leave regulations and administrative regulations which are closely tied to his work (Alyafie & Purnawan, 2020; Amudy et al., 2020; Iryadi et al., 2021).

The problem faced by notaries in this digital era is maintaining the security of their data storage from digital interference, such as computers infected with viruses, data theft, data modification, and misuse of electronic signatures. Data manipulation can potentially occur when data is being transmitted, and in the midst of its journey, it is intercepted by the

Received August 30, 2023; Revised September 30, 2023; Accepted October 31, 2023
* Reza Saputra, *rezakosambi5@gmail.com*

unauthorized people (Arrohim & Wahyuningsih, 2020; Sidharta & Dewi, 2023; Syuaib et al., 2021).

The information contained in the deed must maintain its integrity, ensuring that the information within the deed does not undergo in unauthorized people (Anshori et al., 2022; Bayu Pratama & Syarif, 2021; Ryan et al., 2021). Therefore, data security is needed to prevent data theft. One of the methods commonly used to secure this data is cryptography. Its implementation used in this research is RC4. This is a symmetric key cryptographic algorithm and is a stream cipher so that the length of the characters resulting from encryption (ciphertext) has the same character length as the original data (plaintext) (Butarbutar, 2021; Ito & Miyaji, 2021; Kartikadarma et al., 2018; Rahmatulloh, 2017; Riad et al., 2013; Slamet et al., 2021).

The purpose of the design and build application is to design and construct an application that can preserve the security of deed integrity and apply the RC4 Method to

RESEARCH METHODS

The data used in this research refers to information and content relevant to notarial deeds, which will be secured using the RC4 encryption method. This data includes details of the notarial deed, such as name, date, deed number, contents, and other attributes related to the notarial deed that will be protected. In addition, the data also refers to the encryption key used in the RC4 method. The encryption key is confidential information needed to secure the data encryption and decryption process. The encryption key must be kept confidential and only known to authorized parties.

In this research, data obtained from the Academia.edu website can be used as a relevant reference. These data can provide examples of implementation and case studies related to securing notarial deeds using the RC4 method. Data collection was not carried out through direct interviews or questionnaires. As a scientific research and publication platform, Academia.edu provides access to various research and papers that have been conducted by other researchers.

As an important part of the research process, data collection plays a crucial role in obtaining the information required to develop an App. When collecting data, there are several factors that need to be considered, such as the collection method, data source, data collection location, and collection time period. The research was done by conducting a search on the Academia.edu website using relevant keywords to the topic appropriate to the research, such as "notarial deed security", "RC4 method", or other related topics. Then, identifying and selecting of research that suits the research objectives and is relevant to the application to be developed.

The data sources employed research, papers, or scientific articles available on the Academia.edu platform. This is a platform that allows researchers and academics to share their scientific publications. In this research, the physical location of data collection is irrelevant as it involved materials or data from the Academia.edu website. This website is an online platform that can be accessed from anywhere as long as it is connected to the internet. Therefore, there is no specific physical location associated with collecting data from the website. The time span for collecting data in this research using materials or data from the academia.edu website will depend on the availability and accessibility of the required data.

This research involved several important stages. The first stage was a literature study and needs analysis, where information was searched regarding securing notarial deeds, the RC4 method, and Android application development (Chakraborty et al., 2021; Koyama et al., 2016; Rahmatulloh, 2017; Slamet et al., 2021). Next, the system design stage was carried out to design an application architecture that included security components, such as data encryption, user authentication, and access authorization. The application development stage involved implementing the security features that have been designed and creating a responsive user interface using the Android platform. After that, the testing and evaluation phase were carried out to ensure the security, functionality, and performance of the application. Security testing involved penetration testing and attack testing to identify and address potential security gaps. The improvement and adjustment stage were carried out based on the evaluation results to correct the problems and weaknesses. At the end, documentation was carried out by documenting all research steps and results in a research report, which included the background, objectives, methods, results, and conclusions of the research.

RESULTS AND DISCUSSION

System analysis of the Notarial Deed Security Application using the Android-based RC4 Method was carried out to ensure the security and functionality of the application. Analysis of the system running in this research was conducted to ensure security and optimal performance. In this analysis, several main aspects are evaluated. First, the security component is analyzed by checking the RC4 encryption method used and ensuring its strength and security. Next, the information exchange process is evaluated to ensure secure mechanisms for sending and receiving data, such as the use of encryption protocols. In addition, the authorization system and access settings are analyzed to ensure that each user has access appropriate rights to their role. Security testing is carried out to identify gaps or vulnerabilities that may exist, and remedial steps are taken based on the test results. System performance is

also analyzed, including application responsiveness, response time, and capacity to handle users and data. Finally, integration with other components is evaluated to ensure compatibility and security. Through this running system analysis, the objective is to identify potential security gaps, ensure good performance, and improve overall application performance. In this way, the application can operate safely and efficiently in maintaining the security of data and information related to notarial acts.

The system analysis proposed in this research aims to improve application security and performance. In this analysis, several important steps were taken. First, an evaluation of the RC4 encryption method used was carried out to ensure its strength and security. Furthermore, user authentication mechanisms, such as usernames and passwords, were updated to concern the latest security practices. Moreover, additional security measures, such as end-to-end encryption, were implemented to protect data integrity and confidentiality. Furthermore, improvements were made to the information exchange mechanisms between users by strengthening the use of secure encryption protocols and implementing countermeasures against attacks, such as Man-in-the- Middle. The authorization system was strengthened by ensuring that each user has access rights appropriate to their role and by implementing stricter access settings. Additionally, security and performance testing were performed to identify and fix possible security gaps and ensure the application could handle high user load with fast response. Finally, the application's integration with other components, such as database servers and external APIs, was checked and updated to ensure good compatibility and security. By analyzing the proposed system, it is expected that this application can have a higher level of security, protect sensitive information, and provide optimal performance. The following is the model architecture that will be proposed for the application, which can be seen in Figure 1.

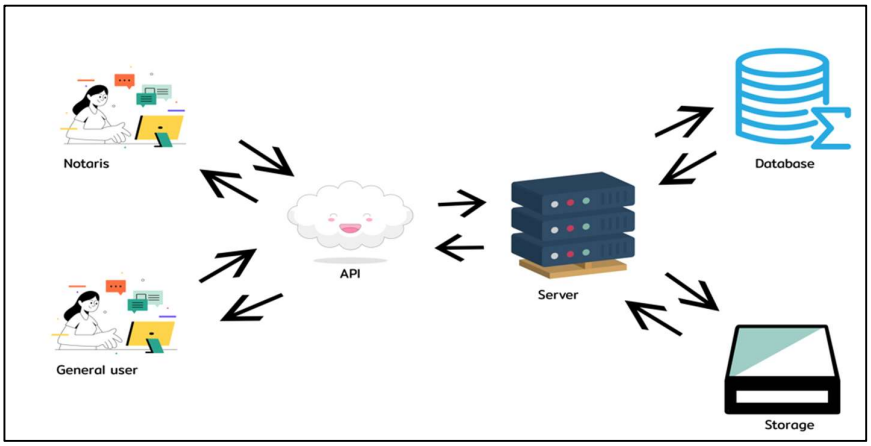


Figure 1System Architecture

Implementation of *Splash Screen Interfaces* page is the implementation page Android used by Ordinary users and Notaries, which contains an initial display that appears while the application is loading and provides a first impression to the user.

Files SplashScreen.kt is scripts to show that when the *SplashScreen* activity is running, the activity_splash_screen layout will be displayed using the *setContentView* method. Then, with the assistance of the *Handler()* object, there is a delay of 2000 milliseconds (2 seconds) using the *postDelayed* method. After the delay is complete, a page redirect is performed by creating an *Intent* object that points to *LoginActivity,* and a call to *startActivity(intent)* is made to start *Login Activity.*

```
package com.example.lockdoc

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Handler

class SplashScreen : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_splash_screen)
        Handler().postDelayed({
            val intent = Intent( packageContext: this, LoginActivity::class.java)
            startActivity(intent)
        }, delayMillis: 2000)

    }
}
```

Figure 2. SplashScreen.kt Script

Home page is the initial screen that appears when the application is loading and provides the first impression to the user. As for *Splash Screen* display can be seen as follows:

Figure 3. Splash Screen Display

Implementation of *Users Interfaces* page is the implementation of *Android* page, used by Ordinary users and Notaries, which contains information about available Menu in the application.

1. urlAPI.kt

urlAPI.kt files is scripts to connect *Android* with API-bridged resource, which contained in *the MySQL database*.

```
package com.example.lockdoc

enum class urlAPI (val url : String){
    endPoint( url: "https://9003-103-92-232-3.ngrok-free.app")
}
```

Figure 4. urlAPI.kt Script

Home page is the main page from the Application for Securing Notary Deeds using the Android-Based RC4 Method. This page works as place to displays available Menus about this application. The Home display can be seen as follows:

Figure 5. Home Display

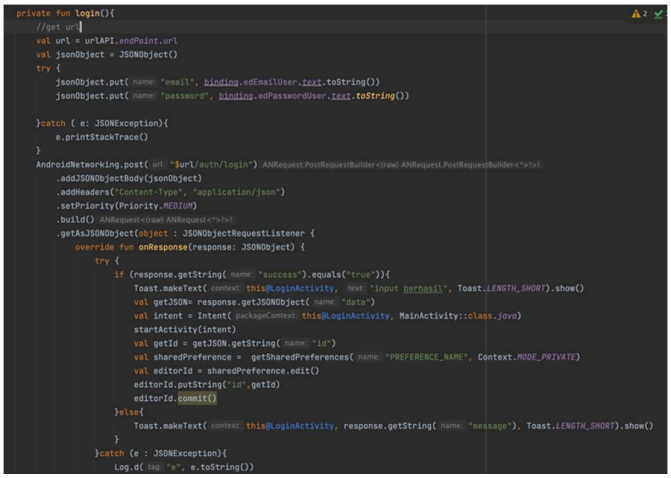
2. Android implementation

a. Login Page Implementation

The Login page is the initial page that will be displayed for the user to enter their email and password to authenticate before accessing the main features of the application. This implementation aims to ensure that only authenticated users can access the application and its features.

b. LoginActivity.kt

Files LoginActivity.kt is the scripts which is done during the login process. First, retrieve the URL of the API endpoint using urlAPI.endPoint.url. Then, the email and password data entered by the user are taken from the user input and stored in the JSONObject object. Next, using AndroidNetworking, a POST request is made to the URL corresponding to the login endpoint. JSON data containing email and password is sent in the request. If the request is successful and gets a success response (success=true), then the user will be notified with the message "input successful" via Toast. With the Login implementation, users can enter their email and password to carry out the authentication process and gain access to the application's main page.



```
private fun login(){
    //get url
    val url = urlAPI.endPoint.url
    val jsonObject = JSONObject()
    try {
        jsonObject.put( name: "email", binding.edEmailUser.text.toString())
        jsonObject.put( name: "password", binding.edPasswordUser.text.toString())

    }catch ( e: JSONException){
        e.printStackTrace()
    }
    AndroidNetworking.post( url: "$url/auth/login") ANRequest.PostRequestBuilder<(raw) ANRequest.PostRequestBuilder<*>!>!
        .addJSONObjectBody(jsonObject)
        .addHeaders("Content-Type", "application/json")
        .setPriority(Priority.MEDIUM)
        .build() ANRequest<(raw) ANRequest<*>!>!
        .getAsJSONObject(object : JSONObjectRequestListener {
            override fun onResponse(response: JSONObject) {
                try {
                    if (response.getString( name: "success").equals("true")){
                        Toast.makeText( context: this@LoginActivity, text: "input berhasil", Toast.LENGTH_SHORT).show()
                        val getJSON= response.getJSONObject( name: "data")
                        val intent = Intent( packageContext: this@LoginActivity, MainActivity::class.java)
                        startActivity(intent)
                        val getId = getJSON.getString( name: "id")
                        val sharedPreference =  getSharedPreferences( name: "PREFERENCE_NAME", Context.MODE_PRIVATE)
                        val editorId = sharedPreference.edit()
                        editorId.putString("id",getId)
                        editorId.commit()
                    }else{
                        Toast.makeText( context: this@LoginActivity, response.getString( name: "message"), Toast.LENGTH_SHORT).show()
                    }
                }catch (e : JSONException){
                    Log.d( tag: "e", e.toString())
```

Figure 6. LoginActivity.kt Script

c. Login Page Display

Through this display, users will be asked to enter login information, such as email and password to authenticate. The Login page display design aims to provide an intuitive and easy-to-use user experience, as well as ensuring security in the user authentication process. The Login page can be seen as follows:

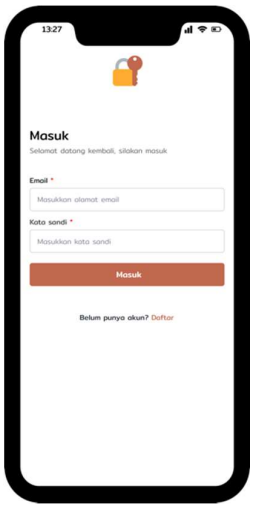


Figure 7. Login Display

3. Android implementation

a. Encryption Page Implementation

The File Encryption page allows users to encrypt the files they want to secure using the RC4 encryption method. This display allows the user to select the files to be encrypted and start the encryption process. The File Encryption page display design aims to provide an easy-to-understand user experience, ensure the security of the encryption process, and make it easy to secure important files to users.

b. EncryptFileActivity.kt

In the presented File Encryption code, it can be seen the implementation of the logic and encryption process in the Notary Deed Security Application using the Android-based RC4 Method. The code takes input from the user in the form of user ID, document description, file name, and plaintext to be encrypted. Next, using Android Networking, an HTTP request is sent to the "process/encrypt" endpoint to send the encryption data to the server. If the request is successful, a success dialog will be displayed to the user with the option to view the encryption history. This code shows how the application interacts with the server and implements file encryption using the RC4 algorithm.

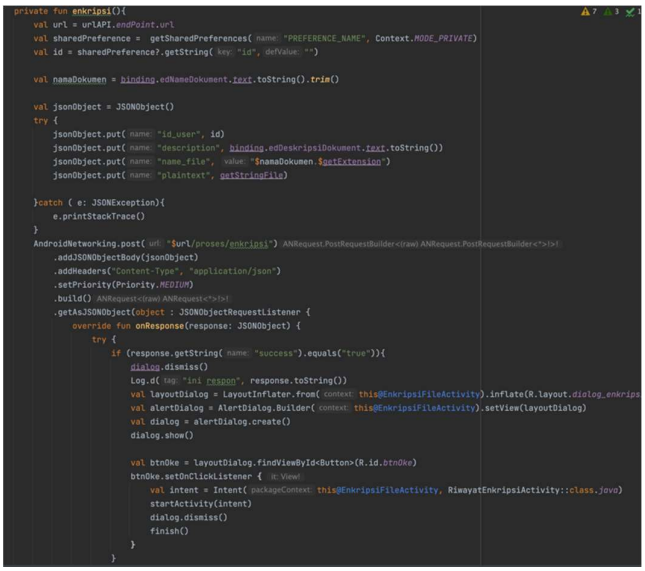


Figure 8. EncryptFileActivity.kt Script

4. File Encryption Page Display

On this page, users can fill in information, such as document description, file name, and files to encrypted. After that, the application will send encryption data to the server via HTTP request. If the encryption process is successful, the user will be provided a notification and the option to view the encryption history. The encryption display can be seen as follows:

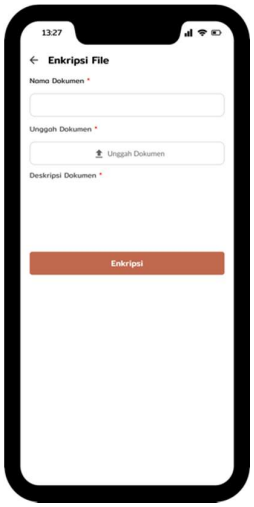


Figure 9. Encryption Display

5. Back-End Implementation

a. Implementation of the Encryption process

After the user sends encryption data from the user interface, the backend will receive the data and carry out the encryption process using the RC4 algorithm. This process involves some steps, such as obtaining user information, document description, file name, and plaintext

to be encrypted. Next, the RC4 algorithm will be applied to encrypt the plaintext into a secure ciphertext. Once encryption is complete, the backend will send a response to the user interface, providing confirmation of encryption success, and providing the option to view encryption history.

b. encryption_process.js

The following is an implementation of the file encryption process using the RC4 method in the backend. At this stage, when the '/encryption' route is accessed, the server will receive a POST request containing plaintext data, user_id, file_name, and description of the user interface. The code will then carry out several steps, such as checking the user's whereabouts based on the given ID, encrypting the plaintext using the RC4 algorithm, saving the encryption results in a file, and saving information related to the file in a database. If all steps are completed successfully, the server will provide a response in the form of JSON with a success status and the message 'Data Encryption Successful'.

```
router.post('/enkripsi', async (req, res)=>{
    const {plaintext, id_user, name_file, description} = req.body
    let connection
    try{
        connection = await pool.getConnection()
        let sql = `SELECT * FROM users WHERE id = ${id_user}`
        const [result] = await connection.query(sql, [], {rowsAsArray: false, nestTables: false})
        if(result == 0){
            throw new Error('User tidak ditemukan')
        }
        const [key,resultEncrytion] = encrypted.encryptRC4(plaintext)
        const date = Date.now() + `-${name_file}.enc`
        fs.writeFileSync(path.join(__dirname + `/../files/encryption/${date}`),resultEncrytion)
        sql = `INSERT into files (id_user, name_file, description, key_file, path, is_private) VALUES (${id_user}, '${name_file}',
        await connection.query(sql)
        res.json({
            success : true,
            message : 'Enkripsi Data Berhasil'
        })
    }catch(error){
        res.json({
            success : false,
            message : error.message
        })
    }finally{
        if(connection){
            connection.release()
        }
    }
```

Figure 10. encryption_process.js Script

The results obtained have several main features that can be used by users. The following are some of the results that can be found in this application:

1. Menu: Users can log in to the application using a registered account. They need to enter a valid email and password to access other features.
2. Register Menu: Users who don't have an account can register through this menu. They will be asked to fill out a registration form with the required information.
3. Send Files: Users can send notary deed files, which will be secured through this application. They need to upload the file they want to send and provide a brief description of the file.

4. Inbox: Users can view the list of files they have sent through this menu. They can see a list of files along with related information, such as file name, delivery date, and encryption status.
5. Encryption: Users can encrypt the files they have sent. They can select the files they want to encrypt and set the encryption key to use.
6. Decryption: Users can carry out the decryption process on files that have been previously encrypted. They need to select the files they want to decrypt and set the decryption key accordingly.
7. File Details: Users can view detailed information regarding files they have sent, including descriptions, encryption status, and encryption keys used.
8. Change Profile: Users can change their profile information, such as name, email address, and password. They can access this menu to update and save the changes made.
9. Encryption History: Users can view the history of the encryption processes they have carried out. They can see a list of files that have been encrypted along with related information, such as the encryption date and the encryption key used.

The application provides these features to ease users to secure their notarial deed files. By using the RC4 encryption method, this application can provide additional security to these files so that only users who have the encryption key are able to open and access the contents of the files.

Discussion

1. Login Menu Testing:
   - Can display the login page
   - Can validate the login data entered by the user. If the user is successful in logging in, the user will be directed to the appropriate page.
2. Register Menu Testing:
   - The registration data submitted by the user has been successfully recorded or recorded in the database.
   - The registration data that has been recorded in the database has been successfully used in the login process.
3. File Send Testing:
   - Can upload valid files and process them correctly.
   - The sent file successfully reaches the user who has been selected via the user list.

4. Inbox Testing:
    - Successfully displays a list of files that have been submitted by the user, including the display of accurate information.
    - Files received in the inbox can be viewed in detail for the decryption process.
5. Encryption and Decryption Testing:
    - Application successfully performs the ability to encrypt and decrypt files correctly.
    - The token used in the decryption process functions properly. When the token is used correctly, the file is successfully decrypted. However, if the token is used incorrectly, the file cannot be decrypted.
6. Testing File Details:
    - The application can display detailed file information that is accurate and in accordance with the data that have been saved.
    - The application can display the encryption status and encryption keys used.
7. Profile Change Testing:

    The application is successful in changing the user's profile information and saving the changes correctly.
8. Encryption History Testing:

    Successfully displays a list of encryption history that is accurate and in accordance with the data that have been stored.

The discussion of black box testing will provide a comprehensive understanding of testing the main features of this application. This will ensure that the app functions properly, responds appropriately, and keeps user data safe.

CONCLUSIONS AND RECOMMENDATIONS

Based on the research and design methods described previously, a conclusion can be drawn that this research uses the RC4 encryption method to secure notary deeds in an Android-based application. The data used in this research includes information and content related to the notarial deed to be protected, including name, date, deed number, content, as well as other relevant attributes. The encryption key is confidential information that must be kept confidential and only known by authorized parties. The data obtained for this research employed relevant references from the Academia.edu website. By searching for appropriate keywords to the research topic, such as "notarial deed security" and "RC4 method", the researchers can identify and select research or papers that are relevant to the research objectives and applications to be developed. Data collection procedures are carried out online via the

Academia.edu platform. This platform allows researchers and academics to share their scientific publications. Therefore, there is no physical location associated with data collection. The research stages include literature study and needs analysis, system design, application development, testing and evaluation, as well as improvements and adjustments. Each stage is carried out with the aim of ensuring security, functionality, and optimal performance of the notarial deed security application. System analysis is performed to ensure the security and functionality of the application. Security components, information exchange between users, authorization and access settings, as well as integration with other components are evaluated to identify security gaps and improve application performance. Overall, this research aims to develop a notarial deed security application using the RC4 encryption method that is safe and efficient. Through the proposed research and design steps, it is expected that the application can maintain the confidentiality, integrity and availability of notarial deed data, as well as provide confidence and reliability for users in using the application. Further researches are able to develop more complex user authorization and management, implementation of email verification, use of push notifications to notify users about changes in encryption status, provision of search and filter features, integration with Cloud storage, analysis and statistics on encryption history, compatibility with other platforms, file archiving and recovery, comment and collaboration features, as well as security report.

REFERENCE

Alincia, D., & Sitabuana, T. H. (2021). Urgency of Law Amendment as Foundation of The Implementation of Cyber Notary. *Law Reform: Jurnal Pembaharuan Hukum*, *17*(2). https://doi.org/10.14710/lr.v17i2.41749

Alyafie, Z. Z., & Purnawan, A. (2020). The Implementation of Legal Responsibilities Of Notary on Authentic Deed Which His Made Based On Act No. 2 of 2014 On the Amendment of Act No. 30 of 2004 On Notary in Kendari. *Jurnal Akta*, *7*(2). https://doi.org/10.30659/akta.v7i2.8104

Amudy, N., Gunarto, G., & Sulchan, A. (2020). Implementation of Legal Presumption Principle for Notary Deed Makes Partij According to Law No. 2 of 2014 on the Amendment of Act No. 30 of 2004 Concerning Notary Position. *Jurnal Akta*, *6*(4). https://doi.org/10.30659/akta.v6i4.7900

Anshori, I., Rahmi, E., & Syamsir, S. (2022). Polemik Penerapan Tanda Tangan Elektronik Dalam Pembuatan Akta Otentik. *Recital Review*, *4*(2).

https://doi.org/10.22437/rr.v4i2.18863

Arrohim, M. B., & Wahyuningsih, S. E. (2020). Analysis of Judicial Application of Criminal Penalty Against Notary / Land Deed Officials Conducting Making Crime of the Fake Authentic Deed in State Court of Semarang. *Jurnal Akta*, *7*(2). https://doi.org/10.30659/akta.v7i2.7891

Bayu Pratama, E., & Syarif, M. (2021). Pemodelan Extreme Programming Untuk Pengarsipan Akta Pada Kantor Notaris Dan Ppat. *Jik)*, *5*(2).

Butarbutar, O. (2021). Use of Rc4 Method In Android-Based Sms Security Application. *Jurnal Info Sains : Informatika Dan Sains*, *11*(2). https://doi.org/10.54209/infosains.v11i2.41

Chakraborty, C., Chakraborty, P., & Maitra, S. (2021). Glimpses are forever in RC4 amidst the spectre of biases. *Discrete Applied Mathematics*, *298*. https://doi.org/10.1016/j.dam.2021.03.021

Iryadi, I., Ansari, T. S., Saputra, J., Afrizal, T., & Thirafi, A. S. (2021). The role of jurisprudence as form of legal prescriptions: A case study of notaries in indonesia. *WSEAS Transactions on Environment and Development*, *17*. https://doi.org/10.37394/232015.2021.17.8

Ito, R., & Miyaji, A. (2021). New iterated RC4 key correlations and their application to plaintext recovery on WPA-TKIP. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, *E104A*(1). https://doi.org/10.1587/transfun.2020CIP0006

Kartikadarma, E., Listyorini, T., & Rahim, R. (2018). An Android mobile RC4 simulation for education. *World Transactions on Engineering and Technology Education*, *16*(1).

Koyama, S., Katagiri, T., Minamikawa, K., Kato, M., & Hayashi, H. (2016). Effects of rice husk charcoal application on rice yield, methane emission, and soil carbon sequestration in andosol paddy soil. *Japan Agricultural Research Quarterly*, *50*(4). https://doi.org/10.6090/jarq.50.319

Rahmatulloh, A. (2017). Source Code PHP dengan Teknik Obfuscation Code pada Extension PHP. *Konferensi Nasional Informatika*, *October 2015*.

Riad, A. E. D., Elminir, H. K., Shehata, A. R., & Ibrahim, T. R. (2013). Security evaluation and encryption efficiency analysis of RC4 stream cipher for converged network applications. *Journal of Electrical Engineering*, *64*(3). https://doi.org/10.2478/jee-2013-0029

Ryan, S., Mohammad Taufan Asri, Z., Yuliadi, & Hanif Priabdul, A. (2021). Rancang Bangun Aplikasi Pencatatan Order Akta Notaris dan PPAT I Wayan Pastika. *JURIKOM (Jurnal*

Riset Komputer), *8*(4).

Sidharta, R., & Dewi, P. E. T. (2023). THE ROLE OF CYBER NOTARY IN THE FIELD OF DIGITAL INTERNATIONAL TRADE IN INDONESIA. *NOTARIIL Jurnal Kenotariatan*, *8*(1). https://doi.org/10.22225/jn.8.1.2023.1-7

Slamet, C., Syaripudin, U., Kaffah, F. M., & Tiasto, B. E. (2021). Implementation of Rivest Cypher 4 algorithm in Security Assertion Mark-up Language protocols on Single Sign-On services. *IOP Conference Series: Materials Science and Engineering*, *1098*(3). https://doi.org/10.1088/1757-899x/1098/3/032109

Syuaib, A. M., Purnawan, A., & Khisni, A. (2021). Legal Certainty of Application of Electronic Archives in Keeping Minutes of Notary Deed as Authentic Evidence. *Sultan Agung Notary Law Review*, *3*(1). https://doi.org/10.30659/sanlar.3.1.250-256

Xiong, A., Liu, G., Zhu, Q., Jing, A., & Loke, S. W. (2022). A notary group-based cross-chain mechanism. *Digital Communications and Networks*, *8*(6). https://doi.org/10.1016/j.dcan.2022.04.012

Yildiz, G., & Cakmak, E. K. (2021). Investigating the distance education process according to the demographic characteristics of the notary and the notary employee. *Contemporary Educational Technology*, *13*(2). https://doi.org/10.30935/cedtech/9583