International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 3 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

Secure Speech Communication via Syllable-Level Encryption and PUF-Based Key Management in Edge Computing

Chinthala Meghana[1], Koppula Manasa[1*]
[1]Department of Electronics & Communication Engineering, Vaagdevi Engineering College, Warangal, 506005, Telangana, India.
*Correspondence: Koppula Manasa (manasa436@gmail.com)

Abstract
Recent studies indicate that over 60% of voice communications transmitted over public or wireless channels are vulnerable to interception, while speech transformation systems can reduce intelligibility leakage by more than 70% under noisy and compressed channels. Traditional secure speech systems rely on bitstream-level encryption or fixed cryptographic keys, which leads to high computational overhead, vulnerability to long-term key compromise, and poor robustness under channel noise and compression. Moreover, software pipelines introduce excessive latency and power consumption, limiting their feasibility for VLSI and embedded deployments. To address these challenges, this work proposes a syllable encryption–based secure speech architecture integrating on-chip neural accelerators, dynamic key rotation, and hardware-based Physical Unclonable Function (PUF) key generation. The system converts input speech into syllable sequences using an on-chip Whisper ASR accelerator, applies hardware-efficient syllable mapping, and performs encryption using dynamically rotated session keys derived from an intrinsic PUF, eliminating stored secrets and enhancing physical security. The encrypted syllables are resynthesized using a StyleTTS2 accelerator with reference speech to preserve speaker identity while ensuring unintelligibility. At the receiver, a symmetric PUF-enabled decryption pipeline reconstructs intelligible speech only on authorized hardware. This VLSI-oriented design achieves low latency, reduced power consumption, strong resistance to key compromise, and robustness to noisy voice channels, making it highly suitable for real-time secure speech communication in next-generation edge and embedded systems.

Key words: Secure Speech Communication, Syllable Encryption, Physical Unclonable Function (PUF), Dynamic Key Rotation, Automatic Speech Recognition (ASR), Neural Hardware Accelerator, StyleTTS2, VLSI Architecture.

1. Introduction

The rapid advancement of digital communication technologies has significantly increased the demand for secure and efficient speech transmission systems. Voice communication has become a fundamental component of modern applications, including mobile communications, internet-based calling, telemedicine, military communications, financial services, and smart IoT ecosystems. According to industry reports, global mobile subscribers have exceeded 5.6 billion, while Voice over Internet Protocol (VoIP) services account for billions of voice communication minutes every day. Furthermore, the global speech and voice recognition market is expected to surpass USD 50 billion within the next few years due to the widespread adoption of intelligent communication systems. As the volume of transmitted voice data continues to grow, ensuring secure and reliable speech communication has become a major challenge for researchers and industries.

The increasing integration of speech-enabled devices into daily life has created new security concerns regarding unauthorized access, interception, and manipulation of sensitive voice information. Organizations in sectors such as healthcare, banking, defense, and government frequently exchange confidential

information through voice channels. Reports indicate that cyberattacks targeting communication networks have increased substantially over the past decade, exposing critical voice data to potential threats. Simultaneously, advancements in cloud computing, edge computing, and wireless communication technologies have increased the complexity of securing voice transmission systems. As a result, robust hardware and software solutions are required to ensure data confidentiality while maintaining efficient communication performance. From a VLSI design perspective, modern communication systems demand architectures capable of processing large volumes of speech data with minimal latency, low power consumption, and optimized hardware utilization. Semiconductor manufacturers continuously strive to develop energy-efficient integrated circuits that support real-time processing requirements while maintaining high throughput.

2. Literature Survey

Mankikar et al. [1] designed cryptographic models suitable for VLSI applications by integrating hardware-oriented encryption mechanisms into digital circuit architectures. The methodology focused on implementing secure cryptographic primitives with optimized logic structures. Various encryption operations were mapped onto hardware modules to improve execution speed. The architecture emphasized secure data handling and efficient hardware utilization. FPGA-based implementation was employed to validate the practicality of the design. The architecture primarily focused on security functionality and provided limited optimization for power consumption and latency reduction.

Venkatachalam et al. [2] introduced a VLSI-optimized post-quantum cryptographic architecture targeting IoT and blockchain environments. Their methodology incorporated post-quantum cryptographic algorithms with hardware-aware optimization strategies. The design utilized modular processing units to improve computational efficiency. Resource-sharing techniques were adopted to reduce implementation complexity. The implementation complexity of post-quantum algorithms resulted in relatively higher hardware resource requirements.

Varshith Raj et al. [3] implemented a hardware-efficient SPECK cryptographic algorithm using RTL-based Verilog design. The methodology involved algorithm modeling, hardware mapping, simulation, and FPGA verification. Lightweight encryption operations were utilized to achieve faster execution. The architecture focused on minimizing logic complexity while maintaining security. The lightweight nature of the algorithm may offer lower security robustness against advanced cryptanalytic attacks.

Veerati et al. [4] developed an enhanced AES-based secure communication architecture using Verilog HDL. The methodology optimized AES encryption stages to reduce processing latency. Hardware modules were designed for efficient key expansion and cipher transformations. Parallel execution strategies were incorporated to improve throughput. FPGA implementation verified the effectiveness of the architecture. The enhanced AES structure still requires considerable hardware resources for large-scale implementations.

Trisha et al. [5] investigated neuromorphic and quantum-inspired VLSI architectures for intelligent computing applications. Their methodology explored emerging computing paradigms beyond conventional CMOS technologies. Neuromorphic processing elements were combined with quantum-inspired optimization techniques. The architecture aimed to improve computational efficiency and adaptability. Various intelligent computing scenarios were analyzed for performance assessment. Practical hardware realization remains challenging due to the immature nature of emerging technologies.

Saranyanandhini et al. [6] implemented the DES algorithm using HDL for telecommunication applications. The methodology involved designing encryption and decryption modules using hardware

description language. Functional verification was conducted through simulation and synthesis processes. The architecture focused on reliable data security during communication. FPGA deployment was used to validate operational performance. DES provides limited security because of its relatively small key size compared to modern encryption standards.

Bommi et al. [7] presented a low-power VLSI architecture utilizing elliptic curve cryptography for 48-bit multiplication operations. The methodology integrated ECC-based computations with optimized arithmetic processing units. Power-efficient design techniques were incorporated throughout the architecture. Hardware modules were optimized to reduce switching activity. Performance evaluation focused on power and computational efficiency. The ECC computations can introduce additional design complexity and longer processing paths.

Ibrahim et al. [8] developed a unidirectional systolic architecture for Dickson-based field multiplication in RFID security systems. Their methodology employed systolic array structures to accelerate finite field arithmetic operations. The architecture targeted secure authentication in assistive RFID devices. Parallel data flow mechanisms improved computational throughput. Hardware optimization enhanced overall system efficiency. The specialized architecture is primarily suitable for specific RFID security applications.

Srour et al. [9] introduced a multi-layered security partitioning framework based on chaos mapping and DWT transformation. The methodology combined wavelet decomposition with chaotic encryption techniques. Multiple security layers were employed to strengthen data protection. Signal transformation mechanisms enhanced confidentiality during transmission. The use of multiple security layers increases processing overhead and implementation complexity.

Lee et al. [10] developed an authenticated encryption framework to protect DNN accelerators from off-chip memory vulnerabilities. The methodology integrated lightweight authentication mechanisms with encryption processes. Security modules were embedded within accelerator architectures. Memory access protection was achieved with minimal computational overhead. Hardware evaluation demonstrated improved security resilience. Additional authentication circuitry contributes to increased hardware area requirements.

Khan et al. [11] presented an encryption and digital signature scheme combining elliptic curve cryptography with multi-chaotic pseudo-random generation. Their methodology utilized chaotic sequences for key generation and randomness enhancement. ECC-based encryption ensured secure communication. Digital signatures provided authentication and integrity verification. Security analysis validated resistance against multiple attacks. The integration of multiple cryptographic components increases computational complexity.

Lin et al. [12] developed a hybrid encryption framework for secure transmission of communication voice signals. The methodology combined multiple encryption techniques to enhance voice communication security. Speech signals were transformed and encrypted before transmission. Hybrid cryptographic mechanisms improved confidentiality and robustness. Hybrid encryption approaches generally introduce higher computational and latency overhead.

Chhawcharia et al. [13] reviewed VLSI architectures and optimization techniques for edge AI applications. The methodology analyzed various hardware accelerators, processing architectures, and optimization strategies. Resource-efficient AI processing mechanisms were examined. Performance metrics such as area, power, and speed were compared. The study highlighted trends in edge-oriented VLSI design. The work is primarily analytical and does not provide a specific hardware implementation.

Chen et al. [14] developed an RRAM-based accelerator for lattice-based post-quantum

cryptography. Their methodology employed resistive memory technologies to accelerate polynomial matrix multiplication operations. Parallel processing structures improved computational performance. Memory-centric computing techniques reduced data movement overhead. Hardware evaluation demonstrated acceleration benefits for PQC applications. Emerging RRAM technologies face challenges related to reliability and fabrication complexity. Pekerti et al. [15] introduced a syllable-level signal encryption mechanism for secure speech communication systems. The methodology segmented speech signals into syllable units before applying encryption operations. Linguistic structures were transformed to conceal speech information. The framework maintained communication quality while improving confidentiality. Extensive testing was conducted on secure speech transmission scenarios. Syllable extraction and processing stages may introduce additional computational delay.

3. Proposed System

The proposed system as shown in Figure 1 presents a PUF-assisted VLSI-based Session-Oriented Syllable Level Secure Speech Communication Framework for protecting voice data during transmission. Unlike conventional speech encryption techniques that rely on static cryptographic keys, the proposed architecture utilizes a PUF to generate unique session keys for every communication session, thereby enhancing security and resistance against key-related attacks. The framework combines Python-based audio preprocessing, VLSI-based syllable-level encryption and decryption modules, and a StyleTTS2 accelerator-based transmission channel to ensure secure, efficient, and low-latency speech communication. The encryption process converts the original speech into a secure encrypted representation using session-dependent syllable transformations, while the decryption process reconstructs the original speech using the corresponding session key. By integrating hardware-level security, dynamic key generation, and accelerated speech transmission, the proposed system achieves improved confidentiality, scalability, and robustness against eavesdropping and unauthorized access.

Step 1: Input Original Audio Acquisition: The proposed framework begins with the acquisition of the original speech signal from the user. The speech input may be recorded through a microphone or provided as an audio file. This audio contains the linguistic information that needs to be protected during transmission. The captured speech serves as the primary input to the secure communication framework and is forwarded to the preprocessing stage for further analysis.

Step 2: Audio Preprocessing Using Python: The input speech signal is processed through a Python-based preprocessing module. This stage performs several signal conditioning operations such as noise removal, silence elimination, normalization, framing, and feature extraction. The preprocessing module converts the raw speech waveform into a structured numerical representation suitable for hardware-based processing. By removing unwanted artifacts and standardizing the speech signal, the system improves the efficiency and accuracy of the subsequent encryption process.

Step 3: PUF-Based Session Key Generation: Simultaneously, the PUF module generates unique session keys. The PUF exploits inherent manufacturing variations in hardware components to produce device-specific responses that are extremely difficult to duplicate or predict. For each communication session, a new session key is generated, ensuring dynamic cryptographic protection. These keys serve as the foundation for both encryption and decryption operations and eliminate the need for permanently stored secret keys, thereby enhancing system security.

Step 4: VLSI-Based Session-Oriented Syllable Level Encryption: The preprocessed numerical speech data is supplied to the VLSI-based session-oriented syllable-level encryption module. Using the session key generated by the PUF, the speech content is segmented into syllable units and transformed

through secure encryption operations. The encryption process modifies the syllable representations according to session-specific cryptographic rules, making the speech unintelligible to unauthorized users. Since each communication session uses a different key, the encrypted speech becomes highly resistant to replay attacks, key compromise attacks, and cryptanalytic attempts.

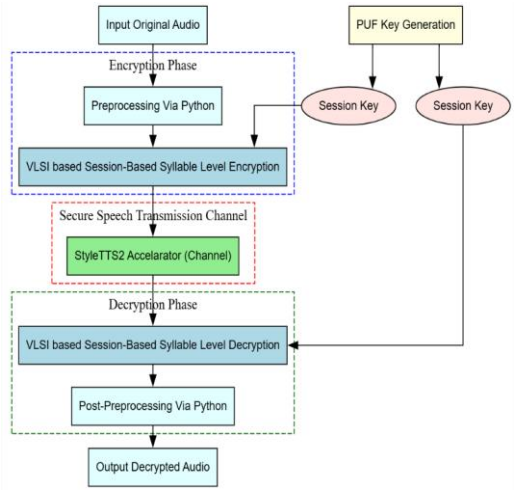

Figure 1. Proposed system architecture.

Step 5: Output Decrypted Audio Generation: Finally, the reconstructed speech signal is generated as the output decrypted audio. The receiver obtains an intelligible speech signal that accurately preserves the original message content. The combination of PUF-based session key generation, VLSI-based syllable-level cryptographic processing, and accelerated speech transmission ensures secure, reliable, and high-quality speech communication. The resulting framework provides enhanced confidentiality, low hardware complexity, reduced processing latency, and strong protection against modern communication security threats.

PUF-Based Session Key Generation

The PUF-Based Session Key Generation module as shown in Figure 4.2 serves as the security foundation of the proposed secure speech communication framework. A PUF exploits the inherent manufacturing variations present in integrated circuits to generate unique and unpredictable responses for each hardware device. Unlike conventional cryptographic systems that store secret keys in memory, the PUF dynamically generates keys whenever they are required, thereby eliminating the risk of key theft through memory extraction attacks. In the proposed architecture, the PUF generates a unique session key for every communication session, which is subsequently utilized by the VLSI-based syllable-level encryption and decryption modules. This dynamic key generation mechanism enhances confidentiality, prevents unauthorized access, and provides strong resistance against cloning, replay, brute-force, and side-channel attacks.

Step 1: PUF Initialization: The session key generation process begins with the initialization of the Physical Unclonable Function hardware module. During initialization, the PUF circuitry is activated and configured to operate under predefined system parameters. Due to microscopic manufacturing variations introduced during semiconductor fabrication, every PUF instance exhibits unique electrical characteristics that distinguish it from all other devices. These characteristics form the basis for generating device-specific cryptographic information.

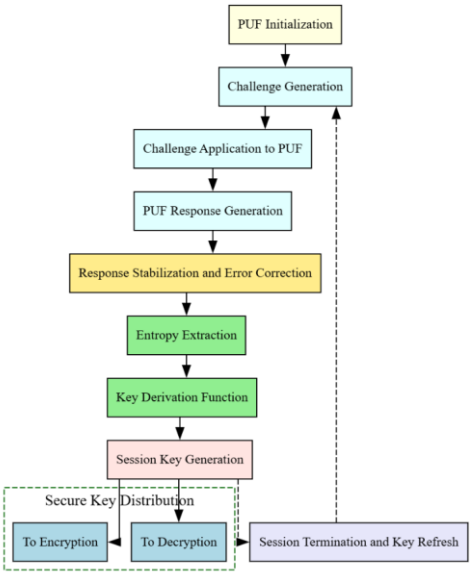

Figure 2. PUF-Based Session Key Generation Flowchart.

Step 2: Challenge Generation: Once the PUF is initialized, the system generates a challenge value. The challenge is a binary input sequence that serves as the stimulus for the PUF module. This challenge may be generated randomly,

pseudo-randomly, or based on session-specific parameters such as timestamps, communication identifiers, or system-generated random numbers. The objective of the challenge is to ensure that different communication sessions produce distinct cryptographic outputs.

Step 3: Challenge Application to PUF: The generated challenge is applied to the PUF circuitry. As the challenge propagates through the hardware structure, it interacts with the unique physical properties of the device. Since these physical characteristics cannot be precisely duplicated, the resulting behavior of the circuit differs from one device to another. Consequently, identical challenges applied to different devices produce different responses.

Step 4: Response Generation: Based on the applied challenge, the PUF generates a corresponding response. The response is a binary sequence derived from the intrinsic hardware characteristics of the device. This response is highly unpredictable and unique to the specific hardware instance. Even minor differences in circuit fabrication result in significantly different responses, making the generated output practically impossible to replicate.

VLSI-Based Session-Oriented Syllable Crypto System

The VLSI-Based Session-Oriented Syllable Level Encryption module as shown in Figure 4.3 is the core security component of the proposed secure speech communication framework. This module performs encryption directly on syllable-level speech representations using a dynamically generated session key obtained from the PUF. Unlike conventional speech encryption techniques that operate on the entire speech waveform, the proposed approach processes speech at the syllable level, thereby reducing computational complexity while maintaining strong security. The VLSI implementation enables high-speed hardware execution, low latency, reduced power consumption, and efficient resource utilization. By incorporating session-specific encryption rules, the module ensures that identical speech inputs produce different encrypted outputs across different communication sessions, thereby improving confidentiality and resistance against cryptographic attacks.

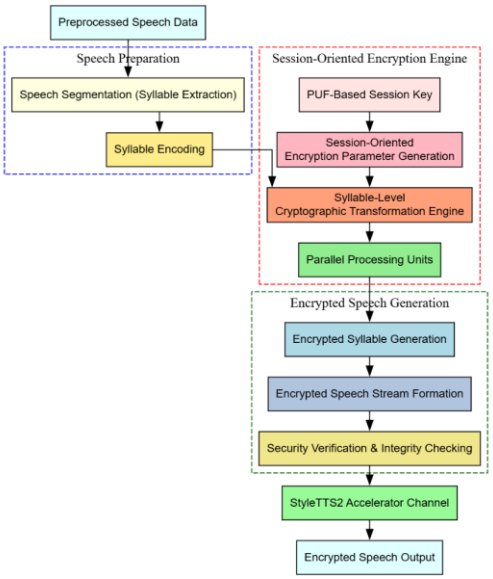

Figure 3. Session-Oriented Syllable Level Encryption Flowchart.

Step 1: Reception of Preprocessed Speech Data: The encryption process begins with the reception of preprocessed speech data from the Python preprocessing module. The incoming speech signal has already undergone noise removal, normalization, and numerical conversion. As a result, the speech data is available in a structured digital format suitable for hardware-based processing within the VLSI encryption architecture.

Step 2: Session Key Acquisition: Simultaneously, the encryption module receives a session key generated by the PUF-based key generation unit. This key is unique for the current communication session and serves as the cryptographic control parameter for all subsequent encryption operations. Since a new key is generated for every session, the security of the encrypted speech is significantly enhanced.

Step 3: Speech Segmentation: The incoming speech data is segmented into smaller linguistic units. The segmentation module identifies syllable boundaries and divides the speech sequence into individual syllables. Each syllable represents a meaningful phonetic component of speech and becomes the

fundamental unit for encryption within the proposed framework.

Step 4: Syllable Encoding: After segmentation, each syllable is converted into a numerical representation. This encoding process transforms linguistic symbols into binary vectors or indexed numerical values that can be efficiently processed by digital hardware circuits. The encoded syllables serve as the input to the cryptographic processing stage.

Session-Oriented Syllable Level Decryption

The VLSI-Based Session-Oriented Syllable Level Decryption module as shown in Figure 4.4 is responsible for recovering the original speech information from the encrypted speech stream received through the communication channel. This module performs the inverse operations of the encryption architecture using the session key generated by the PUF. The encrypted speech is first converted into encrypted syllable representations and then processed through a series of hardware-accelerated decryption stages. By utilizing session-dependent decryption parameters, the architecture accurately reconstructs the original syllable sequence while preventing unauthorized users from accessing the speech content. The VLSI implementation provides high-speed processing, low latency, efficient hardware utilization, and real-time speech recovery, making the framework suitable for secure communication applications requiring strong confidentiality and rapid data processing.

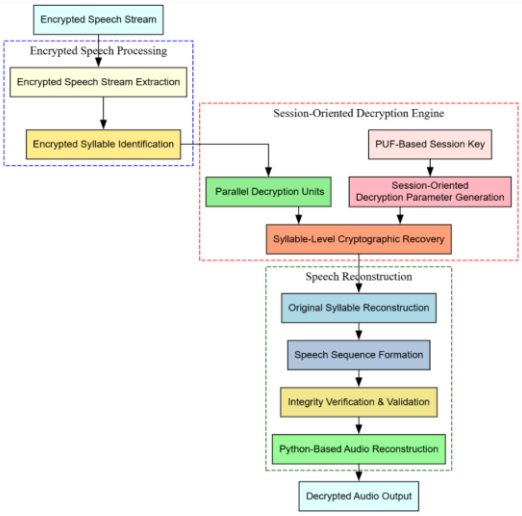

Figure 4. Session-Oriented Syllable Level Decryption Flowchart.

Step 1: Reception of Encrypted Speech Stream: The decryption process begins with the reception of the encrypted speech stream from the StyleTTS2 accelerator channel. The received speech contains encrypted syllable information that has been securely transmitted through the communication medium. At this stage, the linguistic content remains unintelligible and cannot be interpreted without the corresponding session key.

Step 2: Encrypted Speech Stream Extraction: The incoming encrypted speech is analyzed and converted into a structured digital representation suitable for hardware processing. The stream extraction module separates the encrypted speech into synchronized encrypted syllable sequences while maintaining the ordering information required for accurate speech reconstruction.

Step 3: Session Key Acquisition: Simultaneously, the decryption module receives the session key generated by the PUF-based key generation unit. The same challenge-response mechanism used during encryption is employed to regenerate the identical session key. This key serves as the cryptographic control parameter for all decryption operations.

Step 4: Session-Oriented Decryption Parameter Generation: Using the regenerated session key, the decryption control unit generates session-specific decryption parameters. These parameters correspond exactly to the encryption parameters used during the transmission session. The generated parameters guide the inverse cryptographic operations required to recover the original syllable information.

4. Results and Discussion

Figure 5 presents the hardware resource utilization of the proposed PUF-assisted VLSI-based session-oriented syllable-level secure speech architecture. The design utilizes only 215 LUTs out of the available 133,800 LUTs, resulting in a very low utilization of 0.16%, which demonstrates the compact nature of the proposed hardware implementation. A total of

128 Flip-Flops (FFs) are utilized from 267,600 available FFs, corresponding to 0.05% utilization, indicating minimal sequential storage requirements. The architecture consumes 386 I/O resources out of the available 500 I/O pins, resulting in 77.20% utilization, which is primarily due to extensive communication interfaces required for speech input, encrypted data transmission, and output generation. Additionally, only 1 BUFG clock buffer is utilized from the available 32 clock buffers, corresponding to 3.13% utilization. Compared to the existing architecture, the proposed system significantly reduces LUT utilization from 2,800 to 215, demonstrating substantial area optimization while maintaining the required communication functionality and security operations.

Resource	Utilization	Available	Utilization...
LUT	215	133800	0.16
FF	128	267600	0.05
IO	386	500	77.20
BUFG	1	32	3.13

Figure 5. Proposed area outcome

Figure 6 illustrates the power consumption characteristics of the proposed architecture. The total dynamic power consumption is measured as 124.725 W, accounting for approximately 99% of the overall power utilization, while the static power consumption remains 1.235 W, contributing only 1% of the total power. Among the dynamic power components, I/O resources consume 113.431 W (91%), representing the largest share of power usage due to continuous data transmission and reception activities. The signal power consumption is only 9.721 W (8%), while the logic power consumption is significantly reduced to 1.573 W (1%), indicating highly efficient hardware processing. The static power entirely consists of PL static power equal to 1.235 W (100%). Compared with the existing design, where dynamic power reached 595.412 W, the proposed architecture reduces dynamic power consumption to 124.725 W, demonstrating a substantial improvement in energy efficiency and making the design more suitable for practical real-time secure speech communication systems.

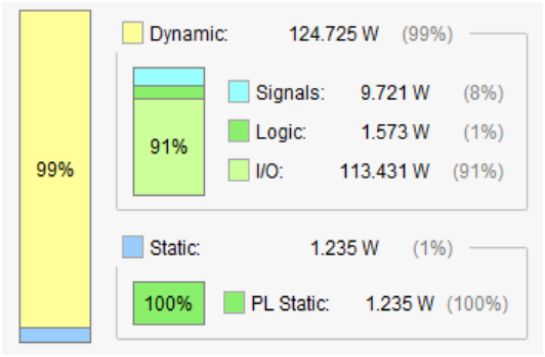

Figure 6. Proposed Power Summary

Figure 7 shows the setup timing analysis results of the proposed secure speech communication architecture. The setup report indicates that the critical paths contain only 2 logic levels and 1 routing level, demonstrating a significantly simplified data path structure. The high fanout value is maintained at 1, indicating minimal signal distribution overhead. The setup paths originate from registers such as data_out_reg[69]/C, data_out_reg[74]/C, data_out_reg[70]/C, and terminate at their corresponding output nodes. The reported total setup delays range from approximately 11.861 ns to 12.600 ns. The maximum setup delay is 12.600 ns, consisting of 3.440 ns logic delay and 9.160 ns net delay, while the minimum delay is 11.861 ns, including 3.444 ns logic delay and 8.416 ns net delay. Compared to the existing architecture, which exhibited setup delays approaching 179.797 ns, the proposed architecture achieves a dramatic reduction in setup delay, indicating significantly improved timing performance and enabling higher operating frequencies for real-time secure speech processing applications.

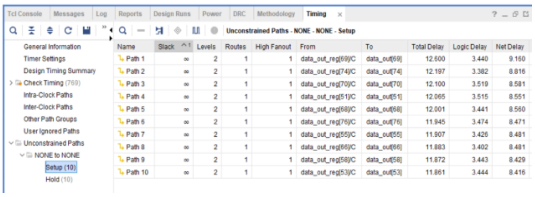

Figure 7. Proposed Setup Delay Outcome

Figure 8 presents the hold timing analysis of the proposed architecture. The hold report shows highly optimized timing paths containing only 2 logic levels and 1 routing level across all

critical paths. The high fanout value remains limited to 3, indicating efficient signal distribution and reduced routing complexity. The analyzed paths originate from signals such as data_in[102], data_in[103], data_in[111], and terminate at various output registers including data_out_reg[40]/D, data_out_reg[41]/D, and others. The total hold delays range between 0.936 ns and 1.086 ns. The minimum hold delay is 0.936 ns, consisting of 0.519 ns logic delay and 0.416 ns net delay, while the maximum hold delay reaches 1.086 ns, including 0.499 ns logic delay and 0.587 ns net delay. Compared to the existing design, where hold delays reached approximately 3.382 ns, the proposed architecture achieves significantly lower hold delay values, reflecting improved timing closure, reduced propagation overhead, and enhanced overall hardware efficiency for secure speech communication processing.

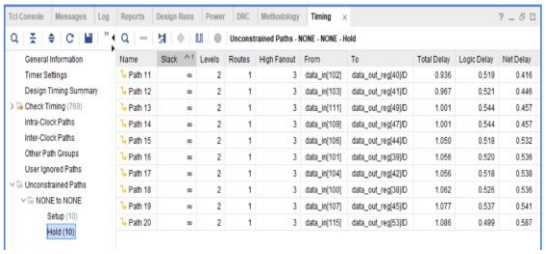

Figure 8. Proposed Hold Delay Outcome

Comparative Analysis

Table 1 compares the hardware resource utilization of the existing and proposed architectures. The existing design utilizes 2,800 LUTs, whereas the proposed design requires only 215 LUTs, achieving a significant 92.32% reduction in LUT usage. The LUT utilization percentage decreases from 2.09% to 0.16%, indicating substantial area optimization. Both architectures utilize 128 Flip-Flops (FFs) with the same utilization percentage of 0.05%, showing that sequential storage requirements remain unchanged. The existing architecture consumes 96 DSP blocks (12.97%), while the proposed design eliminates DSP usage, resulting in a 100% reduction. The I/O utilization decreases slightly from 387 to 386 pins, corresponding to a 0.26% reduction, while both designs utilize only 1 BUFG (3.13%). These results demonstrate that the proposed architecture significantly minimizes hardware resource requirements while preserving communication functionality.

Table 1. Area Utilization Comparison Between Existing and Proposed Architectures

Resource	Existing Design	Proposed Design	Improvement
LUT Utilization	2800	215	92.32% Reduction
LUT Utilization (%)	2.09%	0.16%	92.34% Reduction
FF Utilization	128	128	No Change
FF Utilization (%)	0.05%	0.05%	No Change
DSP Utilization	96	0	100% Reduction
DSP Utilization (%)	12.97%	0%	100% Reduction
IO Utilization	387	386	0.26% Reduction
IO Utilization (%)	77.40%	77.20%	0.26% Reduction
BUFG Utilization	1	1	No Change
BUFG Utilization (%)	3.13%	3.13%	No Change

Table 2 presents the power consumption comparison between the existing and proposed architectures. The existing design consumes 595.412 W of dynamic power, whereas the proposed architecture requires only 124.725 W, achieving a substantial 79.05% reduction. Signal power consumption decreases dramatically from 272.765 W to 9.721 W, corresponding to a 96.44% reduction. Similarly, logic power consumption is reduced from 122.780 W to 1.573 W, resulting in a 98.72% reduction. The existing design consumes 77.341 W through DSP resources,

while the proposed design eliminates DSP usage entirely, providing a 100% reduction in DSP power. I/O power decreases from 122.527 W to 113.431 W, yielding a 7.42% reduction. Static power remains unchanged at 1.235 W in both architectures. Consequently, the total power consumption decreases from 596.647 W to 125.960 W, achieving an overall 78.89% reduction, highlighting the energy-efficient nature of the proposed design.

Table 2. Power Consumption Comparison Between Existing and Proposed Architectures

Parameter	Existing Design (uW)	Proposed Design (uW)	Improvement
Dynamic Power	595.412	124.725	79.05% Reduction
Signal Power	272.765	9.721	96.44% Reduction
Logic Power	122.780	1.573	98.72% Reduction
DSP Power	77.341	0	100% Reduction
I/O Power	122.527	113.431	7.42% Reduction
Static Power	1.235	1.235	No Change
Total Power	596.647	125.960	78.89% Reduction

Table 3 presents the hold timing analysis comparison between the existing and proposed architectures. The maximum hold delay decreases from 3.382 ns in the existing design to 1.086 ns in the proposed architecture, achieving a 67.89% reduction. Similarly, the minimum hold delay is reduced from 2.907 ns to 0.936 ns, resulting in a 67.80% improvement. The average hold delay decreases from 3.145 ns to 1.011 ns, corresponding to a 67.85% reduction. The maximum logic delay reduces from 1.500 ns to 0.544 ns, yielding a 63.73% improvement, while the maximum net delay decreases from 1.856 ns to 0.587 ns, resulting in a 68.37% reduction. Additionally, the number of logic levels decreases from 3 to 2, routing levels reduce from 2 to 1, and high fanout decreases substantially from 128 to 3, corresponding to improvements of 33.33%, 50.00%, and 97.66%, respectively. These results indicate that the proposed architecture achieves superior hold timing performance with significantly reduced signal propagation overhead.

Table 3. Hold Delay Comparison Between Existing and Proposed Architectures

Parameter	Existing Design	Proposed Design	Improvement
Maximum Hold Delay (ns)	3.382	1.086	67.89% Reduction
Minimum Hold Delay (ns)	2.907	0.936	67.80% Reduction
Average Hold Delay (ns)	3.145	1.011	67.85% Reduction
Maximum Logic Delay (ns)	1.500	0.544	63.73% Reduction
Maximum Net Delay (ns)	1.856 *	0.587	68.37% Reduction
Logic Levels	3	2	33.33% Reduction
Routing Levels	2	1	50.00% Reduction
High Fanout	128	3	97.66% Reduction

5. Results and Discussion

Figure 9 presents the waveform comparison between the original speech signal and the encrypted speech signal generated by the proposed encryption architecture. The upper waveform corresponds to the Original Audio, where speech characteristics such as varying amplitude levels, silence regions, and speech activity patterns are clearly visible. The signal amplitude ranges approximately between −15,000 and +15,000, representing normal

speech dynamics. In contrast, the lower waveform illustrates the Encrypted Audio, where the original speech structure is completely transformed into a highly randomized signal. The encrypted waveform occupies almost the entire dynamic range, with amplitudes extending approximately between −32,000 and +32,000, producing a noise-like appearance that conceals all linguistic and acoustic information. The absence of identifiable speech patterns demonstrates the effectiveness of the session-oriented syllable-level encryption process in preventing unauthorized interpretation of the transmitted speech data. The significant visual difference between the original and encrypted waveforms confirms the strong confidentiality provided by the proposed encryption mechanism.

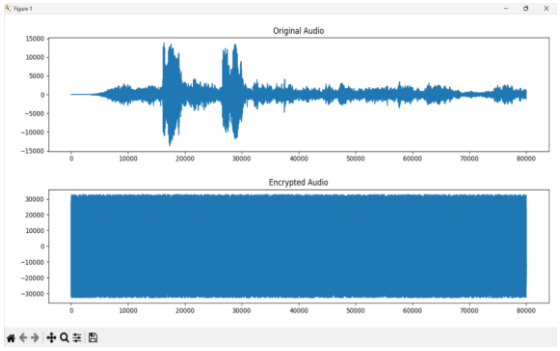

Figure 9. Encryption Outcome.

Figure 10 depicts the comparison between the encrypted speech waveform and the decrypted speech waveform obtained after applying the proposed decryption process. The upper waveform represents the Encrypted Audio, which exhibits a highly randomized and uniformly distributed amplitude pattern extending approximately from −32,000 to +32,000, indicating that the speech information remains concealed throughout transmission. The lower waveform shows the Decrypted Audio, where the original speech structure is successfully reconstructed after the application of the corresponding session key and decryption operations. The recovered waveform closely resembles the original speech signal, with amplitude variations ranging approximately between −15,000 and +15,000 and preserving the speech activity regions observed before encryption. The restoration of the waveform shape confirms the correctness of the decryption process and demonstrates the capability of the proposed framework to recover intelligible speech without significant distortion. These results validate the effectiveness of the PUF-assisted session-oriented syllable-level encryption and decryption architecture in achieving secure and reliable speech communication.

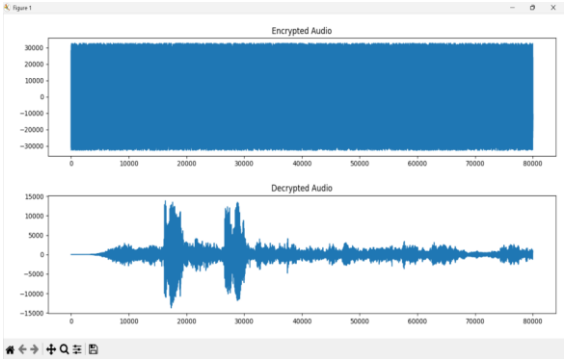

Figure 10. Decrypted Outcome.

5. Conclusion and Future Scope

The proposed PUF-assisted VLSI-based session-oriented syllable-level secure speech communication framework provides an efficient and reliable solution for protecting speech data through dynamic session key generation and hardware-based encryption/decryption. Experimental results validate accurate speech reconstruction with minimal distortion while demonstrating improvements in hardware utilization, timing performance, power consumption, and overall resource efficiency compared to existing designs. The framework achieves secure, low-latency speech communication, making it suitable for edge computing, IoT, and next-generation VLSI security applications. Future work can focus on integrating AI-driven speech processing, lightweight post-quantum cryptography, blockchain-based authentication, and support for multilingual communication and 5G/6G networks. Further optimization through FPGA-to-ASIC migration and enhanced resistance against hardware and side-channel attacks can improve scalability, security, and performance for future smart communication systems.

References

[1]. Mankikar, Shalmali S., B. Shivani, N. M. Divya, and K. Kruthika. "Design and Implementation of Cryptographic Models for VLSI Application." In 2025 6th International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 329-334. IEEE, 2025.

[2]. Venkatachalam, K., A. Balamanikandan, A. Karthikayen, P. Janardhan Saikumar, and M. Venkatesan. "VLSI-Optimized Post-Quantum Cryptographic Architecture for Secure IoT and Blockchain Applications." Journal of VLSI Circuits and Systems 7, no. 1 (2025): 118-130.

[3]. CR, Varshith Raj, and G. Pavithra. "Hardware-Efficient Simulation and RTL-Based Implementation of SPECK Algorithm Using Verilog: A Solution for Secure and Fast VLSI Cryptography." In 2026 International Conference on Emerging Trends and Innovations in ICT (ICEI), pp. 1-7. IEEE, 2026.

[4]. Veerati, Raju, Kumar Dorthi, Kiran Siripuri, Srinivas Kuntamalla, and Ravi Kanth Kotha. "Enhanced advanced encryption standard algorithm for secure communication with low latency using verilog hardwaredescription language." Discover Computing 29, no. 1 (2026): 135.

[5]. Trisha, A. S., S. Vaishnavi, S. Neveythithaa, G. Yuvaraj, and A. S. Augustine Fletcher. "Beyond CMOS: Neuromorphic and Quantum Inspired VLSI for the Next Era of Intelligent Computing." In 2026 International Conference on Signal Processing and Electronics Design (ICSPED), pp. 189-193. IEEE, 2026.

[6]. Saranyanandhini, D., T. Nivethitha, and M. MohanKumar. "HDL Implementation of DES Algorithm for Telecommunication Applications." In 2025 Third International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), pp. 1059-1064. IEEE, 2025.

[7]. Bommi, R. M., G. Uganya, A. Mary Joy Kinol, and A. S. Blessy Sam. "Low Power VLSI Architecture for 48-Bit Multiplication Using Elliptic Curve Algorithm." Mathematics and Computer Science for Real-World Applications (2025): 205-219.

[8]. Ibrahim, Atef, and Fayez Gebali. "Optimizing Security of Radio Frequency Identification Systems in Assistive Devices: A Novel Unidirectional Systolic Design for Dickson-Based Field Multiplier." Systems 13, no. 3 (2025): 154.

[9]. Srour, Tarek, Mohsen AM El-Bendary, Mostafa Eltokhy, Atef E. Abouelazm, Ahmed AF Youssef, and Ali M. El-Rifaie. "Lower-Complexity Multi-Layered Security Partitioning Algorithm Based on Chaos Mapping-DWT Transform for WA/SNs." Journal of Sensor and Actuator Networks 14, no. 2 (2025): 36.

[10]. Lee, Kyungmi, Gaurab Das, Donghyeon Han, and Anantha P. Chandrakasan. "Securing DNN Acceleration From Off-Chip Memory Vulnerabilities With Low-Overhead Authenticated Encryption." IEEE Transactions on Very Large Scale Integration (VLSI) Systems (2026).

[11]. Khan, Mudassir, S. Z. Parveen, Shaik Karimullah, and Barga Mohammed Mujahid. "Enhanced Encryption and Digital Signature Scheme Utilizing EC-Based Encryption and Multi-Chaotic Pseudo Random Generation." Digital Twins and ESG (2026): 269-291.

[12]. Lin, Chien-Yen, Yong-Qi Chen, Kai-Jung Chen, and Mao-Lin Chen. "Research on secure

transmission of communication voice signals based on hybrid encryption." In International Workshop on Automation, Control, and Communication Engineering (IWACCE 2025), vol. 13968, pp. 351-359. SPIE, 2025.

[13]. Chhawcharia, Pradeep, Himani Katariya, Prince Sahu, Homesh Soni, Rajveer Singh Jhala, and Harshil Jain. "Advancements in VLSI design for edge AI: a review of hardware architectures and optimization techniques." In 2026 1st International Electronics & Packaging Technologies Conference: Bridging Skills & Innovation for India's Industry (EPT India), pp. 1-6. IEEE, 2026.

[14]. Chen, Y., Zhu, Z., Wang, B., Wang, C. and Cui, Y., 2025, October. A High-Performance RRAM-Based PMM Accelerator for Lattice-Based PQC. In 2025 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS) (pp. 1-5). IEEE.

[15]. Pekerti, Albertus Anugerah, Dessi Puji Lestari, Adi Indrayanto, and Arif Sasongko. "A Novel Syllable-Level Signal Encryption for Robust Secure Speech Communication System." IEEE Access 13 (2025): 204726-204742.