Designing End-to-End Web-Based Application Encryption with Asymmetric Encryption Using Waterfall Methodology

Teguh Rijanandi#1, Sara Lutami Pardede#2, Mayer Reflino Sitorus*3, Niken Dwi Wahyu Cahyani #4

#*Cybersecurity and Digital Forensics Study Program, Informatics Faculty, Universitas Telkom*
Jl. Telekomunikasi No.1, Terusan Buah Batu, Dayehkolot, Bandung
1teguhnandi@student.telkomuniversity.ac.id
2saralutamip@telkomuniversity.ac.id
4nikencahyani@telkomuniversity.ac.id

**Informatics Study Program, Informatics Faculty, Universitas Telkom*
Jl. Telekomunikasi No.1, Terusan Buah Batu, Dayehkolot, Bandung
3mayerreflino@student.telkomuniversity.ac.id

Abstract— **An in-depth exploration of a robust and systematic approach aimed at enhancing the security and integrity of communication systems within a website is conducted in this research. The focus is strengthening the interaction between the server and the user or client. Thus, secure data transmission can be guaranteed. This research integrates the well-established and widely respected Waterfall Methodology with asymmetric encryption techniques, explicitly using the RSA algorithm, into the overall development process. This method covers data encryption and decryption comprehensively. Blackbox testing validated the results for the application's expectation of increased research reliability. Advanced security measures are integrated into projects that use these insights to protect user data due to fast-growing cyber threats and the importance of data privacy.**

Keywords— **asymmetric encryption, data protection, RSA algorithm, security, web-based applications.**

Abstrak— Eksplorasi mendalam terhadap pendekatan yang kokoh dan metodis yang bertujuan untuk meningkatkan keamanan dan integritas sistem komunikasi dalam sebuah situs web dilakukan dalam penelitian ini. Fokus utamanya adalah memperkuat interaksi antara server dan pengguna atau klien. Dengan demikian, transmisi data yang aman dapat dijamin. Penelitian ini menggunakan Metodologi Air Terjun yang sudah mapan dan dihormati secara luas lalu mengintegrasikannya dengan teknik enkripsi asimetris, khususnya dengan menggunakan algoritme RSA, ke dalam proses pengembangan secara keseluruhan. Metode ini mencakup enkripsi dan dekripsi data secara komprehensif. Pengujian blackbox digunakan sebagai cara untuk memvalidasi hasil untuk ekspetasi aplikasi dalam peningkatan kehandalan penelitian. Langkah-langkah keamanan canggih diintegrasikan ke dalam proyek-proyek yang menggunakan wawasan ini untuk melindungi data pengguna karena ancaman siber yang berkembang cepat dan pentingnya privasi data.

*Kata Kunci— **algoritme RSA, aplikasi berbasis web, enkripsi asimetris, keamanan, perlindungan data.***

I. Introduction

In today's interconnected world, web-based application security has become a significant concern due to the sophisticated increase in cyber threats [1]. Web applications, from e-commerce platforms to cloud-based services, transmit vast amounts of sensitive data, including personal information, financial records, and proprietary business data [2]. Protecting this data during transmission is crucial to maintaining user confidence and complying with data protection regulations.

Traditional security measures, like firewalls and SSL encryption, are essential but lack full end-to-end protection [3]. SSL, for example, secures data in transit between a user's browser and a web server but fails to encompass the application's backend or databases, leaving vulnerabilities open to exploitation by malicious actors [4]. To tackle this problem, asymmetric encryption, specifically the RSA algorithm, has received widespread recognition. Asymmetric encryption involves two keys - public and private - for encryption and decryption purposes [5]. Material that is encrypted with the public key can only be decoded employing its corresponding private key, thereby primarily enhancing security [6]. This encryption method permits data to be encrypted at its source and decrypted exclusively at its intended destination, guaranteeing end-to-end encryption and safeguarding data even within the application's backend [7].

Nevertheless, robust encryption implementation alone is inadequate. Designing, developing, and deploying a secure web-based application necessitates a structured approach to guarantee the integration of security measures at each point [8]. As part of normal (unencrypted) web communications between web servers and clients, data is sent in plain text, making it vulnerable to interception and unauthorized access [9].

In previous research, we have emphasized the importance of securing these communications to protect sensitive information [10] because it is important to discuss in this research. This approach ensures the confidentiality and integrity of data exchanged between servers and clients, improving the overall security of web-based applications.

The waterfall methodology is a reputable framework that provides a structured method for tackling challenges [8]. It divides the software development process into several stages, such as analyzing the requirements, system design, implementation, testing, and maintenance, and each stage relies on the previous one [11]. By incorporating security factors into each of these stages, developers can methodically produce web-based applications that prioritize data protection without compromising functionality [12]. For validating a testing result, this research will use a black-box testing method to test the application because black-box testing is an efficient method and simple to test [13].

This study aims to merge the advantages of asymmetrical encryption, particularly RSA, with the structured discipline of the waterfall methodology to establish a detailed resolution for the encryption of end-to-end web-based applications [14] [15]. By incorporating security measures at each stage of the development process and using state-of-the-art encryption techniques [16], this approach seeks to meet the crucial requirement for strong web application security in an age where data leaks and cyber threats are widespread [17]. The paper will present theoretical perspectives on the proposed system and practical advice on its implementation. As a result, it will make a valuable contribution to web application security and perhaps make a developer focus on website security, such as the web server and the client communication encryption.

II. Methodology

As explained in the previous section, this research uses a waterfall methodology. In Fig. 1 it is explained that the research contains five steps below.

1) Requirement

During this initial phase, our objective is to identify and define the precise requirements for the web-based encryption system that safeguards the entire application. This necessitates a comprehensive analysis of security requirements, data transmission patterns, and user expectations. With extensive stakeholder input and domain expertise, we strive to create a comprehensive set of project requirements [11].

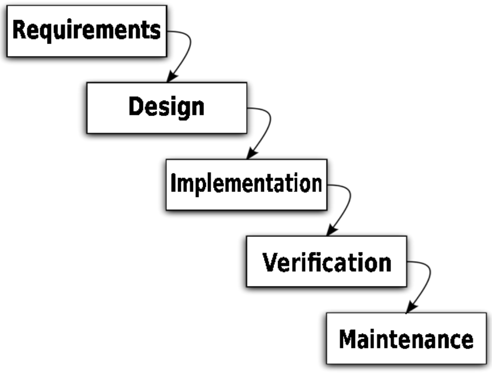


Fig 1. Waterfall methodology

2) System Design

Based on the gathered requirements, the architecture and design of the system are meticulously planned in this stage. This involves defining the encryption mechanisms, data flow, and system components. Technical specifications are created to guide the development process [8][18].

3) Implementation

The implementation phase requires that the web-based application encryption system be developed based on design specifications. Asymmetric encryption algorithms, such as RSA, are integrated into the system, and developers write code, configure servers, and create the necessary database structures [8].

4) Verification (Testing)

Testing is conducted to guarantee the system's functionality, security, and dependability. This stage involves unit testing, integration testing, system testing, and user acceptance testing. Crucially, security testing, comprising penetration testing, is implemented to authenticate the efficiency of the encryption measures [8].

5) Maintenance

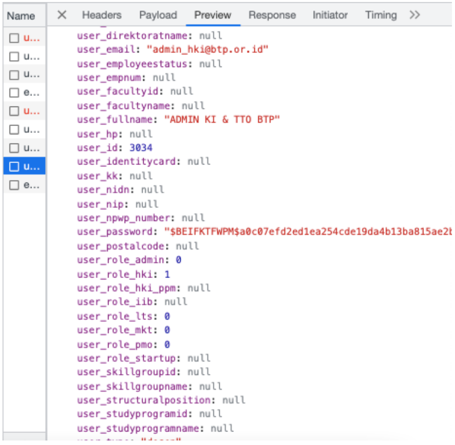
It is crucial to provide ongoing maintenance and support once deployed to tackle any issues, update implementations, and adapt to evolving security threats. Regular maintenance and monitoring activities are carried out to keep the system secure and operational [8] [19].

III. Results and Discussion

In the discussion stages, this research will explain each of the waterfall flow processes from, the requirements, design, implementation, testing, and maintenance stages.

A. Requirements

In this stage, this research analyzes a website communication technique, between the backend and frontend. This research analyzes a Rest API JSON result from a network tab in a browser like in Fig. 2.



```
user_direktoratname: null
user_email: "admin_hki@btp.or.id"
user_employeestatus: null
user_empnum: null
user_facultyid: null
user_facultyname: null
user_fullname: "ADMIN KI & TTO BTP"
user_hp: null
user_id: 3034
user_identitycard: null
user_kk: null
user_nidn: null
user_nip: null
user_npwp_number: null
user_password: "$BEIFKTFWPM$a0c07efd2ed1ea254cde19da4b13ba815ae2
user_postalcode: null
user_role_admin: 0
user_role_hki: 1
user_role_hki_ppm: null
user_role_iib: null
user_role_lts: 0
user_role_mkt: 0
user_role_pmo: 0
user_role_startup: null
user_skillgroupid: null
user_skillgroupname: null
user_structuralposition: null
user_studyprogramid: null
user_studyprogramname: null
```

Fig 2. JSON Response from the backend

Fig. 2 is a JSON response from a backend server, several pieces of information are not encrypted for example email and user ID. This is a piece of private information, if an unauthorized user can brute force a user login form or something dangerous activity is significant to prevent this attack. This research will use an asymmetric encryption technique. The asymmetric encryption technique will use a public and private key for secure website data information. Asymmetric encryption is chosen over symmetric encryption in this research for enhanced security and key management in securing website data information [20]. Unlike symmetric encryption, which employs a single shared key for both encryption and decryption, asymmetric encryption uses a pair of keys: a public key for encryption and a private key for decryption. This dual-key mechanism provides a higher level of security, as the public key can be openly distributed while the private key remains confidential [5]. So, this research will explain more about old design and new design for website communication between the backend and the frontend sides.

B. *Design*

In this stage, this research will explain old and new website communication design. In the first design, this is an old communication design (ordinary or non-encrypted communication). Fig. 3 is an ordinary communication or non-encryption communication. If the user wants a read data, the user must request a website backend and second step server will respond an ordinary data. The data are in Fig. 1.

For a new website, the communication design is in Fig. 4. Fig. 4 is a new technique when the user requests data from a web server. Figure 4 explains that when a user requests data to a web server before the user requests out from the user's phone to a wide network, the data first must be encrypted. An unauthorized user cannot see any user request message to a server.

Fig. 5 is a new technique for server response. Fig. 5 is for the new website and the communication design. Before the response data is out to a wide network, the data will be encrypted first in the server and the data will be delivered to users' mobile phones across the network. In the user browser, the encrypted data will be decrypted by a JavaScript function. So, the user can see the original response from the server.

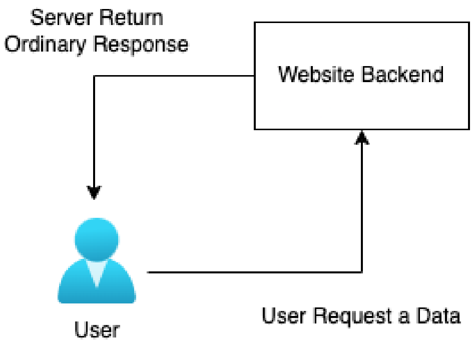


Fig 3. Ordinary communication

Fig. 6 shows how JavaScript encryption or decryption works. Each request or response from the user or server will be encrypted first before it is out from a wide network. So, this research hopes it can be prevented in a middleman attack. For this end-to-end encryption work, we need to save a private key to the user's local storage browser and server. This key is needed to encrypt or decrypt a message to the server.

Fig. 7 is a flowchart from a public and private key generation process from a server and a user's browser. If a user opens a website and no key is available in the client's browser local storage, the server will generate a new key. The server will send the key through the wide internet to a user's browser then local JavaScript engine encryption will be stored in the browser's local storage. Encryption and decryption code will be available in Fig. 8 and Fig. 9 in the implementation stages.

C. *Implementation*

In the implementation stages, this research explains how inside an encryption and decryption process code. The code is written in the JavaScript language.

Fig. 8 is a how inside an encrypt function code. It is the same in JavaScript engine encryption in the server. That encryption function reads the public key from local storage. The key will be used for the encryption process using the RSAES-OAEP/SHA-256 algorithm.

Table I is this research's black-box scenario. This research can be said finished if all scenarios are passed, and not available as a fail-test scenario. The purpose of this research is to secure server and client communication. The testing result is available in Table II.

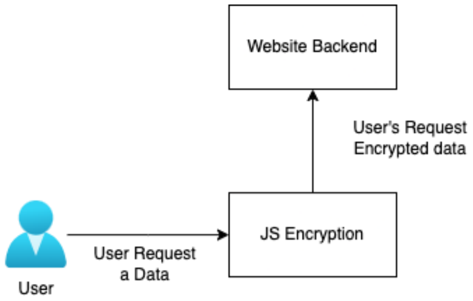


Fig 4. Not encrypted web communication

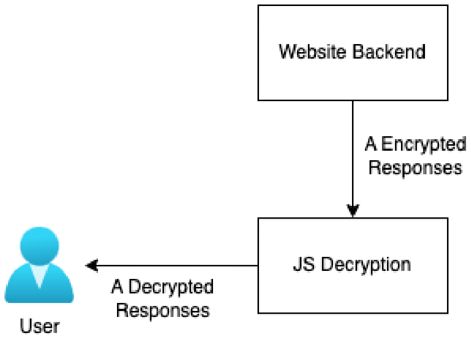


Fig 5. Encrypted communication

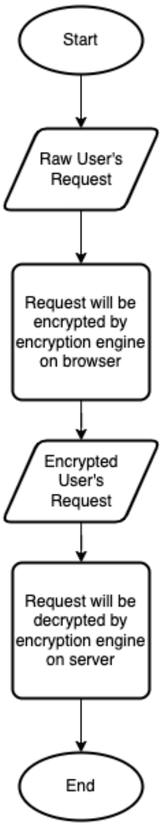

Start
Raw User's Request
Request will be encrypted by encryption engine on browser
Encrypted User's Request
Request will be decrypted by encryption engine on server
End

Fig 6. JavaScript encryption engine works

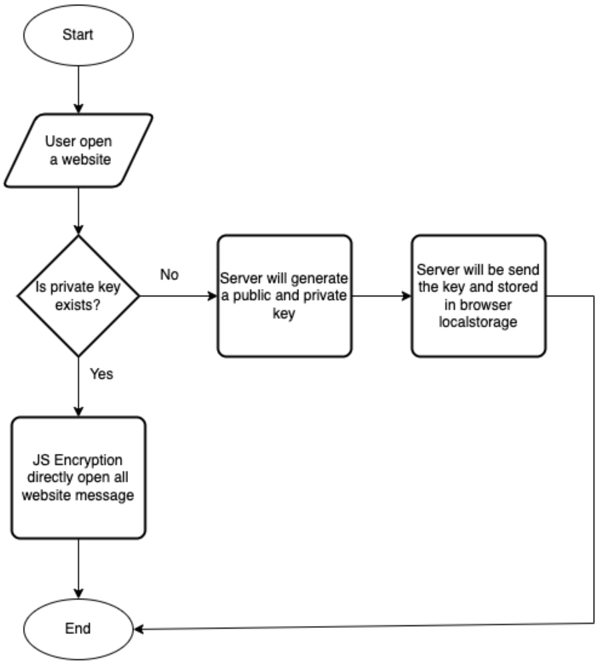
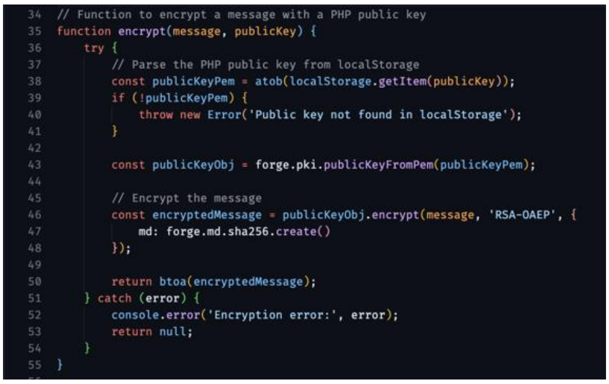

Start
User open a website
Is private key exists?
No
Server will generate a public and private key
Server will be send the key and stored in browser localstorage
Yes
JS Encryption directly open all website message
End

Fig 7. Public and private key flowchart generation process



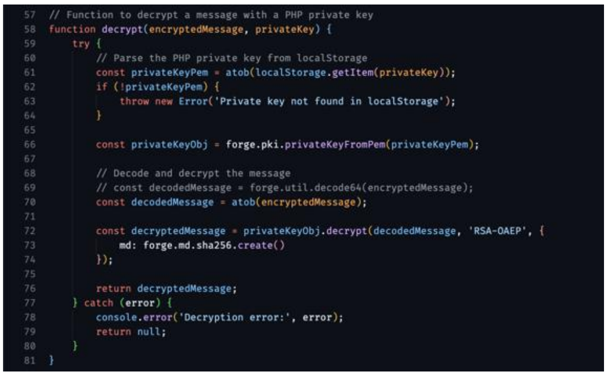
```
// Function to encrypt a message with a PHP public key
function encrypt(message, publicKey) {
    try {
        // Parse the PHP public key from localStorage
        const publicKeyPem = atob(localStorage.getItem(publicKey));
        if (!publicKeyPem) {
            throw new Error('Public key not found in localStorage');
        }

        const publicKeyObj = forge.pki.publicKeyFromPem(publicKeyPem);

        // Encrypt the message
        const encryptedMessage = publicKeyObj.encrypt(message, 'RSA-OAEP', {
            md: forge.md.sha256.create()
        });

        return btoa(encryptedMessage);
    } catch (error) {
        console.error('Encryption error:', error);
        return null;
    }
}
```

Fig 8. Pieces of encryption JavaScript code



```
// Function to decrypt a message with a PHP private key
function decrypt(encryptedMessage, privateKey) {
    try {
        // Parse the PHP private key from localStorage
        const privateKeyPem = atob(localStorage.getItem(privateKey));
        if (!privateKeyPem) {
            throw new Error('Private key not found in localStorage');
        }

        const privateKeyObj = forge.pki.privateKeyFromPem(privateKeyPem);

        // Decode and decrypt the message
        // const decodedMessage = forge.util.decode64(encryptedMessage);
        const decodedMessage = atob(encryptedMessage);

        const decryptedMessage = privateKeyObj.decrypt(decodedMessage, 'RSA-OAEP', {
            md: forge.md.sha256.create()
        });

        return decryptedMessage;
    } catch (error) {
        console.error('Decryption error:', error);
        return null;
    }
}
```

Fig 9. Pieces of decryption JavaScript code

TABLE I
TESTING SCENARIO

No.	Scenario	Expectation
1	Users request a message to a server	The message must be encrypted
2	The server will respond to a secured server response	The response results will be encrypted
3	Server and client Communication are secure	Any unauthorized user can't read any important message

TABLE II
TESTING SCENARIO RESULTS

No.	Scenario	Expectation	Passed/Not
1	Users request a message to a server	The message must be encrypted	Y
2	The server will respond to a secured server response	The response results will be encrypted	Y
3	Server and client Communication are secure	Any unauthorized user can't read any important message	Y

Fig. 9 is a decryption function, written in JavaScript language, the same as Fig. 8. Fig. 9 is a decryption function same in the JavaScript encryption engine in the server. That encryption and decryption function process needs the same public and private key between the server and the user's client browser. If it is not the same, the decryption process will throw a new error exception or a blank white page from the decryption process.

D. Testing

In the testing stages, this research uses black box testing to validate the results of the application if the results are the same based on previous expectations. This research has three scenarios in Table 1. The original message is attached in Fig. 2 above.

Table II is a testing result from that's available scenario. This research has successfully passed all scenario tests. This research attached evidence test results in Fig. 10, Fig. 11, and Fig. 12. For the first test, writer tests a server request. The result is available in Fig. 10. Fig. 10 is an encrypted user's request from a network tab in this research's browser. The result is a user's request is successfully encrypted and, for example, the data breach to an unauthorized person, no one can decrypt without public and private user's key.

Fig. 11 is an encrypted server's response for step two of the scenario, which this research caught from the preview network tab in the browser. This is a long random word from the JavaScript encryption engine in the server. Before the response out as a JSON to a wide network, a user's JavaScript encryption engine will encrypt first and decrypted by a JavaScript encryption engine in the user's local browser.

For the third step, evidence of the scenario can be found in Fig. 10 and Fig. 11, because of from the server and client sides are successfully encrypted.

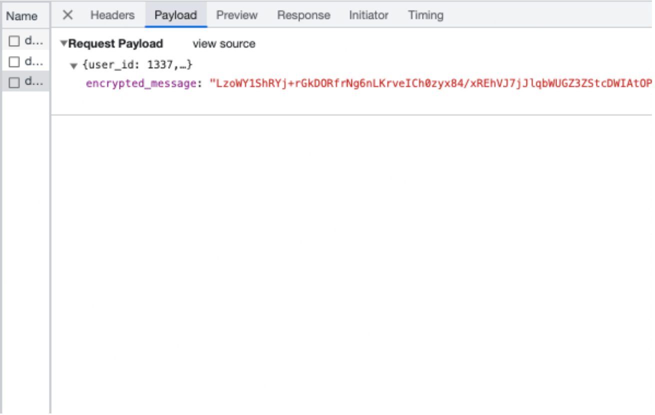
For Fig. 12 above is the decryption illustration process. First, we need the user's private key to open the encryption process and we need the encrypted message. Second, the key and the message will be decrypted by the decryption engine according to Fig. 6, and the result will same as in Fig. 2 previously. This research hopes no one knows what the message from the user is or what is server's response is. If between server and client are successfully encrypted, it can prevent a middleman attack.

E. Maintenance

In the maintenance stages, this research will monitor about encryption and decryption process from an implemented system and will monitor the JavaScript encryption engine between the server side and user side. The encryption and decryption must not fail or error, because this research wants to increase a user's experience for use or access of a website application.

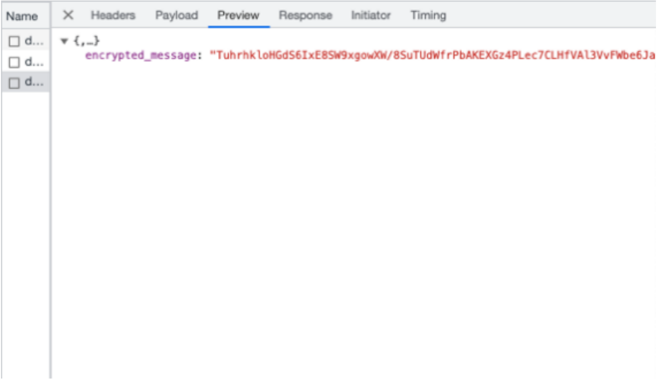
IV. Conclusions

This research highlights the crucial significance of ensuring the protection of user requests and server responses within the extensive domain of network services. The advantage of this research is to enhance and strengthen security to prevent cyber threats and data stolen by integrating asymmetric encryption within the structured framework of the waterfall methodology. This research objective is to explore advanced encryption techniques and further refine key management practices to correct this research's disadvantages. Additionally, the evolving landscape of web-based applications and cloud computing offers prospects for future investigations into secure data transmission across multiple platforms and technologies. By continuously improving our knowledge of web communication security, this research aims to offer better protection for sensitive data in the constantly changing digital landscape.



```
▾Request Payload     view source
  ▾ {user_id: 1337,…}
     encrypted_message: "LzoWY1ShRYj+rGkDORfrNg6nLKrveICh0zyx84/xREhVJ7jJlqbWUGZ3ZStcDWIAtOP
```

Fig 10. Encrypted user's request



```
▾ {,…}
   encrypted_message: "TuhrhkloHGdS6IxE8SW9xgowXW/8SuTUdWfrPbAKEXGz4PLec7CLHfVAl3VvFWbe6Ja
```

Fig 11. Encrypted server's response

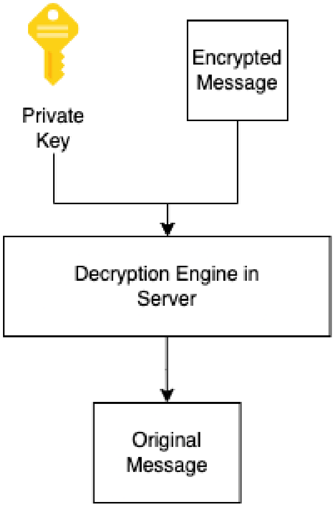


Fig 12. Decrypted process illustration

References

[1] H. M. Alzoubi *et al.*, "Cyber security threats on digital banking," in *2022 1st International Conference on AI in Cybersecurity (ICAIC)*, 2022, pp. 1–4. DOI: 10.1109/ICAIC53980.2022.9896966.

[2] K. Jannes, B. Lagaisse, and W. Joosen, "The web browser as distributed application server: towards decentralized web applications in the edge," in *Proceedings of the 2nd International Workshop on Edge Systems, Analytics and Networking*, in EdgeSys '19. New York, USA: Association for Computing Machinery, 2019, pp. 7–11. DOI: 10.1145/3301418.3313938.

[3] Y. Dun-Yi, "Data encryption method of SSL digital authentication signature system based on privacy protection," in *2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 2020, pp. 40–44. DOI: 10.1109/ICMTMA50254.2020.00016.

[4] B. Dash, M. F. Ansari, P. Sharma, and A. Ali, "Threats and opportunities with AI-based cyber security intrusion detection: a review," *International Journal of Software Engineering & Applications*, vol. 13, no. 5, pp. 13–21, Sep. 2022, DOI: 10.5121/ijsea.2022.13502.

[5] V. Kapoor and R. Gupta, "Hybrid symmetric cryptography approach for secure communication in web application," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 24, no. 5, pp. 1179–1187, 2021, DOI: 10.1080/09720529.2021.1936900.

[6] G. Ribeiro, M. Grabovschi, M. Antunes, and L. Frazão, "Ncryptr: a symmetric and asymmetric encryption application," in *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, 2019, pp. 1–6. DOI: 10.23919/CISTI.2019.8760763.

[7] R. Chatterjee, R. Chakraborty, and J. K. Mandal, "Design of cryptographic model for end-to-end encryption in FPGA based systems," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 2019, pp. 459–465. DOI: 10.1109/ICCMC.2019.8819761.

[8] T. Rijanandi, *et al.*, "Web-based application with SDLC waterfall method on population administration and registration information system (Case Study: Karangklesem Village, Purwokerto)," *Jurnal Teknik Informatika (Jutif)*, vol. 3, no. 1, pp. 99–104, 2022, DOI: 10.20884/1.jutif.2022.3.1.145.

[9] M. R. Royani and A. Wibowo, "*Web Service Implementation in Logistics Company uses JSON Web Token and RC4 Cryptography Algorithm*," *Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi/System Engineering and Information Technology)*, vol. 4, no. 3, pp. 591–600, 2020, DOI: https://doi.org/10.29207/resti.v4i3.1952.

[10] W. Bai, M. Pearson, P. G. Kelley, and M. L. Mazurek, "Improving non-experts' understanding of end-to-end encryption: an exploratory study," in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2020, pp. 210–219. DOI: 10.0.4.85/EuroSPW51379.2020.00036.

[11] W. Steven Dharmawan, *et al.*, "*Penerapan metode* SDLC waterfall *dalam perancangan sistem informasi administrasi keuangan berbasis* desktop," *Jurnal Khatulistiwa Informatika (JKI)*, vol. VI, no. 2, pp. 159-167, Des 2018, DOI: https://doi.org/10.31294/jki.v6i2.5733

[12] E. A. Parn and D. Edwards, "Cyber threats confronting the digital built environment: common data environment vulnerabilities and block chain deterrence," *Engineering, Construction and Architectural Management*, vol. 26, no. 2, pp. 245–266, Mar. 26, 2019. DOI: 10.1108/ECAM-03-2018-0101.

[13] A. Alamgir, A. K. A'ain, N. Paraman, and U. U. Sheikh, "Adaptive random testing with total cartesian distance for black box circuit under test," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 2, pp. 720–726, Nov. 2020, DOI: 10.11591/ijeecs.v20.i2.

[14] L. Chamari, E. Petrova, and P. Pauwels, "An end-to-end implementation of a service-oriented architecture for data-driven smart buildings," *IEEE Access*, vol. 11, 2023, pp. 117261–117281, DOI: 10.1109/ACCESS.2023.3325767.

[15] X. Chen, "Implementing AES encryption on programmable switches via scrambled lookup tables," in *Proceedings of the 2020 ACM SIGCOMM Workshop on Secure Programmable Network Infrastructure, SPIN 2020*, Association for Computing Machinery, Aug. 2020, pp. 8–14. DOI: 10.1145/3405669.3405819.

[16] D. Rachmawati, M. A. Budiman, and F. Atika, "PDF file encryption on mobile phone using super-encryption of variably modified permutation composition (VMPC) and two square cipher algorithm," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Mar. 2018. DOI: 10.1088/1742-6596/978/1/012115.

[17] T. Rijanandi, A. Silvia, B. Abillah Safna, and R. Dias Ramadhani, "Implementation of encrypt national ID card in Sinovi application use waterfall methodology," *RIGGS: Journal of Artificial Intelligence and Digital Business*, vol. 1, no. 2, pp. 11–18, Jan. 2023, DOI: 10.31004/riggs.v1i2.15.

[18] M. Susilo, "*Rancang bangun website toko* online *menggunakan metode waterfall*," *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, vol. 2, no. 2, pp. 98–105, 2018, DOI: 10.30743/infotekjar.v2i2.171.

[19] E. Listiyan and E. R. Subhiyakto, "*Rancang bangun sistem* inventory *gudang menggunakan metode* waterfall (*studi kasus di* CV Aqualux Duspha Abadi, Kudus, Jawa Tengah)." *Jurnal Konstelasi: Konvergensi Teknologi dan Sistem Informasi*, vol. 1, no. 1, pp. 74-82, Juni 2021, DOI: https://doi.org/10.24002/konstelasi.v1i1.4272

[20] Z. Niu, M. Zheng, Y. Zhang, and T. Wang, "A new asymmetrical encryption algorithm based on semitensor compressed sensing in WBANs," *IEEE Internet Things J*, vol. 7, no. 1, pp. 734–750, Jan. 2020, DOI: 10.1109/JIOT.2019.2953519.

Teguh Rijanandi, born in Purwokerto. Teguh is one of the Nine Best Graduates who made Telkom Institute of Technology Purwokerto proud at the XVII Graduation in June 2023 in the Innovation category. Teguh's current activity is joining a Cybersecurity and Digital Forensics Master Program at Telkom University.

Sara Lutami Pardede, a student of Master of Forensic Science at Telkom University Bandung and now focusing on cybersecurity science, especially in social media analysis.

Mayer Reflino Sitorus, a student of the Informatics Study Program at Telkom University Bandung and currently studying cyber and digital forensics and developing remote android acquisition.

Niken Dwi Wahyu Cahyani, a lecturer of undergraduate and master programs at Telkom University in the Forensic Science Master Program and currently focusing on the field of digital forensics, and has contributed a lot to digital investigators in Indonesia.