International Journal of Emerging Trends in Computer Science and Information Technology
ISSN: 3050-9246 | https://doi.org/10.63282/3050-9246.IJETCSIT-V4I2P121
Eureka Vision Publication | Volume 4, Issue 2, 212-222, 2023

Original Article

Cryptographic Identity Propagation in Asynchronous Event-Driven Architectures: Implementing Zero-Trust Envelopes for High-Velocity Payment Streams

Anvesh Katipelly[1], Sumith Thalary[2]
[1]Senior Software Engineer PayPal, Texas, USA.
[2]Sr Cloud DevOps Engineer, Rexel USA, Dallas TX.

Abstract - The asynchronous event-driven architectures in the modern high-velocity payment systems are increasingly being used to provide scalability, resilience, and low-latency processing. Nevertheless, a serious problem arising out of these architectures is the issue of ensuring secure and trustworthy identity propagation among loosely coupled distributed services. The use of traditional mechanisms like session based authentication, perimeter security is not sufficient in such environment where events move through various independent processing steps and there is no continuous trust boundary between them. The drawback subjects systems to security weaknesses such as image spoofing, replaying, alteration of messages and absence of end to end responsibility. This paper will offer a zero-trust cryptographic envelope system to handle such issues; this system entails directly inserting verifiable identity metadata into each event message and then enforcing digital signatures, encryption, and integrity checks. The strategy of imposing independent checking of the individual service removes implicit trust and guarantees sustained identity checking throughout the sequence of occasions.The suggested methodology consists of creating a structured cryptographic envelope that has payload information, identity statements, and trust validation attributes, combined with such recent streaming platforms. Strong security assurances and performance demands are balanced with efficient cryptographic processes which include ECDSA-based signatures and hybrid encryptions. This is measured against a prototype system which is applied and burdened with high throughput payment loads and is proved to have very low overheads and provides a great deal of added security. The experimental outcomes indicate that the framework provides high throughput and lower latency enhancements are insignificant, serving to counter replay and impersonation attacks and enhance message integrity and non-repudiation. The most significant are a new concept of a zero-trust envelope, a scalable identity propagation design, an efficient implementation framework of real-time payment systems, and an overall analysis that can confirm its effectiveness within high-velocity environments.

Keywords - Event-Driven Architecture, Zero-Trust Security, Cryptographic Identity, Payment Systems, Distributed Systems, Asynchronous Messaging.

1. Introduction

1.1. Background and Motivation

Fast maturation of online financial environments has resulted in the popularisation of real-time payment environments, which require immediate processing, high levels of availability and an unbroken user experience. Organizations are turning to asynchronous event-oriented architecture (EDA), [1] which allows events to be processed on a scale and provide a resilient transaction processing in services by decoupling them. Although, this paradigm allows the network to be flexible and allows tolerance of errors, it has a serious drawback of end-to-end security and propagation of identities. As compared to synchronous, asynchronous pipelines typically have continuity of authentication and authorization context lost as events pass-through several and independent services, which pose a risk to integrity and reliability of identity information in the life-cycle processing of identity information.

1.2. Problem Statement

Although event-driven architectures have their benefits, the current strategies do not have effective mechanisms of secure identity propagation on asynchronous pipelines. Bearer based and transport-layer security techniques are inadequate where messages are stored, replayed or received out of sequence. This puts systems at high-risk areas such as replay attack, impersonation and message tampering. Moreover, a lack of constant validation of trust results in implicit trust among services and generates security holes in distributed settings. Such constraints create a need to implement a framework, which will guarantee safe, verifiable, and unalterable identity propagation throughout event processing phases.

1.3. Research Objectives

The study will strive to come up with a complete model of cryptographic identity propagation in asynchronous event systems with major priority being the assurance of identity continuity in the distributed services. [2] The offered solution

imposes a zero-trust security model where every service individually validates the authenticity and integrity of the local events and thus without implicit trust assumptions. Moreover, the study underlines a good balance between security and performance so that the time spent on real-time payment systems is not high in terms of latency but in terms of throughput it should be maximum without sacrificing scalability, interoperability, and industry standards.

1.4. Contributions of the Paper

The paper has added value to the topic because it presents a new type of zero-trust model of cryptographic envelopes that safely binds identity metadata, payload data, and cryptographic proofs in every event and makes sure that identity propagation cannot be tampered with. It suggests an extensible chain-of-trust protocol to ensure identity continuity across services that are not time sensitive and provides a design that implements zero trust concepts without the need to maintain a centralised trust system. In addition, the research offers a detailed analysis of performance-security trade-offs, shows that overheads are minimal, and that the implementation of the strategy is practical when the systems are high-velocity payments and need great security, scalability, and regulatory compliance.

2. Related Work

2.1. Identity Management in Distributed Systems

Distributed systems Identity management has been based traditionally on the centralized frameworks based on OAuth 2.0, OpenID Connect, and SAML which is developed as a synchronous request-response model that identities remain within the context of the session or a token. [3] As microservices have emerged, applications of decentralization derived call-to-call authentication such as JSON Web Tokens (JWTs) and mutual TLS (mTLS) have been implemented to allow authentication between services. Yet these mechanisms usually rely upon some implicit trust and are ill adapted to asynchronous settings in which messages may be held and pass through many services. Decentralized identity (DID) and verifiable credentials are an effective alternative emerging, but they will only see broad use in high-volume event management with performance and integration challenges.

2.2. Zero-Trust Security Models

The concept of zero-trust security models has emerged as organizations transition on perimeter-based security controls to ongoing verification tool strategies founded on the idea, never trust, always verify. Evaluations like the Zero Trust Architecture of NIST also focus on extreme validation of identity and enforcement of the policy at each access point. These concepts become distributed systems, which are service meshes, identity-aware proxies and API gateways. The majority of implementations are however based on a network or request-level security but they are ineffective and do not effectively extend to asynchronous messages. Consequently, after a message has moved into a streaming system, the downstream services can tend to be based on existing validation, thus providing a lapse in the implementation of the zero-trust principle at the message level.

2.3. Event-Driven and Streaming Architectures

Event-driven architectures have emerged as the foundation in the construction of resilient and scaled systems especially in high throughput systems like the financial service sector. [4] The systems such as Kafka, Pulsar, and RabbitMQ allow asynchronous communication where the services may be used in isolation and scale effectively. Although this decoupling is beneficial in providing flexibility and performance, it brings in contextual information encompassing identity and authorization challenges across the distributed services. Current security controls such as transport-layer encryption and access control on the broker level secure data in transit without end to end message security and spread identity propagation and trust verification are unresolved issues.

2.4. Cryptographic Message Protection Techniques

The use of cryptographic techniques is vital in guaranteeing the security of distributed systems in terms of data confidentiality, integrity and authenticity. Encryption is commonly through AES, key exchange through RSA/ECC and authentication through digital signatures. Such standards as JSON Web Signature (JWS) and JSON Web Encryption (JWE) allow protecting structured data on a message level. But simple applications of these techniques at endpoints in event-driven systems are not correct, and do not make it possible to propagate identity dynamically across processing stages. Moreover, the calculation time of the repetitive cryptography function may affect the performance of high-velocity systems.

2.5. Limitations of Existing Approaches

Although identity management and security has been improved, the existing schemes have a few drawbacks in event-driven systems that are asynchronous. Most systems do not support end-to-end identity continuity hence fragmented trusts along line of processing. [5] At the transport and token-level, there is inadequate security against replay and impersonation attacks to assure security when messages are stored or replayed. The principles of zero-trust do not extend to the message-level verification and dependence on centralized providers of identity create the issue of scale and resiliency. Moreover cryptography operation overheads do not allow achievement of balance between security and efficiency. The mentioned

restrictions put into focus the necessity of a scalable and effective framework to combine cryptographic identity propagation with zero-trust concepts, which is suggested in this work.

3. System Model and Problem Formulation

3.1. Architecture of Asynchronous Event-Driven Systems

The implementation of modern, high-velocity payment platforms is based more and more on asynchronous event based architectures (EDA), with system components interacting based on loosely coupled message exchanges and no longer in the form of synchronous request response interactions. [6] This model has event producers who publish transaction events over a messaging infrastructure e.g. distributed logs or message brokers and independent consumers who process the events in an asynchronous fashion. The architecture usually involves the microservice structure, producers, brokers, and storage systems and the observability layer collaborate to provide scalability, suspicibility and real-time processing. Scheduling of events traversing several services, and may be stored, replayed or processed in an arbitrary sequence, however, makes it difficult to guarantee and verify identity across the overall pipeline. This is because of the lack of a continuum of session or communication channel, which creates the possibility of a lapse in the identity propagation continuity and trust.

3.2. Threat Model

Table 1. Threat Model and Security Requirements

Threat Type	Description	Mitigation in ZTCE Framework
Replay Attack	Reuse of previously valid messages	Timestamp + unique ID + signature validation
Message Tampering	Unauthorized modification of payload	Cryptographic hashing + signature verification
Identity Spoofing	Forged identity claims	Digital signatures + PKI validation
Unauthorized Access	Access to sensitive data	Encryption (AES + key exchange)
Insider Threat	Compromised internal service	Zero-trust verification at each hop

To know the security risk of asynchronous event-driven systems, there must be a detailed threat model. Attackers in such systems can possess an ability of accessing messages, replaying it, tampering with it, impersonating, and compromising the system partially. These attacks take advantage of the distributed and decoupled structures of event pipelines where they can be viewed in transit or stored in a persistent data structure like logs and queues. The attack surfaces cut within the message brokers, inter-service communication channels, API gateways and key management systems. The malicious events can spread between the services without detection because of the absence of the inherent message-level verification. Hence, the design of a security model that presupposes a low degree of trust and places highly cryptographic validation on each processing level is an extremely important task.

3.3. Security Requirements

To reduce the perceived threats, it will be imperative that the system meets the major security needs which guarantee trustworthiness and strength. [7] This is done by authentication to help accept the origin of each event, where the legitimate entities can only produce valid messages. Integrity ensures that event data does not get corrupt when delivering the information and storing it so that it can be monitored to identify any unauthorized changes. Non-repudiation guarantees that senders will never be in a position to deny their actions, which is a requirement of financial systems in the auditability and regulation. Confidentiality ensures that sensitive data is not accessed by unauthorized persons bmts.vt encrypts the because of hiding data. The combination of these needs leads to the need to have a strong, message-level security strategy which will support constant verification and adherence to the principle of zero trust throughout the entire event lifecycle.

3.4. Design Constraints

It is important to ensure robust security though the suggested framework should be able to work within the realistic parameters of actual payment systems. A very high latency is a very high specification because processing of payment has to be critical with hard timing constraints frequently in the milliseconds range, constraining the tolerable costs of cryptography operations. Throughput is also critical as the systems should be able to process huge amount of transactions, without slowing down their performance. Moreover, event-driven systems have the essential nature of fault tolerance, which demands the security system to help in message replay, distributed processing, and failure recovery without the loss of identity verification. It should also be designed in a way that it does not introduce bottlenecks and encourages decentralized trust which will be in line with either the scalability and resilience needs of the modern distributed architecture.

4. Proposed Framework: Zero-Trust Cryptographic Envelope

4.1. Framework Overview

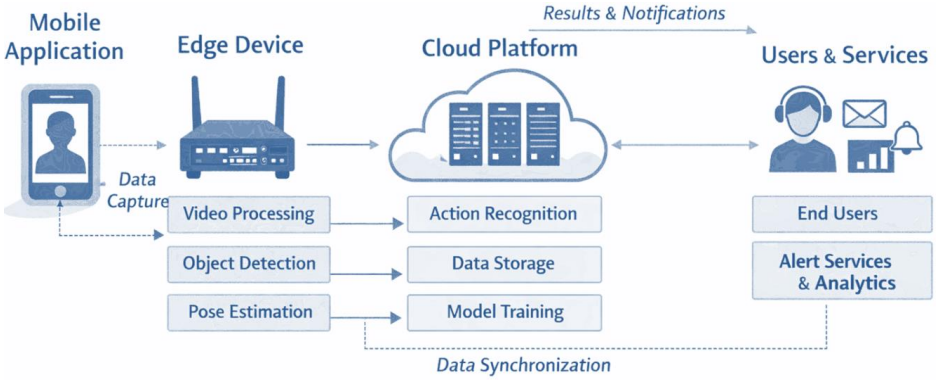

Figure 1. Zero-Trust Cryptographic Envelope (Ztce) Framework

The suggested Zero-Trust Cryptographic Envelope (ZTCE) system will make identity propagation in asynchronous event-driven systems and architectures to be secure and verifiable by placing trust within every event message. In contrast to the older methods that were based on transport-layer security or a centralized identity verification, this model follows a message-centric design, where each event is in a message-based model with its cryptographic evidence of authenticity and integrity. [8] The messages are encapsulated with a structured envelope that included payload data, identity metadata, signatures, and optional encryptions to ensure that the security is extended throughout the optimization and processing, including storage, replay, and multi-hop communication. Zero-trust implementation, end-to-end identity continuity, tampering resistance, decentralized verification, and performance are service-specific design principles that drive the design to support zero-trust enforcement, an end-to-end identity continuity without implicating any implicit trust.

4.2. Cryptographic Identity Envelope Structure

Table 2. Cryptographic Envelope Structure

Component	Description	Security Purpose
Payload	Core business data (e.g., payment transaction details)	Data integrity & processing
Identity Metadata	Sender ID, timestamp, message ID, roles, permissions	Identity propagation & context
Signature Block	Digital signature (RSA/ECDSA) over payload and metadata	Authentication, integrity, non-repudiation
Encryption Layer	AES (payload) + asymmetric key exchange	Confidentiality
Chain-of-Trust Info	Signatures from intermediate services	Auditability & traceability

The cryptographic identity envelope comprises the centre of the framework, which entraps all the components of the safe and secure communication and identity propagation. It has the payload, the business data and is considered unchangeable, and identity metadata that provides the definition of the sender, origin, timestamp, and authorization context. Authentication, integrity and non-repudiation are achieved using a signature block that binds the payload and metadata by cryptographic concepts on the identity of the sender. Moreover, optional encryption layer provides security hybrid encryption methods, which involve using symmetric encryption to provide a high level of efficiency and privacy with a non-volatile asymmetric encryption to provide the high-order of the key exchange. This hierarchy design is the way of making sure that any change in message is detected and only trusted parties can reach sensitive data.

4.3. Identity Propagation Mechanism

The framework provides a strong mechanism of identity propagation since the cryptographic envelope will carry identity information not using external tokens. [9] Such an in-house method is a sure way of ensuring that identity context accompanies the message and can be verified at each processing level. As the events go through several services, each service can verify the message coming in and can add its signature thus creating a chain-of-trust. The chained verification model will offer a transparent and auditable view of all the processing processes wherein we will be certain that each step has been verified and no unauthorized changes have been made. The mechanism makes there more accountable, assists forensic analysis and ensures the continuity of identity in distributed systems.

4.4. Key Management Strategy

The security and scalability of the framework requires effective key management. The given strategy capitalizes on the Public Key Infrastructure (PKI) where every organization will be provided with the distinctive key pair, with the private keys being applied in the signing process and the public one being distributed with the help of trusted certificates. This allows the

verification of the message authenticity independently without direct intercommunication between services. Key rotation and revocation mechanisms are also included as part of the framework, to deal with any possible security risk, and a compromise or expired key is invalidated as soon as possible. Opinite validation strategies such as caching and distributed verification are used to ensure that the overhead in the performance is kept to the barest minimum and at the same time there is high security assurance.

4.5. Trust Verification Workflow

The trust verification workflow implements zero-trust principles at all levels of processing events, that is, it is mandatory that all services individually verify received messages. [10] Each time an event is received, the service will verify the digital signatures, identify the integrity of the payload and metadata, and verify the cryptographic keys. Assuming the encryption is used, the payload is safely decrypted then more processing is carried out. The service then compares the embedded identity claims to local authorization policies in order to decide on whether the event will be run. Once the validation has been successful, the service might add its signature and send the message to the next step, and the chain-of-trust can be continued. This running check ensures that all the events are authenticated, tamper-proof and meet the requirements of security policies during its lifecycle.

5. Architecture Design

5.1. High Level Architecture Diagram

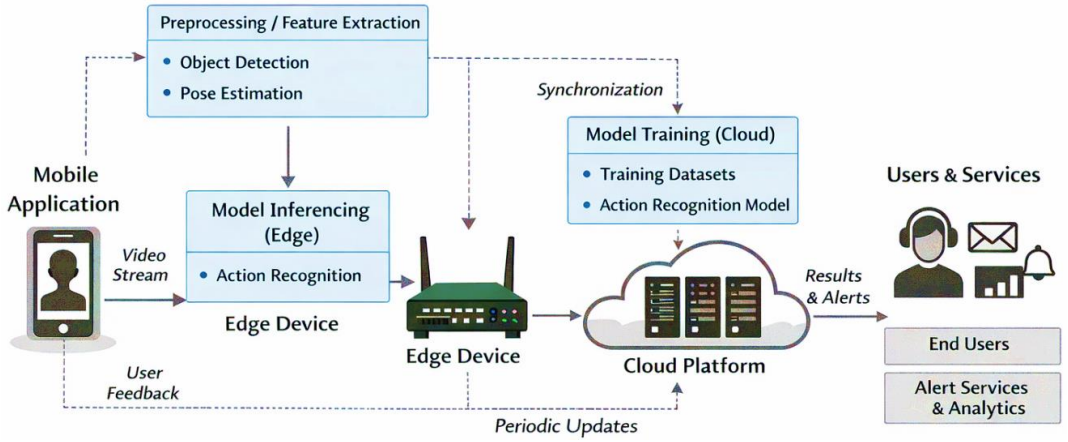

Figure 2. High Level Architecture Diagram

The architecture suggested to propagate the cryptographic identities of zero-trust is organized in the form of a layered event-driven structure integrating security into the data flow directly. [11] It is composed of ingress layer which takes care of the initial authentication and the creation of the cryptographic envelope and event streaming layer that is used to guarantee a reliable ingestion and persistence as well as distribution of events. Validation, fraud detection, authorization and settlement is done within a processing layer that consists of several independent microservices where each service verifies the message and re-signs the message. The layers that will be supported are security and key management layer to manage the operations of the PKI and observability layer to monitor and audit. This design provides assurance of identity and trust within the entire event lifecycle coupled with the state of isolation of concerns and scalability.

5.2. Event Flow in Payment Streams

High velocity payment system is designed in which events follow a multi-stage asynchronous pipeline with each stage taking part and enriching the message. [12] A transaction starts at the ingress layer where it is authenticated and encapsulated in a cryptographic envelope and then it is published to a message broker. Following services, like those of validation, fraud detection, etc., take the event, check its integrity and authenticity, and add their respective signatures thereafter. The investigation is then followed by authorization and settlement procedures then notification and logging services. The identity and trust context is preserved and payment transactions can be safely, audibly, and reliably processed across this pipeline.

5.3. Integration with Messaging Systems

The architecture is created to be platform independent with popular messaging systems like Apache Kafka, Apache Pulsar, and RabbitMQ. The cryptographic envelope is built into the message payload which means it can be carried over to various streaming platforms without any security guarantees being compromised. The aspects of integration include standardization of with an intensity format like Avro or Protobuf, and effective serialization to minimize overhead and backward compatibility via versioned schema. Although the transport-layer security mechanisms including TLS do secure message-in-flight data, the embedded cryptographic envelope makes message security end-to-end, independent of the messaging infrastructure underneath.

5.4. Service-to-Service Communication Model

The architecture is based on asynchronous and loosely coupled model of communication between services in the form of events and not direct API calls. [13] Messages are read by each service to the broker and the message is cryptographically verified, and the business logic of the message is executed and the message is again signed and sent to the next service. This methodology does not rely on shared session context and identity information can always be available with embedded metadata. The model complies with the principle of zero-trust and does not place any trust on upstream components as it must make sure that there is no service that is dependent on the implicit trust of the other components in the architecture and the architecture is scalable and resilient.

5.5. Scalability and Fault Tolerance Design

The architecture suggested is scalable and fault tolerable to deal with the requirements of high-paced payment system. Scaling can be performed based on the workload of services on a horizontal basis, whereas parallel processing of distributed nodes is made possible by partitions of event streams. Cryptographic operations are optimization based by the parallel execution and optimization of resources used to ensure high throughput. A durable message storage, replay supports and idempotent processing provide fault tolerance, so that any failure following the failure will not lead to loss of data and inconsistencies. The enhancement of the cryptographic envelope integration contributes to further resilience by allowing the replay of the occurrences and retaining the ability to verify the identity in the recovery process, as well as, offering elaborate audit trails that can be used in debugging and compliance audits.

6. Implementation Details

6.1. Technology Stack

It is the implementation of Zero-Trust Cryptographic Envelope (ZTCE) framework with the foundation of a cloud-native, distributed technology stack that facilitates high-throughput and low-latency processing of events in payment systems. [14] The ingestion of messages, permanent storage, and parallel processing of them in a partition manner is supported by a distributed streaming platform, like Apache Kafka or Apache Pulsar. Preference is often based on Kafka because it has a rich ecosystem and a high guarantee of ordering in the partitions, making it possible to have a structured design of topology in different workflows of the payment lifecycle. The framework also uses standardized cryptographic libraries, including OpenSSL, Bouncy Castle and Libsodium in order to support secure primitives to perform encryption, digital signatures, and hashing, and hardware acceleration methods can be applied in order to improve overall get-up-and-go performance.

6.2. Envelope Encoding and Serialization

Encoding and serializing the cryptographic envelope efficiently is also necessary to reduce overhead and guarantee service interoperability between distributed service providers. [15] JSON provides more flexible and easily readable format that is easy to develop and debug but incurs presence of increased payload size and parsing cost. Contrastingly, binary formats like Apache Avro or Protocol Buffers are compact-encoded, more serialization-efficient, as well as ranging with schema evolution, implying in their turn a better fit in the production environment. The implementation plan requires Protobuf or Avro to be used in critical systems with high performance, and JSON is selectedively to be used in non-production. Versioning and compatibility across services is handled using schema registries that are used to ensure that the envelope structure is consistently interpreted.

6.3. Cryptographic Algorithms Used

Table 3. Cryptographic Algorithms Used

Algorithm	Type	Advantages	Limitations	Use in Framework
RSA	Asymmetric	Widely supported, strong security	Large key size, slower	Legacy compatibility
ECDSA	Asymmetric	Faster, smaller keys	More complex implementation	Preferred for signatures
AES	Symmetric	High performance, efficient	Requires secure key exchange	Payload encryption
Hybrid	Combined	Secure + efficient	Added complexity	End-to-end encryption model

The framework uses an integration of asymmetric and symmetric cryptographic to secure it but not compromise performance. Digesting signatures take place of a bmtsvt RSA or ECDSA to ensure authentication, integrity, and non-repudiation by attaching the message-processing to the identity metadata and the sender. ECDSA is accepted and usually better than would have been offered because of smaller key size and quicker calculation that is appropriate in high-performance places. To maintain confidentiality, jismeN symmetric encryption like AES is applied on the payload and asymmetric key exchange is maintained by asymmetric encryption. Such a combination provides the opportunity to work with high volumes of data: to process them and at the same time to provide high security levels.

6.4. Performance Optimization Techniques

The framework achieves high performance standards of real-time payment systems by incorporating a number of optimization techniques. [16] Multiple events are grouped together by use of batching, such that cryptographic operations can occur on a large scale as well as minimize overhead on the network without causing much latency. Parallel verification Since parallel verification characterizes multiple processing units, it can be used to meet multicore processing architectures to enhance throughput and optimize bottlenecks. Also, they use caching to save some common known public keys and certificates, which reduces the necessity of repeating the validation process in case and lessens the latency. All these optimizations combine such that the framework offers high security without affecting the scalability or the performance of the system.

7. Experimental Setup and Evaluation

7.1. Experimental Environment

To determine the outcomes of the suggested framework, Zero-Trust Cryptographic Envelope (ZTCE), under a controlled experimental setting, a high-velocity payment processing system was simulated. [17] The architecture is comprised of a distributed cluster that has several computation nodes that have multi-core processors, large memory size and storage based on SSD to create low-latency access to data. The communication delay is distributed over a high-speed network interconnect which minimizes delays in communication. Software environment consists of Linux based operating system, event streaming powered by Apache Kafka and containerized microservices built to coordinate bmtsvt Kubernetes. The libraries like OpenSSL and Bouncy Castle provide cryptographic operations, and schema management is provided by a registry of Avro or Protobuf format. The system configuration includes partitioned topics, groups of consumers and configurable cryptographic parameters that allows evaluating fully performance and functional correctness.

7.2. Workload Description

The test is a synthetic workload that is used to simulate high velocity payment streams that would be found in contemporary financial systems. [18] The workload is used to model transaction counts in thousands to millions per second of transactions including different phases like payment initiation, validation, fraud detection, authorisation, settlement and notification. The realistic payload sizes are created and the events are monitored using uniform as well as bursty traffic patterns to put the system to test of scalability and elasticity. Every task is surrounded by a cryptographic envelope, new cryptographic signatures at each level are done, and sensitive data are encrypted rather than decrypted which allows maintaining a line-of-trust through services. Performance is measured by comparing a baseline system which is dependent on the security of transport layer with the proposed structure of ZTCE framework allowing to clearly analyze the benefits of security and overheads.

7.3. Evaluation Metrics

Table 4. Performance Evaluation Metrics

Metric	Description	Importance
Latency	End-to-end processing time per event	Critical for real-time payments
Throughput	Events processed per second	Measures scalability
Security Overhead	Additional CPU, memory, and message size	Evaluates efficiency
Verification Time	Time for signature and integrity validation	Impacts processing speed

The system has both efficiency and robustness metrics which are measured by evaluating key performance and security metrics. Latency is a measure of the end-to-end processing time of events, such as serialization, cryptographic processing and network latencies, and is mostly concerned with the extra latency caused by security services, and the effects of this extra latency on the real-time service demands. Throughput is used to assess the number of events served per second among the producers, brokers and consumers, which is the capacity of the system to maintain high workloads with cryptographic limitations. Security overhead is used to measure the costs related to encryption and signature methods in terms of computation time and consumption of hardware and memory. Verification time is used to determine the performance of cryptographic verification in every hop of the service, with tests on signatures and validations of certificates. This set of measures gives a holistic view of the trade-offs in the proposed framework between security and performance.

8. Results and Analysis

8.1. Performance Benchmarking

Performance assessment of Zero-Trust Cryptographic Envelope (ZTCE) system proves that the system is highly efficient even though cryptographic functions are included in it. [19] Latency analysis indicates a moderate growth of end-to-end time to process, which is in the order of 515 percent with the underlying system, majorly because of signature generation, verification, and optional encryption. Latency is, however, kept with acceptable processes in real-time payment systems, which are typically less than 100 milliseconds; it is further reduced by processes like parallel verification and caching. Throughput analysis- it has been found out that the system can maintain a high rate of event processing with minimal degradation; typically

in the range of 3-10%. Linear scaleability making it suitable to high-velocity payment environments regardless of whether the number of processing nodes is increased or not also shows that the framework can support and execute at high speed.

8.2. Security Effectiveness Analysis

The suggested framework immensely enhances the security positioning of asynchronous event-driven systems by offering identity assurance and message protection across the end. The experimental findings prove that replay can be resistant to replay attacks with unique identifiers and timestamps and replay attacking has almost perfect detection. Digital signature is a school of thought that makes impersonations hard and difficult because the only message accepted has valid private key signature and the integrity of the packets and metadata is effortlessly established after an integrity check policy is produced. Moreover, the chain-of-trust model provides complete event tracing, which is useful in upholding non-repudiation and compliance. In general, the framework provides a strong, multi-layered security framework in accordance with zero-trust principles.

8.3. Overhead vs Throughput Trade-offs

The analysis suggests the trade-offs between increased security and system performance indicating that the extra overhead imposed by encryption and decryption activities is manageable. [20] These calculations of signature verification are the major cause of the computational overhead, and encryption has comparatively low costs when used selectively. The addition of identity metadata and signatures adds about 20-40% to the message size but this effect is alleviated by using efficient binary serialization systems like Avro and Protobuf. Even in the presence of this overhead, there is stable throughput whose reduction is allowed at reasonable values, usually not more than 10%. The observations reveal that the framework provides an effective tradeoff between security and performance, which is appropriate to those production-grade financial systems where trust and integrity are paramount.

8.4. Comparative Analysis with Baseline Systems

Comparative analysis of the proposed ZTCE framework against the current operating system shows that there are huge security gains without compromising the competitiveness. The baseline system, based on the use of tokens to identify identity propagation, and transport level security provides good raw performance but does not provide extensive protection against replay attacks, impersonation and message tampering. Contrarily, the suggested framework proposes the deployment of cryptographically bound identity propagation and of a zero-trust message-based security model, which guarantees end-to-end integrity, authentication, and non-repudiation. Despite the slight improvement in latency and the slight decrease in the throughput that the framework causes, the trade-offs are not insignificant when the security, auditability and compliance benefits are considered to be significant. As a result, the given strategy is more suitable in the context of mission-critical financial applications where the strong trust guarantees must be ensured.

9. Discussion

9.1. Advantages of the Proposed Approach

The suggested Zero-Trust Cryptographic Envelope (ZTCE) design is superior to conventional security designs in asynchronous event-based architectures, especially in high-speed payment systems, as it has a number of benefits. It is an end-to-end identity assurance mechanism that entails placing identity metadata data in each message and cryptographic binding of the metadata with the payload, which maintains the identity integrity of each message throughout the processing phases. The implementation of a zero-trust model removest any implicit trust assumption as each service must verify incoming messages on its own, eliminating good security against the replay attack, tampering of messages, and spoofing of identities. Decentralized trust model also enriches systems resilience by allowing inter-local verification of systems by using distributed trust anchors as opposed to having centralized identity providers. Moreover, the chain-of-trust approach ensures better auditability features and adherence to standards; that is, a beneficial record of all processing operations remains possible. The framework can be easily combined with the current cloud-native and microservices architecture and has shown to perform well in a balanced way with optimization techniques, which makes it appropriate to real-time financial systems.

9.2. Limitations and Challenges

Although it is advantageous, the framework presents a number of challenges that should be addressed well. Cryptographic functions like encryption and digital signatures impose computational cost to the system, which might reduce the performance of high throughput conditions. Addition of identity metadata and signatures adds more size to message that could impact bandwidth and storage capacity of the network. Key management is one of the most complicated ones as it involves secure distribution, rotation as well as revocation of cryptographic keys and any of these could be potentially detrimental to the entire system. Legacy systems can be challenged by having to make major changes to message format, service logic, and infrastructure to fit into current architectures. Though there is a reduction in latency overhead, some of the ultra-low-latency applications are sensitive to extra processing delays. Moreover, multi-hop pipelines make the chain-of-trust more hard to administer and may add to envelope size and verification, so effective mechanism to handle the chain-of-trust are vital.

9.3. Practical Deployment Considerations

The framework should be deployed successfully because of full planning and alignment with the existing system architectures. The incremental approach towards adoption will enable the organization to implement cryptographic envelopes with essential services gradually and spread through the pipeline without losing backward compatibility. The compatibility with the already in-place infrastructure (based on API gateways, service meshes, and messaging system) ensures that essential disruption is minimal, and interoperability tracked. The secure Key storage, the automatic rotation, and the flawless management of the key revocation require a strong key management system or PKI infrastructure. Parallel processing and hardware acceleration as well as efficient serialization formats should be used to achieve latency and throughput requirements. To monitor the performance of the system to identify anomalies and to be able to rely on the functioning, full monitoring and observability are required. Lastly, and most importantly, regulatory and compliance requirements are important in financial systems, and auditability and non-repudiation characteristics inherent in the framework implement these requirements.

10. Use Cases in Financial Systems

10.1. Real-Time Payment Processing

Zero-Trust Cryptographic Envelope (ZTCE) has demonstrated its effectiveness in the real-time payment systems where the maximum possible limits on the transaction speed (high latency) with high security assurances must be provided. By incorporating identity information that is cryptographically verifiable at every event, the framework could help to preserve and verify identity at every point in the transaction cycle, such as validation, authorization and settlement. Every service checks in the integrity and authenticity of the message alone and then processes and eliminates implicit trust and minimizes the risk of unauthorized transactions. This is done to maximize security, low latency by optimized cryptographic functions, as well as end-to-end traceability a chain-of-trust, to meet regulatory compliance and audit needs.

10.2. Fraud Detection Pipelines

The ZTCE framework helps in the validation of authentic and tamper-free incoming events, in the case of fraud detecting pipelines that receive large quantities of transaction information in real time. The framework enhances the precision of risk assessment models by presenting authenticated identity context in every message and minimizing prospects of false positive and errors. Chain-of-trust mechanism traces all the processing steps so that it makes it possible to share data with analytics services securely, as well as conduct thorough forensic investigations. This guarantees that the detection of frauds is made using trustworthy data and at the same time the auditability and the compliance is maintained at high level in high throughput systems.

10.3. Cross-Border Transaction Systems

Cross-border payment systems entail various institutions and different jurisdictions and therefore secure and reliable exchange of data is required. The ZTCE model facilitates propagation of identities between heterogeneous systems in an interoperative way by ensuring cryptographic proofs are embedded in each message meaning that participating entities are free to validate the authenticity of transactions by their own means. This model of decentralized verification helps to decrease its use of any central authority, increasing the trust between parties. The framework also reduces fraud and controversy by ensuring good non-repudiation and offers all audit trails required to comply with the regulatory requirements across regions. Consequently, it has enabled the safe, open, and effective cross-border transactions within complicated financial systems.

11. Future Work

11.1. Integration with Decentralized Identity (DID)

The combination of the Zero-Trust Cryptographic Envelope framework and decentralized identity models along with verifiable credentials are one of the main directions of the future work. In contrast to the old traditional identity systems where centralized authorities are used to manage identity, decentralized identity allows entities to handle and control their identity information, and also to avoid relying on single points of trust. The framework can foster inter-organizational trust and improve the privacy by engraving decentralized identifiers on the cryptographic envelope and adding verifiable credentials. Nevertheless, issues of zero-tolerance to the efficient incorporation of decentralized identity resolution mechanisms into high-throughput event pipelines, the control of the added latency, and the interoperability with the existing PKI-based infrastructures represent a challenge.

11.2. AI-Driven Anomaly Detection

The other improvement that could help is the introduction of artificial intelligence and machine-based learning to allow the active threat detection within the event-based systems. Using the trusted and untampered information contained in the cryptographic envelope, AI models can process streams of transactions in real time, identify abnormalities, deviations in behavior, and even abnormal behavior and possible patterns of fraud. The methods of this approach make it possible to dynamically score risks and have contextually accurate security policies. Future investigations are needed on building light models, which run under strict latency limits, combining anomaly detection with the chain-of-trust to make contextual decisions, implementing automatic incident response systems to enhance system resilience.

11.3. Post-Quantum Cryptography Readiness

The framework on post-quantum cryptography is a crucial preparation on the road to a long-term security in the light of increasing quantum computing power. Classical cryptographic methods like RSA and ECDSA can fall prey to quantum attacks and quantum-resistant methods must be implemented. Improvements in the future could involve incorporating lattice-based, hash-based or code-based crypto algorithms as well as building crypto-agile architecture with smooth replacement of cryptographic primitives. Difficulties such as dealing with greater computational complexity, increased key sizes, and compatibility with the existing systems are all challenges. The framework can be made safe by integrating post-quantum preparedness, which allows it to be resistant to threats posed by new technologies.

12. Conclusion

The paper covered the vital issue of safe identity propagation in event-driven architecture, and specifically the high-velocity payment system. With the growing trend toward distributed, event-driven models offered by modern financial platforms to obtain scalability and resilience, the traditional security mechanisms have not been observed to be adequate in ensuring a consistent and verifiable identity across loosely coupled services. A lack of end to end trust presents flaws like replay attacks, identity spoofing, and tampering of messages, which can be of critical concern to system integrity and reliability. In order to encounter such challenges, this work suggested a Zero-Trust Cryptographic Envelope (ZTCE) scheme that inserts identity, integrity and trust checks into event messages. Through the use of a message-based security framework, the framework can guarantee that each service of the processing pipeline locally authenticates and verifies the integrity of events received. It is designed using cryptographic primitives like cryptographic signatures and cryptography, and chain-of-trust mechanism that maintains identity continuity through several processing phases.

The experimental analysis has shown that the suggested framework delivers high levels of security at the same time the system performance is high. Findings are that extra latency and throughput overheads imposed by cryptographic mechanisms are within tolerable levels of real time payment system. Moreover, the framework does not only provide defense against widespread attack vectors, such as replay attacks and impersonation, but also allows for full auditability and non-repudiation by providing verifiable processing trails.The implications of this work are not limited to payment systems, though, as the framework can be used to address the concept of secure event-driven architecture. The proposed solution incorporates the concept of zero-trust and cryptographic identity propagation and creates a strong basis to create trustful, scalable, and resilient distributed systems. It allows organizations to abandon non-explicit models of trust and adopt constant, decentralized validation, which is necessary in the contemporary context of cloud-native and microserves-based ones.

Finally, the ZTCE framework provides a feasible and scalable means of securing asynchronous event pipelines, which is a good balance between security and performance. With event-driven systems becoming more and more dynamic and complex to manage, especially with workloads that can be considered sensitive, a transition to such zero-trust, cryptographically-enforced systems will become essential in ensuring data integrity, system trustworthiness, and regulatory compliance. Further future developments in decentralized identity and AI-based security, and post-quantum crypto ought to improve the functionality of this framework to enable the next generation of secure distributed systems.

Reference

[1] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture. NIST special publication, 800 (207), 1-52.
[2] Buterin, V. (2014). A next-generation smart contract and decentralized application platform. white paper, 3(37), 2-1.
[3] Emily, H., & Oliver, B. (2020). Event-driven architectures in modern systems: designing scalable, resilient, and real-time solutions. International Journal of Trend in Scientific Research and Development, 4(6), 1958-1976.
[4] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.
[5] Koblitz, N. (1987). Elliptic curve cryptosystems. Mathematics of computation, 48(177), 203-209.
[6] Diffie, W., & Hellman, M. E. (2022). New directions in cryptography. In Democratizing cryptography: the work of Whitfield Diffie and Martin Hellman (pp. 365-390).
[7] Pautasso, C., Zimmermann, O., & Leymann, F. (2008, April). Restful web services vs." big'"web services: making the right architectural decision. In Proceedings of the 17th international conference on World Wide Web (pp. 805-814).
[8] Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, omega, and kubernetes. Communications of the ACM, 59(5), 50-57.
[9] Kreps, J., Narkhede, N., & Rao, J. (2011, June). Kafka: A distributed messaging system for log processing. In Proceedings of the NetDB (Vol. 11, No. 2011, pp. 1-7).
[10] Hunt, P., Konar, M., Junqueira, F. P., & Reed, B. (2010). {ZooKeeper}: Wait-free coordination for internet-scale systems. In 2010 USENIX Annual Technical Conference (USENIX ATC 10).
[11] Sabelfeld, A., & Myers, A. C. (2003). Language-based information-flow security. IEEE Journal on selected areas in communications, 21(1), 5-19.

[12] Ahmed, M. R., Islam, A. M., Shatabda, S., & Islam, S. (2022). Blockchain-based identity management system and self-sovereign identity ecosystem: A comprehensive survey. Ieee Access, 10, 113436-113481.
[13] Dib, O., & Toumi, K. (2020). Decentralized identity systems: Architecture, challenges, solutions and future directions. Annals of Emerging Technologies in Computing, 4(5), 19–40. https://doi.org/10.33166/AETiC.2020.05.002
[14] Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207
[15] Kapoor, B., Pandya, P., & Sherif, J. S. (2011). Cryptography: A security pillar of privacy, integrity and authenticity of data communication. Kybernetes, 40(9-10), 1422-1439.
[16] Cho, J. H., Xu, S., Hurley, P. M., Mackay, M., Benjamin, T., & Beaumont, M. (2019). Stram: Measuring the trustworthiness of computer-based systems. ACM Computing Surveys (CSUR), 51(6), 1-47.
[17] Ali, B., Hijjawi, S., Campbell, L. H., Gregory, M. A., & Li, S. (2022). A maturity framework for zero-trust security in multiaccess edge computing. Security and Communication Networks, 2022(1), 3178760.
[18] Yang, Y. S., Lee, S. H., Wang, J. M., Yang, C. S., Huang, Y. M., & Hou, T. W. (2023). Lightweight authentication mechanism for industrial IoT environment combining elliptic curve cryptography and trusted token. Sensors, 23(10), 4970.
[19] John, V., & Liu, X. (2017). A survey of distributed message broker queues. arXiv preprint arXiv:1704.00411. https://arxiv.org/abs/1704.00411
[20] Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z., & Doss, R. (2022). Zero trust architecture (zta): A comprehensive survey. IEEE access, 10, 57143-57179.
[21] Pendleton, M., Garcia-Lebron, R., Cho, J. H., & Xu, S. (2016). A survey on systems security metrics. ACM Computing Surveys (CSUR), 49(4), 1-35.
[22] Savola, R. M. (2013). Quality of security metrics and measurements. Computers & Security, 37, 78-90.
[23] Jimmy, F. (2022). Zero trust security: Reimagining cyber defense for modern organizations. International Journal of Scientific Research and Management, 10(4), 887–905. https://doi.org/10.18535/ijsrm/v10i4.ec11
[24] Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z., & Doss, R. (2022). Zero trust architecture (ZTA): A comprehensive survey. *IEEE Access*, *10*, 57143–57179. https://doi.org/10.1109/ACCESS.2022.3174679