IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Hardware Secured E-Data System

[1]Vaishnavi J, [2]Adarsh R, [3]Bhuvana G, [4]Niranjan H R ,[4]Chaithanya S

[1]Student, [2]Student, [3]Student, [4]Student, [5]Assistant Professor

Electronics and Communication Engineering,

RajaRajeswari College of Engineering, Bangalore, India

Abstract: With the growing reliance on Electronic Data Records there is an increasing demand for safeguarding the data from cyber threats. So, we propose a Hardware Secured E-Data System with dual securing strategy. We employ biometric authentication technique along with password which are converted into hash codes. After hashing a random fusion of these biometric minutiae which enables to achieve high irreversibility and increased non-reconstruction capability of bio-hash key. The system employs AES encryption for protecting data during write operations and decryption during viewing, optimizing performance while maintaining data integrity. A direct access to files, information for self-review and user experience is provided.

Index Terms - Electronic Data Records, Authentication, Hash codes, biometric minutiae, AES encryption.

I. INTRODUCTION

Biometric authentication utilizes one or more biological traits to grant access to secured digital records or databases. Since each biometric pattern is distinct, it is nearly impossible to replicate or tamper with. The process of extracting biometric features can be divided into three categories: biological, behavioural, and morphological. Biological features, such as DNA and blood samples, are used for access control purposes. Behavioural biometrics rely on characteristics like voice and heartbeat to regulate access. Morphological traits, including fingerprints, facial features, irises, ears, and palms, offer an additional layer of security.

Among these morphological traits, fingerprints are widely used in generating secret keys. A key derived from fingerprint authentication provides extra protection against potential attacks. Securing electronically stored records—such as reports, images, and documents—is essential to prevent unauthorized access. Therefore, it is crucial to implement robust security measures to safeguard this information. Biometric applications have proven effective in areas such as forensic analysis, preventing electoral fraud, securing e-health records, managing revenue records, and also protecting banking transactions.

In addition to biometric systems, hardware-secured technologies employ encryption mechanisms to protect data, both when stored and during transmission. Hardware-based encryption significantly enhances security while maintaining operational efficiency, enabling seamless real-time data protection. Access control mechanisms are integral to protecting sensitive data, utilizing biometric authentication, token systems, and multi-factor authentication to restrict access to authorized users only. Furthermore, secure boot procedures verify the integrity of software and firmware at startup, blocking the execution of unauthorized or harmful code.

Section II of this paper gives an overview of the currently existing record systems and further proceeds to section III which discusses the proposed model. Section IV is the hardware implemented and Section V concludes the system level performances of the proposed model.

II. PREVAILING E-RECORD SYSTEMS

Data stored in computers and personal laptops were majorly prone to theft due to insufficient security measures. This became a major concern of the health industry as hospitals contained huge number of details related to patients. It consisted of reports, images, and other documents which if leaked could invade privacy of thousands of patients.

In order to secure those information EU and USA came up with PHR system. PHR stands for Personal Health Record and allowed the users to keep track of their health records which was stored in cloud. It used various policy-based schemes like attribute-based encryption system, blockchain technology to strengthen the data. It succeeded in providing high security but due to high cost it became unaffordable for public access.

Electronic Health Record (HER/EHR) was designed and developed exclusively for hospitals and health care centres. This system kept the patients informed about their regular check-ups with the respective physicians. Apart from storing the e-data this system also provided access to both patient and physician to review the reports through digital signature validation thus providing an additional layer of security. The blockchain technology used in HER facilitated the accurate database management. As both the private and public keys were being stored in the same server, manipulation and misuse of data by the adversary led to potential attack.

EMR i.e. Electronic Medical record was the improvised version of the HER which was widely recognised and implemented by countries like USA and EU, due its effective monitoring and constant assessment of patients, response to treatments. EMR systems were protected by passwords, which could be accessed by anyone within health department which made it less secure and more vulnerable to various attacks. Later the privacy was enhanced through the use of ECC and DSA algorithms.

Bio-Hash secured e-data records used the Hashed Minutiae Random Fusion (HMRF) logic which focused on random fusion of the minutiae obtained from the fingerprint images of both the patient as well as the medical practitioner. This model includes three different types of access to the records which are Write mode, View mode and Read-only mode. The patient can view reports only in read-only mode, whereas the medical practitioner can select the mode from mode selector. If write mode is selected then new documents or the already existing ones can be modified, and if view mode is selected only the patient's history can be viewed.

Later the SHA-3 hashing algorithm is applied which is pseudo-cascaded, followed by AES encryption algorithm which encrypts the stored data and gives access when correct bio-hash key is generated. But it was not suitable for hand-held applications and not portable which became a setback.

Thus, it is necessary to store and secure the electronic data of all kinds in every applicable field. This requires the implementation of robust encryption methods, secure access controls, and continuous monitoring of digital systems. It includes providing high security at affordable prices to people belonging to all communities from manipulation and misuse of data by the attackers.

III. PROPOSED E-RECORD SYSTEM

The initial step in securing the stored electronic data is obtaining the fingerprint minutiae of individuals. In this model we have set the number as two individuals who are working at same level and both the individuals are required to access the stored data. If there is no involvement of the second person then one can register two fingerprints. Once the minutiae are extracted, they are sent to SHA block to get hashed sequences, these in turn are fed to the Skimmer block which reduces the number of bits to required bit size. After skimming these bits are fused together to produce a bio-hash key.

As this is a two-factor authentication system it involves both biometric authentication and password authentication to provide high security. After correct bio-hash key generation, the system allows for password entry. The password entered also undergoes the same process to produce a pass-code key. The two keys obtained are fused and a final fused bio-pass key is produced. This key is used in AES encryption to grant access to the stored data as shown in fig 1.

A. Fingerprint Enrolment and Authentication

During the enrolment or registration phase, each individual has to place their fingers on the fingerprint sensor to register their respective biometric trait into the database. The fingerprint sensor captures these unique ridge and groove pattern from the individual's finger and stores these in the form of biometric sample templates. The templates are stored in the database if there is no matching template already registered in the database.

The authentication phase involves verifying the individual's fingerprint against those registered in the database. When there is a match between the entered and registered it indicates that the authenticated person is accessing the data, and if there is no match found it indicates that an unauthorized entity is trying to gain access to view the data.

Once the fingerprints are registered and stored as templates in the database, these are converted into bits. These converted bits are sent as input to SHA-512 block.

B. SHA-512 Hashing Algorithm

Hashing is one-way function, i.e. they take certain data as input and produce a fixed length output. This output, called a hash value, is unique for each input, making it useful for data integrity verification and password storage. The fixed length of output takes the bit size depending on the algorithm used. There are different hashing algorithms like MD5, SHA-0, SHA-1, SHA-2, SHA-3. Each SHA family has various input sizes – 224, 256, 384, 512.

We employed SHA-512 which belongs to SHA-2 family as it can operate on more operations when compared to the SHA-3 family. The reason why hashing is preferred is due to its high collision resistance feature. It means that there are no two distinct inputs which can produce same output. Thus, it provides high irreversibility and increases non-reconstructive property. This hashing function is applied on the data obtained in the form of bits from biometric templates. SHA-512 occurs in four stages: input formatting, hash buffer initialization, message processing, final output.

Input formatting – SHA-512 cannot hash an infinite sequence of bits, it also has a limit of 2^128-1 bits. This step prepares the message for further processing. The message format consists of original message bits, padding bits, size of original message. All combined must have size of 1024 bits or its multiple. Hash buffer initialization – in order to process the first block of 1024 bits we make use of these initialization vectors (IV). IVs are derived from hash buffers which consists of eight sub-registers. They hold the values by taking first 64 bits of fractional parts of square root of initial 8 prime numbers.

Message Processing – the above formatted input is divided into N-blocks of 1024 bits. Each block is then sent to Rounds and Addition block. Rounds takes one-word, previous round output, and a SHA-512 constant. First round does not have previous stage output so it uses the block as its input. These constants are obtained from first 64 bits of fractional parts of cube root of initial 80 prime numbers, as there are 80 rounds in total. The final addition of all rounds outputs produces the Hash bits. Final Output – each block after processing produces a hash, which serves as input to next block processing. Once the last block is processed it produces the final hashed output which is of 512 bits.

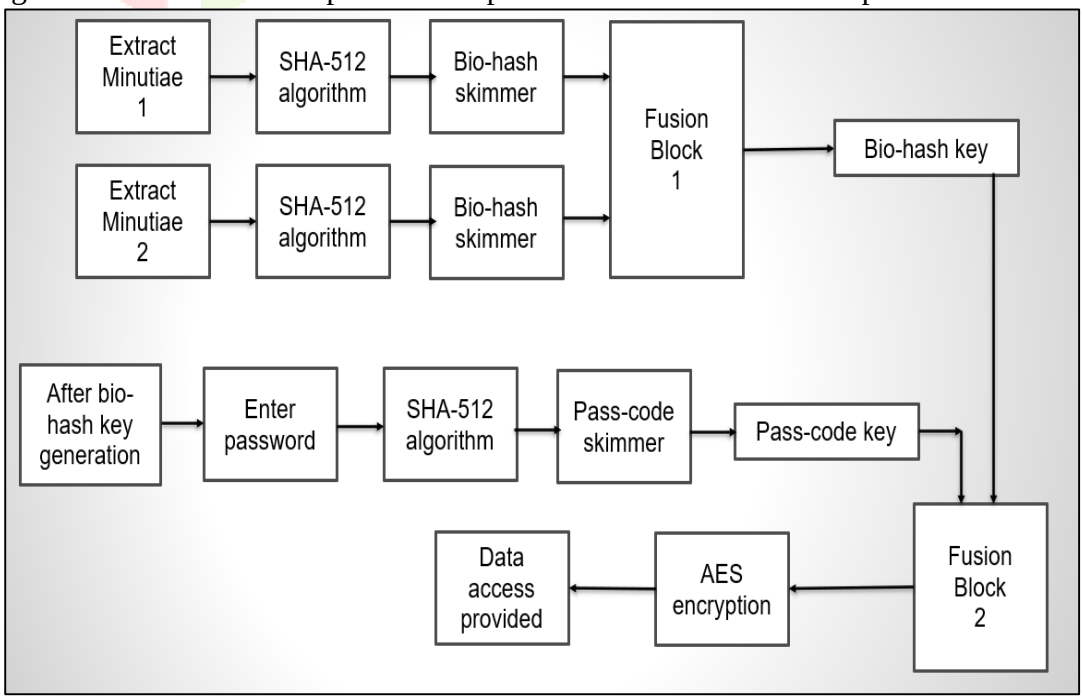

Fig 1. Block Diagram of Key generation for Data Access

C. Skimmer Block

The skimming is necessary to reduce the number of bits to a required size. We skim the hashed bits which are provided as input to this block. First this block checks whether the hashed input is of 512 bits or not. Only if it is true then the hashed input is split into chunks as shown in fig 2. Each chunk is of the length 16 bytes (128 bits). So, the 512 bits hash is divided into four such chunks. We perform XOR operation between the chunks by :

Chunk 1 ^ Chunk 2 = R1
R1 ^ Chunk 3 = R2 (Or) Chunk 1 ^ Chunk 2 ^ Chunk 3 ^ Chunk 4 = Skimmed H
R2 ^ Chunk 4 = Skimmed Hash

Thus, obtained value is called Skimmed Hash which is of 128 bits.

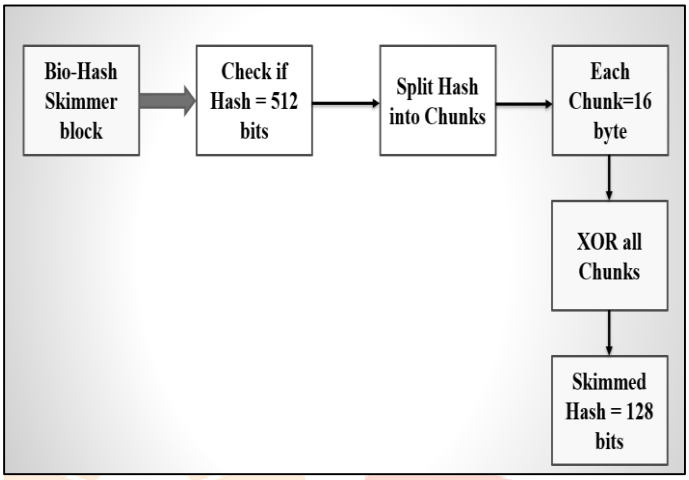

Fig 2. Skimming of Hashed input

D. Fusion Blocks and Password Entry

The skimmed hash of both the individuals is calculated. They are fused to form the Generated Bio-hash key in the Fusion block 1. The fusion step includes XOR operation of skimmed hash 1 and skimmed hash 2. The Original Bio-hash key is already stored in the source code. This generated key as well as the original key has to match in order to proceed. Even if a single bit differs it generates an error message and denies access to further authentication steps.

Once the correct Bio-hash key is generated this allows for password entry. The entered password must be of minimum 8 characters, must contain 1 uppercase and 1 lowercase alphabet, must contain 1 numeric value, must contain a special character. If the password consisting of all the above attributes is given only then it is considered valid. The password entered again passes through SHA-512 block, Skimmer block to produce a generated Pass-code key. The code again verifies this generated key with that of the original key previously stored in source code as shown in fig 3. Only if right match is found it moves onto fusion block 2. The second fusion block is used for fusing the two keys generated i.e. Bio-hash key and Pass-code key. This results in final fused Bio-pass key which is of the length 128 bits. This acts as key for AES encryption process.

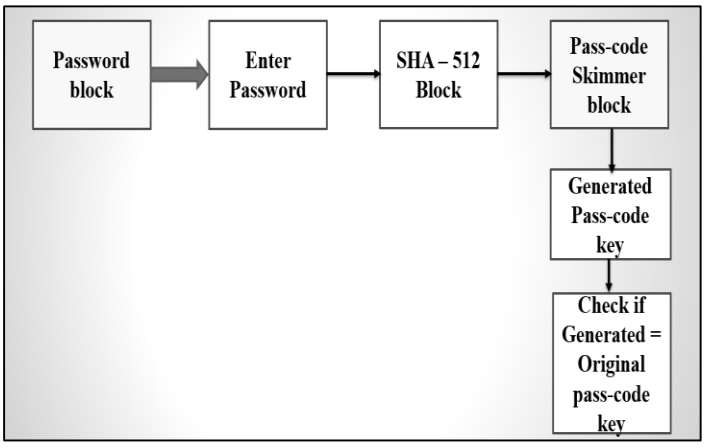

Fig 3. Password Entry block

E. AES-128 Encryption Standard

The encryption of data provides high security against unauthorized access and is vital for securing electronic data. Encryption converts plain text to cipher text with the help of a key. This key remains a secret between the sender and the receiver to preserve confidentiality. DES, AES, and RSA are basic encryption types available. Advanced Encryption Standard (AES) is known to be six times faster than the DES encryption standard. Depending on the key sizes they can operate with, they are named as AES-128, AES-192, and AES-256.

We employ AES-128 encryption though there are two more advanced key sizes utilized as it is much faster, occupies less area and sufficient for small-scale applications. It works on the principle of Substitution-Permutation network. AES prefers to perform calculations on bytes rather than bits and is a better replacement to Triple-DES which was prone to exhaustive key search attack and slowed down on increased computing power.

The AES algorithm creates a matrix of size 4x4, and fills each cell with 1 byte of data (128-bit plain text), in total 16 bytes. The other input to this encryption is 128-bit fused bio-pass key, which is also written in the form of matrix 4x4 with each cell containing 1 byte of data i.e. in total 16 bytes. The number of rounds vary depending on the key size i.e. if key=128 bits, rounds=10; key=192 bits, rounds=12; and key=256 bits, rounds=14. There are three stages at which processing occurs: key expansion, pre-round transformation, and rounds operation. Before these are computed S-box is to be created. It is formed by assigning unique bytes for different combinations of two hexadecimal values in a table (x varies from 0 to F, and y varies from 0 to F) as shown in fig 4.

Key Expansion – it takes up the original key, and starts expanding into multiple keys such that each round is provided with a unique key. As it is a 128-bit size key it has 10 rounds and always produces (rounds +1) keys, here it produces a total of 11 keys. The last column of key matrix is taken and the first byte is shifted downwards. And each byte is substituted using its corresponding byte of data from the S-box. After substitution the last column is used for computation:

Step 1: First column of original key ^ substituted last column ^ Rcon(1)
and the so obtained column is placed as the first column of Round key 1. The round constant is shown in the fig 5.

Step 2: Second column of original key ^ first column of Round key 1
and its result is placed as Second column of Round key 1.

Step 3: Third column of original key ^ first column of Round key 1
and its result is placed as Third column of Round key 1.

Step 4: Last column of original key ^ first column of Round key 1
and its result is placed as last column of Round key 1.

This completes Round 1 key which gets used for generating Round 2 key. The process continues till each round has a unique key.

Pre-Round Transformation – at the start, the original data matrix is just XORed with original key matrix and the resultant matrix is input to the Round 1.

Rounds Operation – it takes previous round output as input and performs the following operations:

1.Sub Bytes:

Input matrix is substituted with its corresponding bytes from the S-box.

2.Shift Rows:

The sub bytes output matrix is taken and first row remains unaltered, the second row is rotated over by 1 byte (first byte is moved to last cell), the third row is rotated over by 2 bytes (first and second bytes are moved to last 2 cells), the last row is rotated over by 3 bytes (first, second and third bytes are moved to the last 3 cells).

3.Mix Columns:

The shift rows output matrix is taken as input and each of its column is multiplied with Galois Field matrix separately to produce this stage output matrix. The Galois Field matrix is shown in fig 6. The last round operation excludes this mix columns operation.

4.Add Round Key:

The mix columns output matrix is XORed with the Round 1 key obtained from key expansion stage.

The resultant matrix is input to the next round and this process continues till the final cipher text is produced.

Once the generated final cipher text is produced it is matched with that of the original cipher text. If a match is found then the file access is provided or else it is denied.

hex	y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

x

Fig 4. Creation of S-box

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

Fig 5. Round constant matrix

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

Fig 6. Galois Field matrix

IV. HARDWARE IMPLEMENTATION

The proposed hardware secured e-data system involves two factor authentication which primarily focuses on securing data stored in PCs and personal laptops. The AES-128 encryption is implemented on the DECA Board with Altera Max 10 FPGA board. Initially to configure Max 10 FPGA on DECA we use JTAG configuration.

We need to install Quartus Prime 22.1 std Lite edition software to program the DECA board. We write the Verilog code for AES-128 encryption by creating a top-level module. After assigning the inputs and outputs we create sub-modules for S-box, Key expansion, AES round operations and provide an AES workflow including Sub Bytes module, Shift Rows module, Mix Columns module, and Add Round Key module.

After compiling the design, we need to select the pin planner and assign the required number of LEDs with pins as shown in fig 7. We have assigned LED[0] to indicate flag 1, LED[1] to indicate flag 2, LED[2] to indicate flag 3, LED[3] to indicate whether input key is taken or not, LED[4] to indicate whether plain text is taken or not. Flag 1 turns ON if correct cipher text is generated. Flag 2 turns ON if correct key is generated. Flag 3 turns ON if correct plain text is generated. If all the five LEDs turn ON then AES-128 encryption is successful as shown in fig 8, and if only two LEDs turn ON then it is unsuccessful.

A website login page is created for providing only authorised access. In fig 9, once the generated IP address is entered it gets directed towards this login page. Here the authorised individuals are required to scan and register their fingerprints. These are stored in the database. If a third party tries to enrol their biometric trait it pops a message as authentication failed. Fig 10 shows the password entry page which opens only if authentication is successful. Fig 11 shows the different types of folders that are stored, and upon selecting any one of the folders it is directed towards drive where files stored under that particular folder can be viewed as in fig 12.

Signal Name	FPGA Pin No.	Description	I/O Standard
LED[0]	PIN_C7	LED [0]	1.2V
LED[1]	PIN_C8	LED [1]	1.2V
LED[2]	PIN_A6	LED [2]	1.2V
LED[3]	PIN_B7	LED [3]	1.2V
LED[4]	PIN_C4	LED [4]	1.2V
LED[5]	PIN_A5	LED [5]	1.2V
LED[6]	PIN_B4	LED [6]	1.2V
LED[7]	PIN_C5	LED [7]	1.2V

Fig 7. Pin Assignment of LEDs

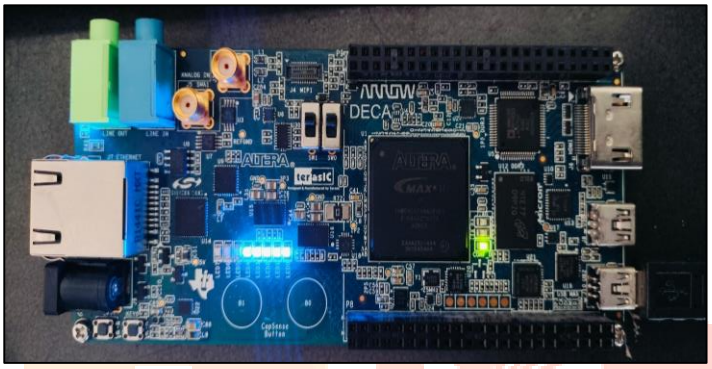

Fig 8. AES Encryption is successful.

Fig 9. Login page for Fingerprint Authentication.

Fig 10. Password Entry.

Fig 11. Selection of Folders to be accessed.

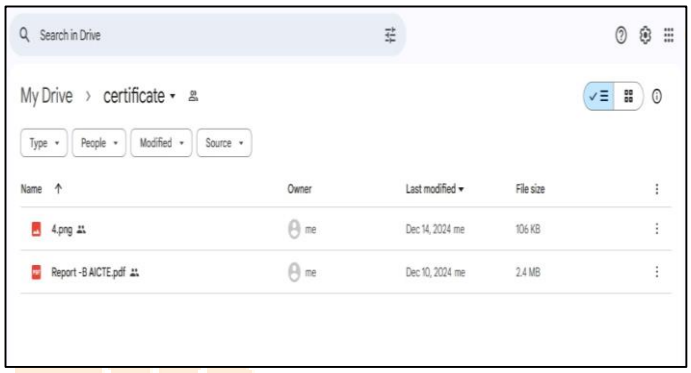

Fig 12. File access granted.

V. CONCLUSION

The proposed Hardware Secured E-Data System leverages dual-layer security through biometric authentication and hashed password fusion, ensuring robust protection against cyber threats. By combining advanced bio-hashing techniques with AES encryption, the system achieves enhanced irreversibility, non-reconstruction capability, and data integrity during storage and access. Furthermore, its design prioritizes user convenience for hand-held applications, offering direct access to files for self-review while maintaining stringent security standards. The implementation of AES encryption in Altera Max 10 FPGA board verifies whether encryption is taking place properly. This innovative approach addresses the increasing need for secure and efficient management of electronic data records, providing a reliable solution for safeguarding sensitive information in a digital world.

REFERENCES

[1] M. M. Sravani and S. Ananiah Durai, “Bio-Hash Secured Hardware e-Health Record System,” in EEE Trans Biomed Circuits Syst. 2023 Jun;17(3):420-432. E-pub 2023 Jul 12.

[2] Alok Tripati, Rajiv Pandey and Amarjeet Singh, “Simulating Tardos finger printing codes under randomized bits collusion attacks,” in 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)

[3] Dr. N. Venkatesan, M. Rathan Kumar, “Finger Print Authentication For Improved Cloud Security,” in 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)

[4] Hassan. M. Elkamchouchi, Abdel-Aty M. Einarah, Esam A. A. Hagras, “A New Secure Hash Dynamic Structure Algorithm,” in 23'" National Radio Science Conference (NRSC 2006)

[5] S. Ribaric and N. Pavesic, “A Finger-based Identification System,” in MELECON 2008 - The 14th IEEE Mediterranean Electrotechnical Conference

[6] Dong, X. Meng, M. Chen, and Z. Wang, “Template protection based on DNA coding for multimodal biometric recognition,” in Proc. IEEE 4th Int. Conf. Syst. Inform., 2017, pp. 1738–1742.

[7] X. Zhang, D. Cheng, P. Jia, Y. Dai, and X. Xu, “An efficient android-based multimodal biometric authentication system with face and voice,” IEEE Access, vol. 8, pp. 102757–102772, 2020.

[8] V. Talreja, M. C. Valenti, and N. M. Nasrabadi, “Deep hashing for secure multimodal biometrics,” IEEE Trans. Inf. Forensics Secure., vol. 16, pp. 1306–1321, 2021.

[9] R. Dwivedi et al., “A fingerprint based crypto-biometric system for secure communication,” J. Ambient Intell. Humanized Comput., vol. 11, no. 4, pp. 1495–1509, 2020.

[10] N. Ansari, P. Sakarindr, E. Haghani, C. Zhang, A. K. Jain, and Y. Q. Shi, "Evaluating electronic voting systems equipped with voter-verified paper records," IEEE Secur. Privacy, vol. 6, no. 3, pp. 30–39, May/Jun. 2008.

[11] J. Galbally, R. Haraksim, and L. Beslay, "A study of age and ageing in fingerprint biometrics," IEEE Trans. Inf. Forensics Secur., vol. 14, no. 5, pp. 1351–1365, May 2019.

[12] F. J. Zareen and S. Jabin, "Authentic mobile-biometric signature verification system," Inst. Eng. Technol. Biometrics, vol. 5, no. 1, pp. 13–19, 2016.

[13] C.-L. Lei and Y.-H. Chuang, "Privacy protection for telecare medicine information systems with multiple servers using a biometric-based authenticated key agreement scheme," IEEE Access, vol. 7, pp. 186480–186490, 2019.