Security & Best Practices

Table of Contents

---

1. Overview

1.1 Intended Security Properties
1.2 Experimental Nature Disclaimer

2. Strength Characteristics

2.1 Resistance to Pattern Analysis
2.2 Stateful Transformation Behavior
2.3 Variable Key Construction

3. Critical Limitations

3.1 No Authentication / Integrity Guarantees
3.2 No Built-in Error Detection
3.3 Verification Ambiguity
3.4 Susceptibility to Incorrect Inputs
3.5 Lack of Formal Cryptographic Analysis

4. Attack Surface Considerations

4.1 Brute Force Characteristics
4.2 Mnemonic Guessing
4.3 Lexicon Exposure Risks
4.4 Known-Plaintext Scenarios
4.5 Statistical Artifacts

5. Operational Risks

5.1 Human Error
5.2 Mnemonic Reuse
5.3 Weak Lexicon Construction

6. Best Practices

6.1 Strong Lexicon Generation
6.2 Mnemonic Selection Strategies
6.3 Index Selection Guidance
6.4 Secure Storage of Lexicons

7. Recommended Use Cases

7.1 Experimental / Educational Use
7.2 Obfuscation Use Cases
7.3 Controlled Environments

8. Not Recommended For

8.1 High-Security Applications
8.2 Situations Requiring Proven Cryptography

9. Future Work

9.1 Formal Analysis Opportunities
9.2 Potential Improvements

---

1. Overview

1.1 Intended Security Properties

Vizini2 is designed to address a core weakness of traditional substitution ciphers: susceptibility to pattern analysis.

Rather than relying on fixed mappings, the system introduces:

- Variable-depth word expansion into Chunks
- Non-uniform weight distribution across Chunks
- Stateful index evolution during transformation

These properties aim to:

- Reduce direct frequency correlations between input and output
- Introduce structural variability across transformations
- Increase the effort required to analyze or reverse outputs without full system knowledge

The system does not attempt to replicate or replace modern, formally analyzed cryptographic algorithms. Its design emphasizes structural complexity and transformation variability rather than provable security guarantees.

1.2 Experimental Nature Disclaimer

Vizini2 is an experimental system.

- It has not undergone formal cryptographic analysis
- Its security properties are not mathematically proven
- Its behavior has not been validated against established attack models

As such:

- It should not be considered a secure encryption system
- It should not be relied upon for protecting sensitive or critical data

This document outlines observed characteristics and best practices, not guarantees of security.

2. Strength Characteristics

2.1 Resistance to Pattern Analysis

A primary design goal is reducing the effectiveness of pattern-based attacks.

This is achieved through:

- Multiple Chunks per word (Depth)
- Variable Chunk weights
- Non-linear selection of Chunks during mnemonic resolution

As a result:

- Identical input values do not consistently produce identical output values
- Output distributions may not directly reflect input frequency

However, resistance is contextual and dependent on lexicon quality, mnemonic structure, and usage patterns.

2.2 Stateful Transformation Behavior

Vizini2 inherits and extends the stateful processing model from earlier versions.

- Each transformation step updates an internal index
- Subsequent operations depend on the evolving state
- Output is influenced by prior transformations

This introduces:

- Order sensitivity
- Dependency across the entire dataset
- Increased difficulty in isolating individual transformations

2.3 Variable Key Construction

The Active Key is derived dynamically from:

- The lexicon
- The mnemonic
- The structural properties of associated words

This results in:

- Non-uniform key material
- Context-dependent resolution of mnemonic inputs
- Increased variability compared to static key systems

The same mnemonic may produce different effective behavior when paired with different lexicons.

3. Critical Limitations

Some limitations are intentional and contribute to ambiguity and resistance to brute-force analysis. However, they also introduce operational and security risks.

3.1 No Authentication / Integrity Guarantees

Vizini2 does not provide:

- Message authentication
- Integrity verification
- Tamper detection

Modified data will still produce output without indication of corruption.

3.2 No Built-in Error Detection

The system does not detect:

- Incorrect parameters
- Corrupted input
- Partial or incomplete transformations

Errors propagate silently through the transformation process.

3.3 Verification Ambiguity

This system does not provide a built-in mechanism to verify the correctness of a result.

Given:

- a Lexicon

- a mnemonic
- starting indices

the Engine will always produce an output. However:

- Incorrect parameters may yield outputs that appear valid or meaningful
- There is no internal checksum, signature, or validation mechanism
- Multiple distinct parameter sets may produce plausible results

As a result:

- Successful transformation does not imply correctness
- External validation is required when correctness matters

This ambiguity increases resistance to brute-force attempts, but also increases risk of user error.

3.4 Susceptibility to Incorrect Inputs

Small changes in:

- Mnemonic
- Lexicon
- Starting indices

can produce entirely different outputs.

There is no safeguard against:

- Typographical errors
- Misremembered parameters
- Mismatched configurations

3.5 Lack of Formal Cryptographic Analysis

The system has not been:

- Peer-reviewed
- Subjected to formal proofs
- Evaluated against standard cryptographic attack models

Its security characteristics should therefore be treated as observational, not guaranteed.

4. Attack Surface Considerations

4.1 Brute Force Characteristics

Brute-force attempts require simultaneous discovery of:

- The correct lexicon
- The correct mnemonic
- The correct starting indices

The search space can be large, particularly with well-constructed lexicons.

However:

- No formal bounds on resistance are established
- Effectiveness depends heavily on user choices

4.2 Mnemonic Guessing

A mnemonic alone is insufficient without the corresponding lexicon.

- The same mnemonic behaves differently across lexicons
- Without lexicon knowledge, mnemonic guessing has limited value

However, weak or predictable mnemonics reduce overall resistance.

4.3 Lexicon Exposure Risks

Lexicons function similarly to key material.

If a lexicon is exposed:

- The search space for attackers is significantly reduced
- Mnemonic guessing becomes more feasible

Lexicons should therefore be treated as sensitive data.

4.4 Known-Plaintext Scenarios

If an attacker has access to both input and output:

- Structural characteristics of the transformation may be partially inferred
- Repeated analysis may reveal patterns tied to lexicon construction

Risk depends on:

- Lexicon complexity
- Mnemonic reuse
- Volume of known data

4.5 Statistical Artifacts

Output may exhibit non-ideal statistical properties, including:

- Occasional compressibility
- Non-uniform byte distribution

These artifacts may:

- Reveal structural characteristics
- Reduce resistance to advanced analysis techniques

5. Operational Risks

5.1 Human Error

The system relies heavily on correct parameter input.

- No recovery mechanisms exist
- No correctness validation is provided

User error is one of the most significant risks.

5.2 Mnemonic Reuse

Reusing the same mnemonic:

- May introduce repeatable structural behavior
- May increase exposure in known-plaintext scenarios

However, impact is moderated by:

- Stateful processing
- Lexicon variability

5.3 Weak Lexicon Construction

Poorly constructed lexicons reduce effectiveness.

Examples include:

- Low depth (few Chunks per word)
- Uniform or predictable weight distribution
- Small or repetitive vocabularies

Lexicon quality directly affects system behavior.

6. Best Practices

6.1 Strong Lexicon Generation

- Use large and diverse text sources
- Favor higher depth with variability
- Use non-uniform weight expressions
- Avoid overly small or uniform lexicons
- Append mode can help create more natural structural variation

6.2 Mnemonic Selection Strategies

- Avoid common phrases or predictable patterns
- Use sufficient length and variability
- Avoid reuse across unrelated data

6.3 Index Selection Guidance

- Avoid trivial or repeated starting indices
- Treat indices as part of the key material
- Vary indices across operations when possible

6.4 Secure Storage of Lexicons

- Treat lexicons as sensitive assets
- Avoid storing alongside transformed data
- Use secure storage practices appropriate to the environment

7. Recommended Use Cases

7.1 Experimental / Educational Use

- Studying transformation behavior

- Exploring alternative cipher constructions

7.2 Obfuscation Use Cases

- Non-critical data masking
- Situations where deterrence is sufficient

7.3 Controlled Environments

- Systems where all parameters are tightly managed
- Scenarios where ambiguity is acceptable

8. Not Recommended For

8.1 High-Security Applications

Vizini2 is not suitable for protecting sensitive data.

This may change only after:

- Formal analysis
- Defined minimum construction standards
- Peer review

8.2 Situations Requiring Proven Cryptography

Do not use Vizini2 where:

- Regulatory compliance is required
- Established cryptographic standards apply
- Failure would have significant consequences

9. Future Work

9.1 Formal Analysis Opportunities

- Statistical evaluation of output distributions
- Resistance testing against known attack models
- Comparative analysis with established ciphers

9.2 Potential Improvements

- Formally defined lexicon construction standards
- Schema/version enforcement
- Additional tooling for validation and testing

If you have feedback, suggestions, or would like to contribute, please reach out via the project contact channel.