CryptoGL Cryptographic Algorithms Specifications

CryptoGL Project Documentation

July 2, 2025

Contents

1 Introduction

This document provides comprehensive specifications, official links, test vectors, and detailed attack specifications for all cryptographic algorithms implemented in the CryptoGL project. The project includes a wide range of cryptographic primitives including block ciphers, stream ciphers, hash functions, MAC algorithms, and classical ciphers.

2 Attack Specifications

2.1 Differential Cryptanalysis

Description: A chosen-plaintext attack that analyzes how differences in input affect differences in output. **Complexity**: Typically requires 2^{n+1} chosen plaintexts for n-bit block ciphers. **Method**:

1. Choose plaintext pairs with specific differences
2. Analyze ciphertext differences through the cipher rounds
3. Use differential characteristics to recover key bits
4. Apply statistical analysis to distinguish correct key guesses

References: https://en.wikipedia.org/wiki/Differential_cryptanalysis

2.2 Linear Cryptanalysis

Description: A known-plaintext attack that exploits linear approximations of the cipher. **Complexity**: Requires approximately $2^{n/2}$ known plaintexts. **Method**:

1. Find linear approximations of S-boxes and round functions
2. Construct linear expressions involving plaintext, ciphertext, and key bits
3. Collect many plaintext-ciphertext pairs
4. Use statistical analysis to determine key bits

References: https://en.wikipedia.org/wiki/Linear_cryptanalysis

2.3 Brute Force Attack

Description: Exhaustive search through all possible keys. **Complexity**: 2^k operations where k is the key length in bits. **Method**:

1. Try all possible key values systematically
2. For each key, encrypt known plaintext and compare with ciphertext
3. Stop when correct key is found

2.4 Collision Attack (Hash Functions)

Description: Finding two different inputs that produce the same hash output. **Complexity**: Birthday paradox - approximately $2^{n/2}$ operations for n-bit hash. **Method**:

1. Generate many random inputs and compute their hashes
2. Store hash values in a hash table
3. Look for hash collisions
4. Verify that colliding inputs are different

2.5 Length Extension Attack

Description: Exploits the iterative structure of hash functions to extend messages. **Complexity**: Depends on hash function structure. **Method**:

1. Obtain hash of unknown message
2. Use internal state to continue hashing additional data
3. Generate valid hash for extended message without knowing original

2.6 Meet-in-the-Middle Attack

Description: Used against double encryption schemes. **Complexity**: 2^{k+1} operations instead of 2^{2k}. **Method**:

1. Encrypt plaintext with all possible first keys
2. Decrypt ciphertext with all possible second keys
3. Look for matches between encryption and decryption results

2.7 Related-Key Attack

Description: Exploits relationships between different keys. **Complexity**: Varies based on key schedule weaknesses. **Method**:

1. Choose keys with specific relationships (e.g., XOR differences)
2. Analyze how key differences propagate through the cipher
3. Use differential analysis on key schedule

2.8 Impossible Differential Attack

Description: Uses impossible differentials that cannot occur in the cipher. **Complexity**: Often more efficient than standard differential cryptanalysis. **Method**:

1. Find impossible differential characteristics
2. Filter out keys that would make impossible differentials possible
3. Reduce key space significantly

2.9 Algebraic Attack

Description: Formulates the cipher as a system of equations. **Complexity**: Depends on equation system complexity. **Method**:

1. Express cipher operations as algebraic equations
2. Collect many plaintext-ciphertext pairs
3. Solve the resulting equation system

2.10 Side-Channel Attack

Description: Exploits physical implementation characteristics. **Types**: Timing attacks, power analysis, electromagnetic analysis. **Method**:

1. Measure physical characteristics during encryption/decryption
2. Correlate measurements with key-dependent operations
3. Use statistical analysis to extract key information

3 Block Ciphers

3.1 AES (Advanced Encryption Standard)

- **Standard**: FIPS 197 (2001)
- **Block Size**: 128 bits
- **Key Sizes**: 128, 192, 256 bits
- **Rounds**: 10, 12, 14 (depending on key size)
- **Official Specification**: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf
- **Test Vectors**: https://csrc.nist.gov/Projects/cryptographic-standards-and-guidelines/example-values
- **Security Level**: 128, 192, 256 bits respectively

Attack Specifications:

- **Biclique Attack**:
  - **Complexity**: $2^{126.1}$ for AES-128, $2^{189.7}$ for AES-192, $2^{254.4}$ for AES-256
  - **Method**: Uses biclique structures to reduce key search space
  - **Reference**: https://eprint.iacr.org/2011/449
- **Related-Key Attack**:
  - **Complexity**: $2^{99.5}$ for AES-256
  - **Method**: Exploits key schedule weaknesses with related keys
  - **Reference**: https://eprint.iacr.org/2009/317
- **Side-Channel Attacks**:
  - **Types**: Cache timing, power analysis, electromagnetic analysis
  - **Countermeasures**: Constant-time implementation, masking, randomization

3.2 DES (Data Encryption Standard)

- **Standard**: FIPS 46-3 (1999)
- **Block Size**: 64 bits
- **Key Size**: 56 bits (64 with parity)
- **Rounds**: 16
- **Official Specification**: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.46-3.pdf
- **Test Vectors**: https://csrc.nist.gov/Projects/cryptographic-standards-and-guidelines/example-values
- **Security Level**: 56 bits (broken)

Attack Specifications:

- **Brute Force Attack**:
  - **Complexity**: 2^{56} operations
  - **Method**: Exhaustive key search using specialized hardware
  - **History**: First broken in 1997 by DESCHALL project
- **Differential Cryptanalysis**:
  - **Complexity**: 2^{47} chosen plaintexts
  - **Method**: Uses differential characteristics through S-boxes
  - **Reference**: https://en.wikipedia.org/wiki/Differential_cryptanalysis
- **Linear Cryptanalysis**:
  - **Complexity**: 2^{43} known plaintexts
  - **Method**: Linear approximations of S-boxes and round functions
  - **Reference**: https://en.wikipedia.org/wiki/Linear_cryptanalysis

3.3 Triple DES (3DES)

- **Standard**: SP 800-67 (2017)
- **Block Size**: 64 bits
- **Key Size**: 112 or 168 bits
- **Rounds**: 48 (3×16)
- **Official Specification**: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-67r2.pdf
- **Test Vectors**: https://csrc.nist.gov/Projects/cryptographic-standards-and-guidelines/example-values
- **Security Level**: 112 bits (112-bit key), 168 bits (168-bit key)

Attack Specifications:

- **Meet-in-the-Middle Attack**:
  - **Complexity**: 2^{112} operations (instead of 2^{168})
  - **Method**: Encrypt with all possible first keys, decrypt with all possible third keys
  - **Reference**: https://en.wikipedia.org/wiki/Meet-in-the-middle_attack
- **Sweet32 Attack**:
  - **Complexity**: 2^{32} operations
  - **Method**: Birthday attack on 64-bit block size
  - **Reference**: https://sweet32.info/

3.4 Serpent

- **Standard**: AES finalist
- **Block Size**: 128 bits
- **Key Size**: 128, 192, 256 bits
- **Rounds**: 32
- **Official Specification**: https://www.cl.cam.ac.uk/~rja14/serpent.html
- **Test Vectors**: https://www.cl.cam.ac.uk/~rja14/serpent.html
- **Security Level**: 256 bits

Attack Specifications:

- **Linear Cryptanalysis**:
  - **Complexity**: 2^{118} known plaintexts
  - **Method**: Linear approximations through S-boxes and linear transformation
  - **Reference**: https://www.cl.cam.ac.uk/~rja14/serpent.html
- **Differential Cryptanalysis**:
  - **Complexity**: 2^{131} chosen plaintexts
  - **Method**: Differential characteristics through S-boxes
  - **Reference**: https://www.cl.cam.ac.uk/~rja14/serpent.html

3.5 Twofish

- **Standard**: AES finalist
- **Block Size**: 128 bits
- **Key Size**: 128, 192, 256 bits
- **Rounds**: 16
- **Official Specification**: https://www.schneier.com/academic/twofish/
- **Test Vectors**: https://www.schneier.com/academic/twofish/
- **Security Level**: 256 bits

Attack Specifications:

- **Related-Key Attack**:
  - **Complexity**: 2^{256} operations
  - **Method**: Exploits key schedule with related keys
  - **Reference**: https://www.schneier.com/academic/twofish/
- **Impossible Differentials**:
  - **Complexity**: 2^{256} operations
  - **Method**: Uses impossible differential characteristics
  - **Reference**: https://www.schneier.com/academic/twofish/

3.6 Blowfish

- **Standard**: Designed by Bruce Schneier
- **Block Size**: 64 bits
- **Key Size**: 32-448 bits
- **Rounds**: 16
- **Official Specification**: https://www.schneier.com/academic/blowfish/
- **Test Vectors**: https://www.schneier.com/academic/blowfish/
- **Attacks**: Birthday attack, weak keys
- **Security Level**: 64 bits

3.7 CAST-128

- **Standard**: RFC 2144
- **Block Size**: 64 bits
- **Key Size**: 40-128 bits
- **Rounds**: 16
- **Official Specification**: https://tools.ietf.org/html/rfc2144
- **Test Vectors**: https://tools.ietf.org/html/rfc2144
- **Attacks**: Linear cryptanalysis, differential cryptanalysis
- **Security Level**: 128 bits

3.8 CAST-256

- **Standard**: RFC 2612
- **Block Size**: 128 bits
- **Key Size**: 128, 160, 192, 224, 256 bits
- **Rounds**: 48
- **Official Specification**: https://tools.ietf.org/html/rfc2612
- **Test Vectors**: https://tools.ietf.org/html/rfc2612
- **Attacks**: Linear cryptanalysis, differential cryptanalysis
- **Security Level**: 256 bits

3.9 Camellia

- **Standard**: RFC 3713, ISO/IEC 18033-3
- **Block Size**: 128 bits
- **Key Size**: 128, 192, 256 bits
- **Rounds**: 18, 24 (depending on key size)
- **Official Specification**: https://tools.ietf.org/html/rfc3713
- **Test Vectors**: https://tools.ietf.org/html/rfc3713
- **Attacks**: Impossible differentials, higher-order differentials
- **Security Level**: 256 bits

3.10 IDEA (International Data Encryption Algorithm)

- **Standard**: RFC 3058
- **Block Size**: 64 bits
- **Key Size**: 128 bits
- **Rounds**: 8.5
- **Official Specification**: https://tools.ietf.org/html/rfc3058
- **Test Vectors**: https://tools.ietf.org/html/rfc3058
- **Attacks**: Linear cryptanalysis, differential cryptanalysis
- **Security Level**: 128 bits

3.11 RC2

- **Standard**: RFC 2268
- **Block Size**: 64 bits
- **Key Size**: 1-128 bits
- **Rounds**: Variable
- **Official Specification**: https://tools.ietf.org/html/rfc2268
- **Test Vectors**: https://tools.ietf.org/html/rfc2268
- **Attacks**: Differential cryptanalysis, linear cryptanalysis
- **Security Level**: 128 bits

3.12 RC5

- **Standard**: RFC 2040
- **Block Size**: 32, 64, 128 bits
- **Key Size**: 0-2040 bits
- **Rounds**: 0-255
- **Official Specification**: https://tools.ietf.org/html/rfc2040
- **Test Vectors**: https://tools.ietf.org/html/rfc2040
- **Attacks**: Differential cryptanalysis, linear cryptanalysis
- **Security Level**: Variable

3.13 RC6

- **Standard**: AES finalist
- **Block Size**: 128 bits
- **Key Size**: 128, 192, 256 bits
- **Rounds**: 20
- **Official Specification**: https://www.nist.gov/
- **Test Vectors**: https://www.nist.gov/
- **Attacks**: Linear cryptanalysis, differential cryptanalysis
- **Security Level**: 256 bits

3.14 Skipjack

- **Standard**: FIPS 185 (1994)
- **Block Size**: 64 bits
- **Key Size**: 80 bits
- **Rounds**: 32
- **Official Specification**: https://csrc.nist.gov/csrc/media/publications/fips/185/archive/1994-07-30/documents/fips185.pdf
- **Test Vectors**: https://csrc.nist.gov/csrc/media/publications/fips/185/archive/1994-07-30/documents/fips185.pdf
- **Attacks**: Impossible differentials, truncated differentials
- **Security Level**: 80 bits

3.15 Misty1

- **Standard**: ISO/IEC 18033-3
- **Block Size**: 64 bits
- **Key Size**: 128 bits
- **Rounds**: 8
- **Official Specification**: https://www.iso.org/standard/54531.html
- **Test Vectors**: https://www.iso.org/standard/54531.html
- **Attacks**: Higher-order differentials, impossible differentials
- **Security Level**: 128 bits

3.16 Noekeon

- **Standard**: NESSIE submission
- **Block Size**: 128 bits
- **Key Size**: 128 bits
- **Rounds**: 16
- **Official Specification**: https://gro.noekeon.org/Noekeon-spec.pdf
- **Test Vectors**: https://gro.noekeon.org/Noekeon-spec.pdf
- **Attacks**: Linear cryptanalysis, differential cryptanalysis
- **Security Level**: 128 bits

3.17 XTEA (eXtended TEA)

- **Standard**: Academic paper
- **Block Size**: 64 bits
- **Key Size**: 128 bits
- **Rounds**: 64
- **Official Specification**: https://en.wikipedia.org/wiki/XTEA
- **Test Vectors**: https://en.wikipedia.org/wiki/XTEA
- **Attacks**: Related-key attacks, differential cryptanalysis
- **Security Level**: 128 bits

4 Stream Ciphers

4.1 RC4

- **Standard**: RFC 7465 (deprecated)
- **Key Size**: 1-256 bytes
- **Official Specification**: https://tools.ietf.org/html/rfc7465
- **Test Vectors**: https://tools.ietf.org/html/rfc7465
- **Security Level**: Broken

Attack Specifications:

- **Biased Output Attack**:
  - **Complexity**: 2^{16} operations
  - **Method**: Exploits statistical biases in early keystream bytes
  - **Reference**: https://en.wikipedia.org/wiki/RC4#Security
- **Related-Key Attack**:
  - **Complexity**: 2^{40} operations
  - **Method**: Exploits key schedule weaknesses
  - **Reference**: https://eprint.iacr.org/2007/120
- **Distinguishing Attack**:
  - **Complexity**: 2^{30} operations
  - **Method**: Distinguishes RC4 output from random
  - **Reference**: https://eprint.iacr.org/2001/070

4.2 Salsa20

- **Standard**: RFC 8439
- **Key Size**: 128 or 256 bits
- **Nonce Size**: 64 bits
- **Official Specification**: https://tools.ietf.org/html/rfc8439
- **Test Vectors**: https://tools.ietf.org/html/rfc8439
- **Attacks**: Differential cryptanalysis
- **Security Level**: 256 bits

4.3 ChaCha20

- **Standard**: RFC 8439
- **Key Size**: 256 bits
- **Nonce Size**: 96 bits
- **Official Specification**: https://tools.ietf.org/html/rfc8439
- **Test Vectors**: https://tools.ietf.org/html/rfc8439
- **Attacks**: Differential cryptanalysis
- **Security Level**: 256 bits

4.4 Rabbit

- **Standard**: RFC 4503
- **Key Size**: 128 bits
- **IV Size**: 64 bits
- **Official Specification**: https://tools.ietf.org/html/rfc4503
- **Test Vectors**: https://tools.ietf.org/html/rfc4503
- **Attacks**: Distinguishing attacks
- **Security Level**: 128 bits

4.5 HC-256

- **Standard**: eSTREAM finalist
- **Key Size**: 256 bits
- **IV Size**: 256 bits
- **Official Specification**: https://www.ecrypt.eu.org/stream/hc256.html
- **Test Vectors**: https://www.ecrypt.eu.org/stream/hc256.html
- **Attacks**: Distinguishing attacks
- **Security Level**: 256 bits

4.6 SNOW 3G

- **Standard**: 3GPP TS 35.216
- **Key Size**: 128 bits
- **IV Size**: 128 bits
- **Official Specification**: https://www.3gpp.org/DynaReport/35216.htm
- **Test Vectors**: https://www.3gpp.org/DynaReport/35216.htm
- **Attacks**: Algebraic attacks
- **Security Level**: 128 bits

4.7 SEAL

- **Standard**: Academic paper
- **Key Size**: 160 bits
- **Official Specification**: https://en.wikipedia.org/wiki/SEAL_(cipher)
- **Test Vectors**: https://en.wikipedia.org/wiki/SEAL_(cipher)
- **Attacks**: Related-key attacks
- **Security Level**: 160 bits

4.8 Scream

- **Standard**: Academic paper
- **Key Size**: 128 bits
- **Official Specification**: https://en.wikipedia.org/wiki/Scream_(cipher)
- **Test Vectors**: https://en.wikipedia.org/wiki/Scream_(cipher)
- **Attacks**: Differential cryptanalysis
- **Security Level**: 128 bits

5 Hash Functions

5.1 SHA-1

- **Standard**: FIPS 180-4
- **Output Size**: 160 bits
- **Block Size**: 512 bits
- **Official Specification**: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
- **Test Vectors**: https://csrc.nist.gov/Projects/cryptographic-standards-and-guidelines/example-values

- **Security Level**: 80 bits (broken)

Attack Specifications:

- **Collision Attack**:
  - **Complexity**: $2^{63.1}$ operations (theoretical), $2^{61.2}$ (practical)
  - **Method**: Differential path construction and message modification
  - **History**: First collision found in 2017 by Google and CWI
  - **Reference**: https://shattered.io/
- **Length Extension Attack**:
  - **Complexity**: 2^{160} operations
  - **Method**: Exploits Merkle-Damgård construction
  - **Countermeasure**: Use HMAC or similar construction

5.2 SHA-2 Family

- **Standard**: FIPS 180-4
- **Output Sizes**: 224, 256, 384, 512 bits
- **Block Size**: 512 or 1024 bits
- **Official Specification**: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
- **Test Vectors**: https://csrc.nist.gov/Projects/cryptographic-standards-and-guidelines/example-values
- **Attacks**: Length extension attacks
- **Security Level**: 112, 128, 192, 256 bits respectively

5.3 SHA-3 Family (Keccak)

- **Standard**: FIPS 202
- **Output Sizes**: 224, 256, 384, 512 bits
- **Block Size**: Variable
- **Official Specification**: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
- **Test Vectors**: https://csrc.nist.gov/Projects/cryptographic-standards-and-guidelines/example-values
- **Attacks**: None known
- **Security Level**: 112, 128, 192, 256 bits respectively

5.4 MD2

- **Standard**: RFC 1319
- **Output Size**: 128 bits
- **Block Size**: 128 bits
- **Official Specification**: https://tools.ietf.org/html/rfc1319
- **Test Vectors**: https://tools.ietf.org/html/rfc1319
- **Attacks**: Collision attacks
- **Security Level**: Broken

5.5 MD4

- **Standard**: RFC 1320
- **Output Size**: 128 bits
- **Block Size**: 512 bits
- **Official Specification**: https://tools.ietf.org/html/rfc1320
- **Test Vectors**: https://tools.ietf.org/html/rfc1320
- **Attacks**: Collision attacks
- **Security Level**: Broken

5.6 MD5

- **Standard**: RFC 1321
- **Output Size**: 128 bits
- **Block Size**: 512 bits
- **Official Specification**: https://tools.ietf.org/html/rfc1321
- **Test Vectors**: https://tools.ietf.org/html/rfc1321
- **Security Level**: Broken

Attack Specifications:

- **Collision Attack**:
  - **Complexity**: 2^{20} operations (practical)
  - **Method**: Wang's differential attack with message modification
  - **History**: First collision found in 2004 by Wang et al.
  - **Reference**: https://en.wikipedia.org/wiki/MD5#Security
- **Preimage Attack**:
  - **Complexity**: $2^{123.4}$ operations
  - **Method**: Meet-in-the-middle attack
  - **Reference**: https://eprint.iacr.org/2009/223

5.7 RIPEMD Family

- **Standard**: ISO/IEC 10118-3
- **Output Sizes**: 128, 160, 256, 320 bits
- **Block Size**: 512 bits
- **Official Specification**: https://www.iso.org/standard/39876.html
- **Test Vectors**: https://www.iso.org/standard/39876.html
- **Attacks**: Collision attacks
- **Security Level**: 64, 80, 128, 160 bits respectively

5.8 Whirlpool

- **Standard**: ISO/IEC 10118-3
- **Output Size**: 512 bits
- **Block Size**: 512 bits
- **Official Specification**: https://www.iso.org/standard/39876.html
- **Test Vectors**: https://www.iso.org/standard/39876.html
- **Attacks**: None known
- **Security Level**: 256 bits

5.9 Tiger

- **Standard**: Academic paper
- **Output Size**: 192 bits
- **Block Size**: 512 bits
- **Official Specification**: https://www.cs.technion.ac.il/~biham/Reports/Tiger/
- **Test Vectors**: https://www.cs.technion.ac.il/~biham/Reports/Tiger/
- **Attacks**: Collision attacks
- **Security Level**: 192 bits

5.10 BLAKE

- **Standard**: RFC 7693
- **Output Sizes**: 256, 512 bits
- **Block Size**: 1024 bits
- **Official Specification**: https://tools.ietf.org/html/rfc7693
- **Test Vectors**: https://tools.ietf.org/html/rfc7693
- **Attacks**: None known
- **Security Level**: 256, 512 bits respectively

6 MAC Algorithms

6.1 CBC-MAC

- **Standard**: ISO/IEC 9797-1
- **Block Size**: Variable
- **Official Specification**: https://www.iso.org/standard/50375.html
- **Test Vectors**: https://www.iso.org/standard/50375.html
- **Attacks**: Length extension attacks
- **Security Level**: Variable

6.2 CMAC

- **Standard**: NIST SP 800-38B
- **Block Size**: Variable
- **Official Specification**: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38B.pdf
- **Test Vectors**: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38B.pdf
- **Attacks**: None known
- **Security Level**: Variable

6.3 HMAC

- **Standard**: RFC 2104, FIPS 198-1
- **Block Size**: Variable
- **Official Specification**: https://tools.ietf.org/html/rfc2104
- **Test Vectors**: https://tools.ietf.org/html/rfc2104
- **Attacks**: Length extension attacks
- **Security Level**: Variable

7 Asymmetric Cryptography

7.1 RSA

- **Standard**: PKCS#1, RFC 8017
- **Key Size**: 1024-4096 bits
- **Official Specification**: https://tools.ietf.org/html/rfc8017
- **Test Vectors**: https://tools.ietf.org/html/rfc8017
- **Attacks**: Factorization attacks, side-channel attacks
- **Security Level**: 80-256 bits (depending on key size)

7.2 Hellman-Merkle Knapsack

- **Standard**: Academic paper
- **Key Size**: Variable
- **Official Specification**: https://en.wikipedia.org/wiki/Merkle-Hellman_knapsack_cryptosystem
- **Test Vectors**: https://en.wikipedia.org/wiki/Merkle-Hellman_knapsack_cryptosystem
- **Attacks**: Lattice reduction attacks
- **Security Level**: Broken

8 Classical Ciphers

8.1 Caesar Cipher

- **Type**: Substitution cipher
- **Key Space**: 25
- **Official Specification**: Historical
- **Test Vectors**: Standard
- **Attacks**: Brute force, frequency analysis
- **Security Level**: None

8.2 Vigenère Cipher

- **Type**: Polyalphabetic substitution
- **Key Space**: Variable
- **Official Specification**: Historical
- **Test Vectors**: Standard
- **Attacks**: Kasiski examination, frequency analysis
- **Security Level**: None

8.3 Playfair Cipher

- **Type**: Digraphic substitution
- **Key Space**: 25!
- **Official Specification**: Historical
- **Test Vectors**: Standard
- **Attacks**: Frequency analysis, pattern analysis
- **Security Level**: None

8.4 Hill Cipher

- **Type**: Polygraphic substitution
- **Key Space**: Variable
- **Official Specification**: Academic
- **Test Vectors**: Standard
- **Attacks**: Known plaintext attacks
- **Security Level**: None

8.5 Transposition Ciphers

- **Type**: Permutation cipher
- **Key Space**: Variable
- **Official Specification**: Historical
- **Test Vectors**: Standard
- **Attacks**: Anagramming, frequency analysis
- **Security Level**: None

8.6 ADFGVX Cipher

- **Type**: Fractionated transposition
- **Key Space**: Variable
- **Official Specification**: Historical
- **Test Vectors**: Standard
- **Attacks**: Frequency analysis, pattern analysis
- **Security Level**: None

9 Test Vectors and Validation

9.1 Standard Test Vectors

All algorithms should be validated against their respective standard test vectors. These can be found in the official specifications and standards documents referenced above.

9.2 Common Test Vector Sources

- NIST Cryptographic Standards: https://csrc.nist.gov/Projects/cryptographic-standards-and-gu example-values
- RFC Test Vectors: Available in respective RFC documents
- Academic Papers: Available in original research papers
- ISO Standards: Available in respective ISO documents

10 Security Considerations

10.1 Key Management

- Use cryptographically secure random number generators
- Implement proper key derivation functions
- Use appropriate key sizes for security requirements
- Implement secure key storage and transmission

10.2 Implementation Security

- Protect against timing attacks
- Implement constant-time operations where required
- Use secure coding practices
- Validate all inputs and outputs

10.3 Algorithm Selection

- Choose algorithms based on security requirements
- Consider performance requirements
- Stay updated with cryptanalysis results
- Use standardized algorithms when possible

11 References

References

[1] National Institute of Standards and Technology (NIST), *Cryptographic Standards and Guidelines*, https://csrc.nist.gov/Projects/cryptographic-standards-and-guidelines

[2] Internet Engineering Task Force (IETF), *Request for Comments (RFC)*, https://tools.ietf.org/

[3] International Organization for Standardization (ISO), *Information Technology Standards*, https://www.iso.org/

[4] eSTREAM Project, *Stream Cipher Project*, https://www.ecrypt.eu.org/stream/

[5] AES Competition, *Advanced Encryption Standard*, https://www.nist.gov/

[6] Bruce Schneier, *Cryptography Resources*, https://www.schneier.com/