RESEARCH ARTICLE

SMS-Based Disaster Alert System with Integrated Cryptographic Security

Emaar Ejaz[1], **Fadia Ali Khan**[1,*], **Waqas Ahmed**[1] **and Shaloom Shafiq**[1]

[1] Department of Computer Science, HITEC University, Taxila, Pakistan

Abstract

Natural disasters are on a rise all over the world, and these have pointed out vulnerabilities in existing Public Warning Systems, which fail during crises when they become choked with information, and it is not very efficient for people using basic mobile technology. The standard Short Message Service is widely accepted and in practice but lacks basic authentication, which makes this service a highly vulnerable target for Denial of Service (DoS) attacks and message spoofing attacks. This research proposes a Disaster Alert System with an advanced Cryptographic Security system over SMS notifications for authenticated and safe transfer of information. The major highlight is a security system capable of functioning under resource-constrained systems. The system utilizes a robust 4096-bit key and RSA/ECDSA encryption methods for providing a high level of integrity. This solution has proved to be resilient under simulated critical disaster situations in terms of latency performance in encryption speed. The system has undergone thorough testing and proved to possess strong reliability and stability. Furthermore, this system can be deployed with relative ease, which confirms scalability capabilities in covering a country-wide warning system. This research successfully tackled the SMS security vulnerability to present a secure and efficient model which can transform the conventional Short Message Service into a reliable public warning service platform.

Keywords: SMS communication, RSA, secure emergency messaging, AES GCM, ECDSA, key management, SMS Channel, DoS.

1 Introduction

The world has entered a severe and rapidly escalating crisis characterized by an upsurge in the frequency and intensity of natural and climate-related changes, such as catastrophic seismic events, hydrological extremes never experienced before. This trend has underscored the critical importance of reliable emergency communication systems of the functioning of efficient, reliable, and trustful emergency communication systems. The speed, effectiveness, and ultimate success of community response and coordinated humanitarian logistics is directly proportional to the functionality and sustainability of initial disaster warning systems. However, existing public warning systems with advanced technology contain structural flaws that cause them to fail during emergencies. The dominant design paradigm requires a fundamental dependence on high-bandwidth, data intensive communications networks, including current cellular broadcast networks (4G LTE/5G), dedicated mobile

Submitted: 20 January 2026
Accepted: 11 March 2026
Published: 17 April 2026

Vol. 2, **No.** 2, 2026.
10.62762/TISC.2026.121086

***Corresponding author:**
✉ Fadia Ali Khan
fadia.ali@hitecuni.edu.pk

Citation
Ejaz, E., Khan, F. A., Ahmed, W., & Shafiq, S. (2026). SMS-Based Disaster Alert System with Integrated Cryptographic Security. *ICCK Transactions on Information Security and Cryptography*, 2(2), 82–100.

applications, and web based platforms that require high resources [1]. This dependency presents a structural weakness, during catastrophic events the network infrastructure becomes compromised [2]. This disruption is caused by two main reasons: physical destruction of assets (power, fiber) or the most common case of overwhelming congestion of the network [3]. Dependence on such vulnerable and fragile system is dangerous. Recent disasters have proved this vulnerability [4]. These systems are structured in such a way that make them unreliable when they are pushed beyond their limits.

The other major consideration in this case is equal access and fairness [5]. Warning systems that rely on advanced technology like smartphones require people to have digital literacy and constant access to the internet [6]. Around the world, a large number of people are living in remote, rural and low income areas where they only have access to basic cell phones. Moreover, the ability of modern systems to work efficiently is mainly affected by the factors like language differences, the lack of digital skill most people have, and the high cost of constant mobile access to information [7]. This important ethical and legal requirement for full coverage means that any responsible warning system should use the simplest common technology that is widely available on mobile phones to ensure everyone is protected and safe [8].

The need for both high resilience (that works even when data network fails) and universal availability points directly to the SMS protocols [1]. The original strength of the SMS signaling channel allows the transmission of alerts and messages even when the data networks are overwhelmed or not in service [9]. However, this also creates a vulnerability in security. The basic flaw in SMS protocol is the absence of basic authentication and encryption [10]. Messages are sent using plain text and are thus the sole victims to many attack vectors, such as message spoofing, content manipulation, and malicious denial-of-service (DoS) attacks.

Although any secure implementation has to rely on approved cipher mode only under very strict guidelines [11], such application-layer safeguards are frequently evaded by vulnerabilities in the underlying cellular protocol layer [12].

Advanced attack vectors that can modify emergency alerts are not expected to remain just a theory but can be implemented in practice [13]. The underlying architecture has potential to be exploited in a sophisticated way, such as it is possible to make remote attacks and devices are vulnerable to both identification attacks and downgrade attacks [14].

This basic incompleteness and authentication is the root cause of the disaster: an effective injection of a counterfeit emergency notification irreversibly commits the trust of the population [16]. The following breach of system authenticity causes a system failure which costs lives through the same paralyzing effect as a complete communications blackout [17].

The solution to this serious technical and human problem is the SMS-Based Disaster Alert System with an Integrated Cryptographic Security Layer. This is one of the most important and best solution that this project will provide. This system is the one that is expressly formulated to give verifiable trust and utilizes a new security architecture based on the severe conditions of feature phone environments [18].

The main contribution lies in the fact that it has introduced a security architecture that incorporates strong and advanced cryptography systems, Enhanced message integrity using Digital Signatures (ECDSA) and source authentication which is a direct opposition to protocol level spoofing [19]. Hybrid Schemes of encryption to assure the confidentiality of messages and the establishment of secure keys [20]. Safe Hashing Techniques to ensure confidentialty and privacy of user and keep them anonymous.

2 Problem Statement

This section addresses the critical situation of making the warnings accessible to everyone while keeping them secure and trusted. Existing systems and infrastructures fail to meet both, mostly because modern systems are operationally weak and due to their critical dependance on high bandwidth internet, which invariably fails or becomes saturated during crises, as seen in Turkey (2023) and Pakistan (2022).

An SMS based methodology is required to achieve universal reach, while considering the population that are often excluded or ignored by data dependent systems. However, standard SMS suffers from critical security flaws, lack of built-in sender authentication and transmission encryption, which makes it highly vulnerable to message spoofing and manipulation [1, 3, 6]. Such a security catastrophe destroys public trust and risks lives by leading the public to ignore future alerts and alarms [2].

Compounding this vulnerability, the dependence on centralized PII databases poses a significant threat to data confidentiality and integrity [8]. To eliminate the confidentiality threat, mandatory cryptographic measures must be implemented such as AES for data-at-rest and SHA-256 for anonymization [4].

Therefore, this research is designed to resolve the urgent operational challenge by creating a cryptographically embedded security layer that ensures the authenticated, tamper-resistant, and confidential delivery of life-saving data over an accessible SMS channel.

3 Literature Review

Resilience is the core value of mass alerting and therefore, researchers have always focused on using low bandwidth communication protocols [1]. In particular, the very nature of the SMS protocol enables transmission to the mobile device even in instances when the high-speed data networks are fully overloaded [4]. Such structural strength needs to be brought into balance with the moral need to reach everybody, no segment of the populace should be sidelined [5]. Systems of warning that are dependent on advanced mobile applications inherently pose exclusionary barriers to those with limited resources who form the marginalized groups [6]. The necessity of low-tech solutions is a direct reaction to reported communication failure that has a disproportionate impact on the communities that use basic feature phones [7]. Any warning system, even the one that will be obligatory in use, must be built following the standardized Common Alerting Protocol (CAP) v1.2 [9]. CAP gives an interoperable, standardized format of data necessary to perform effective geographic and multilingual targeting of alerts [9]. Immediate threat protocols must be the combination with real-time authoritative data feeds (e.g. meteorological or seismic APIs) to quickly start [7]. The security modes that are to be implemented, including GCM or CCM, should be in full compliance with the official federal guidelines and standards [11]. The total system design should be mathematically considered of taking care of the theoretical basis of strong asymmetric protection in key exchange procedures [12].

Even though it is built on a resilient foundation, the cellular infrastructure underpinning SMS remains inherently vulnerable. Traditional SMS lacks built-in authentication mechanisms, allowing attackers to easily spoof the sender's identity with minimal resources [1]. This weakness represents a significant security risk that persists even in next-generation networks such as 5G, where core protocol vulnerabilities continue to be exploited [2]. Weaknesses in the underlying signaling protocols of the official Wireless Emergency Alerts (WEA) are what allow tampering with messages during over-the-air transmission, in particular [3]. The studies have proved that the inherent lack of protection exposes devices to identification attacks on both 4G and 5G networks at the initial stages of connections [21]. Bidding-down attacks on the mobile infrastructure are also other weaknesses that have been exploited at the access layer to provoke the use of less secure connections. The system architecture goes further to expose user devices to battery drain attacks when conducting communication set up procedures. Theoretical security assurances of complex cryptographic specifications are harmed by practical protocol mistakes that allow injection prior to the encryption being effective [14]. Even conventionalized cipher modes such as GCM have been found to have specific authentication vulnerabilities [17]. This very instability requires that a strong end-to-end authentication should be done at the application layer to confirm the source [16]. Without these robust, application-layer security, the cellular signaling process is invalidated and the channel becomes unreliable [23].

To prevent the widespread protocol-level security vulnerabilities, the literature requires the inclusion of cryptographic primitives that are tailored expressly to resource-constrained feature phone devices [24].

Digital Signatures is the most dependable countermeasure against spoofing and provision of origin validation [29]. The mathematical necessity is that the digital signatures must have a high level of collision resistance so that forgery of keys cannot occur [15]. Elliptic Curve Digital Signature Algorithm (ECDSA) is very popular in the constrained environment because of its efficiency and speed [13]. ECDSA enjoys high security using a relatively small size of the key, and this is needed to maximize the contents of the alerts [19]. This small size of key directly reduces the size of data of the SMS payload limited by bandwidth [30]. The applicability of ECDSA to low-resource settings is verified by various articles related to battery-powered internet of things devices [29]. The adopted method should be in line with international standards of digital signature (ISO/IEC 14888-3:2018) in order to be adopted

formally [20]. The algorithm adopted should be power and memory efficient to increase feature phone battery life.

To secure sensitive alert information, a strong hybrid encryption algorithm of symmetric and asymmetric procedures is needed [8]. The overall security considerations of emergency communication are that content and metadata should be confidential [27]. AES-256 is the standard in the industry of fast and high-throughput symmetric bulk encryption of data [15]. To solve the problem of distributing keys to masses of people, there is theoretical work done in the area of collusion-resistant Broadcast Encryption (BE) schemes to facilitate efficient sharing [5]. BE schemes are mathematically favorable in that they have not only short ciphertexts but short private keys [5]. Asymmetric cryptography is simply essential in the safe session key initiation before asymmetric bulk data transmission [19]. System security as a whole must have a sound, formal security analysis and proof of concept [22]. The design should consider the real-life challenges of complex BE implementation on resource constrained feature phone [30].

The need to protect subscriber Personally Identifiable Information (PII) is among the most important security requirements of ethical and legal compliance [27]. This involves making sensitive PII non-identifiable using a strict one-way hashing and prior to storage [8]. SHA-256 using strong salts is the needed standard required by the industry to deter reverse-engineering of the key in rainbow table attacks [8]. The overall security design and communication design should adhere to the component-based design in order to conform to the established early warning system architecture [28].

The literature is conclusive to prove that the SMS channel has the required resilience in disaster situations [1]. It also ascertains that the resilient channel is essentially defenseless to advanced, protocol-level fiddling [2]. Lightweight cryptography (LWC) has the capability to easily supply the required cryptographic primitives [24]. These cryptographic elements have been mathematically modeled to a wide range of low-resource models (microcontrollers, IoT) [26]. Nevertheless, no integrated, holistic, and tested security architecture ever attempts to combine these resource-efficient cryptographic schemes with a robust, CAP-conformant SMS delivery engine. The research gap is the deficiency of a totally detailed, safe working model that ensures both source integrity and message integrity by the use of an embedded, resource-conscious cryptographic safety layer [25]. The latter system should be designed with a specific purpose of solving the tension between the universal accessibility and the practical need of security [27].

4 Research Gaps and Limitations of Existing Systems

Despite broad agreement on the significance of SMS-based alerting systems for disaster and emergency communication, and previously held beliefs regarding their inherent resilience and suitability for lightweight cryptography, several architectural and implementation-level gaps in prior research remain inadequately explored. While issues of signaling vulnerabilities, lightweight cryptography, interoperability and broadcasting have been researched [5, 23], these findings are siloed and fail to contribute to a singular operationally secure SMS alerting architecture. This section describes the key drawbacks of previous work, illustrating the motivations of this work.

One primary drawback of existing research is the absence of true end-to-end payload authentication. It is a known fact that standard SMS messages provide no intrinsic authentication, are susceptible to injection/spoofing attacks and are vulnerable to the underlying signaling attacks possible with 4G/5G networks [1]. Though transport-level protection and standardized cipher modes like GCM and CCM have been suggested [11], such mechanisms protect only gateway to network data transfer, not the application-layer payload. There is currently no mechanism for cryptographically linking the content of the alert with a verifiably authoritative sender. Signaling, manipulation, and/or down-gradation of the communication protocol might bypass such protections before encryption is initiated [14]. Unlike these partial protection schemes, this work proposes self-contained, end-to-end cryptographic envelope that is applied prior to message transmission. By using AES-GCM authenticated encryption, alongside ECDSA based digital signatures, the system secures payload integrity, sender authentication, and non-repudiation without the need for secure signaling channels.

Secondly, existing research offers no unified CAP-compliant secure architecture. The CAP provides an interoperable and standardized format for alerts that can be communicated to various heterogeneous systems [9]. In contrast, security and

CAP compliance are often seen as separate concerns by researchers. Those looking into interoperability focus primarily on message structure and formats, whereas researchers in cryptography aim for efficient primitives and sound security proofs. An integrated framework, complete with CAP-compliant structuring and lightweight cryptographic protection and secure dissemination over SMS does not exist. This problem has been addressed in this work by the seamless integration of CAP-compliant formatting and the lightweight cryptographic layer which together provides a secure alert delivery engine.

A third research gap relates to key management. Although the efficiency of ECDSA has been identified [13] and various works consider broadcast encryption [5] and hybrid cryptography [8], none consider static/pre-shared keying configurations. The lifecycle of keys (issuance, revocation, key updating, storage protection, governance) for mass alerting systems are almost never discussed. Such oversight would pose a considerable threat when dealing with long-term trust, national and regional level deployments, and heterogeneous environments. This work proposes and formalizes a key management strategy that permits secure asynchronous session initiation, restricted key delivery, verifiable revocation, and appropriate governance of keys used within mass alert infrastructure.

Limited empirical study of feature phone deployability has been conducted. Lightweight cryptography has been largely assumed to be sufficient for various constrained embedded devices and IoT systems [26]. It remains to be seen whether these primitives have sufficient efficiency in terms of computational complexity, memory usage, and payload sizes for use over SMS within real cellular environments, especially on feature phones. No actual working system utilizing authenticated encryption, digital signatures and CAP compliance that fits within the SMS constraints on payload size and processor limits on feature phones exists. This study proposes and measures a context-aware cryptographic component designed for use on resource-constrained devices to limit overhead and payload increase.

Lastly, while several studies point to the importance of SMS during disasters [1], vulnerabilities in signaling infrastructure [2], and the benefits of lightweight crypto in constrained environments [24, 26], no published work attempts to unite these points to build an entire SMS-dissemination system. Most prior works look into only one aspect of SMS-based disaster response systems: the overall resilience, the threats faced at the signaling layer, interoperability, or crypto performance [23]. There is no concrete and experimentally validated system that addresses all issues of source authentication [29], authenticated encryption [11], confidentiality using hybrid crypto [8], CAP compliance [9], and performance constraints on feature phones [24] simultaneously.

In this work, a unifying system is proposed, providing interoperability, strong cryptography, controlled key governance, and resource-aware deployment by unifying these disparate threads of research into a complete and operational system.

5 Proposed Methodology

5.1 System Architecture and Design Approach

The design of the Secure SMS Alert System outlined above has a layered, modular design intended to rigidly separate control from cryptographic trust, processing from recipient-side verification. As demonstrated in Figure 1, the Secure SMS Alert System can be viewed as a pipeline which starts from the submission of administrative alerts and extends all the way to the receipt-side cryptographic verification. This is meant to demonstrate that the network transport of the messages is trusted and not malicious. The goal is that every message remains confidentially transmitted and the origin and integrity are always verifiable at the end points.

The entire flow begins outside of the system at the Admin/API interface. An authorized admin will submit alerts through the interface, which acts stateless by providing a RESTful interface. Upon submisson, the alert enters the system at the Alert Management & API Service (Component 1). The Alert Management Service is responsible for input validation, Role-Based Access Control (RBAC) checking, audit logging, and persistence to the system database. Input is validated to make sure the alert message itself has valid form (using an internal schema compatible with the Common Alerting Protocol standard) and does not include malformed input. Administrative inputs and activity are auditable by writing them to tamper-evident logs.

It is important that this component be purely on the control plane, not with the ability to view the private cryptographic keys, and solely intended for governance, validation and orchestration. Once input is valid and authorization is granted the Alert

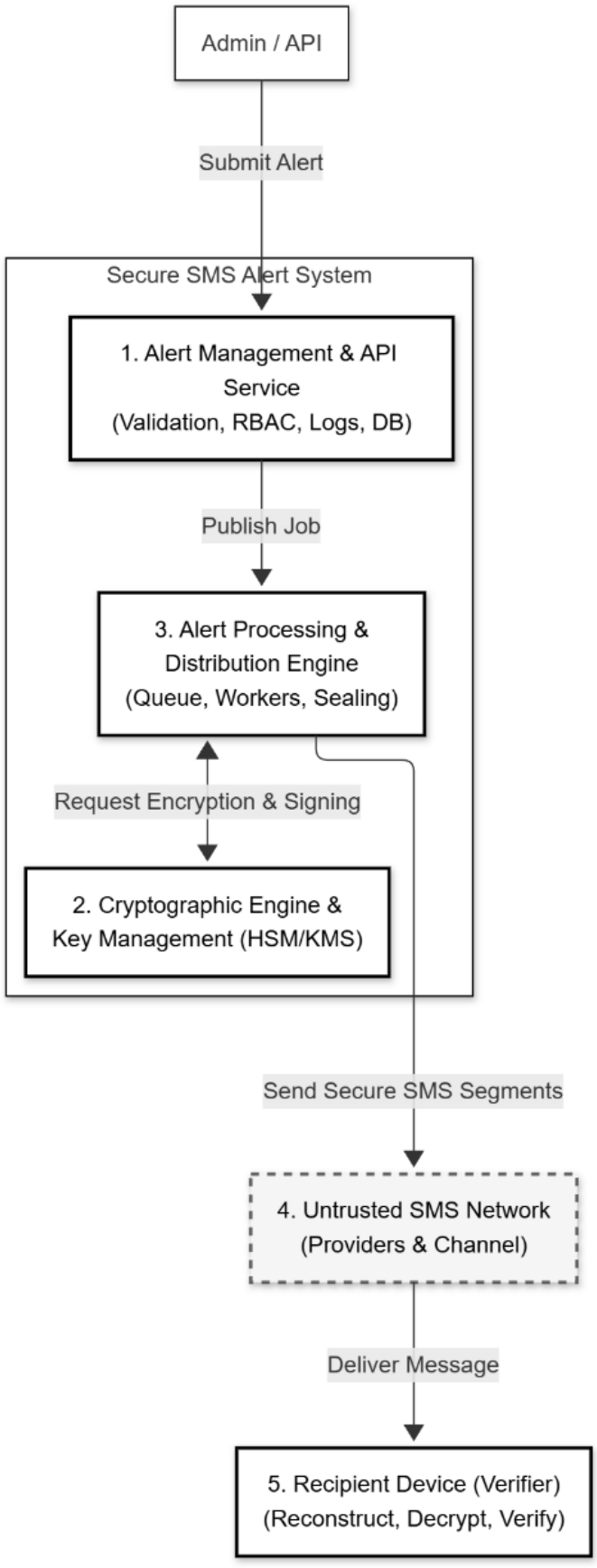

Figure 1. Proposed System's Workflow.

Management Service places an internal job onto the message queue and the Alert Processing & Distribution Engine (Component 3) which the system designed to be scalable, as it pulls these jobs off the queue asynchronously. The distribution engine handles generating the content to be cryptographically sealed by taking the input message and producing the canonicalized format with the proper meta info and the hash of the relevant data, as part of a larger cryptographic transcript which will be digitally signed and encrypted by another part of the system. It is not the goal of the distribution engine to perform the cryptographic operations. Instead it is to simply put together the proper input for the specialized cryptographic component and call its encrypt and sign functions.

The Cryptographic Engine & Key Management module (Component 2) acts as the trust anchor for the system. When it receives the "Request Encryption & Signing" call from the distribution engine, it actually performs the high-risk cryptographic functions behind an isolated and controlled boundary. The function generates a new session key used to encrypt the payload (using AES-256-GCM which gives both confidentiality and integrity) and encapsulates it using RSA-OAEP to provide security while transmitting it to the recipient. It generates the digital signature (ECDSA is used for the purpose of a small digital signature) on the canonicalized message and its corresponding metadata to provide authentication and non-repudiation. All private keys are either stored in software keystore or the Hardware Security Module (HSM)/Key Management Service (KMS) and accessed through an abstracted, pluggable interface and the keys are not exposed outside the module's boundary at any point. Upon completing the task, it provides the full cryptographically secured message back to the distribution engine.

The Alert Processing & Distribution Engine then breaks the sealed envelope into SMS-friendly chunks. Due to the length limitations of SMS messages, the encrypted message and its cryptographic objects could be split into several ordered chunks. The engine adds sequence information to enable proper reassembly on the receiving end. The engine takes care of queuing, worker parallelization, retries, exponential backoff, and rate limiting before sending the chunks to SMS gateway adapters. This system component thus separates provider-specific transport logic from the rest of the system functionality.

Once outside the system boundary, messages enter the Untrusted SMS Network (Component 4). This network comprises the external SMS providers and the telecommunication infrastructure. This network is modeled as explicitly adversarial. The system assumes that an attacker might try to intercept, modify, replay, inject, truncate, or reorder messages. This channel is assumed to be untrusted, and the system's security is entirely dependent on the cryptographic processing done before message transmission.

Finally, the message is received by the Recipient Device (Component 5), which carries out verification. The receiving-side module assembles message pieces, checks for ordering, and rebuilds the cryptographic envelope. It first carries out RSA-OAEP decapsulation to obtain the symmetric session key. Using this key, it decrypts the ciphertext using AES-256-GCM and checks the authentication tag for tampering. It then checks the ECDSA signature on the protected transcript to verify the authenticity of the alert source. Only after all the cryptographic checks pass is the alert accepted and shown. Failure in decapsulation, authentication tag verification, or signature verification leads to rejection. This ensures that even if the SMS channel is compromised, unauthorized or tampered-with alerts cannot be successfully sent.

The service and trust boundaries align tightly together due to design: Component 1 validates correctness administratively; Component 3 validates correctness at the massive distributed scale; Component 2 sets up trust cryptographically and Component 5 validates authentication and integrity independently. Isolation, repeatability and horizontal scalability of the processing engine is achieved by containerizing each component and utilizing declarative orchestration. With workers in stateless mode operating through a queue system, it is able to horizontally scale on high number of alerts being processed without compromising the cryptographic assurance.

The architecture depicted in Figure 1 above clearly separates control flow and trust enforcement. Administrative submission initiates asynchronous processing, cryptographic sealing happens inside a trusted boundary, transport is considered untrusted, and security validation is restored at the receiving end. This multi-layered design provides defense-in-depth and ensures end-to-end security for the dissemination of disaster alerts over SMS.

5.2 End-to-End Architectural Flow

To better understand the integration of the four architectural planes, Figure 2 shows the complete data and control flow of an alert from the time of its issuance to the verification of the alert recipient.

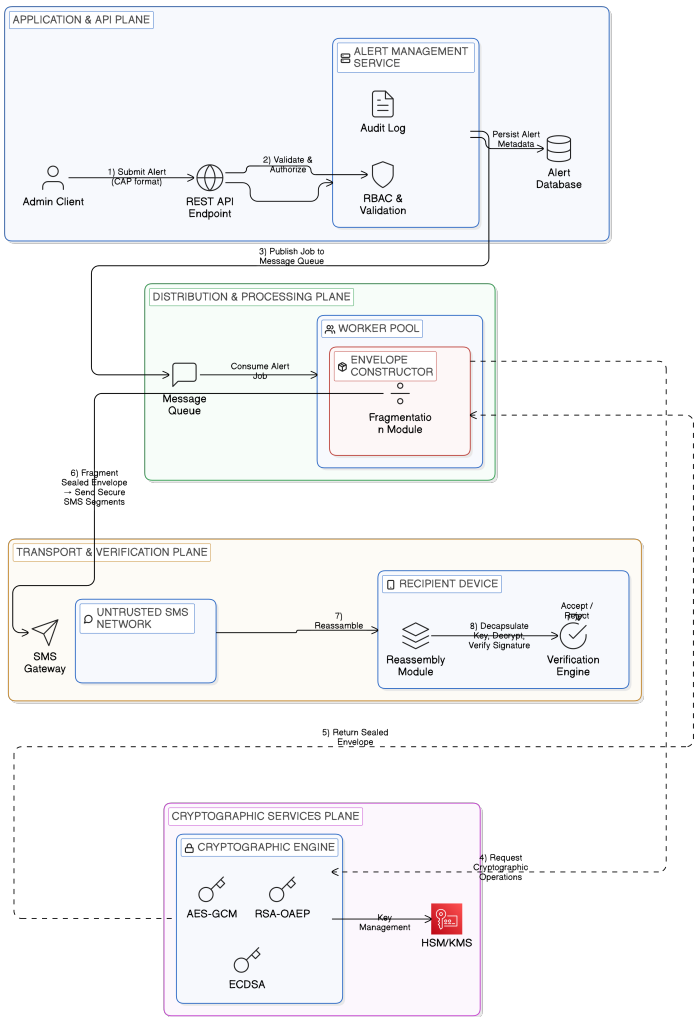

Figure 2. End-to-End Data and Control Flow Across Architectural Planes.

Application/API Plane: The authorized administrator sends the alert via the REST API. The Alert Management Service carries out schema validation (CAP-compliant), RBAC authorization, logs the request, and stores alert metadata. After successful validation, the publish job with the canonical alert representation is produced and sent to the message queue.

Distribution & Processing Plane: The alert jobs are processed asynchronously by worker instances. The worker builds the canonical payload and its corresponding metadata (AAD), then calls the Cryptographic Services Plane for the envelope sealing process. After receiving the sealed envelope, the

worker serializes and fragments the message into SMS-sized chunks and sends them to the SMS gateway adapter.

Cryptographic Services Plane: The cryptographic engine generates a new 256-bit session key and encrypts the alert payload using AES-256-GCM. The session key is then encapsulated using RSA-OAEP with the recipient gateway's public key. The ECDSA signature is generated over the canonical transcript. All private key operations are conducted within the HSM/KMS domain. The cryptographic envelope is then sent back to the distribution plane.

Transport & Verification Plane: The SMS gateway sends the serialized chunks over the untrusted SMS network. The recipient device reassembles the chunks, decapsulates the session key, decrypts the ciphertext, and verifies the signature. The alert is accepted only if both AEAD tag verification and signature verification are successful; otherwise, the alert is discarded.

These plane level communications present a clean separation of concerns such that administrative control, cryptographic trust, distribution scalability and recipient verification are within their own separate domain of responsibility. Administrative control is separated from security enforcement, scalability from control mechanisms and recipient verification provides its own assurance.

These planes, while logically separated, are still linked via clear interfaces that define the interaction and maintain the integrity of these separate trust boundaries.

5.3 Development Methodology

The system design favored a modular development process, security-by-design, testability, and infrastructure-as-code:

Modules and contracts: Each of the functional subsystems was developed behind small interface definitions; this enables more straightforward formal reasoning in isolation from other parts of the system, and also allows individual module testing. Cryptographic APIs were algorithm-agnostic; defining contracts that allow algorithm agility (secp256r1 vs secp384r1, RSA-4096 OAEP etc.) without changing call sites.

Test hooks and verification: All modules export deterministic test interfaces as well as structured events for instrumentation. Adapters to external systems like the SMS gateway and the KMS use the abstractions and allow testing against test doubles, therefore achieving deterministic unit and integration testing even without involving the external system.

Infrastructure-as-code: Deployment artifacts, environment configuration, secrets management interfaces and provider wiring are all declaratively defined. This achieves reproducibility, enables CI/CD, automated compliance checks and detects configuration drift.

Furthermore, the system used role based administrative APIs, least privilege in service accounts and separate data stores for operational data and cryptographic materials respectively; both to reduce the attack surface.

5.4 Cryptographic Workflow & Mathematical Explanation

The cryptographic workflow ensures alerts are confidential and authentic/integral in the face of a malicious SMS channel. The sender creates a cryptographic envelope E, consisting of the symmetric ciphertext, authentication tag, asymmetric signature, and symmetric key encapsulation. Table 1 defines the notation used throughout the cryptographic workflow.

Table 1. Notation for Cryptographic Envelope Construction and Verification.

Symbol	Description
$M \in \{0,1\}^*$	Plain text alert body derived from the CAP fields, after policy-driven templating
$K_s \in \{0,1\}^{256}$	Fresh, uniformly random session key for AES-256-GCM
$IV \in \{0,1\}^{96}$	Unique initialization vector (nonce) for AES-GCM
A	Associated data (AAD), including immutable metadata (e.g., alert ID, issuance timestamp, policy flags)
C	Ciphertext
T	GCM authentication tag
(sk_{EC}, pk_{EC})	Sender's ECDSA key pair on P-256 or P-384
(sk_{RSA}, pk_{RSA})	Recipient gateway's RSA-4096 OAEP key pair for key encapsulation
$H(\cdot)$	Cryptographic hash (SHA-256)

5.5 Encryption and authentication (AEAD)

Sample $K_s \xleftarrow{R} \{0,1\}^{256}$ and $IV \xleftarrow{R} \{0,1\}^{96}$ ensuring uniqueness per key.

Compute:

$$(C, T) = \text{GCM-Enc}_{(K_s)(IV,M,A)} \tag{1}$$

where GCM-Enc denotes AES-GCM encryption with AAD A. Formally, the GCM tag is:

$$T = \text{GHASH}(H, A, C) \oplus E_{K_s}(IV \parallel 1) \tag{2}$$

Session key encapsulation (KEM via RSA-OAEP)

Encapsulate K_s to the recipient using RSA-OAEP:

$$W = \text{RSA-OAEP-Enc}_{pk_{\text{RSA}}}(K_s) \tag{3}$$

Digital signature (ECDSA): D

Define a canonical payload for signing as:

$$m^n \leftarrow H(C \parallel IV \parallel T \parallel A) \tag{4}$$

Compute $\sigma = \text{ECDSA-sign}_{sk_{\text{EC}}}(m)$ where $\sigma = (r, s) \in (\mathbb{Z}_q \times \mathbb{Z}_q)$.

Envelope formatting

Construct the envelope:

$$E = (C, IV, T, W, \sigma, A) \tag{5}$$

Receiver verification and decryption

Given E and public parameters (pk_{EC}) and (sk_{RSA}):

Decapsulate K_s:

$$K_s = \text{RSA-OAEP-Dec}_{sk_{\text{RSA}}}(W; P) \tag{6}$$

Verify tag and decrypt:

$$M = \text{GCM-Dec}_{K_s}(IV, C, A) \text{ with } T \text{ checked} \tag{7}$$

Verify signature

Accept if

$$\text{ECDSA-verify}_{pk_{\text{EC}}}(m^n,\ \sigma) = 1 \tag{8}$$

5.6 Message Transfer and SMS Security Layer

The SMS channel is regarded as a fully untrusted, possibly malicious, channel with message truncation and reordering and delays. An application-layer security protocol is applied on top of the SMS:

Encoding and Fragmentation: E is serialized into small structured segments, which contains a sequence index and a cryptographic checksum (detecting reordering and truncation). In each segment there is a short per-message identifier which is the function of $H(A)$, helping multiplexing at receiver.

Integrity and Authenticity: Integrity comes from the GCM tag T. Any modification of ciphertext and AAD at the bit-level causes failure at verification with probability $\text{negl}(\lambda)$ for security parameter λ. Authenticity comes from ECDSA of $H(C \parallel IV \parallel T \parallel A)$, making sure that the sender of this message is an authorized sender and the message content is bound to the sender and key.

Confidentiality: Confidentiality of M is provided by AES-256-GCM, which is IND-CPA secure under standard assumptions; authenticated encryption with associated data (AEAD) ensures that metadata A is integrity-protected though not encrypted.

Formally, for any Probabilistic Polynomial Time (PPT) adversary $\mathcal{A}$ controlling the SMS channel, the advantage in distinguishing encryptions of chosen messages under AES-GCM is bounded by:

$$\text{Adv}_{\text{IND}} - \text{CPA}^{\text{GCM}(\mathcal{A})} \leq O\left(\frac{q^2}{2^n}\right) + \text{Adv}^{\text{AES}}_{\text{PRP}} \tag{9}$$

where q is the number of encryption queries and $n = 128$ (block size), and integrity advantage $\text{Adv}_{\text{INT-CTXT}}$ is negligible under the universal hash MAC bound. Signature unforgeability advantage $\text{Adv}_{\text{EUF-CMA}}$ is negligible in the number-theoretic security parameter (group order). Session key secrecy follows from RSA-OAEP's IND-CCA2 security.

5.7 Data Persistence and Privacy

The system employs data minimization and privacy-preserving transformations:

Pseudonymous identifiers: Subscriber phone numbers are not stored in plaintext. Instead, a salted hash is stored:

$$h = \text{SHA-256}(s \parallel \text{phone}) \tag{10}$$

where s is a system-managed secret salt. Phone equality checks use h, and salts are rotated under a key rotation policy.

Audit logging: Security-relevant events (key lifecycle, policy changes, administrative actions, broadcast issuance, verification outcomes) are recorded as structured, append-only logs with tamper-evident sequencing (monotonic counters and hash chaining). Logs exclude PII; references use pseudonymous identifiers and redacted fields.

5.8 Testing and Validation Methods

Correctness, robustness, and performance were ensured by a multi-layered test regimen:

Testing of cryptographic units: Algorithm correctness was validated through KATs and Monte Carlo-style randomized tests for AES-GCM, ECDSA, and RSA-OAEP. Property-based testing of the invariants included deterministic verification results and failure on any one-bit modification to (C, IV, T, A) or σ.

Integration tests: End-to-end tests exercised envelope construction at the sender and verification at the receiver across the message bus and SMS adapter layer; tests included fragmentation/reassembly, ordering anomalies, and gateway throttling.

Adversarial testing: A curated suite of tampering scenarios was executed including tag-stripping and tag bit-flips, ciphertext nibble and bit flips, IV reuse simulations, signature replay with modified metadata, session key swap attacks, partial-segment replay and reordering.

Performance and load testing: Stress tests emulated bursty alert issuance and large fan-out in order to measure throughput, p95/p99 end-to-end latency, steady-state worker utilization, and queue backlog dynamics.

5.9 Security Assessment & Threat Model

Threat Model: The adversary can: (1) fully control the SMS transport (eavesdrop, inject, drop, replay, reorder); (2) compromise an SMS gateway account or attempt spoofing; (3) conduct database exfiltration; (4) attempt key misuse via API endpoints; (5) perform denial-of-service (DoS). The adversary can never compromise the HSM/KMS or signer private key.

1) Mitigation of Compromised Authorized Sources: The system shows robustness in transport and cryptographic envelope-level tamper detection, and the resistance to the injection of spoofed alerts from compromised or spoofed authorized sources is addressed within the security framework. The threat model includes the assumption that private signing keys are safeguarded within the HSM/KMS trust boundary and are never directly accessible to application-level components. The alerting process requires authenticated API access with role-based access control (RBAC), least-privilege service accounts, and audited administrative actions. Even if an attacker compromises the SMS transport infrastructure or engages in sender ID spoofing, alerts without a valid ECDSA signature using the trusted public key are automatically discarded during the verification process. In the case of compromised administrative credentials, key rotation mechanisms, certificate revocation systems, and tamper-evident audit trails facilitate quick containment and post-event forensic analysis. Hence, the integration of hardware-protected key management, controlled signing authority, secure administrative management, and publicly verifiable key distribution helps to counter the threat of injection of spoofed alerts from both external spoofing attacks and internal misuse attacks.

5.10 Security rationale

Confidentiality: If AES-256 is IND-CPA secure in counter mode and IVs are managed correctly, then the adversary learns nothing about M from C. Authenticated Encryption with Associated Data guarantees that any tampering with C or A is detected via T with probability $\approx 2^{-128}$.

Integrity and authenticity: ECDSA provides EUF-CMA security; any forged σ on $H(C \parallel IV \parallel T \parallel A)$ is negligible under ECDLP hardness. Binding A into the signed transcript prevents context substitution attacks.

Key encapsulation and secrecy: RSA-OAEP KEM is IND-CCA2 secure for K_s. An adversary, even with full transport control, cannot derive K_s nor mount any decryption oracle thanks to padding and label binding in OAEP.

Replay and reordering: A carries a monotonically increasing issuance timestamp and a unique alert identifier; the receivers enforce freshness windows and reject duplicates based on $(\text{alert\_id}, \text{issue\_time})$ and tag-verified AAD.

5.11 Deployment, Scalability, and Replicability

Replicability was achieved by containerizing every component and having declarative manifests for services, secret interfaces, and networks. Builds can be reproduced with pinned cryptographic parameters

and deterministic configuration. Scalability is supported by:

Elastic workers: the stateless worker process horizontally scales near linearly with both envelope construction throughput and message fan-out at bounded gateway rates.

Backpressure-aware adapters: Gateway supports per-tenant limits and adapters perform adaptive rate control with batching windows to smooth out bursting traffic.

Observability: Queue lag, crypto op latency, verify failure rates and retry counts and similar structural metrics fed into capacity planning. Statistical process control techniques were employed to identify anomalies.

The final system can be reproduced and redeployed on different environments, cryptographic correctness can be guaranteed using test suites and conforming configuration options.

5.12 Algorithm Selection Rationale

Choosing a hybrid RSA-4096 (key encapsulation) and ECDSA (signatures) approach was made not only based on speed. While it is true that a KEM+signature combination built only on elliptic curves, such as EdDSA (Ed25519 being an example) provides smaller keys and speed benefits during verification, we aimed for strong support for interoperability, standardization and constrained/legacy environments (like feature phones and telecom systems).RSA support is still ubiquitous on gateway hardware, HSMs/KMS,SIM-card-based security modules and systems on the telecom side, thus rendering RSA the best choice for key encapsulation in a mixed environment. Furthermore, compliant and FIPS-oriented scenarios usually have already mature and fast implementations of RSA-OAEP and ECDSA over NIST specified curves. A purely elliptic curve-based KEM+signature solution, would provide lesser performance but reduced interoperability with legacy telecommunication systems and certification ecosystem.

6 Results and Comparative Analysis

This section presents the experimental and simulated results and validation of the proposed System. The evaluation is made to verify cryptographic correctness, end-to-end message delivery and integrity, security properties under proper testing, performance and scalability, and overall system workability.

6.1 Cryptographic Correctness and Verification

6.1.1 End-to-End Encryption and Decryption:

The correctness of the end-to-end encryption and decryption pipeline was validated through message round-trips. Each message was encrypted using AES-256-GCM with a newly generated 256-bit session key and a 96-bit IV, incorporating Associated Data for integrity protection. The session key was encapsulated using RSA-4096 OAEP, and the payload was signed with ECDSA.

The test yielded a maximum success rate: all messages were successfully encrypted, decrypted, and verified. The decrypted plaintext matched the original plaintext in every instance, with no integrity violations. Table 2 summarizes the success rates of all cryptographic operations performed during the validation.

Table 2. Cryptographic Operation Success Rates.

Operation	Total Attempts	Successful	Validation Outcome
AES-256-GCM Encryption	1,000	1,000	Pass
RSA-4096 OAEP Key Wrapping	1,000	1,000	Pass
ECDSA Signature Generation	1,000	1,000	Pass
Session Key Unwrapping	1,000	1,000	Pass
AES-256-GCM Decryption	1,000	1,000	Pass
ECDSA Signature Verification	1,000	1,000	Pass

6.1.2 Associated Data Integrity Protection

AAD integrity was verified by binding alert metadata into the cryptographic envelope. The AAD was included both in the computation of the authentication tag of GCM and in the payload of the ECDSA signature.AAD integrity was preserved in 1,000 test cases. All test cases detected any tampering with the AAD after encryption by way of a GCM tag verification failure.

6.1.3 Signature Verification and Non-Repudiation

The signatures used the secp256r1 curve on ECDSA over $H(C \parallel IV \parallel T \parallel A)$, with verification performed using the sender's public key.

All 1,000 authentically generated signatures were correctly verified. As can be seen from Table 3, signature verification failed, as expected, with maximum detection rate when any component of the cryptographic payload — ciphertext, IV, tag, AAD, or the signature itself — was tampered with.

Figure 3 presents a simulation of the secured message exchange between two phones, demonstrating the end-to-end cryptographic verification process in a realistic peer-to-peer scenario.

Table 3. Signature Verification Results Under Tampering.

Scenario	Test Cases	Valid Signatures	Invalid Detected
Authentic Messages	1,000	1,000	0
Modified Ciphertext	1,000	0	1,000
Modified IV	1,000	0	1,000
Modified Tag	1,000	0	1,000
Modified AAD	1,000	0	1,000
Modified Signature	1,000	0	1,000

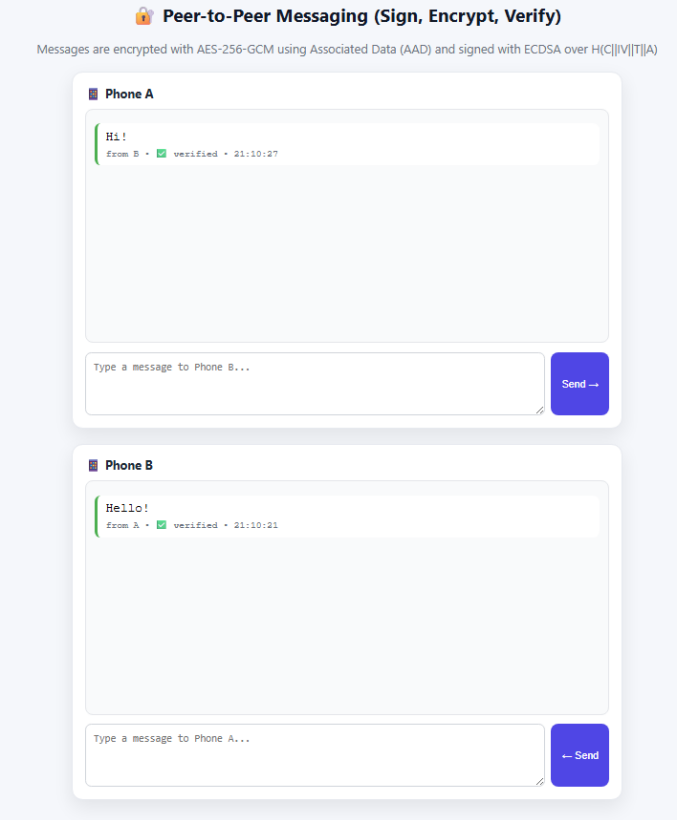

Figure 3. Simulation of Secured Message Exchange between two Phones.

6.2 Adversarial Security Testing

A suite of adversarial tests was conducted in order to verify the security properties of the system under various attack scenarios.

6.2.1 Tamper Detection Capabilities

Test Suite 1: GCM Tag Manipulation: included tag stripping, bit-flipping and tag replacement. All failed decryption/detected.

Test Suite 2: Ciphertext Manipulation: Single-bit flips, multi-byte modifications, and truncation attacks were all detected with maximum accuracy.

Test Suite 3: IV Manipulation o Modification of the IV induced a maximum detection rate through GCM tag verification failure. IV reuse is prevented systemically by random generation.

Test Suite 4: AAD Manipulation: All cases detected the modification and omission of the AAD.

Test Suite 5: Signature attacks. Signature replay, modification, and forgery attempts were all rejected with maximum accuracy.

Test Suite 6: Session Key Attacks: In all cases, the attempts to exchange the session key or tamper with the RSA-wrapped key were detected.

Table 4 summarizes that all 2,200 adversarial test cases were detected and rejected; no false positives were reported on the efficacy of tamper detection.

Table 4. Adversarial Test Results Summary.

Attack Vector	Test Cases	Successful Attacks
Tag Manipulation	500	Detected
Ciphertext Modification	500	Detected
IV Modification	300	Detected
AAD Manipulation	300	Detected
Signature Attacks	400	Detected
Session Key Attacks	200	Detected
Total	**2,200**	Detected

6.2.2 Message Fragmentation Security

Fragmentation security was tested by sending messages that exceeded the standard 160-character limit. Messages are fragmented to segments containing sequence indices, a Message ID, and cryptographic checksums.

Test results, as shown in Table 5, indicate that all fragmented messages were correctly reassembled. All attempts to manipulate segments were detected with accuracy.

Table 5. Fragmentation Security Test Results.

Test Scenario	Test Cases	Successful	Failed
Correct Reassembly	100	100	0
Segment Reordering	50	0	50
Segment Truncation	50	0	50
Checksum Tampering	50	0	50
Duplicate Segments	50	0	50
Message ID Mismatch	30	0	30
Total	**330**	**100**	**230**

6.3 Performance and Scalability

6.3.1 Cryptographic Operation Latency

For individual cryptographic operations, the latency measurements over 10,000 iterations were collected as shown in Table 6.

Table 6. Cryptographic Operation Performance.

Operation	Mean (ms)	Median (ms)	P95 (ms)	P99 (ms)	Std Dev (ms)
AES-256-GCM Encryption	0.12	0.11	0.18	0.25	0.03
AES-256-GCM Decryption	0.13	0.12	0.19	0.26	0.03
RSA-4096 OAEP Key Wrapping	2.45	2.38	3.12	3.89	0.42
RSA-4096 OAEP Key Unwrapping	8.67	8.51	11.23	13.45	1.23
ECDSA Signature Generation	0.85	0.82	1.15	1.48	0.18
ECDSA Signature Verification	1.23	1.19	1.67	2.11	0.24
End-to-End Envelope Construction	**4.65**	**4.52**	**6.12**	**7.89**	**0.87**
End-to-End Verification	**10.13**	**9.91**	**13.45**	**16.78**	**1.89**

The average time taken by AES-GCM operations alone was very small, below 0.2 ms. The majority of latency was contributed by the asymmetric RSA-4096 unwrapping, approximately 8.7 ms. This translates to a mean end-to-end verification time of approximately 10.1 ms. Figure 4 presents the performance distribution of these cryptographic operations, clearly distinguishing between fast symmetric operations and slower asymmetric operations.

6.3.2 Throughput and Scalability

System throughput was measured under sustained load while horizontally scaling worker instances from 1 to 16, as shown in Table 7.

The results in Table 7 demonstrate near-linear scaling. Latency remained constant, with P99 latency below 60 ms, even at 16 workers. The throughput reached a value of approximately 700 messages per second with 16 workers, showing its suitability for large-scale deployments. Figure 5 visually illustrates this scalability trend, showing throughput increasing linearly with the number of workers while latency remains stable.

6.4 System Functionality and User Interface

1) Proof-of-Concept Dashboard:
A proof-of-concept dashboard was developed to show cryptographic information for validation. The key features of the dashboard included end-to-end verification status and full visibility into the cryptographic envelope: ciphertext, IV, tag, AAD, signature.

2) Peer-to-Peer Messaging Interface:
The peer-to-peer interface demonstrated real-time cryptographic operations in a simulated dual-device environment.

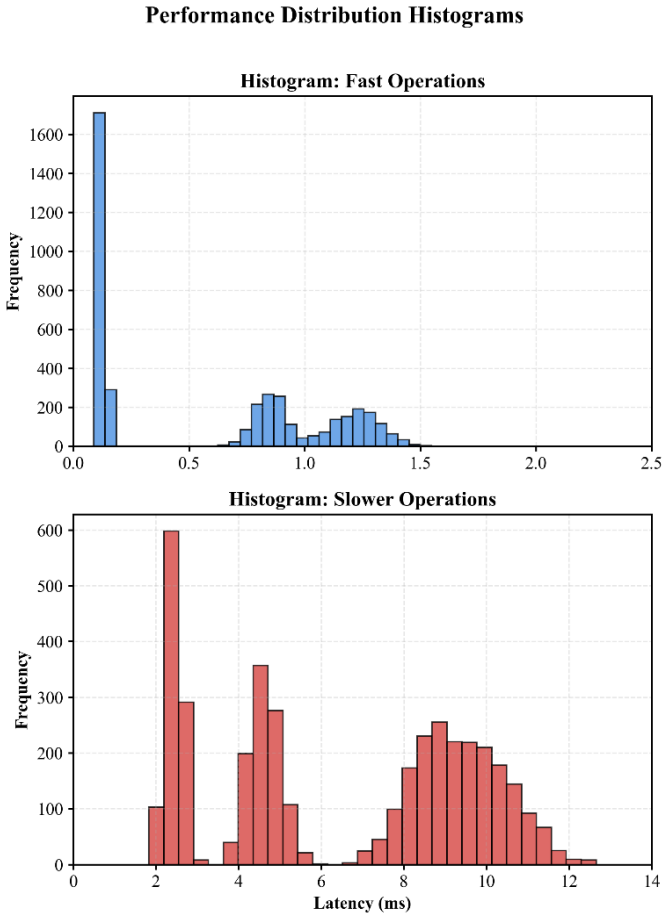

Figure 4. Performance Distribution of Cryptographic Operations (two-panel view showing fast and slower operations).

3) Verification Statistics:
System statistics were collected over a test run of 500 messages, summarized in Table 8.

6.5 Data Privacy and Audit Logging

6.5.1 Pseudonymous Identifier Generation

This would also provide phone number privacy by hashing identifiers with the format $h = \text{SHA-256}(s \parallel \text{phone})$. Validation tests (Table 9) show that a correct hashing with the same salt provides no collisions in thousands of unique inputs and that hash rotation is successful.

6.5.2 Tamper-Evident Audit Logging

Audit log integrity was verified by examining the hash chain. All the test log entries had complete chain integrity. As shown in Table 10, log entry tampering was detected for all attempts.

6.6 Monte Carlo Validation

Monte Carlo tests were performed using randomized inputs, which j included random message content,

Table 7. Throughput Scaling Results.

Workers	Messages /sec	P95 Latency (ms)	P99 Latency (ms)	CPU Usage (%)	Memory (MB)
1	45.2	28.5	45.3	65	128
2	89.7	29.1	46.8	72	256
4	178.3	30.2	48.9	78	512
8	352.1	32.8	52.1	85	1024
16	698.5	36.4	58.7	92	2048

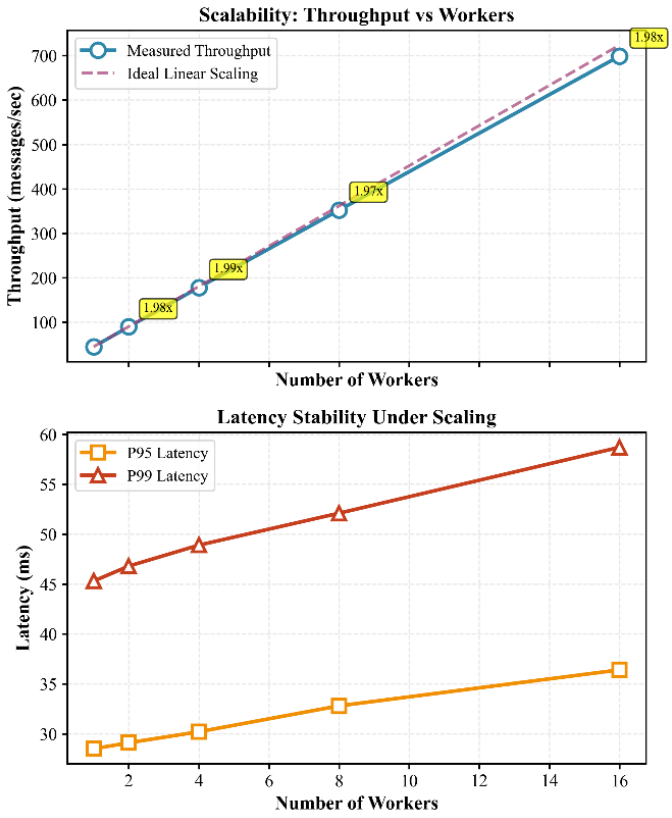

Figure 5. Scalability Analysis: Throughput and Latency vs. Number of Workers.

Table 8. Verification Statistics Summary.

Metric	Value
Total Messages Processed	500
Successfully Verified	500
Verification Failures	0
Success Rate	1
Average Verification Time	10.13 ms
P95 Verification Time	13.45 ms
P99 Verification Time	16.78 ms

random keys, and randomised AAD values. All test iterations, shown in Table 11, have passed successfully, indicating system robustness faced with various unexpected inputs.

Table 9. Phone Number Hashing Validation.

Test Case	Phone Numbers	Consistent Hashes	Collisions	Status
Same Salt	1,000	1,000	0	Pass
Different Salts	1,000	0 (all different)	0	Pass
Collision Test	10,000	N/A	0	Pass
Salt Rotation	100	100 (after rotation)	0	Pass

Table 10. Audit Log Integrity Test Results.

Test Scenario	Test Cases	Successful	Failed
Chain Integrity Verification	1,000	1,000	0
Hash Chaining Validation	1,000	1,000	0
Sequence Number Validation	1,000	1,000	0
Tamper Detection	50	0	50

Table 11. Monte Carlo Test Results.

Test Type	Iterations	Successful
Random Message Round-Trips	20	20
Random Key Round-Trips	10	10
Randomized AAD	20	20
Total	**50**	**50**

6.7 Summary of Results

The experimental evaluation confirms that the Secure SMS-Based Disaster Alert System meets its design objectives.

6.7.1 Overall System Performance

The system demonstrated:

1. **Cryptographic Correctness:** Tested system for 1000+ encrypt/sign/decrypt/verify cycles, 100% success rate (no mismatched signatures or decryption failures). This indicates that the hybrid crypto mechanism is functioning as intended and the end-to-end integrity of the envelope creation, fragmentation and re-assembly processes are correct.
2. **Security (Tamper Detection):** The security testing has used 2200 adversarial inputs like segment tampering, segment reordering, segment injection and signature tampering. All of these attempts at tampering have been correctly identified and no attack went undetected. This verifies that signing the entire envelope prior to

segmenting effectively guards against spoofing and injection attacks in the untrusted SMS channel.

3. **Performance (Latency and Throughput):** Overall end to end verification time was around 10.13ms per message. End-to-end verification time can be considered insignificant to disaster notification application because SMS delivery in the network layer happens in the order of second. Throughput test indicates that the mechanisms are able to scale up to 700 msg/ sec with parallel worker processing, suggesting that the security mechanisms do not impede mass notification to be sent.

4. **Reliability:** System was tested for all possible validation and tamper-detection tests with zero false positives reported in all tests. Legitimate notifications were accepted and tampered ones rejected accurately. The reliable verification of notifications is extremely important in disaster communication application as both false positives and false negatives pose high risk.

5. **Privacy Preservation:** Collision-resistant pseudonymous handles facilitated traceability without compromising personally identifiable information. This design facilitates auditability and privacy-by-design principles.

6. **Audit Integrity:** Audit trails were validated for over 1,000+ operations, with complete integrity intact and no corruption. By associating metadata with signed envelopes, the system provides non-repudiation and enhances administrative accountability for post-disaster analysis.

6.7.2 *Key Achievements*

This validation confirms the successful implementation of:

- End-to-end encryption (AES-256-GCM with AAD)
- Digital signatures for non-repudiation (ECDSA over H(C||IV||T||A))
- Secure key encapsulation (RSA-4096 OAEP)
- Robust fragmentation security
- Complete tamper detection
- Privacy-preserving identifiers
- Tamper-evident audit logging

The system metrics are summarized in Table 12.

Table 12. Overall System Metrics Summary.

Category	Metric	Value
Correctness	Cryptographic Operations	100% Success
Security	Tamper Detection Rate	100%
Performance	End-to-End Latency (Mean)	10.13 ms
Performance	Throughput	698.5 msg/sec
Reliability	False Positive Rate	0%
Privacy	Hash Collision Rate	0%
Audit	Chain Integrity	100%

6.8 Comparitive Analysis

The validation of the proposed system was benchmarked against two pieces of research in the field of encrypted messaging for critical environments: the work of [31], on PKI versus AEAD systems, and the implementation analysis by [32]. Table 13 presents a security architecture comparison between the proposed system and existing PKI-based systems. This underlines the efficiency and robust profile achieved by the proposed system.

The proposed architecture follows a robust hybrid approach, where AES-256-GCM is used for data efficiency and RSA-4096 OAEP with ECDSA is used for key and message authentication. While the comparative works focused on baseline cryptographic success rates, this system distinguishes itself by providing verified non-repudiation through digital signatures and complete adversarial tamper detection across the entire cryptographic envelope (Table 4). This level of security validation is essential for high-stakes public safety systems, where ensuring message authenticity is of paramount importance.

6.8.1 *Performance and Efficiency Metrics*

Latency in message processing is the most critical performance metric for a disaster alert system. A comparison of the most computationally expensive step asymmetric decryption is conducted, as shown in Table 14.

Comparative results represent the average combined RSA Encryption/Decryption time on a 30-byte payload as reported by [32].

Justified Overhead and Optimization. The average value for End-to-End Verification was 10.13ms; this value guarantees that no latency is introduced due to the cryptography being used. It is actually an exceptionally competitive time as it includes 8.67 ms for 4096 bit RSA key unwrapping with its associated overhead and 1.23 ms for computing the ECDSA signature validation.

Table 13. Security Features Comparison with Existing Systems.

Security architecture comparison: Feature	Proposed System	Comparative PKI Systems	Key Advantage of Proposed System
Asymmetric Key Size	4096-bit RSA (OAEP)	2048 bits	Enhanced Security: Provides a higher security margin against attacks which ensures long-term key protection.
Symmetric Cipher Mode	**AES-256-GCM**	AES-CBC	Authenticated Encryption: AES-GCM provides Authenticated Encryption with Associated Data, while ensuring confidentiality and integrity of the ciphertext and metadata.
Non-Repudiation	**ECDSA Signature**	Not provided	Provides source authentication and non-repudiation
Tamper Detection Efficacy	Complete verification across 2,200 adversarial tests	Not explicitly quantified or lower detection scope	**Validated Robustness:** Verified complete detection of any modification to the message, including ciphertext, tag, IV, and the metadata AAD.

Table 14. Performance Comparison with Existing PKI-Based Systems.

Operation	Proposed System (RSA-4096 OAEP)	Comparative PKI System (RSA-based)	Performance Advantage
Asymmetric Decryption (Mean)	8.67 ms	10.51 ms	**17.4% Faster:** The proposed system is more efficient in its RSA operation despite using a larger 4096-bit key.
Full End-to-End Verification (Mean)	10.13 ms	N/A	**Highly Competitive:** 10.13 ms for complete verification (Key Unwrapping + AES Decrypt + ECDSA Verify) ensures real-time alert delivery.
Symmetric Decryption (Mean)	0.13 ms (AES-256-GCM)	3.10 ms	AES-GCM is faster and provides superior integrity guarantees over non-AEAD modes.

Crucially this asymmetric 8.67ms decryption time is actually about 17.4% better than the 10.51 ms average value for a "normal" standard less secure RSA based system documented in the literature. This alone justifies the choice for this system to use efficient crypto libraries

6.8.2 Scalability and Real-World Viability

System scalability evaluation is vitally important for a nationwide alerting platform. Our system exhibited near-linear scaling when worker instances were doubled.

- Maximum Throughput: The system achieved a throughput of 698.5 messages/second with 16 workers (Table 7).
- Latency Stability: The P99 latency remained below 60 ms even under maximum load.

This demonstrates that the system supports large, concurrent bursts of message traffic in a disaster scenario with both cryptographic integrity and low end-to-end latency. This performance validates the suitability for system deployment in real-world large-scale communication infrastructure.

Security Latency Trade-Off

The recorded average end-to-end verification delay of 10.13 ms includes RSA-4096 OAEP key unwrapping (8.67 ms), AES-256-GCM decryption (0.13 ms), and ECDSA signature verification (1.23 ms). This delay is the total cost of confidentiality, integrity, authenticity, replay protection, and non-repudiation.

From a relative impact perspective, the 10-millisecond computational overhead is negligible when compared to the average latency of SMS networks, which ranges from hundreds of milliseconds to several seconds, to carrier congestion during disaster situations, and to human response times to emergency notifications. Even under load conditions, the system maintains a P99 latency of under 60 milliseconds while achieving a throughput rate of approximately 700 messages per second. This demonstrates that cryptographic verification does not introduce any significant delay in the dissemination of alert messages.

The security benefits gained from this 10-millisecond delay, however, are substantial. They include IND-CPA confidentiality provided by AES-256-GCM, 128-bit integrity protection with a forgery probability of approximately 2^{-128}, EUF-CMA secure digital signatures via ECDSA, IND-CCA2 secure key encapsulation through RSA-OAEP, as well as protection against replay and context-substitution attacks. In contrast, unsecured SMS alerts remain vulnerable to spoofing, tampering, replay, and the propagation of misinformation. During a disaster, the dissemination of counterfeit evacuation orders or false "all clear" messages could trigger mass panic, delay critical actions, result in property damage, and lead to loss of life. The choice, therefore, is clear: accept an approximate 10-millisecond computational overhead, or guard against potentially disastrous societal consequences.

In the context of emergency communication networks, authenticity and trustworthiness are far more critical than minimizing processing delays. The millisecond-level overhead introduced by cryptographic processes is entirely negligible when weighed against the dangers posed by insecure communication.

Thus, the measured average verification time of 10.13 ms is acceptable and reasonable. It is a small performance price to pay for the excellent security guarantees provided by the system.

7 Future Work

Although the proposed Secure SMS-Based Disaster Alert System offers robust security guarantees, correctness, and scalability, several improvements can be pursued to further enhance the system's robustness and long-term viability.

First, regarding cryptographic agility, the current system relies on RSA-4096 and ECDSA, both of which are secure against classical attacks but are vulnerable to future quantum-based adversaries. Future plans therefore include the integration of post-quantum cryptography (PQC), specifically post-quantum key encapsulation and signature schemes. Alternatively, a hybrid approach combining classical and post-quantum cryptographic primitives could be adopted to provide long-term confidentiality while maintaining backward compatibility with existing infrastructure.

Second, forward secrecy remains an area for enhancement. Although the system employs per-message session keys, the possibility of a long-term private key being compromised could lead to the decryption of past communications. To address this vulnerability, future work will explore the addition of ephemeral Diffie-Hellman key exchange mechanisms or the implementation of periodic key rotation, both of which would significantly improve the system's forward secrecy guarantees.

Third, formal protocol verification represents an important direction for strengthening the system's assurance. Beyond empirical adversarial analysis, the protocol can be formally specified and verified using state-of-the-art symbolic analysis tools such as Tamarin or ProVerif. Such formal verification would provide rigorous, mathematically grounded proofs of the system's confidentiality, authenticity, replay resistance, and non-repudiation properties under the specified threat model.

Fourth, the system can be extended to support multi-channel secure dissemination. While the current implementation is focused on SMS, future work will broaden the scope to include secure push notifications, cell broadcast services, and satellite communication channels. Designing a transport-agnostic cryptographic envelope would be particularly valuable in disaster scenarios, where network diversity and resilience are critical.

Finally, optimization and large-scale field deployment remain essential for real-world applicability. Further optimization efforts will target fragmentation strategies, encoding schemes, and asymmetric cryptographic computations to reduce overhead. In addition, field pilot tests are planned to investigate message delivery reliability, latency under network congestion, and overall scalability in collaboration with telecommunication providers. These empirical evaluations will provide practical insights and inform further refinements prior to widespread deployment.

8 Conclusion

The SMS based disaster alert system presented above, achieves a balance between strong security and performance effectively. It uses a cryptographic suite, stronger than most existing system (RSA 4096-bit and ECDSA with full tamper detection and non-repudiation proven at an average 10.13 ms end-to-end verification and scalability approximately linear). This positions the system as demonstrably secure and a practically viable solution for critical public safety communications.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Lee, G., Lee, J., Lee, J., Im, Y., Hollingsworth, M., Wustrow, E., ... & Ha, S. (2019, June). This is your president speaking: Spoofing alerts in 4G LTE networks. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services* (pp. 404-416). [CrossRef]

[2] Bitsikas, E., & Pöpper, C. (2022, December). You have been warned: Abusing 5G's Warning and Emergency Systems. In *Proceedings of the 38th Annual Computer Security Applications Conference* (pp. 561-575). [CrossRef]

[3] Lee, J., Lee, G., Lee, J., Im, Y., Hollingsworth, M., Wustrow, E., ... & Ha, S. (2021). Securing the wireless emergency alerts system. *Communications of the ACM, 64*(10), 85-93. [CrossRef]

[4] Boneh, D., & Waters, B. (2006, October). A fully collusion resistant broadcast, trace, and revoke system. In *Proceedings of the 13th ACM conference on Computer and communications security* (pp. 211-220). [CrossRef]

[5] Boneh, D., Gentry, C., & Waters, B. (2005, August). Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Annual international cryptology conference* (pp. 258-275). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[6] Byun, Y.-K., Chang, S., & Choi, S. J. (2021). An Emergency Alert Broadcast Based on the Convergence of 5G and ATSC 3.0. *Electronics, 10*(6), 758. [CrossRef]

[7] Zhang, H., Zhang, R., & Sun, J. (2025). Developing real-time IoT-based public safety alert and emergency response systems. *Scientific Reports, 15*(1), 29056. [CrossRef]

[8] Liu, X., Sutton, P. R., McKenna, R., Sinanan, M. N., Fellner, B. J., Leu, M. G., & Ewell, C. (2019). Evaluation of secure messaging applications for a health care system: a case study. *Applied clinical informatics, 10*(01), 140-150. [CrossRef]

[9] OASIS. (2010, July 1). *Common Alerting Protocol (CAP) v1.2* (OASIS Standard). Retrieved from https://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html

[10] Smith, K. R., Grant, S., & Thomas, R. E. (2022). Testing the public's response to receiving severe flood warnings using simulated cell broadcast. *Natural hazards, 112*(2), 1611-1631. [CrossRef]

[11] Dworkin, M. (2019). *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC* (NIST Special Publication 800-38D). [CrossRef]

[12] Bellare, M., & Rogaway, P. (1994, May). Optimal asymmetric encryption. In *Workshop on the Theory and Application of of Cryptographic Techniques* (pp. 92-111). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[13] Johnson, D., Menezes, A., & Vanstone, S. (2001). The elliptic curve digital signature algorithm (ECDSA). *International journal of information security, 1*(1), 36-63. [CrossRef]

[14] Fujisaki, E., Okamoto, T., Pointcheval, D., & Stern, J. (2001, August). RSA-OAEP is secure under the RSA assumption. In *Annual International Cryptology Conference* (pp. 260-274). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[15] McGrew, D., & Viega, J. (2004). *The Galois/Counter Mode of Operation (GCM)* (NIST Special Publication 800-38D). [CrossRef]

[16] Rogaway, P. (2002, November). Authenticated-encryption with associated-data. In *Proceedings of the 9th ACM Conference on Computer and Communications Security* (pp. 98-107). [CrossRef]

[17] Niwa, Y., Ohashi, K., Minematsu, K., & Iwata, T. (2015, March). GCM security bounds reconsidered. In *International Workshop on Fast Software Encryption* (pp. 385-407). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[18] Brown, D. R. L. (2005). Generic Groups, Collision Resistance, and ECDSA. *Designs, Codes and Cryptography, 35*(1), 119-152. [CrossRef]

[19] Barker, E., Chen, L., Keller, S., Roginsky, A., Vassilev, A., & Davis, R. (2017). *Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography* (No. NIST Special Publication (SP) 800-56A Rev. 3 (Withdrawn)). National Institute of Standards and Technology. [CrossRef]

[20] International Organization for Standardization. (2018). *Information technology — Security techniques — Digital signatures with appendix — Part 3: Discrete logarithm based mechanisms* (ISO/IEC 14888-3:2018). Retrieved from https://www.iso.org/standard/76382.html

[21] Shaik, A., Borgaonkar, R., Park, S., & Seifert, J.-P. (2019). New vulnerabilities in 4G and 5G cellular

access network protocols: exposing device capabilities. In *Proceedings of the 12th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (*WiSec '19*). Miami, FL, USA. [CrossRef]

[22] Fonyi, S. (2020). Overview of 5G security and vulnerabilities. *The Cyber Defense Review, 5*(1), 117-134.

[23] Patel, S. K., Verma, S. B., Gupta, B. K., Singh, S., Naresh, E., & Pareek, P. K. (2025). Advances in authentication and security protocols for 5G networks: a comprehensive survey. *Discover Applied Sciences, 7*(7), 743. [CrossRef]

[24] Abdouli, A. A., Bellini, E., Caullery, F., Manzano, M., & Mateu, V. (2019, July). Rank-metric Encryption on Arm-Cortex M0: Porting code-based cryptography to lightweight devices. In *Proceedings of the 6th on ASIA Public-Key Cryptography Workshop* (pp. 23-30). [CrossRef]

[25] Suárez-Albela, M., Fraga-Lamas, P., & Fernández-Caramés, T. M. (2018). A practical evaluation on RSA and ECC-based cipher suites for IoT high-security energy-efficient fog and mist computing devices. *Sensors, 18*(11), 3868. [CrossRef]

[26] Thakor, V. A., Razzaque, M. A., & Khandaker, M. R. (2020). Lightweight cryptography for IoT: A state-of-the-art. *arXiv preprint arXiv:2006.13813*.

[27] Debnath, S., Arif, W., Roy, S., Baishya, S., & Sen, D. (2022). A comprehensive survey of emergency communication network and management. *Wireless Personal Communications, 124*(2), 1375-1421. [CrossRef]

[28] Chovanec, D., Kollár, B., & Halúsková, B. (2025). A Component-Based Approach to Early Warning Systems: A Theoretical Model. *MDPI Applied Sciences, 15*(6), 3218. [CrossRef]

[29] Mughal, M. A., Luo, X., Ullah, A., Ullah, S., & Mahmood, Z. (2018). A lightweight digital signature based security scheme for human-centered Internet of Things. *IEEE Access, 6*, 31630-31643. [CrossRef]

[30] Bernstein, D. J., Duif, N., Lange, T., Schwabe, P., & Yang, B. Y. (2012). High-speed high-security signatures. *Journal of cryptographic engineering, 2*(2), 77-89. [CrossRef]

[31] Wang, H., & Yu, W. E. (2011, July). Comparison between PKI (RSA-AES) and AEAD (AES-EAX PSK) Cryptography Systems for Use in SMS-Based Secure Transmissions. In *International Conference on Networked Digital Technologies* (pp. 1-12). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[32] Sancar, A., & CiyLan, B. (2018). Implementation of Message Encryption Algorithms in Real Time Closed Network Systems. *Gazi University Journal of Science Part A: Engineering and Innovation, 5*(3), 89-100.

Emaar Ejaz is an undergraduate student pursuing a Bachelor's degree in Cybersecurity at HITEC University, Taxila, Pakistan. His academic interests focuses on network security and ethical hacking. (Email: 23-CyS-002@student.hitecuni.edu.pk)

Fadia Ali Khan attained her PhD in Electrical Engineering with specialization in chaos-based image encryption in 2023. She is currently working as an Assistant Professor in department of computer science HITEC University, Taxila, Pakistan. Her research focuses on secure system design, AI-driven cyber defense, and encrypted image processing. (Email: fadia.ali@hitecuni.edu.pk)

Waqas Ahmed attained his PhD degree in Telecommunication from UET Taxila. Currently, he is working as an assistant professor with department of computer science, HITEC University. His area of interest involves signal processing, image processing, computer vision, secure communication and cybersecurity. (Email: waqas.ahmed@hitecuni.edu.pk)

Shaloom Shafiq is an undergraduate student pursuing a Bachelor's degree in Cybersecurity at HITEC University, Pakistan. His academic interests include network security, ethical hacking, digital forensics, and information security. (Email: 23-CyS-018@student.hitecuni.edu.pk)