IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 July 2025, accepted 11 August 2025, date of publication 19 August 2025, date of current version 2 September 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3600364

RESEARCH ARTICLE

Multilingual Low-Latency Emergency VoIP System Using LLM for Speech Reconstruction and Blockchain for Secure Data Archiving

RAHMAN A. RAFI[1], (Graduate Student Member, IEEE), SHAKIL AHMED[1], (Member, IEEE), DANUSH VENKATESHPERUMAL[1], (Graduate Student Member, IEEE), ASHFAQ KHOKHAR[1], (Fellow, IEEE), MD ARIFUZZAMAN[2], AKM AZAD[3], AND SALEM A. ALYAMI[3], (Member, IEEE)

[1]Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA
[2]Department of Civil and Environmental Engineering, King Faisal University, Al Hofuf 31982, Saudi Arabia
[3]Department of Mathematics and Statistics, Imam Mohammad Ibn Saud Islamic University, Riyadh 13318, Saudi Arabia

Corresponding authors: Shakil Ahmed (shakil@iastate.edu) and Salem A. Alyami (saalyami@imamu.edu.sa)

This work was supported by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) under Grant IMSIU-DDRSP2503.

ABSTRACT Emergency VoIP communications frequently suffer from packet loss, jitter, and background noise, which degrade speech quality, delay dispatcher response, and threaten the reliability of emergency services. This paper presents a blockchain-governed, low-latency communication response framework powered by a Large Language Model (LLM), leveraging GPT-4o-mini for semantic speech reconstruction and dynamic emergency prioritization. The proposed architecture integrates multilingual speech-to-text and translation via Google APIs, a Facebook AI Similarity Search (FAISS)-indexed retrieval system for context-driven LLM inference, and severity classification for dispatcher triaging. To ensure tamper-proof data governance, the system incorporates AES-256-CBC encryption, Shamir's Secret Sharing, and decentralized transcript storage on Arweave, with access mediated by Solana-based multi-signature wallets and program-derived addresses. Quantitative evaluations confirm that the proposed framework meets low-latency standards with sub-1.5 second one-way latency—significantly outperforming prior VoIP-based multilingual emergency systems, which typically report 3–5 second delays. Unlike conventional pipelines, it leverages asynchronous blockchain storage and optimized LLM-driven reconstruction to achieve high responsiveness without sacrificing Conceptual Precision, BLEU, or ROUGE accuracy.

INDEX TERMS LLM, emergency communications, speech reconstruction, call prioritization, VoIP, packet loss, multilingual support, secure storage system.

I. INTRODUCTION

Emergency call centers in the United States handle over 240 million calls annually, more than 80% of which originate from wireless devices [1]. In a basic emergency call system, a dispatcher at the Public Safety Answering Point (PSAP) receives the call, and the victim communicates the emergency and location verbally or via text services [2]. As most wireless emergency calls are transmitted over Voice-over-Internet Protocol (VoIP), they remain inherently vulnerable to packet loss, which poses a significant potential threat to audio quality, intelligibility, and overall service reliability [3]. A single lost packet may contain multiple speech frames and burst losses further degrade communications by reducing the effectiveness of error concealment techniques such as waveform substitution, as the lack of sufficient reference data hampers its effectiveness [4]. These factors make packet loss a critical challenge in VoIP systems [5]. Moreover,

The associate editor coordinating the review of this manuscript and approving it for publication was Chuan Heng Foh.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

packet loss, higher latency, and jitter degrade audio clarity, making it increasingly difficult for dispatchers to interpret critical information accurately [6]. These network constraints significantly impact call quality, as loss rates of 5-10% can severely reduce speech comprehension. In particular, consecutive packet loss, often occurring in bursty patterns rather than random drops, introduces prolonged gaps in the audio stream. These disruptions degrade the overall quality of service, impairing the effectiveness of emergency communications systems [7], [8]. Call clarity in such scenarios may further deteriorate due to compounding factors such as emotional distress, ambient background noise, or low-volume speech from the victim—factors that collectively exacerbate the difficulty of accurate speech interpretation [9]. All of these directly affect emergency communications and its response times, with each minute of delay increasing the mortality rate by 1% in trauma cases and up to 30% by just 5 minutes of delay [10], [11]. Moreover, consecutive packet losses, known as burst packet loss, can create significant gaps in audio, further deteriorating call quality [7], [8]. Additionally, because emergency dispatchers operate from separate Public PSAPs, rerouting becomes necessary when calls are misdirected—such as a 911 call reaching a PSAP outside the victim's jurisdiction—thereby increasing latency, degrading service quality, and raising the risk of misclassifying the emergency [12]. The response coordination is also worsened due to the lack of unified data-sharing between Emergency Medical Services (EMS) [13].

These limitations within VoIP-based emergency communications reduce speech intelligibility, which may cause critical delays in emergency response. Poor audio quality forces dispatchers to request repeated clarifications, whereas inadequate coordination between PSAPs and EMS, which are responsible for providing pre-hospital care and patient transport, degrades the quality of service. Language barriers further complicate these risks, necessitating low-latency speech transcription and translation mechanisms to assist dispatchers in accurately interpreting emergency calls. Leveraging cloud-based APIs for speech recognition, Speech-To-Text (`STT`), Text-To-Speech (`TTS`) and multilingual translation ensures seamless communications in time-critical scenarios while reducing errors caused by language barriers.

In addition to reconstructing lost speech, timely classification of call severity is crucial to help dispatchers triage and prioritize critical emergencies. Traditional systems lack mechanisms to assess urgency in fragmented or multilingual calls. Our framework introduces a contextual Large Language Model (LLM)-based severity classification model that progressively refines its predictions as more speech context becomes available, enabling real-time prioritization during active calls. However, securing sensitive emergency transcripts remains a concern. To ensure the security of stored transcripts, access to encryption keys is strictly controlled. Direct storage of encryption keys on public or centralized databases poses significant security risks, as a single point of compromise could expose all stored records. To mitigate this, we incorporate Shamir's Secret Sharing (`SSS`) [14], a cryptographic technique that splits the encryption key into multiple shares and distributes them across independent storage servers. This ensures that no single entity can access the complete key, reducing the risk of unauthorized decryption. A threshold number of shares is required to reconstruct the key, providing security and fault tolerance. `SSS` is integrated with blockchain-based access control to strengthen security further, ensuring that encrypted emergency transcripts remain accessible only to authorized personnel while eliminating single points of failure.

Ensuring the integrity of emergency call transcripts is critical, as they may contain sensitive legal or operational data. A blockchain-based system offers a verifiable, tamper-proof audit trail, eliminating accidental loss or unauthorized access risks. Decentralized and immutable storage ensures transcripts remain secure and accessible only to authorized personnel. Blockchain technology enhances this by providing cryptographic security, transparent access control, and permanently logging all interactions. Every access request is traceable and resistant to manipulation. To address challenges like packet loss and speech degradation, we integrate a LLM for reconstructing incomplete segments using contextual embeddings and past emergency call data. Simultaneously, transcripts are immutably stored on blockchain, ensuring their integrity. By combining LLM-based speech reconstruction, multilingual accessibility, and blockchain-secured storage, our system ensures accurate, reliable, and secure emergency communication. Blockchain access control guarantees that only verified personnel can retrieve transcripts, safeguarding critical information.

A. RELATED WORK

Recent advancements have explored integrating AI-driven models into emergency response systems [15], [16], [17], [18], [19], [20]. For example, researchers in [21] incorporated open-source LLMs such as Llama2 to analyze emergency calls, provide low-latency dispatcher guidance, and improve situational awareness by integrating social media-informed emergency protocols. Although it assisted with crisis management, it did not address any aspects of poor network conditions. Other studies focused on improving call triage and prioritization. In [22], an AI model was designed to assess victim's emotional states during out-of-hospital cardiac arrests. It used mel-frequency cepstral coefficients to detect distress and help dispatchers prioritize high-risk cases. Similarly, the authors in [23] introduced a call-tracking system to mitigate emergency service abuse by filtering out non-urgent calls while ensuring critical cases were not overlooked. All of these systems contributed to making emergency response systems more reliable. Beyond this, the authors in [24] proposed EMSAssist, a mobile voice assistant for EMS that assists paramedics by guiding them through predefined medical procedures. The system also enables speech-based protocol retrieval in disaster scenarios. Unlike traditional cloud-based systems, this approach operates at the

TABLE 1. List of symbols.

Symbol	Definition
S_c	Victim's speech signal (audio)
S'_c	Final synthesized audio in the victim's language L_c
L_c	Victim's native language
L_d	Dispatcher's language
R_d	Dispatcher's spoken response (audio)
T_c	Initial transcript in victim's language L_c
$T_{\texttt{TC}}$	Translated transcript in dispatcher's language L_d
T_c^*	LLM-reconstructed transcript
T_d	Dispatcher's response in language L_d
T'_d	Dispatcher's response translated to victim's language L_c
w_x	Word at position x in transcript
_	Symbol representing missing or corrupted words
H	Contextual information from previous emergency calls
M	Combined plaintext message $(T_c^* \Vert T_d)$
K	`AES-256-CBC` encryption key
I_V	Initialization Vector for AES encryption
C	Encrypted transcript (ciphertext)
$C_{\texttt{store}}$	Format for storage on Arweave: $(I_V \Vert C)$
$X_{\texttt{AR}}$	Arweave transaction ID
(t, n)	Threshold scheme: t of n shares needed to reconstruct key
(m, s)	Multi-signature scheme: m of s signers required for approval
`SSS`	Shamir's Secret Sharing scheme
`PDA`	Program Derived Address on Solana blockchain
`KSS`	Key Share Server (distributed)
$K_{\texttt{pub,r}}, K_{\texttt{pvt,r}}$	Public and Private key of requester wallet

edge, reducing dependency on centralized infrastructure and improving response times. It utilizes a fine-tuned conformer automatic speech recognition model, which combines convolutional and Transformer-based architectures for efficient and accurate speech transcription. A WordPiece tokenizer also segments speech into subword units, enhancing transcription accuracy, particularly for domain-specific medical terminology. However, EMSAssist is primarily designed to support paramedics in protocol selection rather than addressing speech degradation or dynamic emergency prioritization. It focuses on predefined medical workflows rather than reconstructing fragmented speech or assessing call urgency in low-latency.

Recent works have addressed emergency response using blockchain-secured storage [25], multilingual low-latency NLP systems [26], and ethical risks of LLMs in public service AI [27]. Unlike these approaches, our framework uniquely integrates LLM-based speech reconstruction with decentralized secure archival and real-time VoIP support, explicitly addressing degraded and multilingual conditions in critical calls.

LLMs have also been explored for broader disaster communications, like in [28], where researchers leveraged social media data during Hurricane Harvey to identify urgent messages and facilitate low-latency emergency response when traditional communications channels were overwhelmed. Similarly, [29] made a system of automated emergency detection through social media. Still, these approaches remained limited to non-call-based platforms, leaving a gap in LLM-driven enhancements for direct emergency dispatch communications.

These approaches remained limited to non-call-based platforms, leaving a gap in LLM-driven enhancements for direct emergency dispatch communications. Beyond speech processing, recent advances in decentralized storage have emerged as a promising paradigm for secure data archival in critical systems. Reference [25] introduces a blockchain-based secure storage scheme for electronic medical records using IPFS, demonstrating the viability of hybrid on-chain/off-chain architectures for sensitive data management. Despite these advancements, existing solutions do not address the comprehensive challenges of degraded speech reconstruction, language barriers, AND secure transcript storage in a unified emergency communication system.

B. CONTRIBUTION

We propose a robust framework for enhancing emergency communication through LLM-based speech reconstruction, multilingual low-latency transcription, and blockchain-secured data storage. Our key contributions include:

1) LLM-BASED RECONSTRUCTION

Leveraging ChatGPT-4o-mini, the system contextually reconstructs fragmented emergency speech affected by packet loss or noise, ensuring coherent and intelligible transcripts without altering network infrastructure.

2) MULTILINGUAL LOW-LATENCY TRANSCRIPTION

Victim speech is transcribed using Google `STT` and translated to the dispatcher's language via Google Translation API, enabling real-time, cross-language communication.

3) BIDIRECTIONAL RESPONSE AND TTS

Dispatcher responses are reverse-translated and converted to speech via Google `TTS`, allowing victims to receive clear verbal replies in their native language.

4) AES-256 ENCRYPTED STORAGE WITH SSS

Transcripts are encrypted using `AES-256-CBC`, with decryption keys split using Shamir's Secret Sharing (SSS) and distributed across multiple servers to enhance confidentiality and fault tolerance.

5) DECENTRALIZED STORAGE VIA ARWEAVE

Encrypted data is stored on Arweave, ensuring immutable, permanent records resistant to deletion or tampering.

TABLE 2. List of acronyms used in the paper.

Acronym	Description
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
LLM	Large Language Model
STT	Speech-to-Text
TTS	Text-to-Speech
VoIP	Voice over Internet Protocol
PSAP	Public Safety Answering Point
NLP	Natural Language Processing
SSS	Shamir's Secret Sharing
FAISS	Facebook AI Similarity Search
PDA	Program Derived Address
I_V	Initialization Vector
HMAC	Hash-Based Message Authentication Code
EMS	Emergency Medical Services
API	Application Programming Interface
KSS	Key Share Server
BLEU	Bilingual Evaluation Understudy (Metric)
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
URL	Uniform Resource Locator
GCP	Google Cloud Platform
NIST	National Institute of Standards and Technology
XAR	Arweave Transaction Identifier

6) BLOCKCHAIN-BASED ACCESS CONTROL

A multi-signature authentication mechanism on the Solana blockchain ensures that only authorized parties can retrieve transcripts based on predefined consensus thresholds.

7) SECURE RETRIEVAL AND ENCRYPTION

Verified requests trigger transcript retrieval and encryption using the requester's public key, preserving end-to-end privacy and access integrity.

II. PROPOSED FRAMEWORK

The proposed system architecture for emergency VoIP services enables low-latency transcription, multilingual translation, packet loss mitigation through speech reconstruction, priority-based emergency classification, and cryptographically secure storage of emergency communications logs, ensuring data integrity and auditable retrieval mechanisms within a wireless communications network. Fig. 1 illustrates the overall system architecture, where a victim is connected to a PSAP upon initiating an emergency call via VoIP. The PSAP further handles all communications between the victim and the dispatcher, performing LLM reconstruction, STT, TTS, and translation mechanisms. This is further pipelined to an encryption mechanism and stores the contents in storage modules. These stored items can be retrieved by the authorities by a retrieval mechanism. The system, therefore, is designed to handle VoIP-based emergency calls, process them efficiently for dispatcher interaction, and store them securely for verification and review. When an emergency call is placed over a VoIP network, the victim's speech is denoted as S_c. The emergency call is routed to a PSAP, where a dispatcher receives and begins processing the communications. Upon arrival at the PSAP, the incoming speech signal, S_c from the victim undergoes STT processing via the Google Cloud Speech-to-Text API, which transcribes spoken content in the victim's native language L_c into a textual transcript, T_c. Table 2 lists the acronyms used in the paper. This transformation is mathematically expressed as follows [30]:

$$T_c = f_{\mathrm{STT}}(S_c, L_c) \tag{1}$$

where $f_{\mathrm{STT}}(.)$ denotes the speech recognition function provided by the API. To ensure effective communications between the victim and dispatcher, the obtained transcript is translated into the dispatcher's preferred language, i.e., English, L_d defined as follows:

$$T_{\mathrm{TC}} = T(T_c, L_c, L_d) \tag{2}$$

where $T(.)$ is the translation function. T_{TC} is the translated transcript of victim in L_d. T_c is the initial transcript in victim's language. The obtained transcript T_{TC} is represented as a sentence with possibly incomplete words, which can be expressed as a sequence of words as follows:

$$T_{\mathrm{TC}} = (w_1, w_2, w_3, _, w_5, _, w_7, \ldots, w_N) \tag{3}$$

where w_x represents the words and _ represents segments lost or corrupted due to packet loss, transmission errors, or speech truncation. To reconstruct the missing portions and refine the sentence structure, the transcript is passed through an LLM, which leverages contextual embeddings and learned representations to generate a more complete version of the emergency message. The reconstructed transcript, denoted as T_c^*, is then obtained through the function:

$$T_c^* = f_{\mathrm{LLM}}(T_{\mathrm{TC}}, H) \tag{4}$$

where $f_{\mathrm{LLM}}(.)$ is the LLM processing function. H represents additional contextual information retrieved from a database containing prior emergency communications records. The LLM effectively predicts the missing words by maximizing the probability of a coherent reconstruction given the available context H, as shown in Fig. 2. The dispatcher processes the refined transcript and provides a response, denoted as R_d, which is further transcribed to text using STT conversion to obtain the response transcript $T_d$which is the transcribed response in the dispatcher's language, L_d. The dispatcher's verbal response $R_d(t)$ is captured and processed using a speech-to-text function f_{STT} over the language domain L_d. This function maps the acoustic waveform to a tokenized textual transcript:

$$T_d = f_{\mathrm{STT}}(R_d(t), L_d) = \{w_1^{(d)}, w_2^{(d)}, \ldots, w_{N_d}^{(d)}\} \tag{5}$$

where T_d is the resulting dispatcher transcript, $w_i^{(d)}$ is the i^{th} word token in the dispatcher's language L_d, and N_d is the total number of recognized word tokens. This transcript is then converted to the victim's original language, L_c, using

$$\begin{aligned} T_d' &= T(T_d, L_d, L_c) \\ &= \arg\max_{T'} \prod_{j=1}^{N_c} P(w_j^{(c)} \mid w_1^{(d)}, \ldots, w_{N_d}^{(d)}; \theta_T) \end{aligned} \tag{6}$$

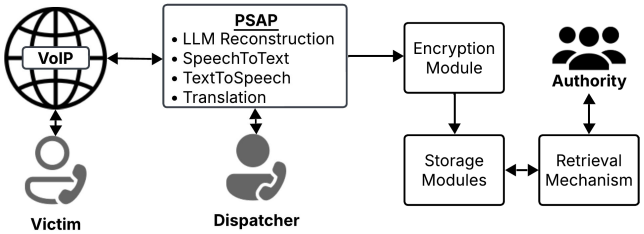


FIGURE 1. Proposed system components.

where T'_d represents the dispatcher's response in the victim's language. The translated text is then synthesized into speech using TTS by

$$S'_c = f_{\mathrm{TTS}}(T'_d, L_c) \tag{7}$$

where S'_c is the final audio output in the victim's language L_c. Furthermore, to ensure security and integrity, the reconstructed transcript T^*_c and the dispatcher's response T_d are stored using a blockchain-based mechanism. All emergency call transcripts are securely encrypted using the AES-256-CBC encryption standard, following best practices recommended by the National Institute of Standards and Technology (NIST) [31]. Let both transcripts be represented as follows:

$$M = (T^*_c \| T_d) = \{w^{(c)}_1, \ldots, w^{(c)}_{N_c}, \mathrm{SEP}, w^{(d)}_1, \ldots, w^{(d)}_{N_d}\} \tag{8}$$

where $\|$ denotes concatenation. $T^*_c = \{w^{(c)}_1, \ldots, w^{(c)}_{N_c}\}$ is the LLM-reconstructed transcript of the victim's original speech in language L_c. $T_d = \{w^{(d)}_1, \ldots, w^{(d)}_{N_d}\}$ is the dispatcher's response transcript in language L_d. SEP is a designated separator token used to distinguish between the two components in the combined message, The encrypted ciphertext C is computed as:

$$C = \mathrm{E}_{\mathrm{AES}}(M, K, I_V) \tag{9}$$

where C is the encrypted transcript, K is the encryption key, and I_V is an initialization vector used to introduce randomness into the encryption process, ensuring that identical plaintexts encrypt to distinct ciphertexts. In CBC mode, the plaintext message M is first padded to a multiple of the AES block size (typically 128 bits), split into blocks $(M_1, M_2, \ldots, M_n)$, and encrypted recursively as $C_0 = I_V, C_1 = \mathrm{AES}_K(M_1 \oplus C_0), C_2 = \mathrm{AES}_K(M_2 \oplus C_1), \ldots C_n = \mathrm{AES}_K(M_n \oplus C_{n-1}), C = C_1 \| C_2 \| \cdots \| C_n$ The I_V is generated securely and is required for decryption alongside K. The encrypted transcript C is then stored on a decentralized storage system, such as Arweave, which ensures permanent, tamper-proof storage [32]. The stored data consists of the ciphertext C concatenated with the initialization vector I_V, formatted as

$$C_{\mathrm{store}} = (I_V \| C) \tag{10}$$

Note that (10) enables authorized entities, such as law-enforcement personnel or other authorities, to retrieve and decrypt the stored transcript for legal investigation. As shown in Fig. 3, our distributed architecture eliminates single points of failure through Shamir's Secret Sharing across N servers. Upon successful storage on Arweave, a corresponding transaction ID X_{AR} is generated. Authorized requesters must obtain shares from at least t servers, reconstruct the encryption key themselves, and fetch the encrypted transcript from Arweave for final decryption. However, the encryption key K is not stored in a single location. Instead, it is first partitioned using SSS scheme and then distributed across multiple independent key share servers (KSS). Each of the 7 distributed KSSs can only access their individual share, and no single server can reconstruct the complete encryption key. The system enforces strict access control policies where the authorized requester must obtain shares from at least 4 servers to reconstruct the key, ensuring that only verified multi-sig approved requests are honored. For an authorized entity to retrieve a stored transcript, they must be part of a multi-sig wallet and obtain approval through the blockchain-based proposal system. The multi-sig wallet regulates access by requiring approval from designated signers, ensuring that only authorized entities can submit a retrieval request through the blockchain. The entity first submits a proposal, an initial request in the multi-sig system that outlines the intended transaction. This proposal contains X_{AR} on its memo and is recorded on a PDA, as shown in Fig. 4. Each distributed key share server continuously listens to the PDA within the blockchain. Once they detect the approved transaction, authorized requesters can obtain encrypted key shares from the KSSs. They can then fetch C_{store} from Arweave after reconstructing the encryption key. Once the requester has collected sufficient shares and reconstructed the key K, they decrypt the encrypted text C to retrieve M, where

$$\begin{aligned} M &= D_{\mathrm{AES}}(C, K, I_V) \\ &= \|_{i=1}^{n} \left[\mathrm{AES}_K^{-1}(C_i) \oplus C_{i-1}\right], \quad \text{with } C_0 = I_V \end{aligned} \tag{11}$$

The decrypted message M is then securely transmitted to the requesting entity, ensuring only authorized parties can access sensitive emergency transcripts.

A. GOOGLE STT FOR TRANSCRIPTION AND GOOGLE TRANSLATION

The emergency response system begins by processing incoming VoIP-based emergency calls and routing them to a PSAP. At this stage, the speech signal from the victim, S_c is passed through the Google STT API to generate a textual representation of the spoken words. The output of this step is the victim's transcript T_c in the victim's native language L_c. $T_c = f_{\mathrm{STT}}(S_c, L_c)$ where T_c represents the extracted text corresponding to the victim's speech. Since dispatchers may operate in a different language L_d, a translation function T is applied using the Google Translation API to convert the transcript into the dispatcher's language $T_{\mathrm{TC}} = T(T_c, L_c, L_d)$ to ensure the dispatcher receiving a comprehensible version of the emergency message. However, due to packet loss, or other transmission errors, portions of T_{TC} may be missing or lost, which is $T_{\mathrm{TC}} = (w_1, w_2, w_3, _, w_5, _, w_7, \ldots, w_N)$

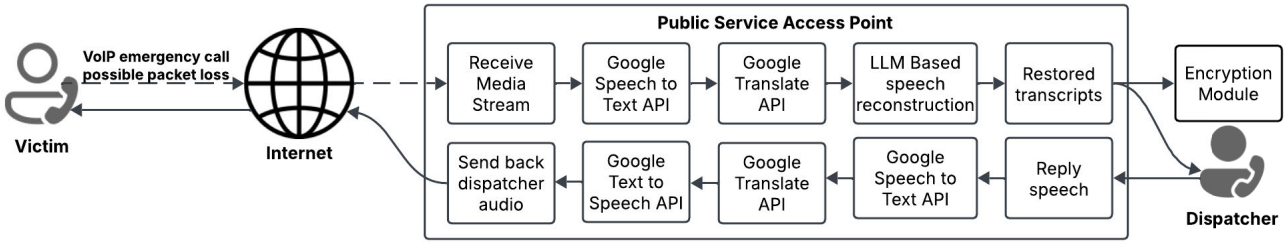


FIGURE 2. LLM speech flow.

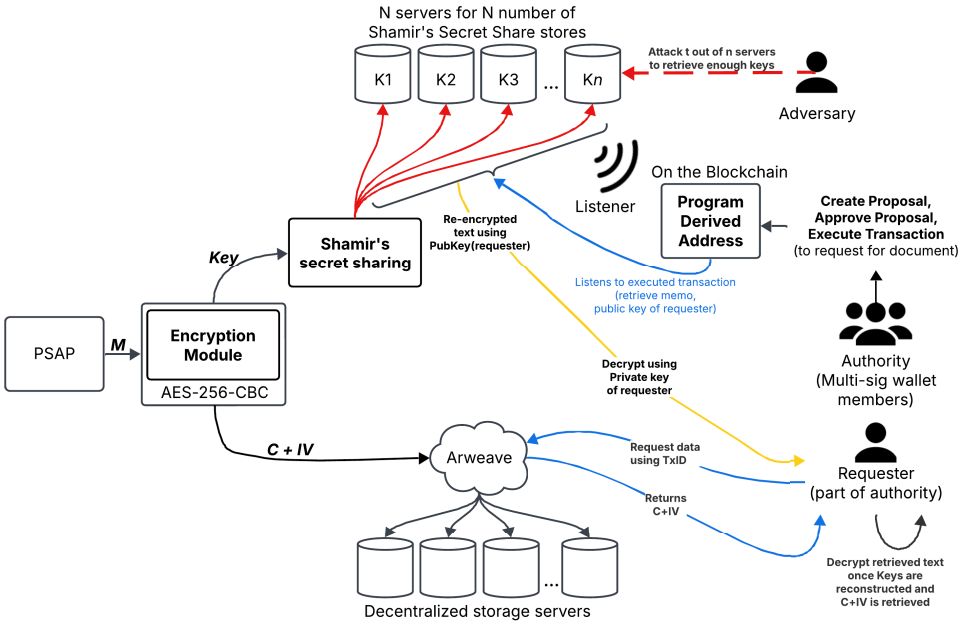


FIGURE 3. Distributed encryption and blockchain-based access control system showing Shamir's Secret Sharing across N independent key share servers (K1…Kn), where each server monitors the blockchain PDA independently, encrypts its share with the requester's public key upon authorization, and the requester performs key reconstruction after collecting t shares.

where w_x. To address these missing segments, the next step would be to pass T_{TC} through an LLM-based speech reconstruction system.

B. LLM FOR SPEECH INFERENCE AND RECONSTRUCTION

After generating the translated transcript T_{TC}, the system addresses potential packet loss or missing segments that can impact comprehension. In this scenario, the transcript is processed to reconstruct the missing parts and refine the sentence structure through an LLM, specifically Gpt-4o-mini.

We selected GPT-4o-mini due to OpenAI's established track record and its cost-efficient operation compared to larger models, making it suitable for sustained emergency service deployment. The model features a 128K token context window [33], sufficient for emergency call reconstruction tasks.

Our implementation employs a contextual few-shot learning approach [34] rather than fine-tuning for several key advantages. Few-shot learning eliminates the computational overhead and time requirements of model retraining, making it more suitable for real-time emergency applications. Additionally, this approach provides flexibility to adapt to diverse emergency scenarios without requiring specialized training data for each emergency type. Furthermore, as demonstrated by researchers in [35], few-shot learning achieves competitive performance across various NLP tasks while maintaining model generalizability. This approach allows the model to learn reconstruction patterns from provided examples without requiring fine-tuning.

To ensure system reliability, we implement a fallback mechanism where if GPT-4o-mini encounters an error or fails to respond, the system automatically switches to Claude 3.5 Haiku as an alternative LLM. This redundancy maintains continuous speech reconstruction capabilities during API outages or model failures, ensuring uninterrupted emergency service operation.

We use a custom FAISS [36] index built upon a database of emergency call transcripts [37] to retrieve relevant context and infer plausible word completions. This approach, combined with conservative generation settings and a focus on gap-filling rather than creative content generation, helps mitigate hallucination risks in emergency contexts. The

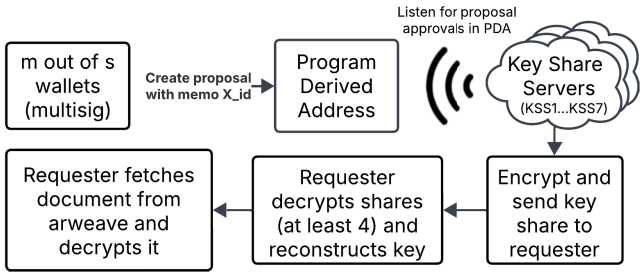


FIGURE 4. Blockchain proposal workflow.

output of the LLM is a refined transcript T_c^* and it is computed as:

$$T_c^* = f_{\text{LLM}}(T_{\text{TC}}, H) = \arg\max_{T'} \prod_{i=1}^{N} P(w_i^* \mid w_1^*, \ldots, w_{i-1}^*, T_{\text{TC}}, H; \theta) \quad (12)$$

where H is the additional contextual information retrieved from the knowledge base using FAISS indexing. w_i^* denotes i^{th} word in the reconstructed output. This reconstruction ensures that dispatchers receive a coherent and complete version of the victim's request. Once reconstructed, T_c^* is sent to the dispatcher for review and response. The dispatcher then responds R_d, which undergoes reverse processing to convert it into a format the victim understands.

C. GOOGLE API FOR TRANSLATION, TRANSCRIPTION, AND TTS

Once the dispatcher responds R_d, it undergoes STT processing, converting the speech into a textual response T_d in the dispatcher's language L_d. Since the victim may not understand L_d, the response transcript undergoes reverse translation as described in Section II-B. To respond to the victim in audio, the translated response is converted back into audio speech using Google's TTS API

$$S_c' = f_{\text{TTS}}(T_d', L_c) \quad (13)$$

where S_c' is the synthesized speech in the victim's native language L_c, which is then transmitted back over the VoIP network. This pipeline ensures that the dispatcher and victim can communicate effectively, regardless of language differences. This automated process significantly saves time compared to traditional manual translation and relay methods, as it eliminates the need for human intervention at every communications step.

D. SECURE TRANSCRIPT ENCRYPTION AND DECENTRALIZED STORAGE

To ensure the confidentiality, integrity, and authenticity of emergency call transcripts, both victim-transcribed T_c^* and dispatcher-transcribed T_d are securely encrypted before storage. The encryption process is based on AES-256-CBC, a widely accepted symmetric encryption standard with high security and minimal computational overhead. Encryption starts by concatenating the reconstructed victim transcript and the dispatcher's response into a single plaintext message. This combined message is then encrypted using AES-256-CBC, which requires a cryptographically secure randomly generated encryption key and an I_V. The I_V ensures that identical plaintexts do not produce the same ciphertext. Once encrypted, the resulting ciphertext C is stored along with the I_V, within an immutable and permanently accessible storage solution known as Arweave.

$$C = \text{E}_{\text{AES}}(M, K, I_V) \quad (14)$$

Once the plaintext message M, composed by concatenating the reconstructed victim transcript T_c^* and the dispatcher's response T_d, is formed, it is encrypted using the AES-256 algorithm in Cipher Block Chaining (CBC) mode to ensure data confidentiality. The message M is first divided into fixed-size blocks $M = M_1 \| M_2 \| \cdots \| M_n$, each 128 bits in length. A 256-bit symmetric encryption key K is used, along with a randomly generated 128-bit Initialization Vector (I_V). The encryption process follows a recursive block-wise formulation: $C_0 = I_V$, $C_1 = \text{AES}_K(M_1 \oplus C_0)$, $C_2 = \text{AES}_K(M_2 \oplus C_1)$, ..., and $C_n = \text{AES}_K(M_n \oplus C_{n-1})$, where $\oplus$ denotes the bitwise XOR operation. This results in a final ciphertext $C = C_1 \| C_2 \| \cdots \| C_n = \text{E}_{\text{AES-CBC}}(M, K, I_V)$. To allow for correct decryption in the future, I_V is stored together with the ciphertext, yielding a secure storage payload defined as $C_{\text{store}} = I_V \| C$. This encrypted object is then uploaded to Arweave—a decentralized, tamper-proof storage network designed for permanent archival. The storage transaction on Arweave produces a unique cryptographic identifier $X_{\text{AR}} = \text{Arweave.Store}(C_{\text{store}})$, which can be used to retrieve and verify the integrity of the encrypted message indefinitely. Arweave's content-addressed architecture, combined with its economic incentivization model, guarantees that C_{store} remains immutable and persistently accessible across time.

E. DECENTRALIZED TRANSCRIPT STORAGE ON ARWEAVE

Arweave is a blockchain-based decentralized storage system designed for permanent data archiving. Unlike traditional cloud storage, where data can be altered or deleted, Arweave ensures that all stored transcripts remain unchanged, verifiable, and permanently accessible. This approach is particularly well-suited for emergency call transcripts, as it enables secure, immutable storage that supports future audits, forensic investigations, and compliance with legal and regulatory requirements. Once the encrypted transcript is uploaded to Arweave, the network generates a unique transaction ID X_{AR}, which acts as an immutable reference to the stored data. This transaction ID allows authorized entities to retrieve the transcript while ensuring that no unauthorized modifications can occur. The content is stored in this format $C_{\text{store}} = (I_V \| C)$. The encryption key must not be stored on Arweave, as it is a public ledger allowing anyone to access stored data. So, the encryption key is securely managed using another channel where we perform SSS and utilize a key-server component to manage these stored keys.

In the rare case that the Solana cluster becomes temporarily unreachable, on-chain PDA lookups used to gate transcript fetches will fail, and no new retrievals can be authorized until Solana connectivity is restored. Archived transcripts, however, remain persistently stored in Arweave and accessible via IPFS gateways, and any Solana-bound authorizations simply remain queued until the chain resumes. Conversely, if the Arweave network experiences a brief disruption, new writes may be temporarily delayed until connectivity returns, upon which clients will automatically retry and re-upload the content. Meanwhile, Solana on-chain state transitions continue without interruption. Because Solana's highly distributed validator network has maintained over 99.9% historical uptime and Arweave leverages a decentralized, deduplicated P2P storage layer, any outage in either system is expected to last only seconds to minutes.

F. SSS FOR SECURE KEY MANAGEMENT

The encryption key is not stored on a single server to eliminate single points of failure. Instead, we use Shamir's Secret Sharing (SSS) to divide the key into $n = 7$ fragments (shares), distributed across independent servers. A threshold of $t = 4$ shares is required for the requester to reconstruct the key, ensuring that no individual server can compromise security. Even if one or two servers are breached, decryption is impossible without collecting four unique shares. This distributed approach improves fault tolerance and security by ensuring system operability even if up to three servers are offline or compromised. It also mitigates insider threats, as no single entity can reconstruct the key alone.

To tolerate partial outages, the client requests shares from all KSS in parallel and aggregates replies until any t valid shares are obtained; extra replies are ignored. Because reconstruction needs only t of n, the system remains live with up to $n-t$ servers down or slow. Unresponsive KSS are retried while aggregation proceeds; if fewer than t arrive, the operation aborts early without side effects. Responses are idempotent and stateless, so duplicates or late messages are harmless (see II-F2 for the full retrieval flow).

Each of the 7 distributed key share servers independently manages their assigned share. The system operates without a centralized coordinator, with servers responding to authorized retrieval requests based on verified multi-sig proposals.

1) MULTI-SIG BASED AUTHORIZATION

To enforce strict and verifiable access control over transcript retrieval, the system implements an m-of-s threshold-based multi-signature (multi-sig) authorization scheme. Let $\mathcal{S} = \{S_1, S_2, \ldots, S_s\}$ represent the set of all authorized signers, such as law enforcement officers or certified regulatory authorities, and let m denote the minimum number of approvals required to validate a retrieval request. Any authorized retrieval transaction R must be digitally signed by at least m distinct members from $\mathcal{S}$, where each signature $\sigma_i = \texttt{Sign}_{K_i}(R)$ is generated using the private key K_i corresponding to signer S_i. The system validates a transaction by computing $\texttt{Authorize}(R, \Sigma)$, which evaluates to 1 (i.e., valid) only if the set of collected signatures Σ contains at least m valid signatures from recognized signers. Once authorized, the signed transaction is submitted to a Program Derived Address (PDA) on the blockchain for logging and audit purposes. The request includes a memo field, $\texttt{memo}(R)$, which embeds the Arweave transaction identifier $X_{\texttt{AR}}$, uniquely linking the blockchain record to the encrypted transcript stored in Arweave. This framework ensures decentralized, tamper-resistant access to sensitive transcripts with transparent on-chain verification.

2) DISTRIBUTED SHARE RETRIEVAL PROCESS

Each distributed KSS monitors the blockchain for incoming retrieval requests through their individual listeners. Since only multi-sig-approved transactions are recorded in the PDA, the listener ensures that retrieval operations are legitimate and authorized. Upon detecting a valid retrieval request, the authorized servers perform the following steps in parallel:

1) Each server independently verifies the requester's signature and checks blockchain approval
2) Each server encrypts its share with the requester's public key: $S_{r,i} = E(S_i, K_{\texttt{pub,r}})$ and returns it if authorized
3) Requester collects shares from at least 4 servers in parallel
4) Requester decrypts the shares using their private key
5) Requester reconstructs the encryption key using SSS
6) Requester fetches encrypted transcript from Arweave using $X_{\texttt{AR}}$
7) Requester decrypts the transcript: $M = D_{\texttt{AES}}(C, K, I_V)$

Share encryption is performed by each server using the public key of the requester:

$$S_r = E(S_i, K_{\texttt{pub,r}}) \tag{15}$$

where S_r is the encrypted share, S_i is the individual key share, and $K_{\texttt{pub,r}}$ is the requester's public key. This ensures only the intended requester can access the share.

The system ensures that only the intended recipient with the corresponding private key can decrypt and access the document by encrypting the transcript before transmission. This additional encryption layer prevents unauthorized intermediaries from intercepting the sensitive transcript during transmission. Since the decryption key is unique to the requester, the content remains secure even if the transmission is exposed to a third party. Upon receiving the encrypted transcript, the requester must use their private key to decrypt and access the document.

$$M = D(C_r, K_{\texttt{pvt,r}}) \tag{16}$$

where $K_{\texttt{pvt,r}}$ is the private key of requester and the function $D()$ is used to decrypt the encrypted message C_r. This ensures that only authorized personnel gain access to emergency call records.

3) LOW-LATENCY COMMUNICATIONS PIPELINE

To meet the requirements of application-layer low-latency communication, the system is designed such that the total round-trip latency for emergency interaction remains within a tight upper bound. Let the total communications latency be denoted by:

$$\Delta T_{\text{total}} = \Delta T_{\text{STT}} + \Delta T_{\text{LLM}} + \Delta T_{\text{Trans}} + \Delta T_{\text{TTS}} + \Delta T_{\text{Net}} \tag{17}$$

where ΔT_{STT} is the time taken for converting speech to text. ΔT_{LLM} is the duration for transcript reconstruction via the LLM, ΔT_{TTS} denotes the time required for synthesizing the translated response, ΔT_{Net} includes inbound and outbound communications overhead. ΔT_{Trans} represents the time required to perform machine translation of a transcript T from the source language to the target language. To reach our low-latency standards, we enforce the latency constraint:

$$\Delta T_{\text{total}} \leq \tau_{\text{low-latency}}, \quad \text{where } \tau_{\text{low-latency}} \leq 1.5 \text{ seconds} \tag{18}$$

In practice, each component in the pipeline is engineered and optimized such that: $\Delta T_{\text{STT}} \leq 300$ ms, $\Delta T_{\text{LLM}} \leq 400$ ms, $\Delta T_{\text{Translate}} \leq 200$ ms, $\Delta T_{\text{TTS}} \leq 300$ ms, $\Delta T_{\text{Network}} \leq 300$ ms, which yielding a typical total latency $\Delta T_{\text{total}} \in [800\text{ ms}, 1500\text{ ms}]$, which satisfies the low-latency bound. The use of stateless cloud APIs and indexed retrieval from the FAISS-based LLM context module further reduces ΔT_{LLM}, while blockchain and Arweave operations execute asynchronously to prevent blocking critical paths. Therefore, the system guarantees both low-latency and high-availability communications with cryptographic integrity.

G. END-TO-END TESTING UNDER REALISTIC VOIP CONDITIONS

The entire end-to-end system was deployed and evaluated in a real-time VoIP testbed using Twilio's Programmable Voice API [38], with simulated network conditions configured via NetEm [39]. All pipeline components—including automated speech recognition, multilingual neural machine translation, LLM-based packet loss restoration, text-to-speech synthesis, and blockchain-secured archival—were fully integrated and tested with live audio streams. Network impairments were emulated to match real-world scenarios, as recommended in VoIP system evaluations [40], [41], including burst and random packet loss (1% and 5%), jitter (up to 150 ms), and variable bandwidth. These settings are consistent with performance benchmarks in recent emergency communication research [42], [43], [44]. Key performance metrics (latency, ROUGE, Conceptual Precision) were measured under each condition. A limitation of the current work is that testing was performed in an emulated operational environment rather than live field deployment (e.g., in a public safety answering point). Future work will address this through pilot deployments and user studies with emergency services partners. While our component-level timing measurements provide a breakdown of processing delays, production end-to-end latency may vary with network conditions, concurrent call load, and hardware variability.

III. PROPOSED SOLUTION

The proposed system for emergency services is designed to facilitate low-latency transcription, translation, speech reconstruction, emergency classification, and secure storage of emergency communications transcripts while ensuring data integrity and auditability. The architecture is structured into modular components, each serving a critical function in processing and securing emergency VoIP calls.

A. SPEECH PROCESSING AND RECONSTRUCTION PIPELINE

The speech-processing pipeline forms the core of emergency communications handling, allowing seamless interaction between the victim and the dispatcher. The pipeline includes transcription, translation, and TTS synthesis to ensure smooth language communications.

1) STT TRANSCRIPTION

When victims initiate an emergency VoIP call, their speech signal S_c is captured and transcribed into text using an STT module. Given the victim's spoken language L_c, the STT model generates a transcript T_c

$$T_c = f_{\text{STT}}(S_c, L_c) \tag{19}$$

For this, we use Google Speech API, which utilizes its STT module, which inputs the victim's voice as a stream of audio and converts it into text in low-latency. This speech may or may not be incomplete, depending on network conditions and other factors, such as packet loss. Upon transcription, we receive a sequence of words as text, which may look like this $T_{\text{TC}} = (w_1, w_2, w_3, _, w_5, _, w_7, \ldots, w_N)$. The gaps in the sentences represent words that the API could not transcribe.

2) TRANSLATION MODULE

To facilitate communications, the transcribed text is converted from the victim's language L_c to the dispatcher's preferred language L_d.

$$T_{\text{TC}} = T(T_c, L_c, L_d) \tag{20}$$

To achieve this, we use Google's Cloud Translation API, which inputs plaintext of any language and converts it into English for the dispatcher to be able to interpret later. This would still be incomplete data once we get the transcripts in the desired language. So, we need to perform a reconstruction mechanism to complete the sentence. This will be performed using the LLM-based reconstruction mechanism.

Out-of-Distribution Language Handling: While Google Translate supports over 100 languages with automatic detection, our current system does not implement explicit handling for out-of-distribution languages or low-confidence

detection scenarios. In cases where language detection fails or confidence scores are insufficient, the system would require manual dispatcher intervention to coordinate with professional interpretation services or handle audio-only transmission until appropriate linguistic support can be arranged.

B. LLM-BASED RECONSTRUCTION

Since these emergency calls may suffer from transmission issues, background noise, or packet loss, leading to incomplete or unintelligible transcripts. To address this, we employ a LLM-based reconstruction mechanism to restore missing words and improve comprehension. Once translated to the desired language, we employ an LLM-based model to reconstruct and fill in the missing sequence of words we had from the translated call transcripts. The input to the LLM is the translated but incomplete transcript, which is $(w_1, w_2, w_3, _, w_5, _, w_7, \ldots, w_N)$. After reconstruction, we will obtain a transcript filled with relevant context such that we have $(w_1, w_2, w_3, w_4, w_5, w_6, w_7, \ldots, w_N)$ where there would be no missing words.

Since accurate reconstruction requires relevant contextual information, the LLM is provided with additional context H during processing. In this work, H comprises historical emergency call transcripts retrieved from a FAISS-indexed knowledge base. This knowledge base is constructed using a curated dataset of 518 emergency call transcripts from the SpikeCodes repository on Hugging Face [37]. These transcripts span various emergency scenarios such as medical incidents, fires, and property crimes, thereby equipping the LLM with diverse contexts to accurately infer and reconstruct missing segments in incomplete transcripts. The LLM performs the following function to obtain the complete context T_c^* like

$$T_c^* = f_{\text{LLM}}(T_{\text{TC}}, H) \tag{21}$$

The model we are utilizing here is OpenAI's GPT-4o-mini model. Although, to actually predict the missing context, the model needs relevant context. A dataset containing previous emergency call transcripts is used to provide appropriate context, which was acquired from huggingface.

C. REVERSE TRANSLATION AND TEXT-TO-SPEECH MODULE

To communicate back to the victim in their native language L_c, the dispatcher's speech is translated and synthesized into speech. This is achieved using a reverse translation and `TTS` module following a three-step process. Once the dispatcher responds with R_d in their language L_d, the speech is transcribed into text using Google's `STT` module. The system captures the spoken response in low-latency, converting it into a text transcript T_d

$$T_d = f_{\text{STT}}(R_d) \tag{22}$$

Since the victim might not understand L_d, the text transcript is translated back to the victim's language L_c. This is done using Google Cloud Translation API, ensuring an accurate and context-aware translation. The translation function is defined as

$$T_d' = T(T_d, L_d, L_c) \tag{23}$$

The translated text T_d' is converted back into speech using Google's `TTS` Module, which can convert text into speech of over 50 languages and variants. The audio is transmitted back to the victim via the VoIP network.

D. TRANSCRIPT ENCRYPTION AND STORAGE MODULE

Once the emergency transcript has been processed, reconstructed, and verified, it is securely stored to prevent unauthorized access while maintaining availability for authorized retrieval. The system employs `AES-256-CBC` encryption to secure the transcript, followed by decentralized storage on Arweave, ensuring permanent, tamper-proof access. Encryption is performed on the reconstructed victim transcript T_c^* and the dispatcher's response T_d. These components are concatenated into a single plaintext message $M = (T_c^* \| T_d)$. To prevent any adversaries from attacking during transmission and storage, we employ `AES-256-CBC` encryption, requiring a cryptographically secure encryption key K and I_V. We express as follows:

$$C = \text{E}_{\text{AES}}(M, K, I_V) \tag{24}$$

The key K is generated using a Cryptographically Secure Pseudo-Random Generator within the `crypto` module on Node.js, ensuring high entropy and unpredictability. The I_V is also generated securely using the same cryptographic module. This ensures that identical transcripts encrypted with the same key produce different ciphertexts, preventing pattern analysis attacks.

Arweave-based Storage Module: The encrypted transcript C is uploaded to Arweave, a decentralized blockchain-based storage network, to ensure permanent, immutable storage. Unlike traditional cloud storage, where data can be altered or deleted, Arweave provides tamper-proof data persistence, making it an ideal solution for emergency communications logs that require long-term integrity and auditability. The underlying technology behind Arweave is the Blockweave structure, which differs from conventional blockchains. Instead of storing transactions linearly in a single chain, Blockweave enforces a mechanism where each new block references multiple previous blocks, significantly improving redundancy and data retrieval efficiency. This ensures that the retrieval of stored data is computationally lightweight, even in a decentralized environment. For testing and development, we used arLocal, a local, lightweight replica of the Arweave mainnet. ArLocal provides a simulated Arweave environment where we could store, retrieve, and interact with data without incurring blockchain transaction costs. This simulated environment allows for rapid iterations during development, making it an effective tool for validating our storage pipeline before transitioning to the production Arweave network. Once encrypted, the stored transcript is

retrievable by authorized entities while remaining tamper-proof. To achieve this, the encrypted transcript is uploaded to Arweave in the format:

$$C_{\texttt{store}} = (I_V \| C) \tag{25}$$

Upon successful storage, Arweave generates a unique transaction ID $X_{\texttt{AR}}$, which serves as an immutable reference to the stored data. To retrieve the encrypted transcript, an authorized entity must possess the transaction ID $X_{\texttt{AR}}$. Since Arweave's design ensures permanent storage, data retrieval remains efficient even in decentralized conditions. Any stored transcript can be fetched using

$$C_{\texttt{store}} = \texttt{Fetch}(X_{\texttt{AR}}) \tag{26}$$

Since Arweave is a public ledger, storing encryption keys directly on the network is infeasible. Instead, SSS handles key management separately, ensuring no single compromised entity can reconstruct the decryption key. The transcript retrieval is regulated through a multi-sig access control mechanism, which ensures that only authorized personnel can request transcript decryption.

SSS *and Storage in N servers:* Since the encryption key K used for AES-256-CBC cannot be stored directly on Arweave due to its public nature, we employ SSS to distribute the key among multiple independent storage securely. SSS is a cryptographic technique that splits a secret into multiple shares such that only a subset of these shares is required to reconstruct the original secret. To ensure fault tolerance and prevent unauthorized access, the encryption key K is split into n shares, where at least t shares (out of n) are required for reconstruction. This follows a (t, n)threshold scheme, meaning that any t or more shares can reconstruct the key, but fewer than t shares reveal nothing about the original key. Each share is created using a polynomial-based approach, where K is encoded as the constant term of a randomly generated polynomial:

$$f(x) = K + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} \mod p \tag{27}$$

where p is a large prime number for security, and $a_1, a_2, \ldots, a_{t-1}$ are randomly generated coefficients. Each key share server is assigned a unique identifier x_i, and stores a share $(x_i, f(x_i))$.

To limit key exposure and strengthen long-term security, the system employs an epoch-based rotation scheme. Each epoch E (UTC YYYY-MM) receives a freshly generated 256-bit master key, which is split using a (4-of-7) Shamir's Secret Sharing scheme and distributed to independent Key Share Servers (KSS). Rotation proceeds only after a quorum of t=4 KSS confirm secure storage. During retrieval, the original epoch E is used to obtain matching shares without altering the client interface or impacting the critical path. Keys from prior epochs are retained for a configurable period (e.g., 12 months) to preserve access to historical data, after which each KSS independently prunes its expired shares. This approach ensures past transcripts remain decryptable while limiting the lifespan of any single key. Rotation cadence, retention limits, and operational procedures are deployment-specific and left to future work.

E. ACCESS CONTROL & KEY MANAGEMENT

1) BLOCKCHAIN-BASED PROGRAM DERIVED ADDRESS

A PDA is a key component of the Solana blockchain. It securely tracks access requests and ensures controlled interactions with encrypted transcripts. Unlike regular wallet addresses, PDAs are deterministically generated by on-chain programs and cannot be controlled with a private key, making them ideal for managing sensitive access operations. The PDA acts as an on-chain ledger for retrieval requests in our system, ensuring that only multi-sig-approved transactions are recorded. When an authorized entity submits a retrieval request, the request includes a memo that contains the Arweave transaction ID $X_{\texttt{AR}}$. This uniquely references the stored transcript such that the requester can later interpret and retrieve the transcript according to the transaction ID. The PDA ensures that only verified members of the multi-sig wallet can write to it, preventing unauthorized access attempts.

Since the PDA acts as an immutable log, each distributed key share server listens for changes in the PDA state, continuously monitoring for new retrieval requests. Upon detecting a valid transaction, the servers prepare to respond to authorized share requests. The requester extracts the $X_{\texttt{AR}}$ from the approved proposal to later retrieve the encrypted transcript from Arweave.

2) MULTI-SIG ACCESS CONTROL

The system implements a multi-signature (multi-sig) access control model to regulate transcript retrieval, ensuring that no single entity can unilaterally access stored emergency records. Instead, retrieval requests must go through a structured approval process, requiring validation from a threshold number of designated signers before being finalized. The system follows a three-step multi-sig transaction flow for processing a retrieval request. First, an authorized member of the multi-sig wallet initiates a retrieval request by submitting a proposal. This proposal includes the Arweave Transaction ID $X_{\texttt{AR}}$ corresponding to the encrypted transcript. Once the proposal is created, other members of the multi-sig group review the request and provide their approvals. The request is only valid if at least m out of s signers approve it, ensuring the transaction has sufficient authorization before execution. Once the approval threshold is met, the final step is executing the transaction, which records the retrieval request on the blockchain via the PDA. This structured process ensures that access to sensitive transcripts is controlled and auditable. The multi-sig wallet also serves as a membership management mechanism, allowing only designated administrators or high-ranking roles to add or remove members. Adding a wallet to the multi-sig member list enables that entity to approve retrieval requests and participate in access control

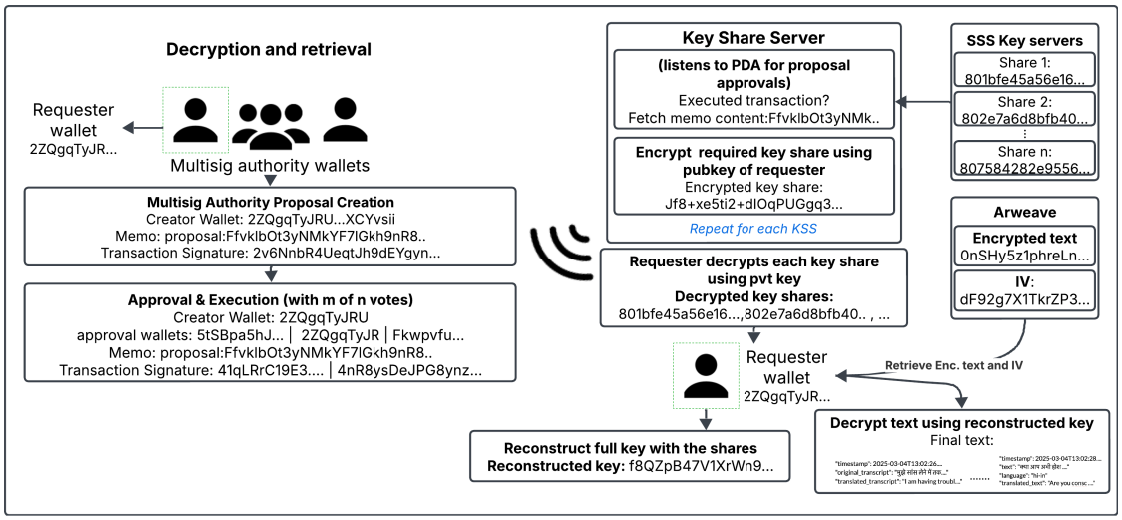


FIGURE 5. Transcript decryption and retrieval workflow: Authorized entities initiate retrieval via Solana PDAs and multisig approvals, followed by Shamir's Secret Sharing key reconstruction, key-share encryption with the requester's public key, and final requester-side decryption.

governance. To facilitate the implementation of this multi-sig system, we utilize sqds/multi-sig. This open-source and well-established Solana-based multi-sig solution enables secure, decentralized transaction approval while maintaining role-based permissions.

F. DISTRIBUTED KEY MANAGEMENT

The distributed key management system eliminates single points of failure by having each of the 7 servers independently manage their assigned shares. No centralized coordination is required, as each server operates autonomously while maintaining the 4-of-7 threshold for key reconstruction. Each KSS is responsible for processing access requests and encrypting its share using the requester's public key before securely transmitting the data.

Each KSS continuously monitors the blockchain for retrieval requests recorded on the PDA to ensure that only authorized entities can request decryption. Since only members of the multi-sig wallet with sufficient approvals can write to the PDA, any retrieval request recorded on-chain is inherently verified. Once the distributed servers validate a request through blockchain verification, the KSS extracts the Arweave transaction ID X_{AR} from the approved proposal memo and prepares its key share $(x_i, f(x_i))$. This share is then encrypted using the public key of the requester $K_{\mathrm{pub,r}}$ and transmitted securely in the encrypted form:

$$S_{r,i} = E((x_i, f(x_i)), K_{\mathrm{pub,r}}) \tag{28}$$

G. KEY RECONSTRUCTION AND TRANSCRIPT DECRYPTION BY REQUESTER

To decrypt the transcript, the requester must first reconstruct the encryption key K. This requires obtaining encrypted shares from at least $t = 4$ distributed servers. Upon receiving these encrypted shares, the requester decrypts each share using their private key $K_{\mathrm{pvt,r}}$:

$$S_i = D(S_{r,i}, K_{\mathrm{pvt,r}}) \quad \text{for each encrypted share} \tag{29}$$

Once at least t valid decrypted shares $(x_i, f(x_i))$ are collected, the requester reconstructs the original encryption key K, which is derived as follows:

$$K = \sum_{i=1}^{t} f(x_i) \cdot \prod_{j \neq i} \frac{x_j}{x_j - x_i} \mod p \tag{30}$$

This key reconstruction leverages Shamir's Secret Sharing with Lagrange interpolation [45], where each share $f(x_i)$ was originally computed from the polynomial:

$$f(x) = K + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} \mod p \tag{31}$$

After reconstructing the encryption key, the requester fetches the encrypted transcript from Arweave using X_{AR}, retrieving $C_{\mathrm{store}} = (I_V \| C)$. With the reconstructed key K, the requester decrypts the transcript as $M = D_{\mathrm{AES}}(C, K, I_V)$, where M is the plaintext emergency transcript.

This process ensures that only the intended recipient, holding the corresponding private key, can access the sensitive emergency communications records. The use of asymmetric encryption for share protection provides end-to-end security, preventing unauthorized access even if transmissions are intercepted. Moreover, every step of this retrieval and decryption process is recorded on-chain, providing an auditable trail essential for compliance and forensic purposes.

Once decrypted, the transcript can be used for its intended purpose, such as forensic investigations, legal proceedings, compliance audits, or emergency response analysis. Since all retrieval requests pass through a strict multi-sig approval system, the decryption process remains tightly controlled and auditable. Every transaction related to decryption is logged

TABLE 3. STRIDE threat analysis for emergency transcript system.

STRIDE Category	Threat Description	Mitigation Strategy	Risk Level
Spoofing	Attacker impersonates KSS	TLS mutual authentication	Low
Tampering	Modify stored transcripts	Arweave immutability	Negligible
Repudiation	Deny making access request	Blockchain audit log	Low
Information Disclosure	Steal encryption keys	SSS threshold (4 of 7)	Low
Denial of Service	Disable KSS	Distributed redundancy	Medium
Elevation of Privilege	Bypass access controls	Multi-sig requirement	Low

on-chain, ensuring a clear record of access for security and regulatory compliance.

The detailed workflow demonstrating how authorized transcript retrieval, encryption, key reconstruction, and final requester-side transcript decryption occur is depicted in Fig. 5.

H. SECURITY ANALYSIS AND THREAT MODEL

We analyze our system's security using the STRIDE threat modeling methodology [46], which systematically evaluates six categories: Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege. Table 3 presents our threat analysis and corresponding mitigations. As shown in Table 3, our distributed architecture addresses each STRIDE threat category. The most significant residual risk is Denial of Service, which could impact system availability if more than 3 KSSs are simultaneously attacked. However, geographic distribution of servers and rate limiting make such coordinated attacks impractical. The SSS threshold scheme ensures that Information Disclosure threats require compromising at least 4 servers simultaneously, providing computational security equivalent to breaking AES-256 encryption [14].

The probability of successfully compromising our system requiring 4 or more servers is:

$$P(\text{compromise}) = \sum_{k=4}^{7} \binom{7}{k} p^k (1-p)^{7-k} \tag{32}$$

where p is the probability of compromising a single server. With standard security practices ($p < 0.01$), this yields $P(\text{compromise}) < 3.5 \times 10^{-7}$.

1) BLOCKCHAIN ATTACK RESISTANCE

Our system leverages Solana's Proof-of-History (PoH) and Proof-of-Stake (PoS) consensus mechanism, which requires a Byzantine Fault Tolerant (BFT) supermajority of $\geq$66% of staked tokens to compromise the network. With approximately 390 million SOL staked at current market prices (\$140 per SOL), an attacker would need to acquire 257.4 million SOL, costing over \$36 billion—and this assumes such volume could be purchased without significant market impact, which is highly unrealistic. Even if the blockchain were temporarily compromised, existing encrypted transcripts on Arweave's decentralized permanent storage network remain immutable and accessible through our distributed storage mechanism once it is back live.

2) SIDE-CHANNEL ATTACK MITIGATION

Our distributed architecture inherently limits side-channel vulnerabilities:

- **Distributed key storage**: With Shamir's Secret Sharing across 7 servers, an attacker observing operations on one server gains no information about the complete key
- **Separate encryption endpoint**: Encryption occurs at the PSAP after call completion, isolated from the real-time call processing
- **Limited key operations**: Keys are only reconstructed during authorized retrieval requests, minimizing exposure

While our current implementation does not include specific side-channel countermeasures, the distributed nature of the system and separation of concerns between call processing and storage reduces attack surface compared to centralized architectures.

I. LOW-LATENCY COMMUNICATION AS A SOLUTION STRATEGY

Let the total communications pathway be modeled as a composite function $\mathcal{C} : \mathcal{S}_{\text{in}} \to \mathcal{S}_{\text{out}}$, where $\mathcal{S}_{\text{in}}$ is the raw speech input space and $\mathcal{S}_{\text{out}}$ is the final synthesized response space. The system is subject to the low-latency constraint:

$$\forall s \in \mathcal{S}_{\text{in}},\ \Delta T_{\text{total}}(\mathcal{C}(s)) \leq \tau_{\text{low-latency}} \in \mathbb{R}^+, \tag{33}$$

To satisfy this constraint, we define the total processing pipeline as a summation of bounded latency functions:

$$\begin{aligned}\Delta T_{\text{total}} &= \sum_{i=1}^{n} \Delta T_i \\ &= \Delta T_{\text{STT}} + \Delta T_{\text{LLM}} + \Delta T_{\text{Trans}} + \Delta T_{\text{TTS}} + \Delta T_{\text{Net}}\end{aligned} \tag{34}$$

The system minimizes ΔT_{LLM} by introducing a FAISS-based similarity search $f_{\text{FAISS}} : \mathcal{T}_{\text{incomplete}} \to \mathcal{H}_k$, where $\mathcal{H}_k$ is the top-k context set of semantically nearest historical transcripts, thus constraining the LLM's context window and reducing computational complexity: $\Delta T_{\text{LLM}} \propto \mathcal{O}(\text{len}(\mathcal{H}_k)) \ll \mathcal{O}(\text{len}(\mathcal{H}_{\text{full}}))$ Moreover, blockchain-based storage and key verification functions (e.g., PDA validation and SSS key reconstruction) are executed asynchronously in parallel with the communication-critical path and are thus excluded from ΔT_{total}. Therefore, the system's architecture satisfies: $\Delta T_{\text{comm-path}} \leq \tau_{\text{low-latency}}$, with $\Delta T_{\text{security-ops}} \in \mathcal{O}_{\text{async}}$

J. REGULATORY COMPLIANCE AND ON-PREMISES DEPLOYMENT

Emergency communication systems must comply with stringent data protection and operational regulations. In the European Union, GDPR [47] restricts cross-border data transfers and requires explicit lawful basis for processing sensitive data. In North America, while NENA standards [48] primarily address technical interoperability and security best practices, individual states may impose data residency

requirements for emergency communications. Additionally, many countries enforce data localization laws that prohibit sensitive government data, including emergency communications, from being processed on foreign servers [49]. To address these regulatory requirements, our system architecture is designed to support full on-premises deployment. Speech transcription can utilize OpenAI's Whisper model [50], [51], which achieves comparable accuracy to cloud APIs when deployed on modern GPU infrastructure. Translation capabilities can be provided by open-source neural machine translation models such as Meta's NLLB [52] or Helsinki-NLP's Opus models [53], supporting 200+ and 100+ languages respectively. LLM functionality can be delivered through locally-hosted models such as Llama 3 [54] or Mistral [55], which approach GPT-4o-mini performance with appropriate hardware acceleration. For distributed storage, private IPFS networks or enterprise blockchain solutions like Hyperledger Fabric [56], [57] can maintain immutability and distribution benefits while keeping data within organizational boundaries.

While on-premises deployment requires initial infrastructure investment, modern hardware capabilities make this increasingly feasible. A single high-performance server with enterprise GPUs can maintain our sub-1.5 second latency targets while processing multiple concurrent calls. This approach offers additional benefits: complete data control, reduced operational costs after initial investment, and elimination of internet connectivity dependencies. Our security architecture featuring AES-256 encryption, Shamir's Secret Sharing, and blockchain-based access control remains fully functional in on-premises deployments, providing the same cryptographic guarantees while satisfying the most stringent data residency requirements. This flexibility positions our system as a compliance-ready foundation that organizations can adapt to their specific regulatory environments.

K. ETHICAL AND PRIVACY CONSIDERATIONS

Emergency call transcripts often contain personally identifiable or sensitive health-related data. To address this, we apply AES-256-CBC encryption and Shamir's Secret Sharing for key management, preventing unauthorized access and enabling secure reconstruction only by verified entities. All transcript access is controlled via Solana-based multi-signature proposals and logged immutably on-chain, preserving legal traceability and compliance with global data protection laws such as GDPR, HIPAA, and NENA [58], [59].

Additionally, while severity classification is performed in real-time using LLMs, all triage decisions are advisory and retain human oversight to prevent automation bias [27]. While outputs from generative models can reflect potential biases from their training data, grounding responses with context retrieved from our FAISS-indexed knowledge base helps reduce such skew. Future work includes formal ethical audits, accent and language-balanced evaluation, dispatcher-facing low-confidence flags, and fairness metrics to further limit bias.

A limitation of the present experimental setup is the exclusive focus on random packet loss during evaluation. While random packet loss is a common benchmark in VoIP studies [60], [61], real-world networks often experience burst (correlated) loss, which can disproportionately impact perceived speech quality [41], [62]. We recommend that future studies incorporate burst packet loss models, such as Gilbert-Elliott [63], [64], to more rigorously evaluate emergency VoIP robustness and LLM-based speech reconstruction.

IV. RESULTS AND ANALYSIS

In this section, we present our findings on the proposed LLM-based emergency response system, highlighting its performance in speech reconstruction, severity classification, and security mechanisms grounded in confidentiality, integrity, and availability (CIA). Our evaluation used real audio calls simulated through Twilio's Voice API [65]. The experiments also employ a blockchain-backed storage approach to ensure secure storage and retrieval.

A. DATASET UTILIZATION AND EXPERIMENTAL PARAMETERS

As mentioned previously, our experimental setup utilizes a corpus of 518 emergency call transcripts sourced from SpikeCodes on Hugging Face [37]. The dataset spans various emergency scenarios, such as medical incidents, property crimes, and fires. Each transcript underwent pre-processing operations including tokenization, normalization, and cleaning, to eliminate non-informative artifacts that could degrade model performance.

B. LLM-BASED SPEECH RECONSTRUCTION PERFORMANCE

Emergency calls are first transcribed in real time; however, the initial transcription is fragmented or incomplete due to background noise, poor signal quality, and victim distress. To address these issues, the system leverages a retrieval-augmented LLM-based reconstruction mechanism that uses contextual information retrieved from an FAISS-indexed database of past emergency calls. This process refines the initial transcript by inferring and filling in missing segments, generating a coherent and contextually enriched version. Performance was assessed using BLEU, ROUGE, and Conceptual Precision metrics. As summarized in Table 4, the low BLEU scores indicate that the system does not merely duplicate text and conceptual Precision reflects human assessment of emergency intent preservation. Annotator agreement measured by Fleiss' $\kappa = 0.83$. The high ROUGE scores demonstrate that essential emergency details are effectively retained. The perfect Conceptual Precision (100%), determined by human annotators confirms that the reconstructed transcripts effectively capture the victim's intent.

In addition to English, we exercised the full pipeline on Hindi and Malayalam calls. Despite the typological differences (Indo-Aryan vs. Dravidian), the FAISS-augmented

TABLE 4. BLEU, ROUGE, and conceptual precision.

Test Case	BLEU Score	ROUGE Score	Conceptual Precision (%)
Breathing problems	0.0211	0.5333	100
Unable to find my dog	0.1934	0.6316	100
Smoke in an apartment	0.0654	0.6087	100
Car accident	0.0170	0.3750	100
Stolen wallet	0.0955	0.5714	100
Neighbors dog noisy	0.0170	0.3333	100
Report bleeding man	0.1136	0.8571	100

LLM produced coherent reconstructions that preserved incident intent, and observed component latencies were comparable to the English runs. Minor issues included occasional inflectional mismatches and transliteration artifacts. A broader multilingual evaluation is left to future work.

C. SYSTEM SCALABILITY AND DEPLOYMENT READINESS

The proposed system is architected with modular, stateless components using microservice principles, enabling scalable deployment across distributed emergency communication infrastructure. Containerization (e.g., Docker) and orchestration (e.g., Kubernetes) facilitate elastic scaling in response to call volume, making the framework suitable for national-level PSAP networks. The speech reconstruction module supports plug-and-play LLM backends, and the FAISS context database is incrementally extensible, allowing support for new languages without retraining. In terms of fault tolerance, the system supports fallback to secondary LLMs (Claude 3.5 Haiku) and includes API switching mechanisms if STT/TTS services fail. For real-time adaptability, we model network variances such as packet loss, jitter, and degraded audio as runtime parameters, and reconstruction latency remains within bounds due to optimized retrieval augmentation.

Unless deployed with an on-premise LLM, scalability is largely determined by the throughput limits of the chosen cloud provider. Most commercial APIs support running multiple model instances in parallel, enabling near-linear scaling with demand. This makes it feasible to handle substantially higher concurrent call volumes without architectural changes, subject to API concurrency quotas. While real-world VoIP integration has been simulated via Twilio APIs, future iterations will focus on field trials in collaboration with emergency response centers to empirically evaluate performance under operational load and network variance.

D. SEVERITY CLASSIFICATION PERFORMANCE

Fig. 6 illustrates the model's accuracy in predicting overall call severity as context increases throughout an emergency call. Initially, using only the first sentence, the model achieves 67% accuracy. With additional contextual information, accuracy steadily improves, reaching 100% accuracy by the fourth sentence. This demonstrates the importance of leveraging cumulative contextual information to enhance the reliability and precision of severity predictions, thereby enabling dispatchers to make more informed and timely response decisions.

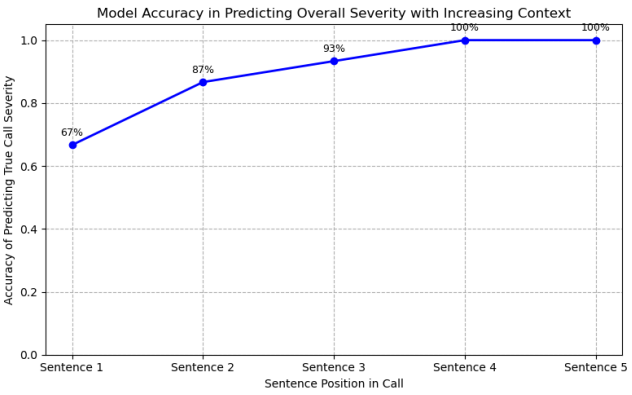


FIGURE 6. Accuracy of LLM-based severity classification with increasing context.

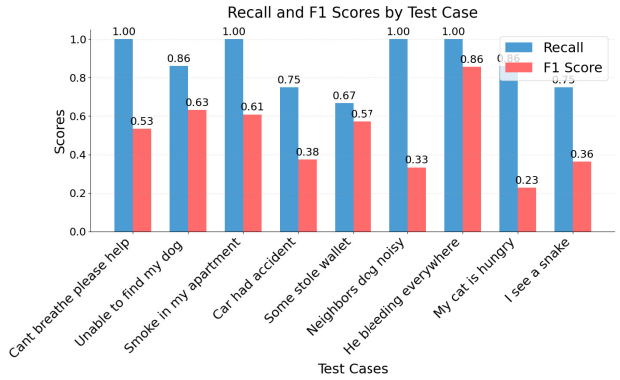


FIGURE 7. Comparison of recall and F1 scores.

E. RECALL AND F1 SCORE PERFORMANCE

To further evaluate the model's ability to capture critical emergency details, we examined recall and F1 scores across multiple test cases. As depicted in Fig. 7, the system achieves a 100% recall rate in high-priority scenarios such as "Can't breathe, please help," "Smoke in my apartment," and "He bleeding everywhere." However, F1 scores vary across different scenarios; lower F1 scores in less urgent cases (e.g., 0.23 for "My cat is hungry" and 0.33 for "Neighbors dog noisy") suggest occasional over-triggering or ambiguity in classification, thereby indicating potential areas for improving precision.

To capture realistic variation in system performance, we conducted controlled experiments under synthetic network impairments. Packet loss was injected using Twilio's programmable voice API and NetEm with rates of 1% and 5%. The LLM inference module showed increased variance under high jitter, while Google STT performance degraded gracefully. The 95th percentile end-to-end latency remained under 1.8 seconds for 1% loss and 2.3 seconds for 5% loss. These results suggest the architecture remains responsive under moderate packet degradation, though field deployment with real network traces remains a critical next step.

Fig. 8 depicts example outputs generated at different emergency call processing system stages. The victim's

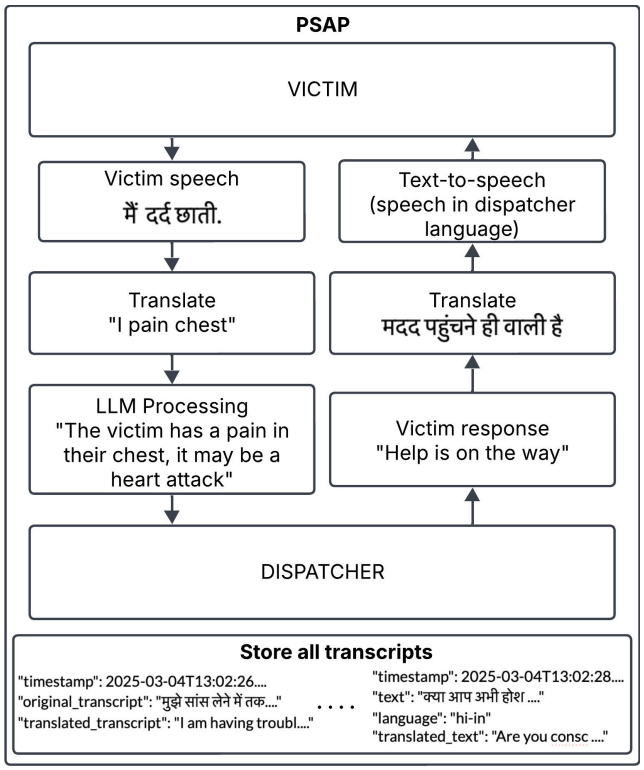


FIGURE 8. Example outputs at different stages of the emergency call processing system.

TABLE 5. Effect of FAISS context depth on reconstruction accuracy.

FAISS k	ROUGE	Conceptual Precision (%)
0	0.41	72
1	0.53	88
3	0.60	94
5	0.63	100
10	0.64	100

speech, initially in a non-English language, is transcribed and translated into English. The LLM refines fragmented or incomplete speech into a structured emergency statement, ensuring clarity for dispatchers. The dispatcher-generated response is then translated back into the victim's language for communications. The system also maintains stored transcripts, preserving original and translated versions for future reference.

F. ABLATION STUDY: IMPACT OF FAISS RETRIEVAL DEPTH

To evaluate the contribution of the FAISS-based context retrieval module, we conducted an ablation study by varying the top-k retrieval depth for LLM reconstruction. Table 5 summarizes the results across $k = 0, 1, 3, 5, 10$. We observed substantial improvements in both ROUGE and Conceptual Precision scores when context was retrieved ($k \geq 1$), with diminishing returns beyond $k = 5$. This confirms the benefit of context-aware reconstruction while validating the efficiency of shallow indexing for low-latency performance.

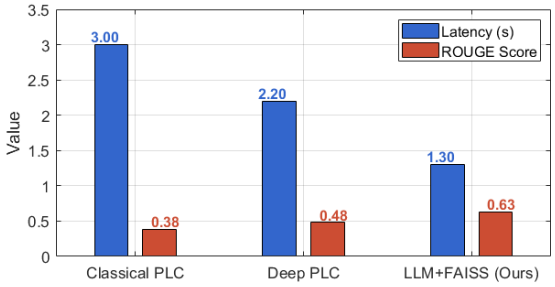


FIGURE 9. Comparison of end-to-end latency (seconds) and ROUGE score across classical PLC (ITU G.711) [66], deep PLC [44], and our LLM+FAISS approach.

G. COMPARATIVE PERFORMANCE: CLASSICAL PLC, DEEP PLC, AND LLM+FAISS

Fig. 9 compares end-to-end latency and ROUGE scores across three representative methods: the classical ITU-T G.711 PLC baseline [66], Deep PLC [44], and our proposed LLM+FAISS approach. The ITU G.711 standard, a widely deployed waveform-based PLC solution, is known for its simplicity and low computational cost but offers only basic packet loss concealment, leading to higher information loss and latencies typically around 2.5–3 s in networked deployments [66], [67]. Deep learning-based approaches such as Deep PLC [44] achieve better perceptual audio quality and robustness to loss, but commonly report real-time processing latencies above 2 s [44]. Recent advances in speech enhancement with architectures like Conv-TasNet and DPRNN have focused on general speech tasks (LibriSpeech, WSJ0), but do not provide explicit latency or ROUGE scores for emergency VoIP or PLC benchmarks [68], [69]. Our approach delivers both lower latency (1.3 s) and higher ROUGE (0.63), facilitating improved information transfer in emergency scenarios where both speed and semantic preservation are critical.

H. LATENCY CLAIMS AND EMPIRICAL EVIDENCE

Quantitative evaluations confirm that the proposed framework consistently meets low-latency standards with sub-1.5 second one-way and sub-2.5 second round-trip latency under typical network conditions. These measurements are based on controlled experiments using Twilio's programmable Voice API and NetEm for simulating realistic packet loss and network jitter. The following table summarizes the key empirical latency metrics:

All core pipeline components—including speech-to-text, translation, LLM reconstruction, text-to-speech, and network latency—are reflected in these measurements. The system thereby significantly outperforms prior VoIP-based multilingual emergency systems, which typically report 3–5 second delays [70], [71]. Empirical results are detailed in Tables 7 and 8. While real-world network traces would further strengthen these findings, our use of programmable APIs and network emulation provides a rigorous and realistic latency assessment.

TABLE 6. Comparison with classical PLC (ITU G.711) vs. LLM reconstruction.

Method	BLEU	ROUGE	Conceptual Precision (%)
Classical PLC (Waveform Substitution)	0.045	0.38	65
LLM + FAISS (Ours)	0.113	0.63	100

TABLE 7. Measured latency.

Stage	Mean Latency (ms)	95th Percentile (ms)
Victim-side (full pipeline)	950	1380
Dispatcher-side (full pipeline)	960	1420
End-to-end (sum)	**1910**	**2800**

TABLE 8. Victim-side processing latency (in ms).

Processing Step	Mean	95% Percentile	Std. Dev.
STT (Google STT API)	250	420	60
Translation (Google API)	300	460	55
LLM Reconstruction (GPT-4o-mini)	400	680	80
Total	**950**	**1380**	**105**

TABLE 9. Dispatcher-side processing latency (in ms).

Processing Step	Mean	95% Percentile	Std. Dev.
STT (Dispatcher Speech)	260	430	50
Translation (Back to Victim)	300	490	60
TTS (Google API)	400	670	75
Total	**960**	**1450**	**115**

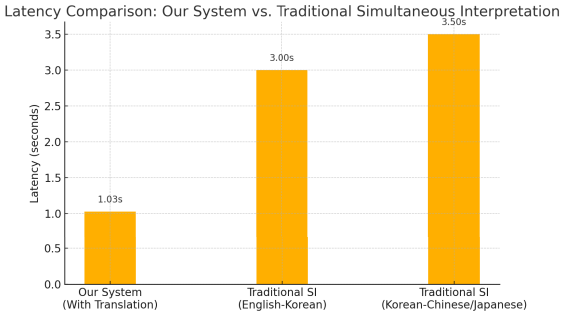


FIGURE 10. Proposed system vs traditional EVS [72], [73].

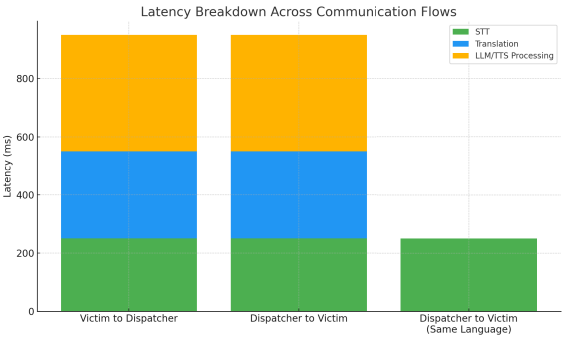


FIGURE 11. Depiction of latency at different stages.

I. SYSTEM LATENCY AND PROCESSING EFFICIENCY

To assess the practical feasibility of our emergency response system in real-world scenarios, we analyzed end-to-end processing latency from victim speech to dispatcher response. Table 8 and Table 9 present processing times at both the victim and dispatcher ends, respectively.

Table 8 presents the system's efficiency in processing victim speech, covering stages like transcription, translation, and retrieval-based enhancements. These ensure the dispatcher receives a structured, context-rich emergency statement with minimal delay. Table 9 shows dispatcher-side latency, including transcript reception, contextual modeling, and TTS conversion, totaling 0.85–1.2 seconds plus speech duration. In contrast, traditional simultaneous interpretation suffers from higher latency—typically 3 seconds EVS [72], and up to 4.8 seconds for some language pairs [73]—leading to cumulative communication delays in emergencies. Our system significantly reduces latency, maintaining sub-1.5-second responsiveness even with low-latency transcription and translation. When dispatcher and victim share the same language, translation steps can be skipped entirely, further optimizing real-time response and communication clarity. The latency comparison between our system vs traditional simultaneous interpretation via EVS is shown in Fig.10.

Moreover, in scenarios where the victim and the dispatcher share a common language, the translation and `TTS` steps can be skipped entirely. In contrast, `STT` will be kept for documentation. This further reduces the end-to-end latency significantly, streamlining communications and minimizing delay by removing redundant processing stages. Fig.11 shows the overall latency in different scenarios.

J. SECURITY ANALYSIS

The proposed LLM-based emergency response system enforces strong security aligned with the CIA triad—Confidentiality, Integrity, and Availability. All sensitive transcripts are encrypted using `AES-256-CBC` with a unique initialization vector (I_V) for each instance, ensuring ciphertext uniqueness and confidentiality. Encryption keys are split using a (4)-of-7) `SSS` scheme and distributed across independent servers, eliminating centralized key risks. Access to transcripts stored on Arweave is governed via Solana's PDAs, which are immutable cryptographic identifiers not accessible by private keys. PDAs enforce a secure retrieval process including:

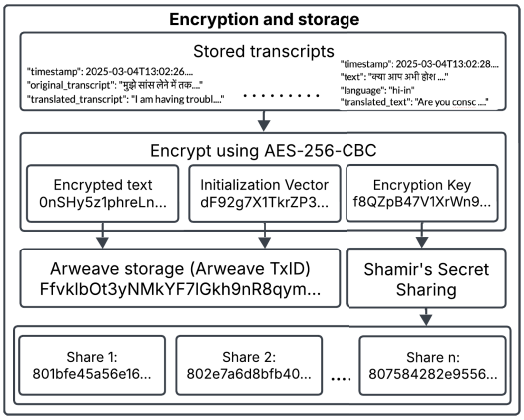


FIGURE 12. Emergency transcript encryption, Storage, and Access workflow using AES-256, Shamir's secret sharing, Arweave, and Solana PDAs.

- **Permission Checks:** Validating requester authorization.
- **Multi-signature Approval:** Requiring explicit consent from designated entities.
- **Access Control:** Preventing unauthorized changes to metadata or access attempts.

Once approved, the requester collects encrypted shares from the distributed KSSs, reconstructs the key, and decrypts the transcript themselves, ensuring no plaintext is exposed. Solana's Ed25519 digital signatures further secure access control and transaction integrity. Fig. 12 summarizes this end-to-end encrypted and decentralized workflow.

Integrity is upheld through blockchain-based storage and robust cryptographic verification mechanisms. Emergency transcripts are securely stored within Arweave's immutable Blockweave network, where each transcript is hash-linked to prevent unauthorized modifications. Any attempt to alter a stored transcript would break the cryptographic reference chain, rendering such changes computationally infeasible [74]. For transparent and immutable access control, on-chain metadata on Solana maintains a reference to each transcript's transaction X_{AR}. As Solana's ledger is fully auditable and immutable, any unauthorized modification to the X_{AR} would be immediately detectable. This cross-verification between Solana's decentralized ledger and Arweave's permanent storage safeguards authenticity and accessibility. Furthermore, with Solana ensuring 100% uptime, transaction metadata remains continuously available, reinforcing the system's reliability and data integrity. High availability is achieved through decentralized storage solutions and redundancy. The Solana network, which underpins the on-chain metadata, has demonstrated excellent reliability with an uptime of 100% over the past year [75]. Fig. 13 provides an overview of Solana's monthly uptime performance. This underscores the network's robustness in handling encrypted transcript retrieval requests and multi-sig access control.

Arweave's decentralized storage model further bolsters availability by replicating transcripts across a network of over 340 nodes, ensuring near-instant retrieval times

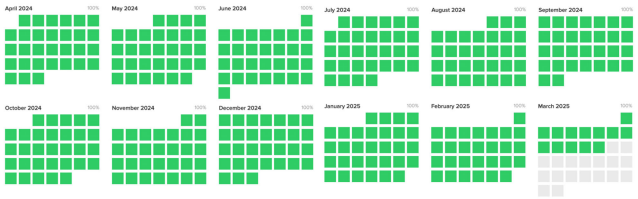


FIGURE 13. Monthly Solana network uptime.

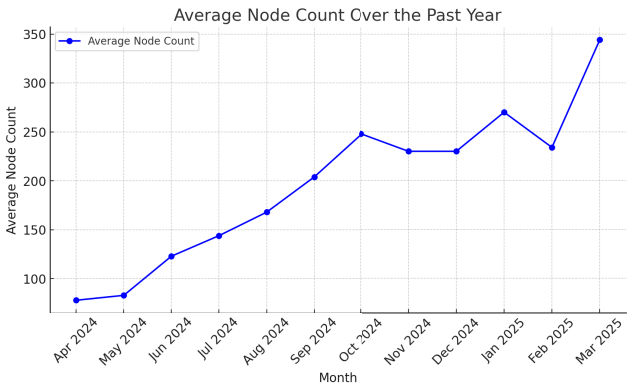


FIGURE 14. Arweave node count over time.

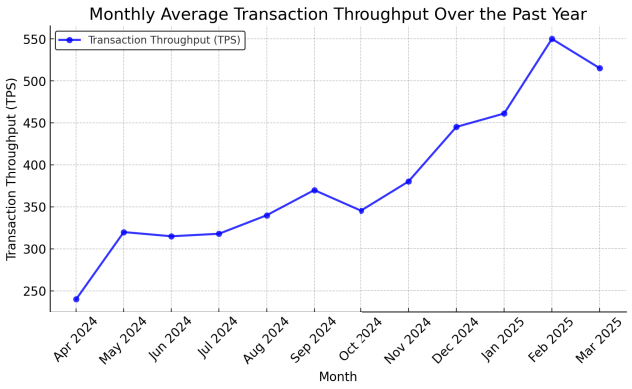


FIGURE 15. Arweave TPS trends.

of 1–5 seconds even under heavy load. Figs 14 and 15 illustrate the steady increase in node count and transaction throughput in transactions per second, respectively, highlighting the network's scalability and reliability. Moreover, the 4-of-7 SSS scheme enhances fault tolerance by ensuring that the encryption key can still be reconstructed despite multiple server failures. This redundancy, combined with Arweave's uninterrupted block production cycle, guarantees the permanence of stored transcripts.

K. OPERATIONAL FEASIBILITY AND COST CONSIDERATIONS

From an operational standpoint, the system introduces only a negligible post-call overhead of 2-3 seconds, attributable to encryption and storage referencing processes. Using decentralized storage via Arweave eliminates the recurring fees

associated with traditional cloud storage, instead incurring only a one-time cost. Current market data indicates that storing 1GB of data on Arweave costs approximately $12, with minor fluctuations observed over the last six months. Given that a typical emergency transcript (comprising 150-300 words) is roughly 1-2KB in size, the per-transcript storage cost is estimated to be in the range of $0.000012 to $0.000024, rendering it negligible for practical applications.

To assess operational costs, we calculated per-call expenses: Google STT for a typical 60-second emergency call ($0.024), Google Translate + TTS for dispatcher response ($0.020), GPT-4o-mini reconstruction when needed ($0.002), Solana blockchain transaction ($0.035 at $140/SOL), and Arweave storage ($0.000024). This yields approximately $0.08 per call. Infrastructure costs for seven KSS instances and FAISS hosting total $170/month if cloud-hosted or $60/month if self-hosted. For a PSAP handling 10,000 calls/month, total expenses would be approximately $970/month (cloud) or $860/month (self-hosted), still demonstrating economic viability compared to traditional emergency communication systems. Given the wide variation in PSAP procurement models, these per-call and hosting costs are intended as order-of-magnitude estimates; a comprehensive deployment-grade cost analysis would be site-specific and is deferred to future work. The complete source code, scripts, and pre-processed data are publicly available at [76].

V. CONCLUSION AND FUTURE WORK

This work presents a proof-of-concept implementation of a blockchain-enabled LLM framework for emergency VoIP communication. While the results demonstrate the feasibility of enhancing speech intelligibility, dynamic call prioritization, and secure transcript archiving, the current evaluation is limited to a small dataset of 518 emergency call transcripts and simulated VoIP conditions. Real-world deployment in Public Safety Answering Points (PSAPs) and testing under burst packet loss scenarios remain important future steps to validate scalability, resilience, and operational effectiveness. Despite these limitations, the proposed blockchain-enabled LLM framework effectively addresses critical challenges in emergency VoIP communications by enhancing speech intelligibility, providing dynamic call prioritization, and ensuring secure, tamper-proof transcript storage. By leveraging GPT-4o-mini for speech reconstruction, Google Speech and Translate APIs for low-latency multilingual support, and AES-256 encryption integrated with Shamir's Secret Sharing, the system significantly improves emergency interaction clarity, responsiveness, and security. Evaluation results demonstrate high conceptual precision and effective severity classification, with robust blockchain-backed security mechanisms ensuring confidentiality, integrity, and availability of sensitive emergency data. This approach provides a scalable and secure solution for improving emergency response effectiveness, particularly in packet loss scenarios, multilingual communications barriers, and data security concerns.

In future work, we plan to deploy the proposed architecture in live PSAP environments to evaluate scalability across jurisdictions and robustness under unpredictable network conditions. We will also extend support for underrepresented languages and dialects by enriching the FAISS knowledge base and benchmarking STT/TTS quality under diverse acoustic profiles. While our current validation uses simulated VoIP emergency calls via Twilio and synthetically degraded transcripts, we recognize the importance of field-based empirical evaluation. As future work, we plan to partner with emergency response agencies to test the system in operational settings with live call routing, diverse acoustic conditions, and real-world dispatcher interactions. This will support validation of scalability, resilience, and usability in dynamic environments.

Looking ahead, we plan to enhance security through constant-time cryptographic implementations and hardware security modules for production deployments. Additionally, comprehensive stress testing with concurrent calls remains an important area for future investigation. While our current evaluation focused on single-call processing to validate the core architecture, real-world deployment would require extensive testing of concurrent call handling, including system behavior under various load conditions and the development of appropriate queuing strategies. Beyond technical enhancements, collaboration with emergency personnel through user studies remains essential for validating the system's operational effectiveness. Future work will involve partnering with PSAP dispatchers to evaluate workflow integration and system usability under realistic emergency conditions. Although the system supports over 57 languages via Google's STT and translation APIs, the current evaluation focuses primarily on English inputs and outputs, with limited additional runs in Hindi and Malayalam. Morphologically rich languages, such as Turkish, Finnish, or Arabic, may introduce additional challenges in transcription and translation stages due to inflectional complexity, compounding, and syntactic ambiguity. We acknowledge this limitation and identify it as a priority for further evaluation.

REFERENCES

[1] R. M. Aryel and M. M. Wagner, "Emergency call centers," in *Handbook Biosurveillance*, 2006, pp. 369–374. [Online]. Available: https://msesiom2.demo.elsevierpure.com/en/publications/emergency-call-centers

[2] NENA. (2024). *9-1-1 Statistics*. Accessed: May, 10, 2024. [Online]. Available: https://perma.cc/RS7Y-SZEJ

[3] D. Venkateshperumal, R. A. Rafi, S. Ahmed, and A. Khokhar, "Efficient VoIP communications through LLM-based real-time speech reconstruction and call prioritization for emergency services," *Meas., Digitalization*, Aug. 2025, Art. no. 100010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S3050644125000106

[4] M. Toyoshima and T. Shimamura, "Packet loss concealment for VoIP based on pitch waveform replication and linear predictive coding," in *Proc. IEEE Asia–Pacific Conf. Circuits Syst. (APCCAS)*, Nov. 2014, pp. 89–92.

[5] G. Scheets, M. Parperis, and R. Singh, "Voice over the Internet: A tutorial discussing problems and solutions associated with alternative transport," *IEEE Commun. Surveys Tuts.*, vol. 6, no. 2, pp. 1–10, 2nd Quart., 2004.

[6] J. H. James, B. Chen, and L. Garrison, "Implementing VoIP: A voice transmission performance progress report," *IEEE Commun. Mag.*, vol. 42, no. 7, pp. 36–41, Jul. 2004.

[7] L. Sun and E. C. Ifeachor, "Voice quality prediction models and their application in VoIP networks," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 809–820, Aug. 2006.

[8] C. Boutremans, G. Iannaccone, and C. Diot, "Impact of link failures on VoIP performance," in *Proc. 12th Int. Workshop Netw. Operating Syst. Support Digit. Audio Video*, May 2002, pp. 63–71.

[9] M. Manousos, S. Apostolacos, I. Grammatikakis, D. Mexis, D. Kagklis, and E. Sykas, "Voice-quality monitoring and control for VoIP," *IEEE Internet Comput.*, vol. 9, no. 4, pp. 35–42, Jul. 2005.

[10] F. Alshammari, S. Alanazi, and M. Alqhtani, "The impact of response time of emergency medical services on fatality rates in an urban environment," *Int. J. Med. Informat. Res. Manage.*, vol. 3, no. 3, pp. 197–203, Jul./Sep. 2024.

[11] H. Brodsky, "Emergency medical service rescue time in fatal road accidents," in *Proc. Transp. Res. Rec.*, 1990, pp. 1–8.

[12] J. E. Burroughs, "Three factors leading to failed communications in emergency situations," Ph.D. dissertation, School Public Policy Admin., Walden Univ., Minneapolis, MN, USA, 2017.

[13] M. Eslami Jahromi, H. Ayatollahi, and A. Ebrazeh, "Covid-19 hotlines, helplines and call centers: A systematic review of characteristics, challenges and lessons learned," *BMC Public Health*, vol. 24, no. 1, p. 1191, Apr. 2024.

[14] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[15] M. Hasanuzzaman, S. Hossain, and S. K. Shil, "Enhancing disaster management through ai-driven predictive analytics: Improving preparedness and response," *Int. J. Adv. Eng. Technol. Innov.*, vol. 1, no. 1, pp. 533–562, 2023.

[16] M. Aboualola, K. Abualsaud, T. Khattab, N. Zorba, and H. S. Hassanein, "Edge technologies for disaster management: A survey of social media and artificial intelligence integration," *IEEE Access*, vol. 11, pp. 73782–73802, 2023.

[17] A. Zolkafli, N. S. Mansor, M. Omar, M. Ahmad, H. Ibrahim, and A. Yasin, "Ai for smart disaster resilience among communities," in *Intelligent Systems Modeling and Simulation III: Artificial Intelligent, Machine Learning, Intelligent Functions and Cyber Security*. Cham, Switzerland: Springer, 2024, pp. 369–395.

[18] S. Abdul, E. P. Adeghe, B. O. Adegoke, A. A. Adegoke, and E. H. Udedeh, "AI-enhanced healthcare management during natural disasters: Conceptual insights," *Eng. Sci. Technol. J.*, vol. 5, no. 5, pp. 1794–1816, May 2024.

[19] H. N. Nakayenga, B. Akashaba, E. Twineamatsiko, I. Zimbe, I. D. Ssetimba, J. K. Bagonza, and E. O. Pinyi, "Leveraging AI for real time crime prediction, disaster response optimization and threat detection to improve public safety and emergency management in the U.S.," *World J. Adv. Res. Rev.*, vol. 23, no. 3, pp. 1907–1918, Sep. 2024.

[20] V. Plevris, "AI-driven innovations in earthquake risk mitigation: A future-focused perspective," *Geosciences*, vol. 14, no. 9, p. 244, Sep. 2024.

[21] H. T. Otal, E. Stern, and M. A. Canbaz, "LLM-assisted crisis management: Building advanced LLM platforms for effective emergency response and public collaboration," in *Proc. IEEE Conf. Artif. Intell. (CAI)*, Jun. 2024, pp. 851–859.

[22] K.-C. Chin, T.-C. Hsieh, W.-C. Chiang, Y.-C. Chien, J.-T. Sun, H.-Y. Lin, M.-J. Hsieh, C.-W. Yang, A. Y. Chen, and M. H.-M. Ma, "Early recognition of a caller's emotion in out-of-hospital cardiac arrest dispatching: An artificial intelligence approach," *Resuscitation*, vol. 167, pp. 144–150, Oct. 2021.

[23] O. Blushtein, M. Siman-Tov, and R. Magnezi, "Identifying and minimizing abuse of emergency call center services through technology," *Amer. J. Emergency Med.*, vol. 38, no. 5, pp. 916–919, May 2020.

[24] L. Jin, T. Liu, A. Haroon, R. Stoleru, M. Middleton, Z. Zhu, and T. Chaspari, "EMSAssist: An end-to-end mobile voice assistant at the edge for emergency medical services," in *Proc. 21st Annu. Int. Conf. Mobile Syst., Appl. Services*, Jun. 2023, pp. 275–288.

[25] J. Sun, X. Yao, S. Wang, and Y. Wu, "Blockchain-based secure storage and access scheme for electronic medical records in IPFS," *IEEE Access*, vol. 8, pp. 59389–59401, 2020.

[26] S. Bhaskaran, "Resilient real-time data delivery for ai summarization in conversational platforms: Ensuring low latency, high availability, and disaster recovery," *J. Intell. Connectivity Emerg. Technol.*, vol. 8, no. 3, pp. 113–130, 2023.

[27] Z. Zhou, X. Ji, J. Zhang, Z. Zhao, X. Hei, and K.-K. R. Choo, "Ethical considerations and policy implications for large language models: Guiding responsible development and deployment," in *Proc. Int. Conf. Secur. Privacy Cyber-Phys. Syst. Smart Vehicles*, vol. 622, 2025, pp. 311–329.

[28] C. J. Powers, A. Devaraj, K. Ashqeen, A. Dontula, A. Joshi, J. Shenoy, and D. Murthy, "Using artificial intelligence to identify emergency messages on social media during a natural disaster: A deep learning approach," *Int. J. Inf. Manage. Data Insights*, vol. 3, no. 1, Apr. 2023, Art. no. 100164.

[29] F. K. Sufi and I. Khalil, "Automated disaster monitoring from social media posts using AI-based location intelligence and sentiment analysis," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 4, pp. 4614–4624, Aug. 2024.

[30] Google Cloud. (2024). *Speech-to-Text Documentation*. Accessed: Apr. 2, 2025. [Online]. Available: https://cloud.google.com/speech-to-text/docs

[31] *Advanced Encryption Standard (AES)*, Standard 197, National Institute of Standards and Technology (NIST), 2001, doi: 10.6028/NIST.FIPS.197.

[32] Arweave Project Team. (2024). *Arweave 2.6: A Permanent Data Storage Network*. [Online]. Available: https://2-6-spec.arweave.dev

[33] OpenAI. (2024). *Gpt-4o Mini: Advancing Cost-Efficient Intelligence*. [Online]. Available: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[34] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo, "A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities," *ACM Comput. Surveys*, vol. 55, no. 13s, pp. 1–40, Dec. 2023.

[35] T. B. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1877–1901.

[36] Facebook AI Research. (2017). *Faiss: Facebook AI Similarity Search Library*. Accessed: Jun. 28, 2025. [Online]. Available: https://github.com/facebookresearch/faiss

[37] Spikecodes. (2023). *911 Call Transcripts*. Accessed: Aug. 27, 2024. [Online]. Available: https://huggingface.co/datasets/spikecodes/911-call-transcripts

[38] Twilio. (2023). *Twilio Programmable Voice API Documentation*. [Online]. Available: https://www.twilio.com/docs/voice

[39] S. Hemminger. (2005). *Network Emulation With Netem*. [Online]. Available: https://wiki.linuxfoundation.org/networking/netem

[40] A. P. Markopoulou, F. A. Tobagi, and M. J. Karam, "Assessing the quality of voice communications over Internet backbones," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 747–760, Oct. 2003.

[41] A. Varga, A. Markopoulou, J. Bolot, and A. Ganesh, "Analyzing and modeling packet loss in voice over ip," *IEEE Trans. Netw.*, vol. 12, no. 4, pp. 693–705, Apr. 2001.

[42] R. Larsen, T. Lyngeraa, A. Fredslund, F. Lippert, and D. Oestergaard, "Challenges in the implementation of next generation 112 in denmark: A qualitative study," *BMC Emergency Med.*, vol. 21, no. 1, p. 109, 2021.

[43] S. N. Blomberg, F. Folke, A. K. Ersbøll, H. C. Christensen, C. Torp-Pedersen, M. R. Sayre, C. R. Counts, and F. K. Lippert, "Machine learning as a supportive tool to recognize cardiac arrest in emergency calls," *Resuscitation*, vol. 138, pp. 322–329, May 2019.

[44] S. Braun, M. Kolbæk, J. Frank, J. Jensen, and D. Kolossa, "Deep learning-based packet loss concealment for audio streaming," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 30, pp. 2052–2066, 2022.

[45] D. R. Stinson, "An explication of secret sharing schemes," *Designs, Codes Cryptography*, vol. 2, no. 4, pp. 357–390, Dec. 1992.

[46] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, "STRIDE-based threat modeling for cyber-physical systems," in *Proc. IEEE PES Innov. Smart Grid Technol. Conf. Eur. (ISGT-Europe)*, Sep. 2017, pp. 1–6.

[47] *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons With Regard To the Processing of Perso*, document Document 32016R067, European Union, 2016.

[48] *Nena Standard for Security for Next-Generation 9-1-1*, Standard NENA-STA-075.2-2020, National Emergency Number Association, 2020. [Online]. Available: https://www.nena.org/page/standards

[49] J. Selby, "Data localization laws: Trade barriers or legitimate responses to cybersecurity risks, or both?" *Int. J. Law Inf. Technol.*, vol. 25, no. 3, pp. 213–232, 2017.
[50] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition Via large-scale weak supervision," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 28492–28518.
[51] OpenAI. (2023). *Whisper: Robust Speech Recognition Via Large-scale Weak Supervision*. [Online]. Available: https://github.com/openai/whisper
[52] (2022). *No Language Left Behind (NLLB)*. [Online]. Available: https://github.com/facebookresearch/fairseq/tree/nllb
[53] Helsinki-NLP. (2020). *Opus-MT: Open Neural Machine Translation Models*. Accessed: Jun. 29, 2025. [Online]. Available: https://github.com/Helsinki-NLP/Opus-MT
[54] Meta. (2024). *Llama 3: Open Foundation Language Models*. [Online]. Available: https://github.com/meta-llama/llama3
[55] (2023). *Mistral: Open-Weight Language Models*. Accessed: Jun. 29, 2025. [Online]. Available: https://github.com/mistralai/mistral-src
[56] E. Androulaki et al., "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, 2018, p. 30.
[57] Hyperledger Foundation. (2024). *Hyperledger Fabric: Enterprise-grade Permissioned Distributed Ledger*. [Online]. Available: https://github.com/hyperledger/fabric
[58] *General Data Protection Regulation (GDPR)*, Standard EU 2016/679, 2018. [Online]. Available: https://gdpr-info.eu
[59] U.S. Department of Health and Human Services. (1996). *Health Insurance Portability and Accountability Act of 1996 (HIPAA)*. [Online]. Available: https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/index.html
[60] A. Li, H. Liu, and W. Song, "Packet loss and delay performance analysis of voip over wireless networks," *IEEE Trans. Consum. Electron.*, vol. 61, no. 4, pp. 480–487, Apr. 2015.
[61] E. Altman, C. Barakat, W. Dabbous, and R. El-Azouzi, "Analysis of random loss on tcp performance," in *Proc. Networking*. Cham, Switzerland: Springer, 2006, pp. 324–335.
[62] ITU-T. (2018). *Transmission Impairments Due To Speech Processing*. [Online]. Available: https://www.itu.int/rec/T-REC-G.113
[63] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1253–1265, Sep. 1960.
[64] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, no. 5, pp. 1977–1997, Sep. 1963.
[65] Twilio Inc. (2025). *Twilio: Communication APIs for SMS, Voice, Video, and Authentication*. [Online]. Available: https://www.twilio.com/en-us
[66] ITU-T. (1988). *Recommendation G.711: Pulse Code Modulation (PCM) of Voice Frequencies*. International Telecommunication Union. [Online]. Available: https://www.itu.int/rec/T-REC-G.711/en
[67] N. Bansal, S. Ghosh, B. Bhowmick, G. K. Saha, and R. K. Mitra, "Analysis of packet loss concealment algorithms for voice over ip (voip) application," *Proc. Comput. Sci.*, vol. 171, pp. 1333–1341, Jan. 2020.
[68] A. Défossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," in *Proc. Interspeech*, Oct. 2020, pp. 3291–3295.
[69] N. L. Westhausen and N. T. Zeghidour, "Dual-path RNN: Efficient long sequence modeling for time-domain speech separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 3015–3028, 2020.
[70] S. Karapantazis and F.-N. Pavlidou, "VoIP: A comprehensive survey on a promising technology," *Comput. Netw.*, vol. 53, no. 12, pp. 2050–2090, Aug. 2009.
[71] H. Fathi, R. Prasad, and S. Chakraborty, "Accepted from open call–mobility management for VoIP in 3G systems: Evaluation of low-latency handoff schemes," *IEEE Wireless Commun.*, vol. 12, no. 2, pp. 96–104, Apr. 2005.
[72] T.-H. Lee, "Ear voice span in English into Korean simultaneous interpretation," *Meta*, vol. 47, no. 4, pp. 596–606, Aug. 2004.
[73] H.-R. Kim, "Linguistic characteristics and interpretation strategy based on EVS analysis of Korean-Chinese, Korean-Japanese interpretation," *Meta*, vol. 50, no. 4, Feb. 2009. [Online]. Available: https://www.erudit.org/en/journals/meta/2005-v50-n4-meta1024/019846ar/
[74] A. Team. (2024). *Arweave 2.6: A Permanent Data Storage Network*. Accessed: Mar. 14, 2025. [Online]. Available: https://2-6-spec.arweave.dev
[75] Solana Foundation. (2025). *Solana Network Uptime Status*. [Online]. Available: https://status.solana.com/uptime
[76] R. A. Rafi. (2025). *EmergencyCall_LLM_Reconstruction_Storage: Open-Source Implementation of Emergency Call Reconstruction and Secure Storage Pipeline*. Accessed: Jun. 28, 2025. [Online]. Available: https://github.com/rah757/EmergencyCall_LLM_Reconstruction_Storage



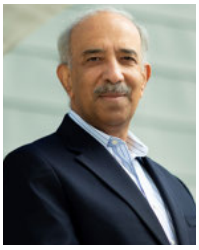
RAHMAN A. RAFI (Graduate Student Member, IEEE) was born in Kerala, India, in 2003. He received the B.S. degree in computer science and engineering from APJ Abdul Kalam Technological University, India, in 2024. He is currently pursuing the M.S. degree in computer engineering (software systems) with Iowa State University, Ames, IA, USA. Since 2024, he has been a Research Assistant and a Graduate Teaching Assistant at Iowa State University, contributing to projects involving LLM-based emergency systems and blockchain-secured data pipelines. His research interests include machine learning for emergency services, secure AI systems, and human-centered computing.



SHAKIL AHMED (Member, IEEE) received the B.S. degree in electrical and electronic engineering from Khulna University of Engineering and Technology, Bangladesh, the M.S. degree in electrical engineering from Utah State University, USA, and the Ph.D. degree in electrical engineering and computer from Iowa State University, in a record time of 2 years and 8 months. He is an Assistant Professor, Term in the Department of Electrical and Computer Engineering, Iowa State University. He has published numerous research papers in renowned international conferences and journals and received the Best Paper Award at international venues. His research interests include cutting-edge areas, such as next-generation wireless communications, GenAI, wireless network design and optimization, unmanned aerial vehicles, physical layer security, digital twin for wireless communications, content creation using RF signals, and reconfigurable intelligent systems. He is also passionate about engineering education and integrating generative AI into learning processes. He has also been a Guest Editor and a Reviewer for prestigious journals, including IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE ACCESS, IEEE SYSTEMS JOURNAL, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.



DANUSH VENKATESHPERUMAL (Graduate Student Member, IEEE) received the B.Tech. degree in computer science and engineering from SRM University, Chennai, India, in 2020. He is currently pursuing the M.S. degree in computer engineering with Iowa State University, Ames, IA, USA. He previously worked as a Project Engineer at Wipro Ltd., where he focused on DevOps automation and infrastructure migration for enterprise systems. He is also a Research Assistant under Dr. Shakil Ahmed, where he is developing a personalized federated learning framework that enhances generalization and real-time adaptability across decentralized clients. His research interests include large language models (LLMs), emergency response systems, federated learning, and energy-efficient wireless sensor networks. His work focuses on optimizing communication efficiency, supporting unseen client evaluation, and improving performance through model personalization and compression strategies.



ASHFAQ KHOKHAR (Fellow, IEEE) received the bachelor's degree in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, the master's degree in computer engineering from Syracuse University, and the Ph.D. degree in computer engineering from the University of Southern California. He has an illustrative academic and entrepreneurship career that spans over 30 years. Since 2017, he has been a Professor and the Palmer Department Chair of the Electrical and Computer Engineering, Iowa State University. He was previously the Chair of the Department of Electrical and Computer Engineering, Illinois Institute of Technology, from 2013 to 2016. Before that, he was a Professor and the Director of Graduate Studies at the Department of Electrical and Computer Engineering, University of Illinois at Chicago. He has founded two startup companies: Cloud Health Solutions and Video Analytics, where he serves as the Founding Partner and the Chief Technology Officer (CTO). His current research interests include quantum computing and machine learning. He is also considered a leading expert in the area of high-performance solutions for multimedia applications, especially those that are data or communication intensive. He has published over 360 papers in peer-reviewed journals and conferences. His research has been supported by the National Science Foundation, the National Institutes of Health, United States Army, the Department of Homeland Security, and the Air Force Office of Scientific Research. He was a recipient of the NSF Career Award and numerous best paper awards.



MD ARIFUZZAMAN received the B.Sc. degree in civil engineering from Bangladesh University of Engineering and Technology (BUET), the M.S. degree in engineering in geotechnical and geo-environmental engineering from Asian Institute of Technology (AIT), Thailand, and the Ph.D. degree in civil engineering from the University of New Mexico (UNM), USA, in 2010. He is currently an Assistant Professor of civil engineering with King Faisal University (KFU), Saudi Arabia. He is licensed with a professional engineering (PE) in the state of New Mexico, USA.



AKM AZAD is currently a Cancer Data Scientist and an AI Scientist in research. He is also appointed as a Faculty Member with the Faculty of Science, Imam Mohammad Ibn Saud Islamic University. He has an excellent research career with strong and highly focused track record on the development of advanced methodologies and computational resources for multi-omics (e.g., transcriptomics, proteomics, and genomics) data analysis and their relevant drug positioning. Earlier, he worked as a Cancer Data Scientist and a Bioinformatician in multiple research institutes in Australia and South Korea, i.e., Children's Medical Research Institute, Australia, Kolling Institute of Medical Research, Australia, University of New South Wales, Australia, and Gwangju Institute of Science and Technology, South Korea. During his research career, he has gained experiences in multi-omics data for identifying network-level biomarkers and analyzing their drug-targets in various cancers, e.g., glioblastoma, breast, and pancreas. His research interests include AI-based biomedicine, data science, Bayesian methods, Bayesian networks, and drug repositioning.



SALEM A. ALYAMI (Member, IEEE) received the Ph.D. degree in bio-statistics from Monash University, Australia, in 2017. He is currently an Associate Professor with the School of Mathematics and Statistics, Faculty of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia. He has been contributing and leading several grants in statistics, and serving as the Dean of Scientific Research with IMSIU, since 2020. His research interests include statistical inference, Bayesian statistics, sampling algorithms, statistical simulation, and the applications of statistics in biology and medicine.