Digest Mining Ciphers: Novel Hash-Based Pre-Image Security Primitives

Cris, DOSAYGO

December 27, 2024

Abstract

This note introduces a novel hash-based encryption scheme named **Digest Mining Encryption** (DME), which leverages the pre-image resistance and nonlinear properties of cryptographic hash functions. Unlike traditional encryption methods that utilize P-boxes, S-boxes, or nonlinear algebraic transformations, DME relies on mining plaintext blocks from a nonce and key by solving for digest mappings. The scheme supports multiple digest selection modes (prefix, sequence, scatter) and optional compression, offering flexibility and robust resistance against brute-force and timing attacks. Additionally, DME aligns with hash-based post-quantum cryptographic techniques, presenting potential for quantum-resistant asymmetric systems. Challenges such as ciphertext expansion and mining speed are addressed through optimization and compression. A proof-of-concept implementation is provided to facilitate further exploration and analysis by the cryptographic community.

Introduction

Traditional cryptographic primitives, such as block ciphers, achieve security through carefully designed P-boxes (permutation), S-boxes (substitution), and algebraic nonlinear transformations. These components introduce diffusion and confusion, creating ciphertext that resists cryptanalysis.

In contrast, **Digest Mining Encryption** (DME) relies solely on the one-way, nonlinear properties of cryptographic hash functions. Encryption involves solving a *digest mining problem*, where plaintext blocks are *mined* from the output of a keyed hash function and a random nonce. This inversion of normal cryptographic principles introduces a new paradigm, where security derives from the difficulty of pre-image search.

The scheme encrypts plaintext by mapping blocks into keyed hash digests. Each plaintext block is constructed by selecting specific parts of the digest using a mapping function, defined by the chosen encryption mode. The process incorporates optional compression to reduce redundancy and mitigate ciphertext expansion.

Disclaimer: This document introduces an experimental concept intended for research and analysis by cryptographers. The scheme presented here has not been formally proven secure or subjected to rigorous cryptanalysis.

Motivation

Digest Mining Encryption is inspired by the inherent cryptographic properties of hash functions, which include:

1. Nonlinearity and One-Wayness

Cryptographic hash functions are inherently nonlinear and computationally one-way. These properties ensure that:

- Given a hash output, recovering the input is computationally infeasible (pre-image resistance).
- Small changes to the input produce unpredictable and significant changes to the output (avalanche effect).

By leveraging these properties, DME avoids the need for traditional nonlinear components like S-boxes and instead relies on hash digests as the foundation for security.

2. Resistance to Timing Attacks

The encryption process involves stochastic mining of a nonce such that the resulting digest matches the desired mapping. This inherently probabilistic process ensures that:

- Encryption time varies unpredictably.
- Fixed execution patterns, which are often exploited in timing attacks, are eliminated.

3. Q-Day Resistance

Hash functions are well-regarded for their resistance to quantum adversaries:

- Grover's algorithm reduces brute-force complexity from 2^n to $2^{n/2}$, but sufficiently large hash outputs remain secure.
- Cryptographic hash functions are widely adopted in post-quantum designs, including hash-based signature schemes like SPHINCS+.

DME aligns with this philosophy by building its security on the same primitives, making it inherently resistant to quantum adversaries.

4. Potential for Integration with Hash-Based Key Exchange

Hash-based constructions like SPHINCS+ demonstrate the viability of hash functions in asymmetric cryptography. DME could potentially integrate with these schemes, enabling the development of quantum-resistant asymmetric encryption systems that extend the principles of digest mining.

Encryption: Core Equation

The encryption process solves the following equation:

$$\text{Map}(H(K \parallel N)) = P$$

where:

- H: A cryptographic hash function (e.g., SHA-256, BLAKE3).
- K: A secret key with sufficient entropy (e.g., 128 bits).
- N: A per-block random nonce (e.g., 64+ bits; larger nonces increase attack hardness).
- P: The plaintext block (e.g., 3 bytes, though any length is permitted).
- Map: A mapping function selecting parts of the digest $H(K \parallel N)$ to construct P.

Digest Mining as Encryption

Plaintext is derived from the digest via:

$$P = H(K \parallel N) \otimes \text{Map}$$

where $\otimes$ represents the selection operation. Encryption requires mining a nonce N such that the hash digest satisfies the mapping constraints defined by Map.

Ciphertext Encoding

Each encrypted block is represented as:

$$C = (N, \text{indices})$$

where N is the nonce and indices defines the mapping used to reconstruct P from $H(K \parallel N)$.

Encryption Modes

The mapping Map can operate in the following modes:

- **Prefix Mode:** Matches the first n-bytes of the digest.
- **Sequence Mode:** Matches a contiguous substring of the digest.
- **Scatter Mode:** Matches bytes at arbitrary indices in the digest, ensuring no duplicates.

Scatter mode ensures that plaintext redundancy does not reveal patterns in the digest indices.

Decryption

Decryption reconstructs the plaintext block P as:

$$P = H(K \parallel N) \otimes \text{indices}$$

The hash digest is computed with the provided N and K, and the mapping defined by indices extracts the plaintext P.

Security Analysis

Pre-Image Security

The security of the scheme relies on the pre-image resistance of H. Specifically, given P, N, and indices, an attacker must solve:

$$H(K \parallel N) = P$$

The effective brute-force effort is:

$$2^{|K|+|\text{nonce}|}$$

where $|K|$ and $|\text{nonce}|$ are the bit lengths of the key and nonce, respectively.

Preventing Redundancy Patterns

Scatter mode ensures that each index is unique, preventing repeated plaintext bytes from being mapped to the same digest index. This avoids revealing plaintext patterns and ensures uniform utilization of the digest.

Ciphertext Expansion

Ciphertext expansion is determined by:

$$\text{Overhead} = |\text{nonce}| + |\text{indices}|$$

Scatter mode incurs higher overhead due to the need to store an index for each byte of P. Optional compression reduces redundancy in the plaintext before encryption, offsetting this overhead.

Timing Security

Mining the random nonce N involves a probabilistic search for a valid mapping $\text{Map}(H(K \parallel N)) = P$. The stochastic runtime behavior ensures that timing attacks are mitigated, as the encryption time varies unpredictably due to secure randomness in the nonce.

Tradeoffs

- **Efficiency vs. Expansion:** Prefix and sequence modes minimize ciphertext size but take longer to mine than the looser matching of scatter mode. Scatter mode is faster to mine but increases expansion.
- **Compression:** Reduces plaintext redundancy, often resulting in smaller effective ciphertext size despite normal expansion overhead.
- **Mining Speed:** Smaller blocks are faster to mine encrypt, but larger blocks reduce the total number of blocks and nonces.
- **Timing Security vs. Constant Time Operations:** While stochastic runtime enhances timing security, it means encryption is probabilistic and does not execute in constant time.

Tradeoff Summary

Aspect	Advantage	Limitation
Block size	Smaller blocks speed up mining	Larger ciphertext due to nonce overhead
Ciphertext expansion	Mitigated by compression	Compression adds preprocessing complexity
Mining modes	Scatter mode provides faster mining	Requires storing an index for every plaintext byte
Timing security	Stochastic runtime eliminates fixed patterns	Mining is probabilistic and not constant time

Proof of Concept

The repository contains a working proof-of-concept toy implementation demonstrating the feasibility of this hash-based encryption scheme. This toy is not attacked or analyzed and should not be used by third parties (i.e., you!) for securing valuables. The toy cipher does not use established cryptographic hash functions but instead utilizes the hashes defined in this repository, which are high quality and fast hash functions that pass SMHasher3. The code supports all defined digest search modes (prefix, sequence, scatter) and uses a constant key per session, without a key schedule. It also incorporates compression for improved storage efficiency.

The repository can be found at: https://github.com/DOSAYGO-Research/rain.

To build and run the example:

```
# Clone the repository
git clone https://github.com/DOSAYGO-Research/rain
cd rain

# Build using make
make

# or if encountering problems, try
./scripts/build.sh

# Example usage
# Encrypt a file with sequence mode
./rain/bin/rainsum -m enc --search-mode sequence input_file.txt

# Decrypt the file
./rain/bin/rainsum -m dec encrypted_file.rc

# Verify the decrypted file matches the original
diff input_file.txt decrypted_file.txt
```

The repository includes a suite of test scripts to validate the correctness of the implementation across various configurations, ensuring decrypted contents match the originals. These tests systematically evaluate different combinations of hash functions, digest sizes, nonce sizes, block sizes, and input files. The test suite can be executed as follows:

```
./scripts/test_cipher.sh
```

Future Work

Future enhancements include:

- Introducing a key schedule to vary K across blocks, improving resistance against related-key attacks, potentially using the hash function in an extendable-output (XOF) mode.
- Optimizing for parallelized mining to accelerate encryption.
- Extending compatibility with alternative hash functions for performance tuning.
- Formalizing security proofs under standard cryptographic assumptions, including resistance to differential and pre-image attacks.
- Investigating integration with hash-based key exchange schemes like SPHINCS+ to develop quantum-resistant asymmetric encryption systems.
- Evaluating the impact of different compression algorithms on ciphertext size and encryption speed.

Conclusion

Digest Mining Encryption (DME) flips traditional cryptographic principles by leveraging cryptographic hash functions for nonlinear one-way transformations. By relying on pre-image search for plaintext block construction, DME provides robust resistance to brute-force and timing attacks. The scheme's alignment with hash-based post-quantum techniques offers a foundation for exploring quantum-resistant cryptographic systems. While challenges such as ciphertext expansion and mining speed remain, ongoing improvements aim to balance efficiency and security. This experimental concept invites scrutiny and analysis from the cryptographic community.