ENQRYPTED

Zero-Knowledge One-Time QR Messaging

enqrypted.net

Abstract

A zero-knowledge messaging system would allow organizations and privacy-conscious individuals to communicate without trusting service providers with message content or cryptographic keys. While many current platforms implement end-to-end encryption, they still present vulnerabilities through centralized key management, metadata collection, and reliance on platform integrity for security guarantees. We propose a solution using client-side cryptography where all encryption occurs in the user's browser before transmission, operating on a zero-trust architecture where service providers never access keys or plaintext. The system supports both secure messaging and file sharing, storing only encrypted data on servers without decryption keys, using Argon2id key derivation and XChaCha20-Poly1305 authenticated encryption. Messages and files automatically self-destruct after 24 hours, successful reading or three failed decryption attempts, with QR code sharing enabling cross-device access. As long as users maintain password security, the system provides mathematical privacy guarantees independent of service provider integrity.

1. Introduction

Modern communication depends heavily on platforms that require trust in service operators. While end-to-end encryption prevents servers from reading message content, centralized infrastructure still introduces vulnerabilities through metadata collection, key distribution, and potential backdoors. Even with strong operational security, users remain exposed to risks from infrastructure compromise, targeted key exchange attacks, and legal coercion of service providers.

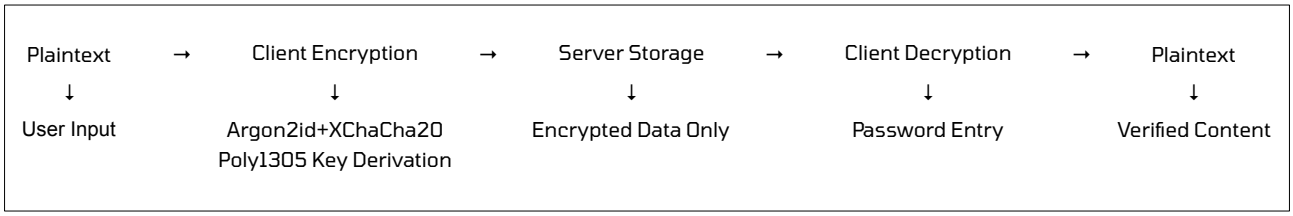
What is needed is a messaging system based on cryptographic proof instead of operational trust, allowing parties to communicate without requiring confidence in infrastructure providers. A zero-knowledge architecture where servers cannot access any meaningful information - neither content nor comprehensive metadata - would eliminate these risks, while automatic deletion would satisfy retention requirements. We propose client-side cryptography that ensures mathematical privacy guarantees without depending on server behavior or policy promises.

This approach serves both enterprise organizations requiring secure business communications and privacy-conscious individuals who value protection of their personal conversations and sensitive information.

2. Zero-Knowledge Architecture

We define secure messaging as a system where message content never exists in decryptable form on servers. The sender encrypts content using password-based key derivation before transmission, and only the recipient with the correct password can decrypt the message.

Message Flow:



The server only stores encrypted ciphertext along with minimal lifecycle metadata (attempts, createdAt, and a messageId generated with UUID). Without the correct password, no one can decrypt the ciphertext. Even with the password, no information about the sender or recipient can be inferred, since messageIds are generated randomly and never linked to any user identity.

3. Cryptographic Implementation

The system uses battle-tested primitives to achieve IND-CCA2 security:

Key Derivation: Argon2id transforms user passwords into cryptographic keys using 256MB memory cost and 3 iterations, preventing GPU-based attacks while maintaining usability [1].

Authenticated Encryption: XChaCha20-Poly1305 AEAD provides confidentiality and integrity with 192-bit nonces, ensuring unique encryption for each message [2].

Random Generation: Cryptographically secure randomness powers salts, nonces, and message identifiers using Web Crypto API [3].

Message Structure

```
{
    "salt": "base64-encoded-32-bytes",
    "nonce": "base64-encoded-24-bytes",
    "ciphertext": "base64-encoded-encrypted-data",
    "attempts": 3,
    "createdAt": Unix timestamp
}
```

The salt ensures unique keys per message, the nonce prevents replay attacks, and the ciphertext contains authenticated encrypted content. Attempt limits prevent brute-force guessing while expiration enforces temporal boundaries.

4. Lifecycle Security

Messages enforce strict lifecycle controls to minimize long-term risk:

Time-Based Expiration: 24-hour maximum lifetime prevents indefinite retention regardless of access patterns.

Consumption-Based Deletion: Successful decryption immediately triggers message destruction, ensuring single-use semantics.

Attempt Limiting: Maximum 3 decryption attempts prevents systematic password guessing while allowing for user error.

Automated Cleanup: Background processes continuously remove expired content, maintaining system hygiene and compliance.

5. Cross-Device Sharing

QR codes solve the cross-device sharing problem while maintaining anonymity. Unlike traditional platforms requiring user accounts, contact lists, or phone number verification, Enqrypted enables instant message access through scannable codes that contain only encrypted message identifiers.

Identifier Generation: Each message receives a UUID v4 identifier with 122 bits of entropy, preventing enumeration attacks [4].

QR Code Creation: Message identifiers are encoded into QR format client-side, eliminating server knowledge of sharing patterns.

Device Compatibility: Standard QR scanners on any device can access messages through web browsers, requiring no app installation.

Access Control: Only users with both the QR code (or identifier) and the correct password can decrypt content.

6. Privacy Guarantees

The system provides computational zero-knowledge privacy through cryptographic design:

Server Blindness: Infrastructure cannot access message content, sender identity, or recipient information beyond encrypted data size.

Unlinkability: Messages cannot be correlated with users or organizations due to anonymous identifiers and minimal metadata.

Forward Secrecy: Per-message keys prevent historical compromise from affecting other communications.

Metadata Minimization: Only essential cryptographic parameters are stored, eliminating unnecessary data collection.

Association Prevention: No social graphs, contact lists, or persistent user profiles exist in the system.

7. System Architecture

Browser-Based Operation: JavaScript client with libsodium-wrappers-sumo performs all cryptographic operations locally.

Serverless Infrastructure: Stateless functions handle encrypted data storage and retrieval without maintaining user state.

HTTPS Transport: TLS 1.3 encryption protects data in transit while CSRF protection secures API interactions.

Compliance Integration: Automatic deletion, minimal data retention, and zero logging satisfy regulatory requirements like GDPR.

Cost Efficiency: Serverless architecture scales with demand while reducing infrastructure overhead compared to traditional messaging platforms.

8. Security Analysis

Threat Model: The system withstands attacks from compromised servers, state-level surveillance, and quantum-enabled adversaries through symmetric-only cryptography.

Attack Vectors:

- **Server Compromise:** Only encrypted data accessible, no decryption capability
- **Network Interception:** HTTPS transport protection with authenticated encryption
- **Brute Force:** Memory-hard key derivation and attempt limits prevent practical attacks
- **Quantum Threats:** 256-bit symmetric keys maintain 128-bit effective security against quantum computers

Trust Assumptions: Security depends on client-side cryptographic implementation, password strength, and Web Crypto API integrity rather than server operator behavior.

9. Use Cases and Comparative Analysis

Different communication scenarios reveal the limitations of trust-based platforms:

Secure Credential Sharing: IT teams distributing API keys or access credentials require single-use delivery that prevents historical access. Traditional platforms store messages indefinitely on devices, creating long-term exposure risks. Enqrypted's consumption-based deletion ensures credentials cannot be retrieved after successful viewing.

Legal Communications: Attorney-client privilege demands both confidentiality and unlinkability. While Signal provides strong encryption, phone number registration creates identity correlation risks under legal discovery. Enqrypted eliminates identity associations entirely through UUID-based anonymous messaging.

Whistleblower Protection: Source protection requires absolute anonymity and message deniability. Platforms requiring registration or maintaining social graphs compromise source security through metadata analysis. Enqrypted provides mathematical guarantees that communications cannot be traced to individuals or organizations.

Enterprise Compliance: Regulated industries face strict data retention requirements that conflict with security needs. WhatsApp and Telegram maintain indefinite message storage, violating data minimization principles. Enqrypted's automatic deletion ensures compliance while maintaining operational security.

Privacy-Conscious Individuals: Personal users sharing sensitive information (financial details, passwords, private documents) face the same trust vulnerabilities as enterprises. Traditional platforms store message histories accessible to device compromises or service provider breaches. Enqrypted's ephemeral architecture ensures personal communications leave no permanent digital footprint.

Comparative Security Properties:

Property	Conventional Platforms (Signal / WhatsApp / Telegram)	Enqrypted
Registration	Requires phone number or account binding	Anonymous access, no personal identifiers
Persistence	Stores message history and backups	Single-use, ephemeral consumption
Metadata	Collects usage patterns and communication graphs	Only minimal cryptographic parameters
Trust Model	Relies on operational security and provider promises	Mathematical privacy guarantees (zero-knowledge)

10. Implementation Results

Performance benchmarks on modern devices:

Key Derivation: ~2-3 seconds on average hardware using 256MB memory

Encryption: ~100ms for typical message sizes using XChaCha20-Poly1305

QR Generation: <50ms client-side processing for message identifiers

11. Conclusion

We have proposed a messaging system that eliminates server-side access to communication content through client-side cryptography. The system provides mathematically guaranteed privacy by storing only encrypted data that cannot be decrypted without user passwords. Temporal controls enforce automatic deletion while QR codes enable cross-device sharing without persistent accounts.

The architecture proves that zero-knowledge messaging is practically achievable using proven cryptographic primitives. Organizations gain absolute privacy guarantees rather than depending on operational security measures, with built-in compliance and cost advantages from serverless infrastructure.

Privacy is transformed from a service provider promise into a mathematical certainty that cannot be compromised by infrastructure vulnerabilities, surveillance demands, or malicious operators.

References

[1] Biryukov, A., Dinu, D., & Khovratovich, D. (2015). "Argon2: the memory-hard function for password hashing and other applications"

[2] Arciszewski, S. (2018). XChaCha: eXtended-nonce ChaCha and AEAD_XChaCha20_Poly1305. IETF Internet-Draft, draft-irtf-cfrg-xchacha-03

[3] World Wide Web Consortium (W3C). (2017). "Web Cryptography API". W3C Recommendation

[4] Leach, P., Mealling, M., & Salz, R. (2005). "A Universally Unique IDentifier (UUID) URN Namespace". RFC 4122