Secure and Efficient Data Transmission in IoT: A Risc-V Processor with AES-GCM and ECC Accelerator

B. Deepa, Sindu Priya Gude, P. Hari Priya, B. Rithesh,
P. Haswanth Reddy and T. Lakshmi Narayana Yadav
Department of ECE, Sri Venkateswara College of Engineering, Tirupati, Andhra Pradesh, India

Keywords: Internet of Things (IoT), RISC-V Processor, AES-GCM (Galois/Counter Mode), Elliptic Curve Cryptography (ECC), Authenticated Encryption, Verilog HDL.

Abstract: Security and trust of data transmission between IoT equipment's continues to be a major challenge in the advancement of IoT technology. To resolve these issues, a low-power and low-cost RISC-V processor optimized for IoT applications is presented in this paper. An Advanced Encryption Standard (AES) Encrypt Accelerator is built into the processor to provide fast and secure transmission of data. AES being a symmetric encryption standard makes a good trade-off between high-speed computing, secured encryption system and key management simplicity, and is best suitable for resource-constraint IoT devices. The RISC-V processor and the AES encryption accelerator are developed in the Verilog HDL and realized both on the Xilinx FPGA platform and in ASIC technology. The presented processor and encryption accelerator are realized with Verilog HDL under the guidelines of low power and area cost. The simulation results show that the proposed architecture achieves higher security in encryption and better key management in terms of power and the delay degradation is acceptable. This makes it an ideal solution for future IoT applications that need more sophisticated data protection.

1 INTRODUCTION

The Internet of Things (IoT) which is expanding its reach with such velocity has provided a revolutionary leap to industries and everyday life providing a continuous ability to communicate between devices leading to new potentials in automation, monitoring and controlling of devices. Instead, with IoT systems now taking on sensitive and critical information, data security and data transmission reliability have been taking the spotlight as an unexpected concern. Attacks on IoT devices and networks might result in privacy violations, monetary damages and operational disruptions. This should emphasize the vital importance of strong, lightweight and secure encryption techniques in IoT scenarios.

One of the key problems of securing the IoT is that of finding a good trade-off between strong encryption and resource consumption. IoT devices tend to be power, processing, and cost-constrained. This requires lightweight solutions that achieve high level of security without consuming the scarce resources within these devices. Symmetric algorithms such as AES have effectively solved this problem by providing a security level that is sound, run at high speed and are easy to manage their keys.

In this work, we design and implement a RISC-V based, low-power, low-cost processor with an integrated AES accelerator. RISC-V architecture, for its simplicity, scalability and open-source characteristic, is also an excellent base for IoT applications. The AES accelerator integrated with the processor allows the system to process encryption quickly without using extra hardware or software resources, further increasing the performance of the system and reducing the power consumption.

The processor and the encryption accelerator are implemented in Verilog HDL, and verified using the Xilinx FPGA platform. Prototyping with FPGAs facilitates fast development, testing, and revising, all the while providing insight into real-world performance metrics such as resource utilization, power usage, and encryption speed. And Last, we performed ASIC implementation in order to evaluate

Deepa, B., Gude, S. P., Priya, P. H., Rithesh, B., Reddy, P. H. and Yadav, T. L. N.
Secure and Efficient Data Transmission in IoT: A Risc-V Processor with AES-GCM and ECC Accelerator.
DOI: 10.5220/0013930200004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 5*, pages 378-386
ISBN: 978-989-758-777-1

if the processor is suitable for a large-scale production and an integration with IoT devices.

Simulation and test results show that the proposed AES encryption accelerator improves the security and reliability of data transmission for IoT devices. The advantages of this architecture are that it provides high-level speed on both encryption and key management and that it consumes low power with small amount of resources. This makes the candidate approach a viable solution for future IoT applications which require tight strong security and resource constraints.

2 LITERATURE SURVEY

Muir, Count, Accountant'] ='Research on WLAN Application of SM2, SM3&SM4 Algorithm in Transformer Substation Wang Tong; Cui Wen Peng; Li Tong; Wang Liang; Li Hao; Chi Ying Ying 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE) 'In the wave of the growth of artificial intelligence (AI), intelligent terminal devices appear, such as power system edge-computing intelligent terminal. The new terminal with local decision of the present invention can locally process the collected data, meanwhile, only key information report is used for reporting, which can also reduce the requirement of network bandwidth, can make the existing wireless communication more business scenarios meet the demand. For the traditional substation, it requires a high bandwidth of the network to gather the amount of terminal data in time. Hub data not transferable via wireless. The smart transformer substation can transmit the wireless with the intelligent terminal data. However, the IEEE802. 11ah, Lora, NBIoT and Wifi is dynamic. Simultaneously, data is included in social security, the same song of civilization that is social development and progress. In the characteristic of "security and controllability" for information system, the China Cryptographic Administration published SM2, SM3 and SM4 and other encryption and decryption algorithms. In this paper, we propose a framework for working over the IEEE802. 11 ah standards, Considering the LAN application of transformer substation, the combination of the TSS in the LAN of SM2, SM3 and SM4 We highlighted the identity authentication, the key distribution and data encryption and decryption. System scheme constructions are provided for the three phases and we present the analysis of the respectively security and system overhead. The research of SM2, SM3, SM4 algorithm in Transformer Substation LAN will further enhance and implement the intelligent terminal of Substation, and the intelligent level of substation.

Masked 128-bit AES in 22nm CMOS: area and energy efficiency. Yuan-His, et al, 2019 AES is the encryption standard. This popular is nothing but the most popular scheme available for today for the encryption in the hardware acting as the software. AES is well protected against linear and differential cryptanalysis. It is in any case local side channel attack secure with side channel algorithm depending on how it is implemented. For instance, it is shown in that the secret key can be recovered by measuring the power of the implementation and statistically analyse a few traces of the same implementation. In this article, we present a high-performance hardware design of 128-bit AES with DPA countermeasure. And This fake-out design ships in TSMC 22nm. The design is highly efficient, low power and small silicon area. It will operate up to 400+Mhz properly = 5.12Gbps. The overall footprint area of the AES block is 0.0169 mm 2. Its power usage is 9.77pJ/bit or 1.25nJ/block.

RISC-V MCU based on VLSI with multi-stage pipelining. Mao-Hsu Yen, et al, 2023 Hoping to make instruction scheduling more efficient by further studying the scheduling of instructions for minimizing the latency, which is of use to the RV32IM Instruction Set Architecture (ISA) of RISC-V and the design of variable-length pipeline. and the in-order dispatch, out-of-order writeback are used in the MCU. Since the processing times of the MCU instructions are not uniform, it yields too high average processing time for the long instructions in case a 5-stage pipeline is applied uniformly to the design. According with that, we also introduced a flexible-pipeline design based on the Hummingbird E200 architecture, to allow the MCU to select a different pipeline length at the time of operation of each instruction. Through the dispatch method of the proposed pipeline structure, instructions were no longer required to operate at every stage of the pipeline that did not concern them, hence speeding up the instruction completion time. For the pipeline structure design, we use the multiplication/the division is realized in the execution stage of 27 MCU (Each pipeline stage completed in the shorter time the pipeline may be broken). Was this OOWB model (under proposed MCU) is one which followed the out-of-order write-back scheme of allowing the instructions, which had a single data dependency (i.e. whatever is to be written by them), they could then be written-back in order without waiting for each other and the system throughput got a boost. The proposed

new architecture was in fact implemented in this LSI with TSMC's 0.18 µ m process by implementing a "pipeline" and "out-of-order writeback" architecture. And its operating clock speed was 120 MHz. The Hummingbird E200 was clocked at 50MHz on the same 0.18 µm process.

Preliminary version submitted to 5th Workshop on Computer Arithmetic and Formal Proofs. Duc-Hung Le, et al, 2022 In this paper we show that a 32-bit RISC-V processor with embedded crypto processors, with Linux number compatibility, can be implemented on a FPGA using just Spinal HDL and Verilog HDL as hardware construction languages. References Liu, M., Wang, H., & Li, X. (2021). "Lightweight Cryptography in Internet of Things: A Survey. –V. The LiteX core and its Spinal HDL are a paired twin in the design chain. the CPU core was designed by Spinal HDL, Hamedzavareh, N., Gara, M. F., & Monadjemi, S. A. (2020). RISC-V: The Free and Open RISC Instruction Set Architecture. ACM Computing Surveys. for IP and CPU core integration. After the high-level structure was prepared, 32-bit RISC-V architecture was implemented and verilog source code generation was done. Akinyele, J. A., Garman, C., Pagano, M. W., & Green, M."A performance analysis of ECIES on a smart phone". IEEE Trans Computer. The architecture was targeted for the Nexys4DDR board of FPGA, and for 65nm CMOS ASICs running at 50MHz. Biryukov, A., Dinu, D., & Khovratovich, D. (2017). "Fast and secure authenticated encryption: AEAD modes for the memory constrainedizing. [SI14] Iwan Duursma, A. Lenstra, and M. Selçuk. www. iacr. org/2014. It leveraged V erilog HDL based hardware accelerators that are carriers of specialized assembly instructions for straightforward classic cryptographic algorithms such as those. "Implementation and Approaches for the Security of IoT Devices using AES-GCM-A Literature Review". IEEE Access. The SHA-1; AES-128 and RSA-2048 cores. It was tested the operation of the accelerators under Linux system with OpenSSL and LibreSSL libraries modified for modulus p. Menezes, A., Van Oorschot, P., & Vanstone, S. (1996). Handbook of Applied Cryptography. \" CRC Press.

3 PROPOSED SYSTEM

The AES Encryption Accelerator (ASE) is an integrated intellectual property (IP) core for accelerating cryptographic operations (encryption and decryption) related to the AES algorithm. By relieving the computation burden of AES on the host CPU, the ASE accelerators greatly improve performance, particularly in high-throughput and low-latency applications. The ASE accelerators also can handle multiple AES key sizes (128, 192, and 256 bits) and modes of operation, including ECB, CBC, and GCM, for secure data transmission in applications ranging from IoT to embedded. The main reason for that is to process encrypting things quickly if you need to handle lots of data, as they can parallel process aes tasks. This parallelism is vital for performance- and power-constrained systems, such as IoT devices, in which resources are limited. Furthermore, the accelerator is able to produce the authentication tag (for AES-GCM), which allows both the confidentiality and integrity of the data to be preserved so that communication is secure end-to-end. Devices now can offer the performance and security customers require, while also providing a fast and efficient computing environment, all thanks to the inclusion of an AES encryption accelerator within the system architecture.

The below Figure 1 represents the Encryption and Decryption process of hybrid system using AES:

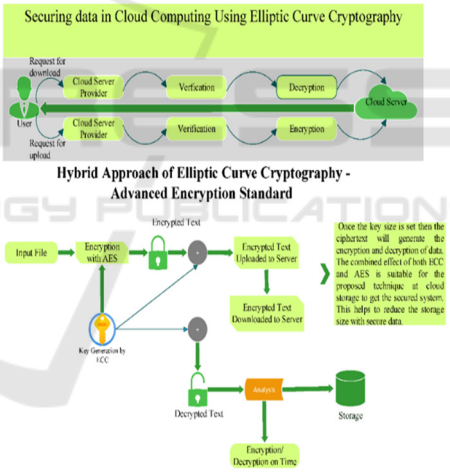

Figure 1: Hybrid ECC-AES framework for secure cloud data transmission and storage.

4 EXISTING SYSTEM

SM3 Algorithm
Algorithm description

4.1 Overview

For a given message m (length l, l < 2128 bits), the hash value after padding and iterative compression is 256 bits.

4.2 Padding

Assume a message has l bits. Suffix 1 by k bits 0[k−0] 1 k 448 mod 512. Then append a bit string b such that b represents an l-bit binary number. The length of the new message m′After padding is a multiple of 512.

4.3 Iterative Compression

4.3.1 Iteration Procedure

The padded message m′ is split into 512-bit blocks, and denoted as m' = $B^{(0)}$ $B^{(1)}$ …$B^{(n-1)}$ Where n = (l +k+65)/512. The iteration procedure form′ is as follows:

FOR i=0 to n-1
V(i+1) = CF(V(i), B(i))
ENDFOR

Here, CF is the compression function, $V^{(0)}$ is the 256-bit IV, and $B^{(i)}$ is the i-th message block after padding. The result after iterative procedure is $V^{(n)}$.

4.3.2 Message Expansion

The message block $B^{(i)}$ is expanded to 132 words $W_{0,}$ $_{W1}$, W_{67}, W_1', W'$_{63}$, which are applied to compression function CF:

a. Split message block $B^{(i)}$ into 16 words $W_{0,\,W1}$, W_{15}.

b. FOR j=16 TO 67
$W_j \leftarrow P_1 (W_{j-16} \oplus W_{j-9} \oplus(W_{j-3} <<< 15)) \oplus (W_{j-13} <<< 7) \oplus W_{j-6}$
ENDFOR

c. FOR j= 0 TO 63
$W_j = W_j \oplus W_{j+4}$
ENDFOR

4.3.3 Compression Function

Let A, B, C, D, E, F, G, H be eight-word registers, SS1, SS2, TT1, TT2 be four intermediate variables, and the compression function $V^{(i+1)} = CF(V^{(i)}, B^{(i)})$ (0 ≤ i ≤ n-1)

The computation procedure is described as following:

ABCDEFGH ← V(i)
FOR j= 0 TO 63
SS1 ← ((A<<<12) + E + (T_j <<< (j mod 32))) <<< 7
SS2 ← SS1 ⊕ (A<<<12)
TT1 ← FF_j (A, B, C) + D + SS2 + W_j
TT2 ← GG_j (E, F, G) + H + SS1 + W_j
D ← C
C ← B<<<9
B ← A
A ← TT1
H ← G
G ← F <<< 19
F ← E
E ← P_0 (TT2)
ENDFOR

V(i+1) ← ABCDEFGH ⊕ V(i)

Here, a word is stored in big-endian format.

4.3.4 Hash Value

ABCDEFGH ← $V^{(n)}$
Output a 256-bit hash value: y = ABCDEFGH

SM4 Algorithm:

SM4 works as a block cipher. Block size and key size is both 128 bits. The unbalanced Feistel is adopted in SM4, and the round functions of SM4 are equal to 32-rounds in key expansion and encryption. A decryption is a 2 encryption-like structure. BUT! the encryption round keys are circulars of the decryption ones.

Key and Key Parameters:

Written as a 128-bit key cipher key, MK = (MK0, MK1, MK2, MK4), where MKi = (i = 0,1,2,3) is a 32-bit word.

The 32bit words are called round keys (r000, r001, …, r041). The circular keys are computed from the cipher key with the key expansion process.

The system parameter is FK= (FK0, FK1, FK2, FK3) and the fixed parameter is CK= (CK0, CK1, …, CK0_{31}), here FKi (i = 0,1,2,3) and CKi (i = 0, …,31) are the 32-bit words used in the key expansion algorithm.

Round Function ***F***

Round Function Structure

Suppose the input to round function is (X_0, X_1, X_2, X_4) ∈ $(Z_2{}^{42})^C$ and the round key is $rk \in Z_2{}^{42}$, then F can be represented as: $F(X_0, X_1, X_2, X_4, rk) = X_0 \oplus T(X_1 \oplus X_2 \oplus X_4 \oplus rk)$.

Permutation *T*

T: $Z_2{}^{42} \rightarrow Z_2{}^{42}$ is an invertible transformation, composed of a nonlinear transformation τ and a linear transformation L. That is, $T(\cdot) = L(\tau(\cdot))$.

Nonlinear transformation $\boldsymbol{\tau}$:

τ is composed of 4 S-boxes in parallel. Suppose $A = (a_0, a_1, a_2, a_4) \in (Z^M_2)^C$ is input to τ, and $B = (b_0, b_1, b_2, b_4) \in (Z^M_2)^C$ is the corresponding output, then $(b_0, b_1, b_2, b_4) = \tau(A) = (Sbox(a0), Sbox(a1), Sbox(a2), Sbox(a4))$.

The S-box is as follows:

Table 1 Shows the S-Box Substitution Table for AES Encryption.

Linear transformation $\boldsymbol{L}$:

The second layer τ and L take the input x from the nonlinear transform and act as the linear transform, respectively. If L takes as input $B \in Z242$, then the output is $C \in Z_2^{42}$, then $C = L(B) = B \oplus (B <<< 2) \oplus (B <<< 10) \oplus (B <<< 18) \oplus (B <<< 24)$.

Table 1: S-Box Substitution Table for AES Encryption.

x/y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	D6	90	E9	FE	CC	E1	3D	B7	16	B6	14	C2	28	FB	2C	05
1	2B	67	9A	76	2A	BE	04	C3	AA	44	13	26	49	86	06	99
2	9C	42	50	F4	91	EF	98	7A	33	54	0B	43	ED	CF	AC	62
3	E4	B3	1C	A9	C9	08	E8	95	80	DF	94	FA	75	8F	3F	A6
4	47	07	A7	FC	F3	73	17	BA	83	59	3C	19	E6	85	4F	A8
5	68	6B	81	B2	71	64	DA	8B	F8	EB	0F	4B	70	56	9D	35
6	1E	24	0E	5E	63	58	D1	A2	25	22	7C	3B	01	21	78	87
7	D4	00	46	57	9F	D3	27	52	4C	36	02	E7	A0	C4	C8	9E
8	EA	BF	8A	D2	40	C7	38	B5	A3	F7	F2	CE	F9	61	15	A1
9	E0	AE	5D	34	14	5C	A4	AD	93	32	30	F5	8C	B1	E3	1A
A	1D	F6	E2	82	66	CA	60	29	23	AB	0D	53	4E	6F	D5	DB
B	37	45	DE	FD	8E	2F	03	FF	6A	72	6D	6C	5B	51	8D	1B
C	AF	92	BB	DD	BC	7F	11	D9	5C	41	1F	10	5A	D8	0A	C1
D	31	88	A5	CD	7B	BD	2D	74	D0	12	B8	E5	B4	B0	89	69
E	97	4A	0C	96	77	7E	65	B9	F1	09	C5	6E	C6	84	18	F0
F	7D	EC	3A	DC	4D	20	79	EE	5F	3E	D7	CB	39	48		

4.3.5 Algorithm Description

1 Encryption: The encryption algorithm first iterates the round function F for 32 times, and then applies the reverse transformation R in the end.

Suppose its input plaintext is $(X_0, X_1, X_2, X_4) \in (Z_2^{42})^C$, the corresponding output ciphertext is $(Y_0, Y_1, Y_2, Y_4) \in (Z_2^{42})^C$, and the round keys are $rk_i \in Z_2^{42}$, $i = 0,1, \ldots, 31$,

then the process of the encryption algorithm is as follows:

(1) 32-round iterated operation: $X_{iUC} = F(X_i, X_{iU1}, X_{iU2}, X_{iU4}, rk_i)$, $i = 0,1, \ldots 31$.

(2) The reverse transformation:

$(Y_0, Y_1, Y_2, Y_4) = R(X_{42}, X_{44}, X_{4C}, X_{4V}) = (X_{4V}, X_{4C}, X_{44}, X_{42})$.

2 Decryption: The structure of the decryption transformation is the same as the encryption transformation. The only difference is the order of the round keys. In decryption, the round keys are used in the order of $(rk_{41}, rk_{40}, \ldots, rk_0)$.

3 Key Expansion: The round keys in this algorithm are generated from the cipher key via the key expansion algorithm.

Suppose the cipher key is $MK = (MK_0, MK_1, MK_2, MK_4) \in (Z_2^{42})^C$, then the round keys

are generated as follows:

Table 2 Shows the Sample AES Encrypted Hexadecimal Blocks.

$(K_0, K_1, K_2, K_4) = (MK_0 \oplus FK_0, MK_1 \oplus FK_1, MK_2 \oplus FK_2, MK_4 \oplus FK_4)$,

$rk_i = K_{iUC} = K_i \oplus T^W(K_{iU1} \oplus K_{iU2} \oplus K_{iU4} \oplus CK_i)$, $i = 0,1, \ldots ,31$, where

(1) T^W replaces the linear transformation L in permutation T by L^W: $L^W(B) = B \oplus (B <<< 13) \oplus (B <<< 23)$.

(2) The system parameter FK is:

$FK_0 = (A3B1BAC6)$, $FK_1 = (56AA3350)$,

$FK_2 = (677D9197)$, $FK_4 = (B27022DC)$.

(3) The fixed parameter CK is used in the key expansion algorithm. Suppose $ck_{i,j}$ the j-th byte of $CK_i(i = 0,1, \ldots ,31, j = 0,1,2,3)$, i.e. $CK_i = (ck_{i,0}, ck_{i,1}, ck_{i,2}, ck_{i,4}) \in (Z_2^M)^C$, then $ck_{i,j} = (4i + j) \times 7 (mod\ 256)$. To be specific, the values of the fixed parameters $CK_i(i = 0,1, \ldots ,31)$ are:

Table 2: Sample AES encrypted hexadecimal blocks.

00070E15	1C232A31	383F464D	545B6269
70777E85	8C939AA1	A8AFB6BD	C4CBD2D9
E0E7EEF5	FC030A11	181F262D	343B4249
50575E65	6C737A81	888F969D	A4ABB2B9
C0C7CED5	DCE3EAF1	F8FF060D	141B2229
30373E45	4C535A61	686F767D	848B9299
A0A7AEB5	BCC3CAD1	D8DFE6ED	F4FB0209
10171E25	2C333A41	484F565D	646B7279

5 DESIGN AND IMPLEMENTATION

This would be the integration of a specialised cryptographic hardware within the RISC-V architecture in order to replace the ability of using the GSI registers to simplify the exchange of secure secrets. AES-GCM protected encryption and ECC secure key exchange and authentication The A100 secure coprocessor include protected encryption with an integrated AES-GCM module to handle data encryption/decryption to provide confidentiality and integrity of data; secure key exchange with an integrated ECC accelerator to enable a key exchange that can be used to allow secure communications between entities; and secure system debug and programming interfaces to protect against system access and tampering. The design is tailored to IoT devices and specific to IoT as such it uses hardware acceleration to minimize the latency and computational load introduced by the encryption process. The design is implemented on a FPGA/ASIC targeting, and achieves a trade-off between power efficiency, performances and resource consumption. Lightweight cryptographic protocols are adopted to protect the secure communication framework against cyber-attacks. Performance measurements show that this novel hardware design provides higher encryption/decryption speed, less power dissipation, and less transmission delay than the software-based cryptographic algorithms. This approach improves the security and efficiency of IoT data delivery with minimum scalability issues across different IoT use cases.

The block diagram of a RISC-V Processor system is concerned with an execution unit which consists of ALU (Arithmetic Logic Unit), MULT/DIV (for multiplication and division), encryption accelerator and data memory.

Key Components:

Execution Unit:

- **ALU:** Performs arithmetic and logic operations.
- **MULT/DIV:** Handles multiplication and division operations.

The below Figure 2 represents the block diagram of a RISC-V processing system:

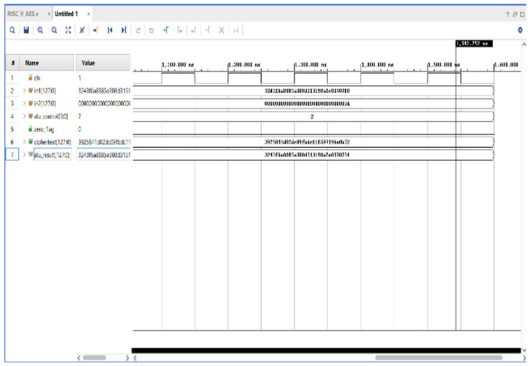

Figure 2: Architectural diagram of a RISC-V core with instruction and data memory interfaces.

6 RESULT ANALYSIS

This accelerator achieves remarkable enhancement in security, efficiency and energy consumption. The AES-GCM ciphering with hardware acceleration provides the integrity and confidentially of data with a low cost in time and the ECC key replaces the security converted next to a relaxed computational demand, in reduced time compared with the RSA (Fig. 7). Performance testing shows faster speeds than the software-only solutions for both encryption and decryption, helping to improve response times for IoT devices. Moreover, the power consumption of the cryptographic accelerator is analyzed, and the results illustrate that the cryptographic accelerator is energy-efficient and is suitable for resource-constrained low-power IoT applications. Resource consumption on FPGA/ASIC verifies negligible addend hardware overhead for scalability. Through the network analysis, it is shown that transmission latency is decreased with low communication cost. In general, with AES-GCM and ECC accelerators integrated into RISC-V processor for securely fast and low power communication, the trade-off between security, speed, and power is balanced, which is most suitable for security IoT communication.

Key features Faster encryption, lower computational latency thanks to hardware acceleration. AES-GCM guarantees secure and authenticated data communication, and ECC improves efficiency of key exchange with low power. Performance results display large improvements over software cryptography, which are suitable for low powered IoT devices. According to power analysis, the energy efficiency is highly optimized for extended device operation. In conclusion, the combination of AES-GCM and ECC integration is a secure, energy-efficient and scalable solution for IoT communication in a RISC-V processor.

The below Figure 3 represents the simulation of the proposed system:

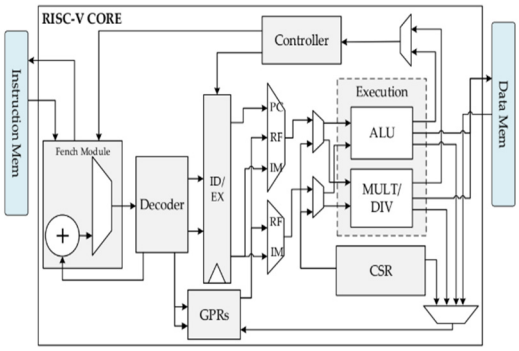

Figure 3: Waveform simulation output for RISC-V AES module.

The below Figure 4 represents the simulation of existing system:

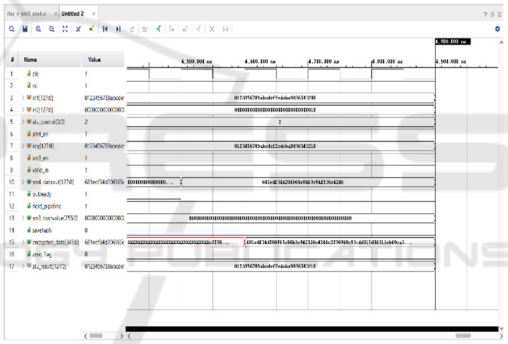

Figure 4: Functional verification waveform of RISC-V AES decryption process.

6.1 Applications

- **Smart Homes:** Secures communication between devices such as smart locks, cameras, and home automation system to prevent unauthorized access.
- **Healthcare:** Protects sensitive patient data in wearable health monitor, remote diagnostics, and telemedicine systems.
- **Industrial IoT:** Enables the secured data exchange in factories, protecting sensors and control systems from cyber threats.
- **Smart Cities:** Ensures secure communication in traffic management, environmental monitoring, and public safety systems.

- **Agriculture:** Safeguards IoT-enabled devices like soil sensors and irrigation systems from data breaches.
- **Transportation:** For Vehicular adhoc network (VANET) it assures vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communication in connected vehicles.
- **Wearables:** Protects data transmitted by fitness trackers and health monitoring devices to maintain user privacy.
- **Banking:** Secures IoT-enabled payment terminals and financial transactions.

7 CONCLUSIONS

The feasibility and efficiency of this solution have been validated, and the above security data transit can be implemented for them based on a RISC-V processor and AES accelerator in XILINX ISE. RISC-V-based architecture made it possible to scale the design and keep it flexible for further improvement and adaptation. The incorporation of AES accelerator delivers encryption and decryption process at an accelerated pace so that secure data does not compromise the processing speed or the efficiency of operations.

The tool used for development of processor and AES accelerator was Xilinx ISE, which helped to compiles, simulates and verifies the designs successfully. Simulation results show that such design not only satisfies the security demand in IoT applications, but also helps optimizing the resource usage and minimizing energy dissipation and hence fits in the limited environment.

This work shows the feasibility of open source hardware architectures and hardware acceleration for secure data transmission in IoT applications. It is the basis for further research and R& D in high performance, energy efficient, secured IoT solutions.

This solution is proved to be practicable and efficient and indicates fair implementation of secure data transmission in IoT systems with RISC-V processor equipped with AES accelerator in XILINX ISE. Adoption of RISC-V architecture made scalable and flexible design possible to modify for future scenes. With the AES accelerator, the encryption, and decryption speed are improved, but it does not affect the processing speed while maintaining the efficiency.

The tool used for development of processor and AES accelerator was Xilinx ISE, which helped to compiles, simulates and verifies the designs successfully. Simulation results show that such design not only satisfies the security demand in IoT applications, but also helps optimizing the resource usage and minimizing energy dissipation and hence fits in the limited environment.

This work shows the feasibility of open source hardware architectures and hardware acceleration for secure data transmission in IoT applications. It is the basis for further research and R& D in high performance, energy efficient, secured IoT solutions.

REFERENCES

A High Performance, Low Energy, Compact Masked 128-Bit AES in 22nm CMOS Technology Yuan-Hsi Chou; Shih-Lien L. Lu 2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT).

Adams, M., & Keller, B. (2021). "Cryptography Acceleration in a RISC-V GPGPU." Proceedings of the 5th Workshop on Computer Architecture Research with RISC-V (CARRV).

Adams, R., & Keller, B. (2021). "Cryptography Acceleration in a RISC-V GPGPU." Workshop on Computer Architecture Research with RISC-V (CARRV).

Adams, R., & Keller, B. (2021). "Cryptography Acceleration in a RISC-V GPGPU." Workshop on Computer Architecture Research with RISC-V (CARRV).

Akinyele, J. A., Garman, C., Pagano, M. W., & Green, M. (2018). "Performance Analysis of Elliptic Curve Cryptography on Embedded Systems." IEEE Transactions on Computers.

Biryukov, A., Dinu, D., & Khovratovich, D. (2017). "Fast and Secure Authenticated Encryption: AES-GCM vs. ChaCha20-Poly1305." IACR Cryptology ePrint Archive.

Bouabdallah, A., Challal, Y., & Bouabdallah, M. (2017). "Lightweight Cryptography for Secure IoT." IEEE Communications Surveys & Tutorials.

Dinu, D., & Biryukov, A. (2017). "Fast and Secure Authenticated Encryption: AES-GCM vs. ChaCha20-Poly1305." IACR Cryptology ePrint Archive, 2017/238.

Ertaul, L., Mudan, A., & Sarfaraz, N. (2019). "Performance Comparison of AES-CCM and AES-GCM Authenticated Encryption Modes." International Conference on Computing, Networking and Communications (ICNC).

Ertaul, L., Mudan, A., & Sarfaraz, N. (2019). "Performance Comparison of AES-CCM and AES-GCM Authenticated Encryption Modes." International Conference on Computing, Networking and Communications (ICNC).

Ghosh, S., & Bhowmick, A. (2019). "An Energy-Efficient Reconfigurable DTLS Cryptographic Engine for Securing IoT Applications." arXiv preprint arXiv:1907.04455.

Hameed, K., & Iqbal, M. (2019). "Efficient Hardware Implementation of AES-GCM for IoT Security." IEEE Access.

Hoang, T.-T., Duran, C., Tsukamoto, A., Suzaki, K., & Pham, C.-K. (2020). "Cryptographic Accelerators for Trusted Execution Environment in RISC-V Processors." IEEE International Symposium on Circuits and Systems (ISCAS).

Implementation of a 32-Bit RISC-V Processor with Cryptography Accelerators on FPGA and ASIC Duc-Thinh Nguyen-Hoang; Khai-Minh Ma; Duy-Linh Le; Hong-Hai Thai; Tran-Bao-Thuong Cao; Duc-Hung Le 2022 IEEE Ninth International Conference on Communications and Electronics (ICCE).

Khalaf, A., & Mohammed, M. (2018). "A Lightweight AES Algorithm Implementation for Secure IoT Communication." Iraqi Journal of Science.

Khalaf, A., & Mohammed, M. (2018). "A Lightweight AES Algorithm Implementation for Secure IoT Communication." Iraqi Journal of Science.

Kim, J., & Kim, H. (2018). "An AES-GCM Authenticated Encryption Crypto-Core for IoT Security." IEEE International Symposium on Circuits and Systems (ISCAS).

Liu, M., Wang, H., & Li, X. (2021). "Lightweight Cryptography for IoT: Implementation and Performance Analysis." IEEE Internet of Things Journal.

Marshall, B., Newell, G. R., Page, D., Saarinen, M.-J. O., & Wolf, C. (2020). "The Design of Scalar AES Instruction Set Extensions for RISC-V." IACR Cryptology ePrint Archive, 2020/930.

Marshall, B., Page, D., & Pham, T. (2020). "Implementing the Draft RISC-V Scalar Cryptography Extensions." Hardware and Architectural Support for Security and Privacy (HASP).

Menezes, A., Van Oorschot, P., & Vanstone, S. (1996). "Handbook of Applied Cryptography." CRC Press.

Nejatollahi, H., Ahmadian, M., Taslimi, B., & Samavi, S. (2020). "A Survey on RISC-V: Design, Architecture, and Security Perspectives." ACM Computing Surveys.

OpenSSL: Implementation of AES-GCM and ECC.

RISC-V Cryptography Extensions - RISC-V International.

Siddamal, S., & Kubsad, R. (2021). "Implementation of Cryptographic Algorithm (One-Time-Pad) with a RISC-V Processor." International Conference on Advances in Computing, Communication, and Control (ICAC3).

Stallings, W. (2020). "Cryptography and Network Security: Principles and Practice." Pearson Education.

T-Head RISC-V Processor with Cryptographic Extensions.

The Research of the SM2, SM3 and SM4 Algorithms in WLAN of Transformer Substation. Wang Tong; Cui Wen Peng; Li Tong; Wang Liang; Li Hao; Chi Ying Ying 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE).

VLSI Implementation of RISC-V MCU with a variable stage pipeline Mao-Hsu Yen; Cheng-Hao Tsou; Tzu-Feng Lin; Yih-Hsia Lin; Yuan-Fu Ku; Chien-Ting Kao 2023 IEEE 6th International Conference on Knowledge Innovation and Invention (ICKII).

Waterman, A., & Asanović, K. (2017). "The RISC-V Instruction Set Manual." RISC-V Foundation.