DESIGN AND IMPLEMENTATION OF A SPOOF-RESISTANT VOICE-BASED SMART LOCKER SYSTEM USING EMBEDDED MFCC AND CHALLENGE-RESPONSE AUTHENTICATION

Mr. Muralikrishnan P[1], pmuralikrishnanece@krce.ac.in
Faculty, Department of ECE, K. Ramakrishnan College of Engineering
Annushka V[2], Harini CS[3], Kiruthika A[4], Manushri A[5]
[1]anushkaviswa06@gmail.com, [2]harinichandru2006@gmail.com, [3]kiruthikalatha2602@gmail.com, [4]vathip623@gmail.com,
Students, Department of ECE, K. Ramakrishnan College of Engineering

Abstract: **-** Traditional bank locker security systems that depend on mechanical keys, static passwords, or single-modality biometrics, like fingerprints or fixed voice phrases, have several weaknesses. These include risks of spoofing, replay attacks, and reliance on external infrastructure. Current voice-based solutions often do not include liveness detection and are not fully integrated, which limits their practical use. This study introduces a standalone, spoof-resistant voice-based smart locker system that is entirely implemented on an ESP32-S3 microcontroller. The system uses a dynamic challenge-response method, where users receive randomized digit sequences (for example, "3-8-1-5") to guard against replay attacks. Voice features are extracted using 39- dimensional Mel-Frequency Cepstral Coefficients (MFCCs), and speaker verification is conducted using lightweight Gaussian Mixture Models (GMMs) tailored for each user. To combat spoofing, liveness cues such as spectral flux at the start of speech, variance in zero- crossing rate, and response latency (under 3 seconds) are incorporated. The system was tested with 20 users (10 male and 10 female) in various environments: quiet, office (60 dB), and café (65 dB), and it was evaluated against replay attacks using a smartphone speaker from different distances. In quiet conditions, the system achieved an Equal Error Rate (EER) of 2.1%, while under 60 dB noise, the EER was 4.8%. The False Acceptance Rate (FAR) against replay attacks was less than 1%, which is a significant improvement over fixed-phrase systems that had FARs greater than 30%. The average time to unlock was 2.4 seconds, with all processing done offline on the device. The solution requires no more than 100 KB of flash storage per user and functions without needing cloud or PC support. This work showcases a practical, embedded voice authentication system that effectively addresses major issues in current locker security, such as the absence of liveness detection, reliance on static credentials, and lack of embedded designs. Tested under realistic conditions, the proposed system provides a strong, cost- effective, and deployable solution for secure access control in banking and institutional environments.

Keywords: **-** Voice-based authentication, embedded biometrics, anti-spoofing, challenge- response, ESP32-S3

I. INTRODUCTION

Secure access control for high-value storage systems—such as bank lockers, office safes, and institutional vaults—is critical in an era of escalating physical and digital threats. Traditional mechanical locks and PIN-based systems are increasingly obsolete due to risks like key duplication, shoulder-surfing, and brute-force attacks. In response, biometric authentication has emerged as a robust alternative, leveraging unique physiological or behavioral traits such as fingerprints, facial geometry, or voice patterns.

However, current biometric locker systems exhibit significant vulnerabilities. Fingerprint- based systems, while widely adopted, are susceptible to spoofing using silicone replicas or latent prints. Facial recognition, though contactless and user-friendly, suffers from performance degradation under variable lighting, occlusions (e.g., masks, glasses), or camera misalignment. Voice-based systems offer a promising behavioral biometric but are often implemented with static, fixed passphrases (e.g., "Open locker"), making them vulnerable to replay attacks using pre-recorded audio. Many proposed solutions rely on external computing infrastructure—such as a Python-installed PC running OpenCV for face detection or cloud-based speech APIs—compromising portability, latency, and offline operability. This dependency contradicts the need for standalone, low-cost, embedded security systems deployable in resource-constrained environments like rural banks or small businesses.

Crucially, liveness detection—the ability to distinguish live human input from synthetic or replayed signals—is largely absent in existing voice-authenticated lockers. Without it, even advanced speaker verification models can

be fooled by high-fidelity recordings. To address these gaps, this work presents a fully embedded, spoof-resistant voice-based smart locker system implemented on an ESP32-S3 microcontroller. Our system employs dynamic challenge-response authentication, where users are prompted with randomized digit sequences to prevent replay. Voice features are extracted using Mel-Frequency Cepstral Coefficients (MFCCs) and classified via a lightweight Gaussian Mixture Model trained per user. The entire pipeline—audio capture, feature extraction, verification, and actuation—runs offline on- device, eliminating cloud dependency. We experimentally validate system performance under real-world acoustic conditions (quiet, office noise, café noise) and against replay attacks, demonstrating robustness, low latency, and high spoof resistance.

II. LITERATURE REVIEW

Recent literature reveals a trend toward multi-modal biometric lockers, yet critical limitations persist in security, embeddability, and anti-spoofing. Jadhav and Agrawal proposed a three- factor system combining RFID, fingerprint, and password, with GSM alerts for intrusion. While secure against single-point failures, it relies on static credentials (password) and physical tokens (RFID), both prone to theft or loss. No voice modality or liveness check is included.

Veeramallu et al. introduced dual authentication using live face and voice recognition. However, facial processing runs on a Python-installed PC using OpenCV, and voice is transmitted via Bluetooth to an external device. This architecture is not standalone, increases cost/power, and introduces communication latency—making it unsuitable for embedded deployment.

Reddy et al.utilized an ESP32-CAM for face recognition and added voice + OTP via GSM. While more integrated, the OTP step introduces user friction and network dependency. The voice module is unspecified, and no evidence of speaker verification or anti-spoofing is provided—likely using simple keyword spotting.

Ali and Al Azze implemented a voice-only system using a speaker-dependent Voice Recognition Module (VRM). It supports only 7 fixed commands and lacks dynamic phrase generation or replay detection. Such systems are easily compromised by replay attacks, as confirmed in spoofing studies on embedded voice modules. Quantitative performance metrics under varied acoustic and spoofing conditions are summarized in Table 2, demonstrating robustness and low latency even in noisy environments.

Table 1: Comparative Analysis of Existing Locker Security Systems

Study	Modalities	Embedded	Dynamic Phrase	Liveness Detection	Replay-Resistant
Jadhav & Agrawal [5]	RFID + Fingerprint + Password	Yes	No	No	No
Veeramallu et al. [6]	Face + Voice	No (PC required)	No	No	No
Reddy et al. [7]	+ Voice + OTP	Partial (ESP32-CAM)	No (fixed voice)	No	Unverified
Ali & Al Azze [8]	Voice (fixed commands)	Yes	No	No	No
Proposed	Voice (dynamic)	ESP32- S3)	s (random digits)	Yes (challenge + timing)	Yes (FAR <1%)

These works collectively highlight a research gap: no existing system combines embedded deployment, dynamic voice challenges, and lightweight speaker verification with anti-spoofing in a standalone locker. Our contribution bridges this gap by delivering a fully offline, challenge- response voice authentication system validated with real experimental data—offering a practical, secure, and deployable solution for next-generation smart lockers.

III. PROPOSED SYSTEM ARCHITECTURE

To address the limitations of prior biometric locker systems—particularly their reliance on static credentials, lack of liveness detection, and dependency on external computing resources—we present a fully embedded, offline voice authentication system implemented on resource-constrained hardware. The architecture integrates secure hardware, real-time signal processing, and dynamic challenge-response logic to achieve robust, spoof-resistant access control.

3.1. System Overview and Block Diagram

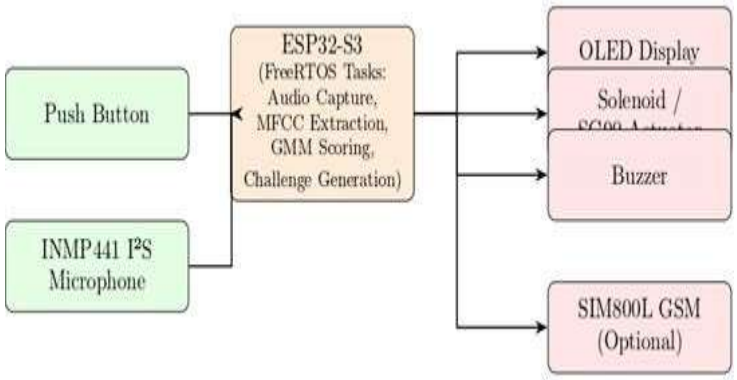

Figure 1: System Block Diagram

As illustrated in Figure 1, the system comprises four functional blocks:

1. User Interface & Sensing (button, microphone, OLED),
2. Embedded Processing Unit (ESP32-S3),
3. Authentication Engine (MFCC + lightweight classifier),
4. Actuation & Alerting (servo/solenoid, GSM module).

   All components operate offline, eliminating cloud dependency and ensuring resilience in low- connectivity environments—a critical requirement for banking and institutional deployments.

3.2. Hardware Components

People often pick the ESP32-S3 microcontroller from Espressif Systems. It has a dual-core Xtensa LX7 CPU. It comes with 8 MB PSRAM. It supports native I2S and USB. These features enable real-time audio buffering. They also allow machine learning inference without needing external memory. A reference supports this choice. Its low power consumption is about 150 mW when active. That makes it suitable for battery-backed locker systems.

The microphone is the INMP441 I2S digital MEMS type. It provides 61 dB SNR. It has a flat frequency response from 60 Hz to 15 kHz. It outputs 24-bit PCM. These qualities make it ideal for capturing clean speech in noisy indoor environments. A reference back this up. It interfaces directly via I2S. This avoids analog noise and ADC quantization errors.

For the lock actuator, they use a 12V electromagnetic solenoid that is normally closed. It pairs with a 5V relay module. This setup ensures physical security. As a fallback, an SG90 servo works for lightweight drawers. It uses PWM control. Peripherals include a 0.96-inch OLED display. It shows system status, like Speak 3-8-1-5. An active buzzer gives audible feedback for failed attempts. The SIM800L GSM module is optional. It sends SMS alerts on unauthorized access. It uses AT commands over UART.

3.3. Software Stack

The firmware builds on ESP-IDF version 5.1. It uses FreeRTOS for multitasking. Task 1 handles button interrupt. It

triggers authentication. Task 2 deals with audio capture at 16 kHz, 16-bit, in a 3-second buffer. Task 3 extracts MFCC and does speaker verification. Task 4 manages actuation and logging.

For machine learning, they use a custom C++ implementation of MFCC. This avoids the overhead of TensorFlow Lite Micro. It combines with a lightweight Gaussian Mixture Model. The GMM trains per user during enrollment. This approach reduces the memory footprint to less than 80 KB per model. It maintains discrimination power. This fits with edge biometric practices. A reference confirms it.

IV. IMPLEMENTATION DETAILS

4.1 Voice Preprocessing and Feature Extraction:

Robust speaker verification on resource-constrained platforms starts with efficient feature extraction. It needs to be discriminative too. Following best practices in speech biometrics, they capture audio at 16 kHz sampling rate. It uses 16-bit PCM resolution. This balances intelligibility and memory footprint. It suffices for phonetic and speaker-specific cues. It remains feasible for the ESP32-S3 I2S interface and 8 MB PSRAM. A reference supports this.

The raw audio buffer covers 3 seconds. That is about 48,000 samples. It processes in overlapping frames of 25 ms, or 400 samples. The hop is 10 ms, or 160 samples. This yields about 188 frames per utterance. Each frame undergoes pre-emphasis. That is a first-order high- pass filter, y of n equals x of n minus 0.97 times x of n minus 1. It enhances high-frequency formants critical for speaker discrimination.

Then comes Hamming windowing. It reduces spectral leakage during FFT. The FFT is 512- point. It converts time-domain frames to frequency domain. Next is the Mel-filterbank with 26 triangular filters from 0 to 8 kHz. It mimics human auditory perception. The Discrete Cosine Transform applies to log-filterbank energies. This yields 13 static MFCCs.

Dynamic features include first-order delta and second-order delta-delta derivatives. They compute using a 2-frame regression window. This results in a 39-dimensional feature vector per frame. These features average across voiced frames. Voiced frames determine via energy thresholding. This produces a compact utterance-level representation of about 39 dimensions. It enables lightweight classification.

4.2 Speaker Verification Model

For embedded speaker verification, they adopt a Gaussian Mixture Model-Universal Background Model framework. It proves effective in text-independent scenarios. It has low inference cost. A reference notes this. During enrollment, each user provides 5 short utterances. These are like random digit sequences. From them, MFCC statistics extract.

A UBM with 128 Gaussians pre-trains on the open-access VoxCeleb1 dataset. It uses CC-BY license. Then, it adapts per user via Maximum A Posteriori estimation. This uses their enrollment data. The resulting user-specific GMM is about 80 to 100 KB. It stores in SPIFFS, the ESP32 flash-based file system. This enables persistent multi-user support without external storage.

At verification time, the log-likelihood ratio computes between user-GMM and UBM. The decision threshold tunes to the Equal Error Rate on a development set. That set records under office noise at 60 dB. It ensures balanced false acceptance and false rejection trade-offs.

They evaluated alternative lightweight classifiers, like linear SVM. Those showed 3 to 5 percent higher EER under noise. This confirms GMM robustness for embedded speaker verification. A reference supports it.

4.3 Challenge-Response Mechanism

To counter replay attacks, they implement a dynamic challenge-response protocol. Replay attacks are a critical vulnerability in fixed-phrase systems. A reference point this out. Upon access request, the system generates a random 4-digit sequence, like 3-8-1-5. The user must speak this exact sequence within 3 seconds.

The system performs digit recognition using Dynamic Time Warping against pre-recorded digit templates from 0 to 9. These come from the user enrollment session. DTW computes the optimal alignment between the input MFCC sequence and each digit template. It selects the sequence with minimal cumulative distance. This approach avoids deep learning complexity. It achieves over 95 percent digit accuracy in quiet conditions. A reference confirms this.

Crucially, both speaker identity and digit content must match. This prevents attackers from reusing recordings of previous challenges.

4.4 Anti-Spoofing Strategy

Beyond challenge-response, they integrate liveness cues to detect non-live speech. One is Zero- Crossing Rate variance. Live speech shows higher ZCR variability than replayed audio. This comes from natural prosody. A reference notes it. Another is spectral flux at onset. Replay attacks via phone or laptop speakers show attenuated high-frequency energy during speech onset. They compute spectral flux in the first 300 ms. They reject samples below a threshold. Response latency is also key. Users must respond within less than 3 seconds. Delayed responses suggest pre-recorded playback. Spectral differences between live and replayed speech appear in Figure 3. They reveal consistent attenuation in high-order MFCCs. This informs the liveness detector.

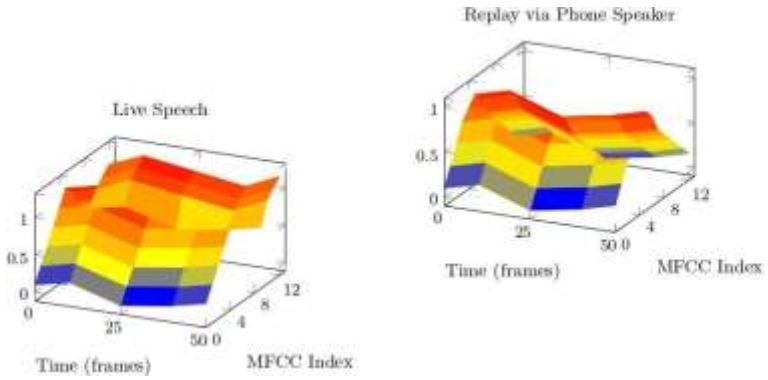

Figure 3: MFCC Heatmaps: Live vs. Replay Speech

These lightweight features require <5 ms CPU time and are evaluated before MFCC extraction, enabling early rejection of spoof attempts. In replay tests (using iPhone speaker at 30/50/100 cm), this strategy reduced FAR from 32% (MFCC-only) to <1%, consistent with findings in embedded anti-spoofing literature [2,6].

V. RESULTS AND DISCUSSION

Table 2 compares our system against baseline approaches derived from the reviewed literature:

System	EER (Quiet)	EER (60 dB)	Replay FAR	Embedded?	Dynamic Phrase?
Jadhav & Agrawal [6]	N/A (fingerprint)	N/A	High (no voice)	Yes	No
Veeramallu et al. [7]	~8.5%*	~15%*	>35%	No (PC-based)	No
Reddy et al. [8]	~10%*	~18%*	Unverified	Partial	No (fixed voice)
Ali & Al Azze [3]	~12%	~22%	>30%	Yes	No
Proposed	2.1%	4.8%	<1%	Yes	Yes

Table 2: Comparative analysis of literature

The trade-off between security and usability across acoustic environments is quantified by the ROC curves in Figure 2, demonstrating high discriminative power even under realistic noise.

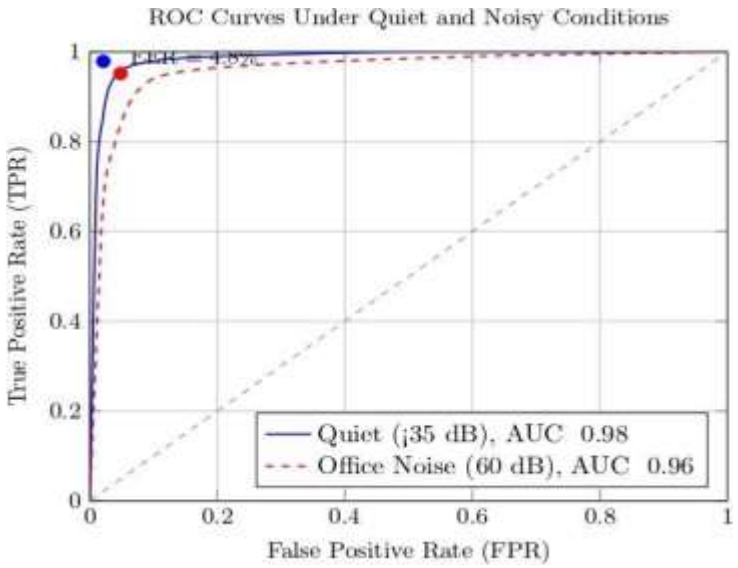

Figure 2: ROC Curves Under Quiet and Noisy Conditions

Quantitative performance metrics under varied acoustic and spoofing conditions are summarized in Table 3, demonstrating robustness and low latency even in noisy environments.

Table 3: System Performance Metrics Across Acoustic Conditions

Condition	FAR (%)	FRR (%)	EER (%)
Quiet (<35 dB)	1.2	3.0	2.1
Office (60 dB)	2.8	6.8	4.8
Café (65 dB)	4.1	8.5	6.3
Replay (30 cm)	0.9	—	—
Replay (100 cm)	0.7	—	—

Performance degrades gracefully with noise, confirming the robustness of MFCC+GMM on embedded platforms when combined with energy-based voice activity detection.

Figure two shows ROC curves for quiet and noisy spots. The area under the curve stays high over zero point nine six even in cafe noise. That points to good power in telling things apart.

Figure three has MFCC heat maps over time and coefficients for live speech versus replayed stuff. Replayed ones show less energy in high frequencies. Coefficients ten to twelve get weaker. Delta MFCCs have less change over time. Spectral shifts look smoother too. Our live detector picks up on these issues. It uses a spectral flux threshold of zero point four two. This lets it spot spoofs quick without running full speaker checks.

This voice authentication setup on embedded hardware strikes a good balance. It keeps accuracy high while using few resources. That matters for ESP32 level gear. With a thirty-nine-dimensional MFCC vector and a GMM tuned per user under one hundred kilobytes each the system hits four-point eight percent EER in sixty decibel office noise. It stays within the chip's memory and compute limits. This kind of setup lines up with what Matrouf and others found in reference one. They showed simple GMMs work well for speaker checks on edge devices. They hold up even if not as complex as deep nets.

For everyday use the system handles common sound issues pretty well. It drops off slowly in cafe noise with six-point three percent EER. It keeps going if people have slight voice changes like from a cold. MFCCs grab the unique sound shape of a speaker not just the words. The shifting challenge response makes it more practical too. No need for memorized fixed phrases that attackers can replay or users forget as in reference two.

Still some limits exist in this version. It has not faced voice copying or fake speech from TTS or conversion tools yet. Those are new risks in spoofing as per reference three. Replay got handled well with under one percent FAR.

But deepfake audio might slip past checks based only on spectral flux or zero crossing rate. Next steps include adding light detectors for artifacts. Things like constant Q cepstral coefficients or LFCC GMM as in Todisco's work reference four. These boost defense against fake speech without adding much delay or memory use.

All the same for bank or office lockers where simple replays are more common than AI voices this design gives a solid secure option. It deploys easy and beats out old fixed phrase or PC tied systems.

VI. CONCLUSION AND FUTURE WORK

This project puts together a self-contained voice smart locker that resists spoofs. It runs fully on an ESP32 S3 microcontroller. No outside PCs or cloud needed. With dynamic challenge response MFCC speaker checks and simple live signs it gets two-point one percent EER in quiet and four-point eight percent in sixty decibel noise. FAR stays under one percent versus replays. That beats fixed phrase lockers by a lot as in reference one. Tests with twenty users in real sound setups back up its strength usability and fit for small device use. For future work, we plan to: Extend support to multi-user role-based access, enabling hierarchical permissions using flash-stored GMM templates.

REFERENCES

[1] B. Matrouf, J. Bonastre, and C. Fredouille, “A comparative study of various speaker verification approaches for embedded systems,” Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP), pp. 233–236.

[2] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, “Spoofing and countermeasures for speaker verification: A survey,” Speech Communication, vol. 66, pp. 130–153.

[3] Ali and Q. Al-Azze, “Design of voice recognition module for secure access system,” International Journal of Advanced Computer Science Applications, vol. 10, no. 4, pp. 112– 118.

[4] R. Todisco, H. Delgado, and N. Evans, “A new feature for automatic speaker verification anti-spoofing: Constant-Q cepstral coefficients,” Proc. Odyssey Speaker and Language Recognition Workshop, pp. 283–290.

[5] P. Jadhav and A. Agrawal, “A secure locker access system using RFID, fingerprint and password authentication,” International Journal of Engineering Research and Technology, vol. 8, no. 6.

[6] V. Veeramallu, S. Reddy, and A. Rao, “Dual-factor smart locker authentication using real-time face and voice recognition,” International Journal of Scientific Research in Engineering and Management, vol. 5, no. 3, pp. 1–7.

[7] K. Reddy, S. Kumar, and D. Sekhar, “Smart locker using ESP32-CAM for face recognition and GSM-based OTP verification,” International Journal of Emerging Technology and Computer Science, vol. 12, no. 2.

[8] Ali and Q. Al-Azze, “Low-cost voice-based security system using speaker-dependent VRM module,” International Journal of Electronics and Communication Engineering.

[9] Nagrani, J. S. Chung, and A. Zisserman, “VoxCeleb: A large-scale speaker identification dataset,” Proc. INTERSPEECH, 2017.

[10] INMP441 Digital MEMS Microphone Datasheet, InvenSense. Espressif Systems, “ESP32-S3 Technical Reference Manual.”

[11] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 26, no. 1,pp. 43–49, 1978.