Indonesian Journal of Electrical Engineering and Computer Science
Vol. 38, No. 3, June 2025, pp. 1957~1969
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v38.i3.pp1957-1969

Ensuring transcript integrity with SHA-3 and digital signature standard: a practical approach

Wa Ode Siti Nur Alam[1], Adha Mashur Sajiah[2], La Ode Muhammad Bahtiar Aksara[2], La Surimi[3], Natalis Ransi[3], Jumadil Nangi[3]

[1]Department of Electrical Engineering, Halu Oleo University, Kendari, Indonesia
[2]Department of Informatics Engineering, Halu Oleo University, Kendari, Indonesia
[3]Department of Computer Science, Halu Oleo University, Kendari, Indonesia

Article Info

Article history:

Received Jul 13, 2024
Revised Dec 17, 2024
Accepted Feb 27, 2025

Keywords:

Digital signature standard
Grade transcript
Halu Oleo University
SHA-3
SIAKAD

ABSTRACT

Academic transcripts are essential documents in higher education, reflecting students' academic performance and capabilities. However, the current management of transcript data at Halu Oleo University (UHO) lacks safeguards against unauthorized alterations, compromising their authenticity. This study proposes a method using the secure hash algorithm 3 (SHA-3) and the digital signature standard (DSS) scheme to ensure the integrity of transcript data. A Python-based web module for managing transcripts and a signing program using SHA-3 and DSS were developed and implemented. This method digitally signs transcript files, ensuring that subsequent changes invalidate the current digital signature. Efficiency tests demonstrated an average signing time of 0.242 seconds, indicating a practical and efficient solution. The study's findings emphasize how SHA-3 and DSS effectively authenticate academic transcript files, preventing unauthorized modifications and safeguarding the integrity of critical educational records. This method presents a robust and efficient solution for educational institutions to strengthen the security and reliability of their academic record management systems.

Corresponding Author:

Wa Ode Siti Nur Alam
Department of Electrical Engineering, Halu Oleo University
Street of H.E.A Mokodompit, Kampus Hijau Bumi Tridharma, Anduonohu
Kendari City, Southeast Sulawesi, 93232 Indonesia
Email: wdsitinuralam@uho.ac.id

1. INTRODUCTION

Academic transcripts are important records of a student's success in classes that act as an indicator of their skills and knowledge. The cumulative achievement index (CAI), which assesses if a student satisfies the requirements to pass, is calculated in part using these papers. Owing to flaws in their digital format, academic transcripts at Halu Oleo University (UHO) are currently not adequately managed, despite their significance in both academic and professional contexts. Transcripts created by the academic information system (SIAKAD) are available for viewing or downloading as common digital files, such as PDFs. But because there aren't enough security safeguards in place to confirm their legitimacy, these files are vulnerable to manipulation and illegal changes.

Transcript manipulation cases involving students fabricating grades and CAI scores for monetary or career advantage have been documented throughout Indonesia. For instance, students in Nagan Raya Regency falsified their transcripts in order to be awarded scholarships; this cost the government IDR 1.15 billion. Similar to this, the selection committee in the Riau Archipelago was duped by the use of manipulated transcripts in civil servant selection procedures. These occurrences highlight a serious issue: there is currently no trustworthy system in place to guarantee the validity and integrity of academic transcripts, which puts the reputation of educational institutions as well as the larger academic and professional community at risk.

One way to overcome the problem of verifying the authenticity of transcript files is to use digital signatures [1], [2]. Digital signatures have been acknowledged as a workable solution to these problems, offering a method to confirm the authenticity of documents and identify changes by encrypting their content with secret keys. A digital signature is a security method that marks the authenticity of a document by breaking it into hash bits [3], using a secret key as encryption to create authentication, which is then agreed upon as a signature [4]–[7]. Digital signatures aim to assure that a document is owned by the signatory [8], [9], and the signer cannot deny ownership of the document. Besides ensuring document ownership, digital signatures can check for changes in document content [10], [11]. Digital signatures utilize a hash function in the form of a message digest to verify any changes or manipulations to the contents of a document [12], [13]. If related to the problem above, digital signatures can prove that the transcript value sent by scholarship applicants and CPNS selection to the committee is an original authenticated document from the campus that has been integrated and free from manipulation so that the possibility of falsifying or changing the authenticity of files can be avoided, especially from irresponsible people [14], [15]. They provide a distinct message digest using cryptographic hash methods like secure hash algorithm 3 (SHA-3), which can reveal any alterations made to the document's content.

A hash function also assists the digital signature scheme in digital signature standard (DSS), and for hash functions, there are various types, including MD5, SHA-1, SHA-2, SHA-3, RIPEMD-160, BLAKE2, and many more [16], [17]. However, judging from the comparison of hash systems, several essential things, such as MD5 and SHA-1, [18] have been considered obsolete because these algorithms have successfully solved and are vulnerable to hash collisions [19] (different data have the same hash value). However, SHA-3 has a different structure than its predecessors (SHA-1 and SHA-2) [20], [21]. SHA-3 uses a sponge structure, while SHA-2 uses a Davies–Meyer structure. There is no user-visible difference here, but it makes a difference to cryptographers' confidence in the design after many hash designs using Davies–Meyer based on MD4 completed completion in the late 90s and early 2000s [22]–[25]. Digital signatures are widely used across various industries, but their specific role in protecting academic credentials requires further exploration. This study aims to address this gap by focusing on the security of academic records. It evaluates the effectiveness of SHA-3 and the DSS in safeguarding academic transcripts against unauthorized modifications.

To tackle the issue of document integrity, several cryptographic techniques have been created. The SHA are well known for their capacity to produce distinct hashes for data, ensuring that alterations made to the original document provide an entirely distinct hash value. Documents and other sensitive data have long been protected by earlier iterations of SHA, such SHA-1 and SHA-2. But studies have demonstrated that SHA-1 has flaws, which is why the national institute of standards and technology (NIST) decided to stop using it in favor of better algorithms like SHA-3 [26], [27]. The newest member of the hash function family, SHA-3, is based on the Keccak algorithm, which has been demonstrated to be resilient to a variety of cryptographic attacks and has enhanced security characteristics [28]. Despite SHA-3 is capable of identifying changes made to a document, it does not offer a way to confirm the document's original source.

Cryptographic digital signatures are used to handle both document integrity and authenticity. In order to confirm document integrity and authenticate its source, NIST's DSS makes use of methods such as the elliptic curve digital signature algorithm (ECDSA) [29]. Sensitive record security using DSS has been investigated in earlier research [30], [31]. Although it is evident that educational institutions need a system like this, there is little study on integrating DSS and SHA-3 for the security of academic transcripts, moreover in practical applications. Regarding the aforementioned, although SHA-3 has proven effective in ensuring data integrity and DSS has been successfully utilized for verifying document authenticity, their combined application specifically for safeguarding academic transcripts has not been extensively studied. This research aims to fill that gap by proposing a practical approach that integrates the SHA-3 hash function with DSS, providing both source authentication and integrity verification within a unified system. By doing so, this study will enable educational institutions to strengthen the security of their digital records and enhance protection against document fraud. Specifically, it will examine the effectiveness of SHA-3 and DSS in verifying the authenticity of transcripts and preventing unauthorized alterations, addressing a critical security concern in academic record management.

2. METHOD

The research methodology began with a system requirements analysis, which included both functional and non-functional requirements. This was followed by the PHP and Python implementation phase, which included testing. Finally, the outcomes of the tests were examined.

2.1. Beginning

At this stage, an analysis is conducted on the method that will be developed. This includes the benefits, objectives, and limitations of the problems to be addressed. Functional requirements analysis is the analysis stage that determines what processes/services must be available in the system to be built. Meanwhile, the functional requirements of the system to be built are shown in Table 1, and the analysis of non-functional system requirements is shown in Tables 2 and 3.

Table 1. System functional requirements

Aspect	Need
User	Users in this transcript data integrity verification system have three categories of users, namely: a. Admin SIAKAD b. Study program admin c. Student
Feature	The facilities/features that this system must have are: a. Generate temporary and permanent transcripts b. Digital signature of the transcript (DSS) c. Verify the integrity of the transcript file

Table 2. Minimum specifications for computer equipment

Device name	Specification
Processor	Intel Core i5
Memory	RAM 8 GB DDR4 L memory

Table 3. Computer software specifications

Software name	Specification
Operating system	Windows 10
XAMPP	Version 3.2.4
Browser	Google Chrome/Mozilla Firefox/Opera GX Browser
Text editor	Sublime/visual studio code

2.2. Elaboration

The system design will utilize the unified modeling language (UML). The use case diagram of this transcript data integrity verification system is shown in Figure 1. It shows the actors involved in the system and the services/functions that can be obtained from the system.

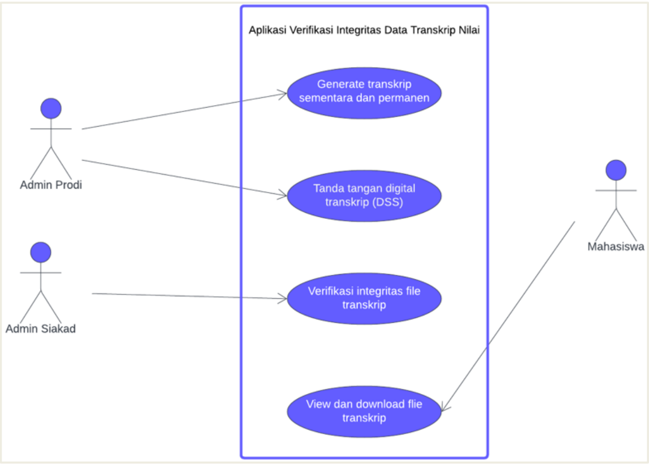

Aplikasi Verifikasi Integritas Data Transkrip Nilai
Generate transkrip sementara dan permanen
Tanda tangan digital transkrip (DSS)
Verifikasi integritas file transkrip
View dan download flie transkrip
Admin Prodi
Admin Siakad
Mahasiswa

Figure 1. Use case diagram of grade transcript data integrity verification system

The transcript data integrity system architecture can be seen in Figure 2, it consists of three users: study program admin, SIAKAD admin, and students. The study program admin is tasked with generating transcript data automatically according to the study results card that has passed the semester and filling in the digital signature. Then, the study program admin requests the SIAKAD admin to verify the integrity of the transcript file. After that, the SIAKAD admin received a request from the study program admin and checked the grades transcript file. After that, if the transcript file is guaranteed to be authentic, the SIAKAD admin can confirm that the file is genuine. Students can download transcript files signed with DSS by the system.

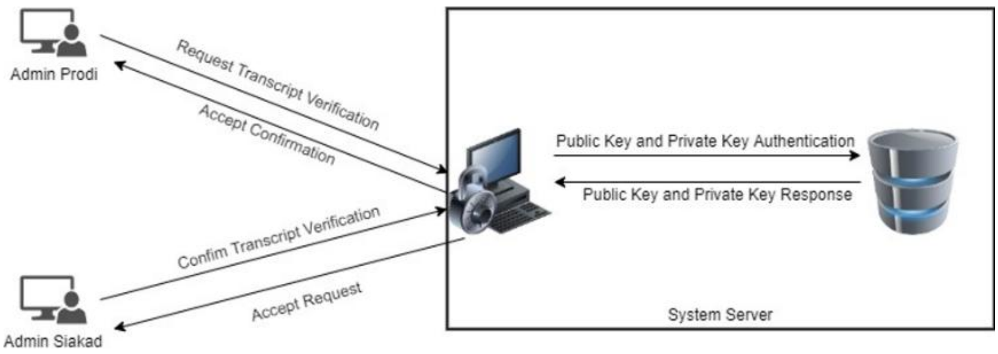

Admin Prodi
Request Transcript Verification
Accept Confirmation
Public Key and Private Key Authentication
Public Key and Private Key Response
Confirm Transcript Verification
Accept Request
Admin Siakad
System Server

Figure 2. Architecture of the grade transcript data integrity verification system

The DSS scheme was implemented using the following key steps: (i) key generation, where a pair of public and private keys will be generated. The head of the study program owns these keys. The key pair is created using the Rivest-Shamir-Adleman (RSA) algorithm and stored in a file with a .p12 extension; (ii) signature, the process schema for the digital signature (digital signature encryption) is illustrated in Figure 3. Digital signature process: Figure 3(a) signing using a hash and private key to generate a signature, and Figure 3(b) verification by comparing the regenerated hash with the signature. In this process, the transcript document and the previously generated .p12 file were hashed using SHA-3. Each file produced a message digest, which was then combined. The combined message digest was stored in the database and encrypted with a passphrase set by each head of study program. This process took place on the digital signature server, generating the digital signature. The digital signature was appended to the student's academic transcript file; and (iii) verification, the verification process for the digital signature is shown in Figure 3(b). The transcript document containing the digital signature was described using a public key, which generated a hash. This hash data was checked by the digital signature server; if the hash file matched the data in the database, the file was considered authentic. If not, it was deemed fraudulent.

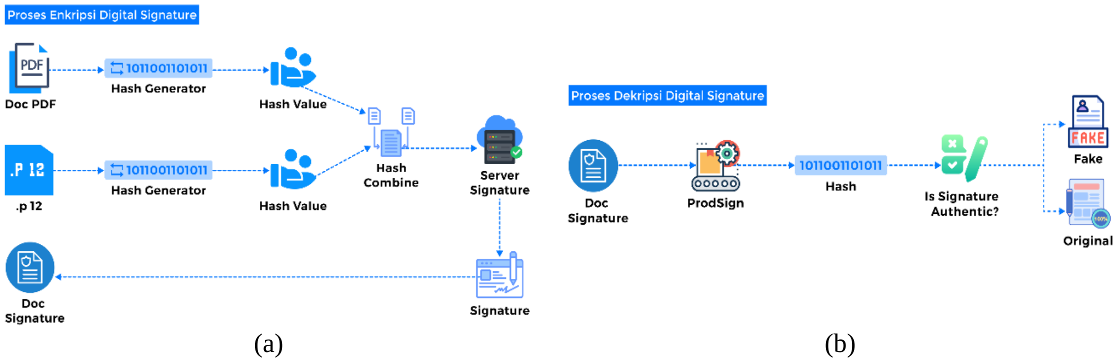


Figure 3. Illustration of the digital signature process (a) encryption/signing, where the document is processed using a hash value and private key to generate a digital signature and (b) decryption/verification, where the document's authenticity is confirmed by comparing the regenerated hash with the original signature

Based on the design above, the system was implemented as a web-based application, following the structure of an existing web application (SIAKAD). This application was deployed on a server capable of running PHP and Python programs. The server used was still monolithic without load balancing. The implementation consisted of modules for transcript creation, downloading signed transcripts, and transcript verification. The transcript management module was developed using PHP with a framework. Meanwhile, the SSH-3 and DSS algorithms were implemented in Python and then compiled for faster processing. The digital signature module, which was implemented in Python, was invoked through the transcript management module using PHP.

In the implemented system, a series of tests were conducted. The testing in this research was carried out to determine whether the system operated according to the desired functionality, using black box testing. In addition, performance testing was conducted to assess system efficiency by measuring computation time. This research utilized a dataset of 10 transcript files from various study programs at UHO to evaluate the proposed method. Quantitative data collected from functionality and performance tests are analyzed to assess the efficiency and reliability of the system.

3. RESULTS AND DISCUSSION

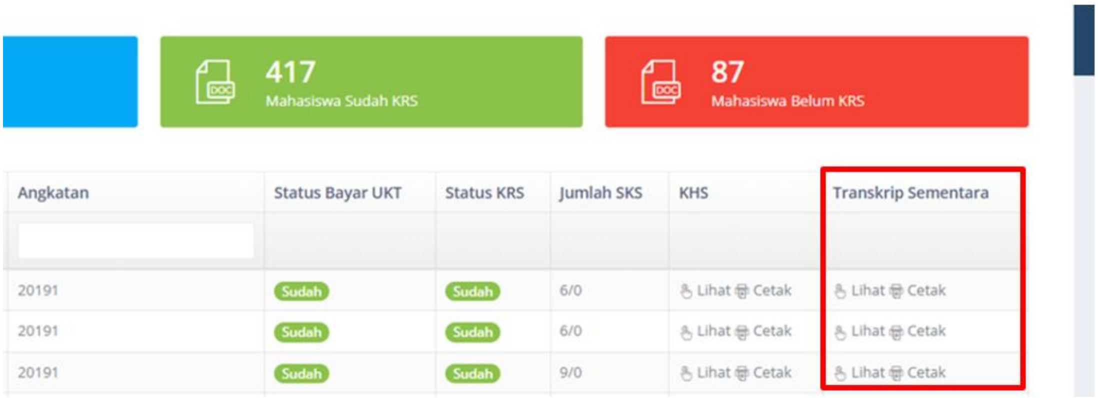
This transcript data integrity verification system is implemented with a MySQL database and PHP and Python programming languages. The PHP programming language is used to create a transcript management module in SIAKAD, which the head of the department will later use. Python is used to implement the designed digital signature algorithm. This Python application is a shell program called via a website-based application.

3.1. Temporary transcript module

The transcript module created is a temporary transcript application module. Departments will use this module to print temporary student transcript data, usually used to arrange scholarships, internships, and advanced requirements for seminar exams. This module is an additional module to the SIAKAD application that is already running. This module can only be accessed by study program admins.

3.1.1. Temporary transcript

The temporary transcript menu is the initial menu to start managing student transcripts. This menu is under the student menu. This page contains a list of students in managed study programs. The final column of the student list is modified by adding a link to the student's transcript. The temporary KHS/transcript page can be seen in Figure 4.



417 Mahasiswa Sudah KRS

87 Mahasiswa Belum KRS

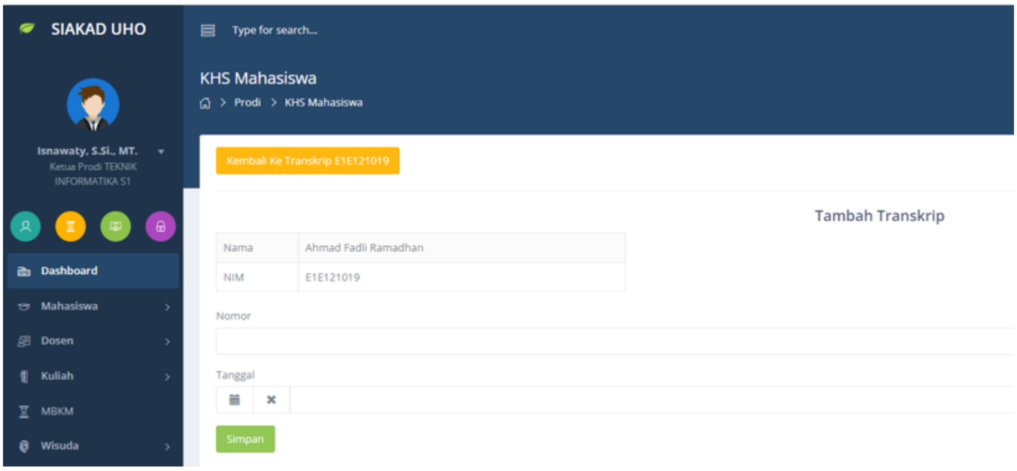
Angkatan	Status Bayar UKT	Status KRS	Jumlah SKS	KHS	Transkrip Sementara
20191	Sudah	Sudah	6/0	Lihat Cetak	Lihat Cetak
20191	Sudah	Sudah	6/0	Lihat Cetak	Lihat Cetak
20191	Sudah	Sudah	9/0	Lihat Cetak	Lihat Cetak

Figure 4. Temporary transcript

3.1.2. Student temporary transcript page

This page, the study program admin, manages transcript data and grades from one student's transcript. This page can perform two main functions. The first is to fill in data regarding the transcript to be made. The second is to manage the value of the transcript. This temporary transcript page can be seen in Figure 5.

The add/edit data transcript page has four main buttons: see the list of grades that can be added, print temporary transcripts, print temporary digital signature transcripts, and edit transcript data. The Edit/Add transcript data page will appear when you click edit transcript data. This page can be seen in Figure 6.



KHS Mahasiswa

Prodi > KHS Mahasiswa

Kembali Ke Transkrip E1E121019

Tambah Transkrip

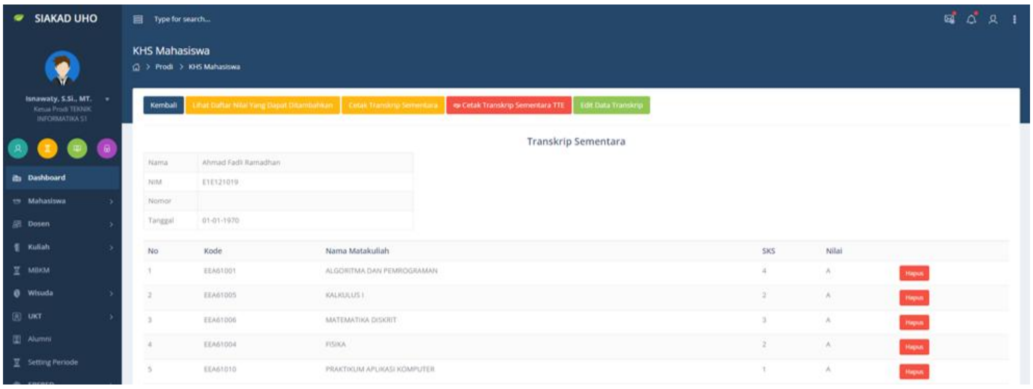
Nama	Ahmad Fadli Ramadhan
NIM	E1E121019

Nomor

Tanggal

Simpan

Figure 5. Student temporary transcript page



KHS Mahasiswa

Prodi > KHS Mahasiswa

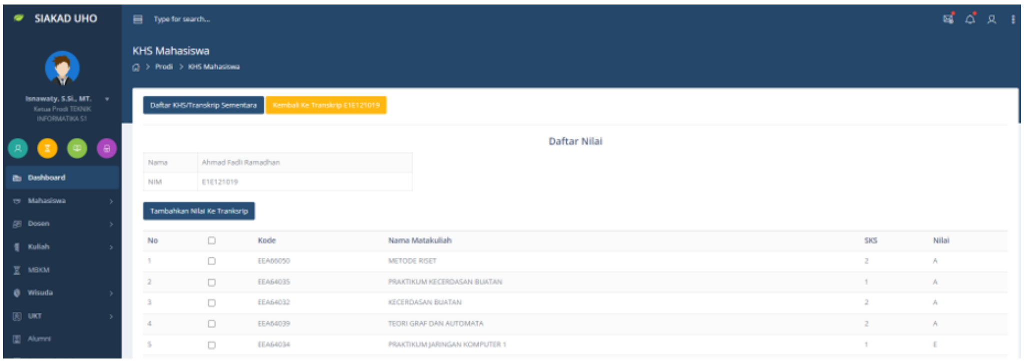
Transkrip Sementara

Nama	Ahmad Fadli Ramadhan
NIM	E1E121019
Nomor	
Tanggal	01-01-1970

No	Kode	Nama Matakuliah	SKS	Nilai	
1	EEA61001	ALGORITMA DAN PEMROGRAMAN	4	A	Hapus
2	EEA61005	KALKULUS I	2	A	Hapus
3	EEA61006	MATEMATIKA DISKRIT	3	A	Hapus
4	EEA61004	FISIKA	2	A	Hapus
5	EEA61010	PRAKTIKUM APLIKASI KOMPUTER	1	A	Hapus

Figure 6. Add/edit data transcript page

When you click the view list of values that can be added button, you will be directed to the list of values that can be added. This page lists the grades for the courses that students have programmed. This page may have repeated courses if the student has programmed a course several times. On this page, the study program admin claims the grades for the classes he wants to include in the transcript by ticking the courses. This page can be seen in Figure 7. Next, the print transcript button will instruct the system to create a temporary transcript in PDF version. The PDF file created is a regular version of the digital/electronic handless sign. An example of the PDF results produced can be seen in Figure 8.


SIAKAD UHO
Type for search...
Isnawaty, S.Si., MT.
Ketua Prodi TEKNIK INFORMATIKA S1
Dashboard
Mahasiswa
Dosen
Kuliah
MBKM
Wisuda
UKT
Alumni
Daftar KHS/Transkrip Sementara
Kembali Ke Transkrip E1E121019
Tambahkan Nilai Ke Tranksrip

KHS Mahasiswa

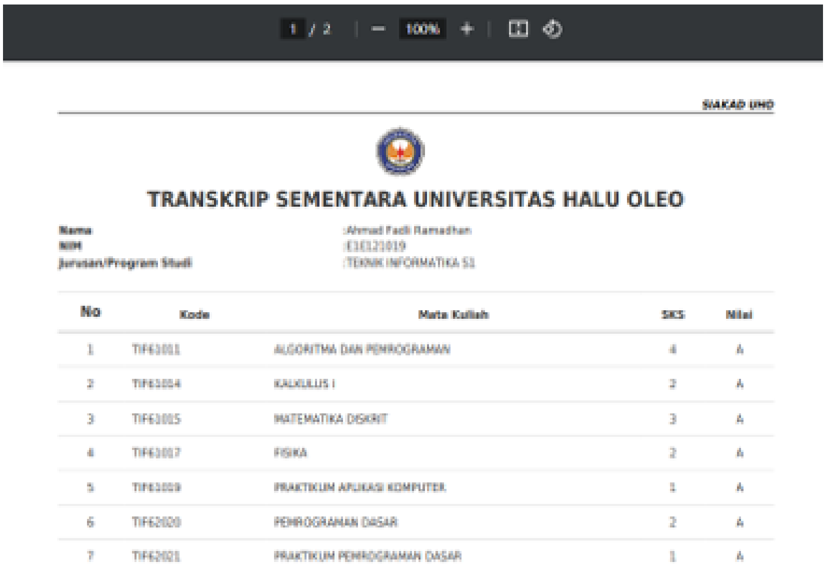
Prodi > KHS Mahasiswa

Daftar Nilai

Nama	Ahmad Fadli Ramadhan
NIM	E1E121019

No	☐	Kode	Nama Matakuliah	SKS	Nilai
1	☐	EEA66050	METODE RISET	2	A
2	☐	EEA64035	PRAKTIKUM KECERDASAN BUATAN	1	A
3	☐	EEA64032	KECERDASAN BUATAN	2	A
4	☐	EEA64039	TEORI GRAF DAN AUTOMATA	2	A
5	☐	EEA64034	PRAKTIKUM JARINGAN KOMPUTER 1	1	E

Figure 7. List of grades that can be claimed on the transcript



SIAKAD UHO

TRANSKRIP SEMENTARA UNIVERSITAS HALU OLEO

Nama	:Ahmad Fadli Ramadhan
NIM	:E1E121019
Jurusan/Program Studi	:TEKNIK INFORMATIKA S1

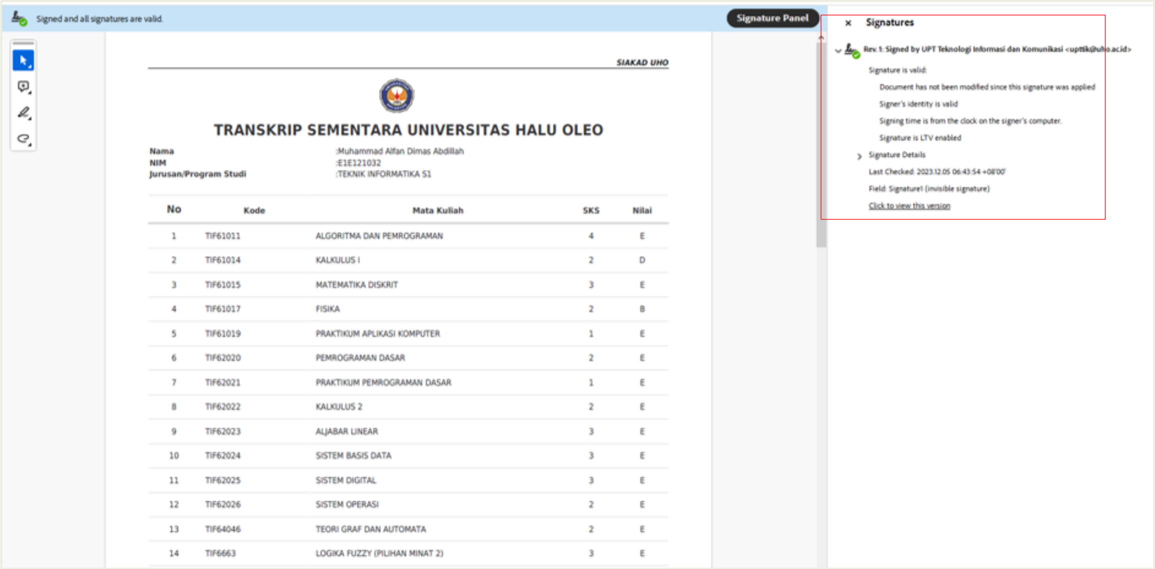
No	Kode	Mata Kuliah	SKS	Nilai
1	TIF61011	ALGORITMA DAN PEMROGRAMAN	4	A
2	TIF61014	KALKULUS I	2	A
3	TIF61015	MATEMATIKA DISKRIT	3	A
4	TIF61017	FISIKA	2	A
5	TIF61019	PRAKTIKUM APLIKASI KOMPUTER	1	A
6	TIF62020	PEMROGRAMAN DASAR	2	A
7	TIF62021	PRAKTIKUM PEMROGRAMAN DASAR	1	A

Figure 8. Printed PDF of the temporary transcript

3.1.3. Digital signature Python program

A Python-based digital signature application was developed for the signing process. The web application generates a PDF from the transcript and gathers the necessary digital signature data. This data is then sent to the Python application for processing.

In the Python application/program, the data sent from the web application is collected, digitally signed, and saved in the file with the name signed-OriginalDocumentName.pdf. An image of the results of the transcript document with digital signature can be seen in Figure 9. The red box is the signature panel visible from the Adobe Reader application.



Signed and all signatures are valid.

SIAKAD UHO

TRANSKRIP SEMENTARA UNIVERSITAS HALU OLEO

Nama	:Muhammad Alfan Dimas Abdillah
NIM	:E1E121032
Jurusan/Program Studi	:TEKNIK INFORMATIKA S1

No	Kode	Mata Kuliah	SKS	Nilai
1	TIF61011	ALGORITMA DAN PEMROGRAMAN	4	E
2	TIF61014	KALKULUS I	2	D
3	TIF61015	MATEMATIKA DISKRIT	3	E
4	TIF61017	FISIKA	2	B
5	TIF61019	PRAKTIKUM APLIKASI KOMPUTER	1	E
6	TIF62020	PEMROGRAMAN DASAR	2	E
7	TIF62021	PRAKTIKUM PEMROGRAMAN DASAR	1	E
8	TIF62022	KALKULUS 2	2	E
9	TIF62023	ALJABAR LINEAR	3	E
10	TIF62024	SISTEM BASIS DATA	3	E
11	TIF62025	SISTEM DIGITAL	3	E
12	TIF62026	SISTEM OPERASI	2	E
13	TIF64046	TEORI GRAF DAN AUTOMATA	2	E
14	TIF6663	LOGIKA FUZZY (PILIHAN MINAT 2)	3	E

× Signatures

Rev. 1: Signed by UPT Teknologi Informasi dan Komunikasi <upttik@uho.ac.id>

Signature is valid:

Document has not been modified since this signature was applied

Signer's identity is valid

Signing time is from the clock on the signer's computer.

Signature is LTV enabled

Signature Details

Last Checked: 2023.12.05 06:43:54 +08'00'

Field: Signature1 (invisible signature)

Click to view this version

Figure 9. PDF document that has been successfully signed

3.2. System testing

At this stage, the system will be tested in two categories. The first test tests the temporary transcript website module and Python application, which have been developed using black box testing. The second test was carried out on the digital signature Python application in the form of performance in the digital signature process.

3.2.1. Blackbox testing

A. Transcript module testing

In testing the transcript module, various scenarios are carried out for each function in the module. The tasks tested are accessing the transcript list, viewing student transcripts, updating transcript data, adding grades to transcripts, and downloading transcripts to determine whether they have digital signature. The system runs as desired, both with standard parameters and invalid parameters. The list of test cases and the results for the black box testing conducted to evaluate the transcript module is shown in Table 4.

Table 4. Results of black box testing of the SIAKAD transcript module

Test cases	Desired outcome	System results	Status
Access the transcript list page with the URL /Prodi-khs/index	Displays a list of students and a link to each student's transcript.	Displays a list of students and a link to each student's transcript.	Success
Access the transcript page with the nim parameter listed /prodi-khs/view-transcript2?id={parameter}	Displays student transcripts according to the NIM parameters sent.	Displays student transcripts according to the NIM parameters sent.	Success
Accessing the transcript page with the unregistered nim parameter/prodi-khs/view-transcript2?id={parameter}	Send bad request response (#400)	Send bad request response (#400) Error KRSM001: student with NIM not found	Success
Access the value list page with the nim parameter listed /prodi-khs/register-value?id={parameter}	Displays a list of grades that can be added to the student's transcript according to the parameters.	Displays a list of grades that can be added to the student's transcript according to the parameters.	Success
Accessing the value list page with the nim parameter not listed /prodi-khs/register-value?id={parameter}	Send bad request response (#400)	Send bad request response (#400) Error KRSM001: student with NIM not found	Success
Add/claim grades to student transcripts.	Grades are successfully claimed and appear on the transcript.	Grades are successfully claimed and appear on the transcript.	Success
	Values claimed do not appear in the list of values that can be added.	Values claimed do not appear in the list of values that can be added.	Success
Accessing the transcript data edit page /Prodi-khs/add-transcript?id={parameter}	Displays the transcript data edit form.	Displays the transcript data edit form.	Success
Save transcript data /Prodi-khs/add-transcript?id={parameter}	Saved data	Saved data	Success
Downloading Transcript Files Without digital signature	The transcript file was downloaded. The transcript file does not have a digital signature.	The transcript file was downloaded. The transcript file does not have a digital signature.	Success
Downloading Transcript Files With digital signature	The transcript file was downloaded. Transcript files have digital signature.	The transcript file was downloaded. Transcript files have digital signature.	Success

The results show that the transcript management system is both functional and dependable in managing the diverse inputs that are expected in real-world application. The use of SHA-3 and DSS digital signatures protects the transcript files' integrity, giving a high level of protection against potential document tampering or manipulation. The black box testing indicates that even when erroneous parameters are entered, the system handles the mistakes effectively while maintaining security, demonstrating its robustness. This is crucial for keeping academic records untampered.

B. Python application testing

Table 5 shows the set of test cases and results from the black box testing used to evaluate the Python application. In this test case, testing is carried out by combining the three primary parameters of the Python application execution. The three parameters are the NIDN of the head of the study program, certificate file name and certificate PIN. The most important thing is ensuring that when modifications are made to a PDF file with digital signature, the modified transcript PDF file will be damaged or invalid.

The Python application testing demonstrates that the digital signature generation and verification operations are quite efficient. The most important conclusion here is that any changes to a transcript file automatically invalidate the document, as indicated by the deletion of the signature panel in Adobe Acrobat. This behavior is consistent with the planned security standards and indicates the effectiveness of the SHA-3/DSS combo in preventing unwanted alterations.

Table 5. Python application black box testing results

Test cases	Desired outcome	System results	Status
The head of the study's NIDN is correct. The certificate name and PIN are accurate.	Sends a success status response with the message: signature successful.	Sends a success status response with the message: signature successful.	Success
The head of the study's NIDN is correct. The certificate name is accurate, but the certificate PIN is incorrect.	Sends an error status response and a Failed message—incorrect PIN.	Sends an error status response and a failed message—incorrect PIN.	Success
The head of the study's NIDN is correct, the certificate name is wrong, and the certificate PIN is correct.	She was sending error status responses and failed messages. The certificate file was not found.	She was sending error status responses and Failed messages. The certificate file was not found.	Success
The head of the study's NIDN is correct, the certificate name is wrong, and the certificate PIN is incorrect.	She was sending error status responses and failed messages. The certificate file was not found.	She was sending error status responses and failed messages. The certificate file was not found.	Success
The NIDN head of study program is wrong, the certificate name is correct, and the certificate PIN is correct.	She was sending error status responses and failed messages. The certificate file was not found.	She was sending error status responses and failed messages. The certificate file was not found.	Success
The head of the study's NIDN is incorrect, the certificate name is correct, and the certificate PIN is incorrect.	She was sending error status responses and failed messages. The certificate file was not found.	She was sending error status responses and failed messages. The certificate file was not found.	Success
The head of the study's NIDN is wrong. The certificate name is incorrect, but the certificate PIN is correct.	She was sending error status responses and failed messages. The certificate file was not found.	She was sending error status responses and failed messages. The certificate file was not found.	Success
Incorrect NIDN of study program head, incorrect certificate name, incorrect certificate PIN	She was sending error status responses and failed messages. The certificate file was not found.	She was sending error status responses and failed messages. The certificate file was not found.	Success
Make modifications to Transcript Files that have digital signature	Digital signature is damaged or invalid	Digital signature is missing from Adobe Panel.	Success

3.2.2. Digital signature application efficiency testing

In this testing scheme, we test the success of the running application and the efficiency of the process. This ensures that the application can dash and is practical for users. This test only tests the signature process running in the Python application. This test runs on a PC with an Intel(R) Core(TM) i5-9400T CPU @1.80 GHz 1.80 GHz and 12.0 GB RAM. The operating system is Linux Ubuntu, and the Python version is 3.12.0. There is only one request to the application at a time. This test was applied ten times; the results are shown in Table 6.

Table 6. Python application execution time test results

2nd test	Signature time (ms)
1	248,99888038635254
2	251,6953945159912
3	238,9659881591797
4	232,00106620788574
5	230,03578186035156
6	236,48890356622267
7	240,12890895863266
8	239,90949358785479
9	258,03809844680743
10	248,05890356935898
Average	242,43214192586300

The observation results in Table 6 show that the time to sign takes an average of 242.43214192586300 milliseconds or around 0.242 seconds. The range of values from 10 tests shows values between 230.03578186035156 – 258.03809844680743 milliseconds. In stormy conditions, we can see that the maximum time is around 0.258 seconds. This time is speedy and efficient even though the signature process using a cryptographic algorithm is very complex and has many iterations. This process is fast because the signature does not rely on the web application, in this case, PHP. The signature cryptography process uses Python, which runs in a computer program shell instead of other applications such as PHP on the Apache server.

Of course, with this fast value, the application can simultaneously sign several files in one program execution. With around 0.25 seconds to sign 1 document, in 1 second, the program can sign four documents. In 1 minute, the application can sign 240 papers. So, this signature process can also be implemented for batch processing in the digital signing of student transcripts. Apart from having an efficient signature process, this system also checks the validity of transcript documents efficiently. Documents signed can be sent via the internet network, and their validity can be confirmed quickly, speeding up business processes in higher education. Previous studies have explored various methods of securing academic documents. For instance, Nadzifarin and Asmunin [32] used Blake2b Hash and ECDA, while Indriyawati *et al.* [33] utilized QR codes for signature verification. However, these methods have limitations in terms of processing speed, document integrity, and ease of implementation within institutional frameworks. This study addresses these gaps by integrating the SHA-3 algorithm and DSS, offering a more efficient, secure, and easily implementable solution.

The results demonstrate that the proposed system achieves both speed and efficiency in the digital signing process, with an average signing time of approximately 0.242 seconds per document. This performance supports batch processing, enabling the signing of up to 240 documents per minute, which is a significant improvement for administrative tasks. Furthermore, the system not only processes signatures efficiently but also validate document authenticity seamlessly over the internet, providing a robust solution for accelerating business processes in higher education institutions.

Compared to previous studies, our system shows superior performance in terms of processing time. For example, the SHA-3 algorithm and DSS implemented here outperformed the Blake2b Hash and ECDA methods used by Nadzifarin and Asmunin [32]. Furthermore, while Indriyawati *et al.* [33] employed a QR code for verification, our method directly integrates DSS, offering immediate detection of document modifications, unlike QR codes, which only reveal changes after accessing the link. Additionally, while Rustemi *et al.* [34] explored blockchain-based systems, our centralized system ensures simpler integration into existing frameworks, though it introduces the potential for a single point of failure.

Although the system performs efficiently, the centralized nature of the architecture could pose a potential limitation. The centralization introduces a single point of failure, which could be a risk in terms of security and system reliability. Future studies could address this by developing a hybrid system combining the benefits of both centralized and decentralized models, increasing redundancy and security. This study’s findings suggest that the system can be expanded to enhance scalability and security. Future research should explore a hybrid model that combines the strengths of both centralized and decentralized systems. Additionally, testing this system at a larger scale across multiple institutions could provide valuable data on its long-term effectiveness. Expanding the system to cover other academic documents, in addition to transcripts, would make it a more comprehensive tool for managing academic credentials.

Recent observations highlight the efficiency and security advantages of the proposed digital signature system. Our findings provide clear evidence that this system significantly enhances document integrity and scalability for academic institutions, reducing processing time while maintaining robust cryptographic security. These benefits are achieved without reliance on traditional paper-based methods, showcasing the system’s potential for streamlining academic records management in the digital era.

4. CONCLUSION

This study addressed the critical issue of safeguarding the integrity of academic transcript data. It proposed an innovative method that integrates the SHA-3 with the DSS scheme. By developing a web-based transcript management module and a digital signature program using SHA-3 and DSS, this research facilitates the secure signing of transcript files, which is essential in today’s digital landscape.

The findings demonstrated that the proposed method effectively prevents unauthorized modifications to transcript files. By digitally signing these files, any subsequent alterations would invalidate the digital signature, thus providing a reliable mechanism for verifying the authenticity of transcript data. Moreover, efficiency testing results reveal an average signature time of just 0.242 seconds, highlighting the practicality and speed of this approach. Such efficiency is crucial for ensuring smooth integration with existing academic information systems and reducing delays in managing academic records.

The implications of this study are significant for educational institutions, as it offers a robust solution to enhance the security and reliability of academic record management systems. By implementing the proposed method, institutions can prevent transcript manipulation or fraud, thereby preserving the integrity of vital educational records and upholding academic standards. This not only strengthens trust in academic qualifications, but it also improves the overall quality of education in the community.

Future research could look into combining this approach with blockchain technology to improve the security and transparency of academic record management. Furthermore, investigating the scalability and performance of this approach in bigger academic information systems or across many institutions could provide useful insights for future deployment. Finally, the study's findings not only contribute to the advancement of data integrity in education, but also lay the groundwork for future innovations that can improve the authenticity of academic records, fostering trust and reliability in educational outcomes for students, institutions, and employers alike.

ACKNOWLEDGEMENTS

On this occasion, the research team would like to thank the Rector of Halu Oleo University for funding the implementation of this research until its publication. We hope that this research will contribute to the advancement of science. Additionally, we trust it will be beneficial to society.

FUNDING INFORMATION

This research was supported by Universitas Halu Oleo. The funding body had no role in the study design, data collection, analysis, interpretation, or manuscript preparation.

AUTHOR CONTRIBUTIONS STATEMENT

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Wa Ode Siti Nur Alam	✓	✓	✓	✓		✓		✓	✓	✓			✓	
Adha Mashur Sajiah	✓	✓	✓	✓	✓				✓	✓	✓			
La Ode Muhammad Bahtiar Aksara						✓		✓					✓	
La Surimi		✓		✓		✓		✓			✓			
Natalis Ransi	✓		✓	✓			✓			✓	✓		✓	
Jumadil Nangi			✓			✓	✓			✓				

C	:	**C**onceptualization	I	:	**I**nvestigation	Vi	:	**Vi**sualization
M	:	**M**ethodology	R	:	**R**esources	Su	:	**Su**pervision
So	:	**So**ftware	D	:	**D**ata Curation	P	:	**P**roject administration
Va	:	**Va**lidation	O	:	Writing - **O**riginal Draft	Fu	:	**Fu**nding acquisition
Fo	:	**Fo**rmal analysis	E	:	Writing - Review & **E**diting			

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflict of interest related to this research. No financial, professional, or personal relationships have influenced the design, execution, or interpretation of this study. Additionally, no external funding bodies have influenced the findings or conclusions presented in this work.

REFERENCES

[1] D. Boneh, "Digital signature standard," in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, pp. 347–347.

[2] J. Andress, "Chapter 5 - Cryptography," J. B. T.-T. B. of I. S. (Second E. Andress, Ed. Boston: Syngress, 2014, pp. 69–88.

[3] L. Sha, "Analysis of an ID-based proxy signature scheme without trusted PKG and a proxy blind multi-signature scheme," in *2014 IEEE/ACIS 15th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2014 - Proceedings*, 2014, pp. 1–4, doi: 10.1109/SNPD.2014.6888700.

[4] W. Stallings, *Network security essentials : applications and standards*, 4th ed. Pearson, 2011.

[5] W. Du, *Internet Security: A Hands-on Approach*, 3rd ed. Wenliang Du, 2022.

[6] V. Trujillo-olaya and J. Velasco-medina, "Hardware implementation of elliptic curve digital signature algorithm over GF(2409) using SHA-3," *International Journal of Machine Learning and Computing*, vol. 12, no. 3, pp. 3–8, 2022, doi: 10.18178/ijmlc.2022.12.3.1082.

[7] National Institute of Standards and Technology, "Digital signature standard (DSS). (Department of Commerce, Washington, D.C.)," *Federal Information Processing Standards Publication (FIPS) NIST FIPS 186-5*, p. 86, 2023.

[8] D. Boneh, "Digital signature standard," *Encyclopedia of Cryptography and Security*, vol. 2, no. July, pp. 347–347, 2011, doi: 10.1007/978-1-4419-5906-5_145.

[9] A. Ismail, V. A. H. F, and A. T. F, "Digital signature system using SHA-3 and ECDSA (in *Indonesian*)," *Unistek*, vol. 10, no. 2, pp. 84–93, 2023.

[10] R. Yap, "Analysis of algorithms magenta with 128 bit key length accompanied by shifting hash securing data," *IOP Conference Series: Materials Science and Engineering*, vol. 725, no. 1, p. 12139, 2020, doi: 10.1088/1757-899X/725/1/012139.

[11] S. Aggarwal and N. Kumar, "Digital signatures," in *The Blockchain Technology for Secure and Smart Applications across Industry Verticals*, vol. 121, S. Aggarwal, N. Kumar, and P. B. T.-A. in C. Raj, Eds. Elsevier, 2021, pp. 95–107.

[12] S. Debnath, A. Chattopadhyay, and S. Dutta, "Brief review on journey of secured hash algorithms," in *2017 4th International Conference on Opto-Electronics and Applied Optics, Optronix 2017*, 2017, vol. 2018-January, pp. 1–5, doi: 10.1109/OPTRONIX.2017.8349971.

[13] R. Dilli and P. C. S. Reddy, "Implementation of security features in MANETs using SHA-3 standard algorithm," in *2016 International Conference on Computation System and Information Technology for Sustainable Solutions, CSITSS 2016*, 2016, pp. 455–458, doi: 10.1109/CSITSS.2016.7779410.

[14] Dale Liu *et al.*, "Chapter 3 - an introduction to cryptography," in *Next Generation SSH2 Implementation*, D. Liu, M. Caceres, T. Robichaux, D. V Forte, E. S. Seagren, D. L. Ganger, B. Smith, W. Jayawickrama, C. Stokes, and J. B. T.-N. G. S. I. Kanclirz, Eds. Burlington: Syngress, 2009, pp. 41–64.

[15] D. Y. W. Liu, G. Z. Xue, Y. Xie, X. P. Luo, and M. H. Au, "Performance of digital signature schemes on mobile devices," in *Mobile Security and Privacy: Advances, Challenges and Future Research Directions*, M. H. Au and K.-K. R. B. T.-M. S. and P. Choo, Eds. Boston: Syngress, 2017, pp. 247–256.

[16] E. Conrad, S. Misenar, and J. Feldman, "Domain 5," in *CISSP Study Guide*, E. Conrad, S. Misenar, and J. B. T.-C. S. G. (Second E. Feldman, Eds. Boston: Syngress, 2012, pp. 213–255.

[17] G. Lee and G. Lee, "Chapter 7 – network virtualization," in *Cloud Networking*, G. B. T.-C. N. Lee, Ed. Boston: Morgan Kaufmann, 2014, pp. 121–137.

[18] T. St Denis and S. Johnson, "Hash functions," in *Cryptography for Developers*, T. St Denis and S. B. T.-C. for D. Johnson, Eds. Burlington: Syngress, 2007, pp. 203–250.

[19] E. Conrad, "Domain 3," in *Eleventh Hour CISSP*, E. B. T.-E. H. C. Conrad, Ed. Boston: Syngress, 2011, pp. 39–54.

[20] H. E. Michail, G. S. Athanasiou, G. Theodoridis, A. Gregoriades, and C. E. Goutis, "Design and implementation of totally-self checking SHA-1 and SHA-256 hash functions' architectures," *Microprocessors and Microsystems*, vol. 45, pp. 227–240, 2016, doi: 10.1016/j.micpro.2016.05.011.

[21] Q. Wang and M. Su, "Integrating blockchain technology into the energy sector - From theory of blockchain to research and application of energy blockchain," *Computer Science Review*, vol. 37, p. 100275, 2020, doi: 10.1016/j.cosrev.2020.100275.

[22] S. K. Black, "CHAPTER 9 - Encryption," in *The Morgan Kaufmann Series in Networking*, S. K. B. T.-T. L. in the I. A. Black, Ed. San Francisco: Morgan Kaufmann, 2002, pp. 327–387.

[23] J. McGovern, S. Tyagi, M. E. Stevens, and S. Mathew, "Security," in *Java Web Services Architecture*, J. McGovern, S. Tyagi, M. E. Stevens, and S. B. T.-J. W. S. A. Mathew, Eds. San Francisco: Elsevier, 2003, pp. 621–688.

[24] Z. Al-Odat, M. Ali, and S. U. Khan, "Mitigation and improving SHA-1 standard using collision detection approach," in *Proceedings - 2018 International Conference on Frontiers of Information Technology, FIT 2018*, 2018, pp. 333–338, doi: 10.1109/FIT.2018.00065.

[25] Z. Al-Odat and S. Khan, "The sponge structure modulation application to overcome the security breaches for the MD5 and SHA-1 hash functions," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, 2019, vol. 1, pp. 811–816, doi: 10.1109/COMPSAC.2019.00119.

[26] Q. H. Dang, "Secure hash standard," *FIBS 180-4 Publication*, vol. 4, no. August, p. 36, 2015, [Online]. Available: http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf%5Cnhttp://thor.info.uaic.ro/~fltiplea/CC/FIPS180-3.pdf%5Cnhttp://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf.

[27] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, "The first collision for full SHA-1," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10401 LNCS, pp. 570–596, doi: 10.1007/978-3-319-63688-7_19.

[28] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "The Keccak reference," *Submission to NIST (Round 3)*, pp. 1–14, 2011.

[29] K. A. Mckay and D. A. Cooper, "Withdrawn NIST Technical Series Publication," no. 2001, pp. 27–28, 2019.

[30] G. Sarath, D. C Jinwala, and S. Patel, "A survey on elliptic curve digital signature algorithm and its variants," in *Computer Science & Information Technology (CS & IT)*, Apr. 2014, pp. 121–136, doi: 10.5121/csit.2014.4411.

[31] Z. Xu, G. Dai, and D. Yang, "Efficient ECDSA-based signature scheme for wireless networks," *Wuhan University Journal of Natural Sciences*, vol. 11, no. 6, pp. 1707–1710, 2006, doi: 10.1007/BF02831856.

[32] A. Nadzifarin and A. Asmunin, "Penerapan Elliptic Curve Digital Signature Algorithm pada Tanda Tangan Digital dengan Studi Kasus Dokumen Surat – Menyurat," *Journal of Informatics and Computer Science (JINACS)*, vol. 4, no. 01, pp. 1–9, Jul. 2022, doi: 10.26740/jinacs.v4n01.p1-9.

[33] H. Indriyawati, T. Winarti, and V. Vydia, "Web-based document certification system with advanced encryption standard digital signature," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 22, no. 1, pp. 516–521, 2021, doi: 10.11591/ijeecs.v22.i1.pp516-521.

[34] A. Rustemi, F. Dalipi, V. Atanasovski, and A. Risteski, "DIAR: a blockchain-based system for generation and verification of academic diplomas," *Discover Applied Sciences*, vol. 6, no. 6, 2024, doi: 10.1007/s42452-024-05984-1.

BIOGRAPHIES OF AUTHORS

Wa Ode Siti Nur Alam is assistant professor at Department of Electrical Engineering, Halu Oleo University, Indonesia. She received the B.Eng. degree in engineering from the Islamic University of Indonesia and the M.Eng. from Gadjah Mada University. She has received several DIKTI-DIKSI scholarships, namely the vocational lecturer short course program at the Southern Taiwan University of Science and Technology, Taiwan, the vocational lecturer retooling program at the Korea University of Technology and Education in the field of solar cell competence, and the vocational lecturer capacity building program at Strathclyde University, Glasgow-UK. Currently, she is secretary of the Information and Communication Technology Center. She has supervised or co-supervised about 50 undergraduate students. She has published more than 20 journals and proceedings. Her research areas are electrical engineering including artificial intelligence, pattern recognition, image/signal processing, and image/signal analysis. She can be contacted at email: wdsitinuralam@uho.ac.id.

Adha Mashur Sajiah received the B.Eng. degree in informatics engineering from Halu Oleo University, Indonesia, and the M.Eng. degree in electrical engineering from Gadjah Mada University, Indonesia. He is currently an assistant professor in the Department of Informatics Engineering at Halu Oleo University, that has been a lecturer since 2018. He supervises undergraduate students and has published 23 journals and proceedings. Additionally, he holds a position as the head of the working group for information system and network security at Information Technology Center Halu Oleo University. He has also obtained several simple patents. His research interests include artificial intelligence, software engineering, soft computing, machine learning, and intelligent systems. He can be contacted at email: adha.m.sajiah@uho.ac.id.

La Ode Muhammad Bahtiar Aksara is a lecturer in the Department of Informatics Engineering, Faculty of Engineering at Halu Oleo University. Completed a bachelor’s degree in informatics at Gunadarma University. Then completed a master’s degree in informatics at the Bandung Institute of Technology. His research area is computer networks, informatics and security. And now he is serving as head of computer network and internet development at the information technology center at Halu Oleo University. He can be contacted at email: bahtiar.aksara@uho.ac.id.

La Surimi received a B.Sc. degree in mathematics from Hasanuddin University, Indonesia, and an M.Cs. degree in computer science from Gadjah Mada University, Indonesia. Currently, he is working as a lecturer in the computer science program at Halu Oleo University, Kendari. He is also currently holding the position of Head of the Information System Development Unit at Halu Oleo University. He has supervised and co-supervised more than 20 undergraduate students. He has authored or co-authored more than 10 publications, including 5 journals. His research interests include cryptography and computer networks. He can be contacted via email: lasurimi@uho.ac.id.

Natalis Ransi received a B.Sc. degree in mathematics from Halu Oleo University, Indonesia, and an M.Cs. degree in computer science from Gadjah Mada University, Indonesia. Currently, he is working as a lecturer in the computer science program at Halu Oleo University, Kendari. He has supervised and co-supervised more than 30 undergraduate students. He has authored or co-authored more than 10 publications, including 5 journals. His research interests include data mining, information system, and data base management system. He can be contacted at email: natalis.ransi@uho.ac.id.

Jumadil Nangi is a lecturer in the Department of Informatics Engineering, Faculty of Engineering at Halu Oleo University. He completed his bachelor’s degree in informatics engineering at STMIK Handayani and his master’s degree in informatics at Hasanuddin University. His research areas include information systems, information technology, and data mining. He can be contacted at email: jumadilnangi@uho.ac.id.