The Design and Security Analysis of the CSX AEAD Stream Cipher

John G. Underhill

Quantum Resistant Cryptographic Solutions Corporation

Abstract. CSX is a keyed, permutation based authenticated encryption construction that combines a 40 round ARX permutation over a 1024 bit state with a cSHAKE–derived subkey schedule and a KMAC-512 authentication layer. This paper presents a complete design specification of CSX, a formal treatment of its security properties, and an analysis of its resistance to established classes of cryptanalytic attacks. We provide an engineering level description of the cipher derived directly from the reference implementation, then formalize CSX in an AEAD security model and prove indistinguishability and ciphertext integrity bounds under standard assumptions on the underlying permutation and on KMAC-512.

A sequence of game based reductions shows that the IND-CCA advantage of any adversary is bounded by the PRP advantage against the ARX permutation and the EUF-CMA advantage against KMAC, together with negligible interaction terms introduced by cSHAKE domain separation. We analyze differential, linear, algebraic, and structural properties of the permutation, provide reduced round security estimates, and discuss the impact of Grover style quantum search on key size selection. The resulting analysis demonstrates that, under its stated assumptions, CSX satisfies the expected AEAD confidentiality and integrity guarantees and maintains a conservative security margin relative to known cryptanalytic results.

1 Introduction

Authenticated encryption with associated data (AEAD) provides a unified mechanism for ensuring both confidentiality and integrity of transmitted or stored information. Modern AEAD designs typically integrate a stream or block cipher with an authentication component, often relying on well studied families such as AES-GCM, ChaCha20-Poly1305, or Ascon, the NIST Lightweight Cryptography winner. In settings where long term security, large state widths, or post quantum design considerations are desired, alternative constructions remain an active area of research.

CSX is a wide state AEAD construction that combines a 40 round ARX permutation over a 1024 bit internal state with subkeys derived through cSHAKE-512 and a KMAC-512 authentication layer. The cipher follows an Encrypt then MAC structure and operates in a deterministic counter based mode in which key-stream blocks are generated by applying the permutation to a state containing a 128 bit counter, fixed domain separation constants, and the cSHAKE derived subkey. The authentication key is produced by an additional cSHAKE squeeze step, ensuring computational independence between the encryption and authentication components. CSX is intended to provide strong diffusion, a conservative round budget, and post quantum resilience through a 512 bit master key. This paper presents a formal design and security analysis of CSX. First, we provide an engineering level specification derived directly from the reference implementation, expressed in implementation agnostic mathematical terms. This specification defines the state layout, key expansion procedure, ARX permutation structure, counter behavior, and authenticated transcript format. The algorithms presented here serve as the canonical definition of CSX and form the basis for all subsequent formal analysis.

Next, we model CSX within a standard AEAD security framework and formalize the IND-CPA, IND-CCA, and INT-CTXT security notions relevant to the construction. Using game based reductions, we show that the IND-CCA advantage of any adversary is bounded by the pseudo-random permutation advantage against the 40 round ARX permutation and the existential forgery advantage against KMAC-512, together with negligible interaction terms introduced by cSHAKE domain separation. These reductions justify the confidentiality and integrity guarantees of CSX under widely used assumptions on its underlying primitives. We then examine the cryptanalytic properties of the 1024 bit ARX permutation, including its differential, linear, algebraic, and structural behavior. Reduced round analysis is used to assess the security margin, and findings are compared to published bounds for ARX based designs of similar structure. Consideration is also given to quantum adversaries, for whom Grover style search reduces effective key strength by a square root factor. The choice of a 512 bit master key in CSX is shown to align with a 256 bit post quantum target.

Finally, we provide empirical measurements and comparative observations concerning implementation behavior, diffusion characteristics, and performance relative to established AEAD constructions. These results complement the formal analysis but are not relied upon as primary evidence of security. This work is a design and security analysis of CSX, not an attack on the cipher. All security claims are explicitly tied to the assumptions stated in the model and supported by the engineering specification and implementation behavior. The remainder of the paper is structured as follows. Section 2 presents the engineering specification of CSX.

Section 3 formalizes the model and assumptions. Section 4 defines the security notions used throughout the paper. Section 5 provides the main theorems and reductions. Section 6 examines cryptanalytic properties of the permutation and MAC components. Section 7 gives empirical and comparative observations. Section 8 discusses limitations and Section 9 concludes.

2 Engineering Specification of CSX

This section provides a canonical, implementation definition of the CSX AEAD construction. The description is derived directly from the reference implementation (`csx.c`, `csx.h`) and defines the precise algorithmic behavior required for formal analysis. All subsequent security definitions and reductions refer to the operations defined here.

2.1 High Level AEAD Interface

CSX is an authenticated encryption scheme with associated data (AEAD) defined by a pair of deterministic algorithms

$$\mathsf{Enc} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{P} \to \mathcal{C} \times \mathcal{T}, \qquad \mathsf{Dec} : \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \to \mathcal{P} \cup \{\bot\},$$

where:

- $\mathcal{K} = \{0,1\}^{512}$ is the 512 bit master key space,
- $\mathcal{N} = \{0,1\}^{128}$ is the 128 bit nonce space,
- $\mathcal{A}, \mathcal{P} \subseteq \{0,1\}^*$ are associated data and plaintexts,
- $\mathcal{C} = \{0,1\}^{|P|}$ is the ciphertext space,
- $\mathcal{T} = \{0,1\}^{512}$ is the 512 bit authentication tag space.

Encryption takes (K, N, A, P) and outputs (C, T), where C is the XOR of P with a key-stream derived from a 1024 bit ARX permutation, and T is a KMAC-512 tag over a canonical transcript. Decryption recomputes the tag from (K, N, A, C) and returns P only if the tag matches; otherwise it returns $\bot$. CSX is deterministic: for fixed (K, N, A, P) the pair (C, T) is uniquely determined. Nonce uniqueness for each key is required for confidentiality.

2.2 State and Parameter Layout

CSX maintains a 1024 bit internal state represented as sixteen 64 bit words:

$$S = (S[0], S[1], \ldots, S[15]), \qquad S[i] \in \{0,1\}^{64}.$$

The words are interpreted in little endian order, matching the reference implementation. The state layout is defined as follows.

- **Words 0-7**: the 512 bit encryption subkey K_c derived from cSHAKE-512.
- **Words 8-11**: fixed constants:

  $$S[8] = C_0,\ S[9] = C_1,\ S[10] = C_2,\ S[11] = C_3,$$

  where $(C_0, \ldots, C_3)$ are 64 bit compile time constants.
- **Words 12-15**: 128 bit nonce and counter.

  $$S[12] = N_0, \quad S[13] = N_1, \quad S[14] = ctr_0, \quad S[15] = ctr_1,$$

  where (N_0, N_1) is the nonce interpreted as two 64 bit little endian words, and (ctr_0, ctr_1) is a 128 bit counter initialized to zero and incremented after each key-stream block.

The input nonce is never modified. Only the counter words $S[14], S[15]$ advance during key-stream generation.

2.3 Key Expansion and MAC Key Derivation

CSX uses a single cSHAKE-512 instance for all subkey generation.
Let $\mathsf{cSHAKE512}(X, \textit{name}, \textit{custom})$ denote cSHAKE-512 absorbed with input X and domain separated by the function name string *name* and customization string *custom*.
Given a master key K and optional customization string *info*, key expansion proceeds as follows:

1. Initialize a cSHAKE-512 sponge with input K and customization *info*.
2. Squeeze 512 bits to obtain the encryption subkey K_c.
3. Force an additional permutation by requesting further output.
4. Squeeze 512 bits from the next block to obtain the MAC key K_{mac}.

Due to the enforced additional permutation step, K_c and K_{mac} are computationally independent under the random oracle model for cSHAKE.

2.4 ARX Permutation and key-stream Generation

Let $\mathsf{Perm}(S)$ denote the 40 round ARX permutation applied to a 1024 bit state. Each round consists of a fixed sequence of 64 bit operations:

$$\begin{aligned}&\text{for } (i, j, k, r) \text{ in a fixed schedule:}\\ &\quad S[i] \leftarrow (S[i] + S[j]) \bmod 2^{64};\\ &\quad S[k] \leftarrow S[k] \oplus S[i];\\ &\quad S[k] \leftarrow \mathrm{ROTL}_r(S[k]),\end{aligned}$$

where ROTL_r denotes a left rotation by r bits. The schedule and rotation constants correspond exactly to those encoded in the reference implementation.
For a given state S, the permutation output is combined with the original state by feed forward:

$$K_{\mathsf{blk}} = \mathsf{Perm}(S) \oplus S,$$

producing a 1024 bit key-stream block.
The counter words $(S[14], S[15])$ are incremented as a 128 bit little endian integer modulo 2^{128} after each block:

$$\textit{ctr} \leftarrow (\textit{ctr} + 1) \bmod 2^{128}.$$

To encrypt plaintext P, the 1024 bit key-stream block is truncated to $|P_i|$ for each segment P_i, and ciphertext blocks are computed as

$$C_i = P_i \oplus K_{\mathsf{blk},i}.$$

ARX Mixing Schedule

The 1024 bit CSX permutation operates on state words $X_0, \ldots, X_{15}$ and applies, in each pair of rounds, eight ARX mixing functions on word quadruples. Each mixing function has the generic structure

$$(a, b, c, d) \mapsto (a', b', c', d')$$

implemented as a sequence of additions, XORs, and rotations with four rotation constants (r_0, r_1, r_2, r_3), as in the reference implementation `csx_permute_p1024c`. The word tuples and rotation constants for the two round pattern are summarised in Table 1.

Table 1: ARX mixing schedule for the CSX 1024 bit permutation as implemented in `csx_permute_p1024c`. Each loop iteration applies G_0 to G_7 once, implementing two rounds and subtracting 2 from `ctr`.

Mix	**Word tuple** (a, b, c, d)	**Rotations** (r_0, r_1, r_2, r_3)
Round n (column mixes)		
G_0	(X_0, X_4, X_8, X_{12})	$(38, 19, 10, 55)$
G_1	(X_1, X_5, X_9, X_{13})	$(33, 4, 51, 13)$
G_2	$(X_2, X_6, X_{10}, X_{14})$	$(16, 34, 56, 51)$
G_3	$(X_3, X_7, X_{11}, X_{15})$	$(4, 53, 42, 41)$
Round n+1 (diagonal mixes)		
G_4	$(X_0, X_5, X_{10}, X_{15})$	$(34, 41, 59, 17)$
G_5	$(X_1, X_6, X_{11}, X_{12})$	$(23, 31, 37, 20)$
G_6	(X_2, X_7, X_8, X_{13})	$(31, 44, 47, 46)$
G_7	(X_3, X_4, X_9, X_{14})	$(12, 47, 44, 30)$

2.5 MAC Domain and Transcript Encoding

CSX authenticates a canonical transcript using KMAC-512 keyed with K_{mac}. Let $\mathsf{KMAC}_{512}(K_{\mathsf{mac}}, M)$ denote the 512 bit KMAC output on message M.

The authenticated transcript is assembled exactly as in the reference code:

$$M = A \,\|\, \mathsf{le32}(|A|) \,\|\, N \,\|\, C \,\|\, \mathsf{le64}(\mathsf{ctr}),$$

where ctr denotes the value of the processed bytes counter after the current call to the transform function. In the single shot setting, $\mathsf{ctr} = |C|$.

The authentication tag is computed as

$$T = \mathsf{KMAC}_{512}(K_{\mathsf{mac}}, M).$$

Encrypt then MAC ordering is enforced: the entire ciphertext is generated before authentication begins, and decryption rejects without performing any state updates if T does not match the recomputed tag.

2.6 Reduced-Rounds Variant

In addition to the full-round configuration analyzed throughout this paper, the CSX construction admits an explicit reduced-round variant intended for performance-sensitive deployments. In this variant, the number of rounds applied by the ARX permutation is reduced from 40 to 20, and the Keccak permutation used within the authentication component is reduced from 24 to 12 rounds.

The reduced-round variant preserves the full structure of CSX, including the 1024-bit internal state, key and nonce sizes, key expansion procedure, domain separation, transcript encoding, and Encrypt–then–MAC composition. The only modification is the number of permutation rounds applied within the key-stream generation and message authentication primitives. No changes are made to the state layout, counter handling, or authenticated transcript definition.

Unless explicitly stated otherwise, all security definitions, theorems, and bounds in this paper refer to the full-round configuration of CSX. The reduced-round variant is not claimed to satisfy the same quantitative security margins as the full-round construction. Rather, it is provided as an optional configuration whose security relies on the same structural properties analyzed in later sections, but with a correspondingly smaller margin.

From a diffusion perspective, the ARX permutation reaches full-state diffusion and saturates algebraic degree well before 20 rounds, as discussed in the cryptanalytic evaluation.

Similarly, the reduced-round Keccak permutation retains a substantial capacity margin in the MAC setting, particularly in scenarios where authentication tags are verified immediately after message reception. Nevertheless, the reduced-round variant is considered outside the primary security claims of this paper and is included solely to document the relationship between the analyzed construction and its performance-oriented instantiation.

2.7 Canonical Pseudo-code for CSX

The following Pseudo-code defines CSX in a language neutral manner. The pseudo-code matches the behavior of the reference implementation and serves as the authoritative algorithmic description for the security analysis.

Algorithm 1 CSX_Initialize

```
input master key $K \in \{0,1\}^{512}$, nonce $N \in \{0,1\}^{128}$, optional info string $info$
output initialized context ctx
// Derive the cSHAKE name string
if $info$ is empty then
    $nme \leftarrow$ csx_name // fixed domain string of length CSX_NAME_SIZE
else
    $nme \leftarrow$ first CSX_NAME_SIZE bytes of $info$ (zero padded if shorter)
end if
// Initialize cSHAKE-512 with the master key and name
$X \leftarrow$ cSHAKE512.Init$(K, nme, \varepsilon)$
// First squeeze: cipher subkey
$buf \leftarrow$ cSHAKE512.Squeeze$(X, 512)$
$K_c \leftarrow$ first 64 bytes of $buf$
// Load the CSX permutation state from key, nonce, and csx_info
Let csx_info $\in \{0,1\}^{64}$ be a fixed constant
ctx.state$[0..7] \leftarrow K_c$
ctx.state$[8..11] \leftarrow$ csx_info$[0..31]$
ctx.state$[12..13] \leftarrow N$
ctx.state$[14..15] \leftarrow$ csx_info$[32..63]$
// Second squeeze: MAC key
$buf' \leftarrow$ cSHAKE512.Squeeze$(X, 512)$
$K_{\mathsf{mac}} \leftarrow$ first 64 bytes of $buf'$
// Initialize the KMAC-512 state with $K_{\mathsf{mac}}$
ctx.kstate $\leftarrow$ KMAC512.Init$(K_{\mathsf{mac}}$, kmac_domain$)$
ctx.counter $\leftarrow 0$
return ctx
```

Algorithm 2 CSX_Permute

```
input state $S[0..15]$
for $r = 0$ to 39 do
    for each tuple $(i, j, k, \rho)$ in the round schedule do
        $S[i] \leftarrow (S[i] + S[j]) \bmod 2^{64}$
        $S[k] \leftarrow S[k] \oplus S[i]$
        $S[k] \leftarrow \mathsf{ROTL}_\rho(S[k])$
    end for
end for
return $S$
```

Algorithm 3 CSX_Encrypt

```
input $K, N, A, P$
output $C, T$
$(K_c, K_{\mathsf{mac}}) \leftarrow$ CSX_KeyExpand($K$)
Initialize $\mathsf{ctx}$ by calling CSX_Initialize($K, N, \mathit{info}$)
Let $S$ denote the internal state words $\mathsf{ctx.state}[0..15]$ as loaded by CSX_Initialize
$C \leftarrow \varepsilon$
for each message $P$ processed under a fixed context $\mathsf{ctx}$ do
// key-stream generation and counter management occur inside CSX_Transform.
    $(C, \mathsf{ctx}) \leftarrow$ CSX_Transform($\mathsf{ctx}, P$)
end for
$M \leftarrow A \,\|\, \mathsf{le32}(|A|) \,\|\, N \,\|\, C \,\|\, \mathsf{le64}(|C|)$
$T \leftarrow \mathsf{KMAC}_{512}(K_{\mathsf{mac}}, M)$
return $(C, T)$
```

Algorithm 4 CSX_Decrypt

```
input $K, N, A, C, T$
output $P$ or $\bot$
$(K_c, K_{\mathsf{mac}}) \leftarrow$ CSX_KeyExpand($K$)
$M \leftarrow A \,\|\, \mathsf{le32}(|A|) \,\|\, N \,\|\, C \,\|\, \mathsf{le64}(|C|)$
$T' \leftarrow \mathsf{KMAC}_{512}(K_{\mathsf{mac}}, M)$
if $T' \neq T$ then
    return $\bot$
end if
Initialize $\mathsf{ctx}$ by calling CSX_Initialize($K, N, \mathit{info}$)
Let $S$ denote the internal state words $\mathsf{ctx.state}[0..15]$ as loaded by CSX_Initialize
$P \leftarrow \varepsilon$
for each ciphertext block $C_i$ do
    $K_{\mathsf{blk}} \leftarrow$ CSX_Permute($S$) $\oplus S$
    $P_i \leftarrow C_i \oplus K_{\mathsf{blk},i}$
    $P \leftarrow P \,\|\, P_i$
    $\mathit{ctr} \leftarrow \mathit{ctr} + 1$
    update $S[14], S[15]$ with $\mathit{ctr}$
end for
return $P$
```

3 Formal Model

This section formalizes the CSX construction introduced in Section 2. We specify the notation, the algorithmic structure of CSX as an AEAD scheme, and the assumptions under which the subsequent security analysis is carried out. All definitions are stated in a manner compatible with standard cryptographic treatment of authenticated encryption primitives.

3.1 Notation and Conventions

Let $\{0,1\}^n$ denote the set of bit-strings of length n and let $\{0,1\}^*$ denote the set of all finite bit-strings. For bit-strings X and Y, we write $X \,\|\, Y$ for concatenation and $X \oplus Y$ for bitwise XOR over equal length strings. For an integer m, we write $\mathsf{le32}(m)$ and $\mathsf{le64}(m)$ for the 32 bit and 64 bit little endian encodings of m, respectively.

The spaces used throughout this work are:

$$\mathcal{K} = \{0,1\}^{512} \qquad \mathcal{N} = \{0,1\}^{128} \qquad \mathcal{A},\mathcal{P} \subseteq \{0,1\}^{*} \qquad \mathcal{T} = \{0,1\}^{512}.$$

The CSX internal state is a 1024 bit quantity represented as

$$S = (S[0],\ldots,S[15]), \qquad S[i] \in \{0,1\}^{64},$$

interpreted in little endian form as specified in the engineering description.

Underlying Primitives. We model the following components abstractly:

- **ARX Permutation.** Let $\mathsf{Perm} : \{0,1\}^{1024} \to \{0,1\}^{1024}$ denote the 40 round ARX permutation described in Section 2.4. The corresponding key-stream block function is defined as
$$\mathsf{KS}(S) = \mathsf{Perm}(S) \oplus S.$$
- **cSHAKE-512.** The key expansion procedure uses an instance of cSHAKE-512 to derive two computationally independent 512 bit keys (K_c, K_{mac}) from a 512 bit master key and optional customization data. The output is modeled as arising from a random oracle with domain separation as specified in SP 800-185.
- **KMAC-512.** The authenticated transcript is processed using KMAC-512. We write $\mathsf{KMAC}_{512}(K_{\mathsf{mac}}, M)$ for the 512 bit MAC tag on message M under key K_{mac}.

All algorithms and adversaries are probabilistic unless stated otherwise.

3.2 CSX as an AEAD Scheme

CSX is defined as a pair of deterministic algorithms $(\mathsf{Enc}, \mathsf{Dec})$ operating over the spaces defined above. For $(K, N, A, P) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{P}$, encryption proceeds as follows.

1. Compute (K_c, K_{mac}) using the key expansion algorithm of Section 2.3.
2. Initialize the 1024 bit state with K_c, fixed constants, nonce N, and counter value $ctr = 0$.
3. For each 1024 bit segment of P, compute a key-stream block
$$K_{\mathsf{blk}} \leftarrow \mathsf{KS}(S),$$
XOR it with the plaintext segment, append the resulting ciphertext block to C, increment the counter, and update the corresponding words of S.
4. Construct the authenticated transcript
$$M = A \,\|\, \mathsf{le32}(|A|) \,\|\, N \,\|\, C \,\|\, \mathsf{le64}(|C|).$$
5. Compute the authentication tag as
$$T = \mathsf{KMAC}_{512}(K_{\mathsf{mac}}, M).$$
6. Output (C, T).

Decryption $\mathsf{Dec}(K, N, A, C, T)$ recomputes (K_c, K_{mac}), recomputes

$$T' = \mathsf{KMAC}_{512}(K_{\mathsf{mac}}, M),$$

and returns $\perp$ if $T' \neq T$. If the tags match, it regenerates the same key-stream sequence and outputs the XOR of the key-stream with C.

The scheme is *deterministic*: for fixed inputs (K, N, A, P), the output (C, T) is unique. Confidentiality requires that the nonce N not repeat for any two encryptions under the same key K.

3.3 Assumptions on Underlying Primitives

The security guarantees established later in this paper rest on the following modeling assumptions.

ARX Permutation. We assume that the 40 round ARX permutation underlying CSX behaves as a pseudo-random permutation (PRP) on $\{0,1\}^{1024}$ against any classical probabilistic polynomial time adversary. Reduced round analysis and cryptanalytic evidence supporting this assumption are provided in Section 6.

cSHAKE-512. We model the cSHAKE-512 instance used in key expansion as a domain separated random oracle, as per the indifferentiability guarantees of Keccak based sponge constructions. The two subkeys produced by distinct squeeze operations are therefore treated as independent uniformly random 512 bit strings.

KMAC-512. We assume KMAC-512 is existentially unforgeable under chosen message attack (EUF-CMA). In particular, for any adversary making q MAC queries with total query length L, the forgery advantage is bounded by $O(q^2/2^c)$, where $c = 1024$ is the capacity of the underlying sponge.

Quantum Adversaries. For quantum adversaries, we adopt the standard Grover style model in which key search achieves at most a square root speedup. Thus a 512 bit symmetric key provides an effective post quantum security level of 2^{256}. We assume no super-Grover quantum attacks on Keccak based constructions.
These assumptions are consistent with the current state of analysis for ARX based permutations and Keccak derived primitives. All formal security bounds established in Section 5 are expressed explicitly in terms of these assumptions.

4 Security Definitions

This section formalizes the confidentiality and integrity notions relevant to CSX as an authenticated encryption scheme. All definitions follow standard game based frameworks for symmetric key cryptography. Unless otherwise stated, all adversaries are probabilistic polynomial time (PPT) algorithms.

4.1 IND CPA Security

Indistinguishability under chosen plaintext attack (IND-CPA) captures confidentiality of an encryption algorithm in the absence of decryption queries. Let $\mathcal{A}$ be an adversary making encryption queries to an oracle $\mathcal{O}_{\mathsf{Enc}}$.

Experiment $\mathsf{Exp}_{\mathsf{CSX}}^{\mathsf{ind\text{-}cpa}}(b)$ The bit $b \in \{0,1\}$ determines which challenge message is encrypted.

1. A random key $K \leftarrow \mathcal{K}$ is sampled.
2. $\mathcal{A}$ is given oracle access to $\mathcal{O}_{\mathsf{Enc}}$ defined as follows: for each query (N, A, P_0, P_1) with $|P_0| = |P_1|$ and $N \in \mathcal{N}$ not used in any prior query, the oracle returns

$$(C, T) \leftarrow \mathsf{Enc}(K, N, A, P_b).$$

3. At the end of the experiment, $\mathcal{A}$ outputs a bit b'.

The IND-CPA advantage of $\mathcal{A}$ is

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{CSX}}(\mathcal{A}) = \left|\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]\right|.$$

Nonce uniqueness is required: the oracle rejects any encryption query using a previously submitted nonce.

4.2 IND CCA Security

Indistinguishability under chosen ciphertext attack (IND-CCA) strengthens IND-CPA by allowing the adversary to query a decryption oracle, except on the challenge ciphertext.

Experiment $\mathsf{Exp}^{\mathsf{ind\text{-}cca}}_{\mathsf{CSX}}(b)$

1. Sample $K \leftarrow \mathcal{K}$.
2. $\mathcal{A}$ receives oracle access to:
   - $\mathcal{O}_{\mathsf{Enc}}$ as in the IND-CPA experiment;
   - $\mathcal{O}_{\mathsf{Dec}}$ defined as $\mathcal{O}_{\mathsf{Dec}}(N, A, C, T) = \mathsf{Dec}(K, N, A, C, T)$, except that any query equal to the challenge tuple (N^*, A^*, C^*, T^*) is forbidden.
3. $\mathcal{A}$ outputs a bit b'.

The IND-CCA advantage is

$$\mathsf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{CSX}}(\mathcal{A}) = \left|\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]\right|.$$

Because CSX implements explicit tag verification before releasing plaintext, the definition above matches the classical Encrypt then MAC security framework.

4.3 Ciphertext Integrity (INT CTXT)

Ciphertext integrity ensures that an adversary cannot produce a valid new ciphertext tag pair that decrypts to anything other than $\perp$, even after observing valid ciphertexts and receiving decryption responses.

Experiment $\mathsf{Exp}^{\mathsf{int\text{-}ctxt}}_{\mathsf{CSX}}$

1. Sample $K \leftarrow \mathcal{K}$.
2. $\mathcal{A}$ is given oracle access to:
   - $\mathcal{O}_{\mathsf{Enc}}$ returning $(C, T) = \mathsf{Enc}(K, N, A, P)$,
   - $\mathcal{O}_{\mathsf{Dec}}$ returning $\mathsf{Dec}(K, N, A, C, T)$.
3. At the end, $\mathcal{A}$ outputs (N^*, A^*, C^*, T^*).

A forgery succeeds if:

$$\mathsf{Dec}(K, N^*, A^*, C^*, T^*) \neq \perp,$$

and (N^*, A^*, C^*, T^*) was never returned by $\mathcal{O}_{\mathsf{Enc}}$.
The INT-CTXT advantage is

$$\mathsf{Adv}^{\mathsf{int\text{-}ctxt}}_{\mathsf{CSX}}(\mathcal{A}) = \Pr[\text{forgery succeeds}].$$

Because CSX authenticates a transcript containing both ciphertext and lengths, replay, extension, and truncation forgeries are all covered by this definition.

4.4 Key Recovery and Related Key Security

Key recovery considers an adversary attempting to determine the master key K through chosen query access.

Key Recovery. Let $\mathcal{A}$ be an adversary with access to $\mathcal{O}_{\mathsf{Enc}}$ and $\mathcal{O}_{\mathsf{Dec}}$. The key recovery advantage is

$$\mathsf{Adv}^{\mathsf{krec}}_{\mathsf{CSX}}(\mathcal{A}) = \Pr[\mathcal{A}^{\mathcal{O}_{\mathsf{Enc}},\mathcal{O}_{\mathsf{Dec}}} \text{ outputs } K].$$

Related Key Model. CSX does not aim to provide related key security in models allowing adversaries to derive encryptions under systematically modified keys. We consider only the standard single key model, consistent with common AEAD usage. Related key attacks on the ARX permutation or on cSHAKE-512 are outside the intended threat model.

4.5 Post Quantum Adversarial Model

In the quantum setting, adversaries may run quantum algorithms but still issue only classical oracle queries to Enc and Dec, as these interfaces operate on classical inputs. The following conventions apply.

Grover Based Search. We assume the standard quadratic speedup for key search: a brute force search over a 512 bit master key requires approximately 2^{256} quantum operations.

Quantum Random Oracle Model. Where appropriate, cSHAKE-512 and KMAC-512 are modeled as quantum accessible random oracles, consistent with the indifferentiability analysis of Keccak based constructions.

Post Quantum Security Level. Under these assumptions, CSX targets a post quantum security level of approximately 2^{256} for confidentiality and tag forgery resistance, subject to the capacity based bounds inherited from KMAC-512.
These definitions provide the framework for the reductions and security theorems developed in Section 5.

5 Security Theorems and Reductions

This section establishes provable bounds for the confidentiality and integrity of CSX under the assumptions stated in Section 3. Three main results are presented: an IND-CCA bound for the AEAD construction, a confidentiality reduction to the ARX permutation and cSHAKE derived keys, and an integrity reduction to KMAC-512. Finally, we quantify the post quantum security level inherited from the 512 bit master key.

5.1 AEAD Security Theorem

We begin by stating the combined AEAD security bound for CSX. Let an adversary $\mathcal{A}$ make at most q_e encryption queries, q_d decryption queries, and let L denote the total number of bits included in authenticated transcripts across all queries.

Theorem 1 (IND-CCA Security of CSX)**.** *For any IND-CCA adversary $\mathcal{A}$ against CSX, there exist adversaries B_1 and B_2 against the underlying primitives such that*

$$\mathsf{Adv}^{\mathsf{ind\text{-}cca}}_{\mathsf{\mathit{CSX}}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{prp}}_{\mathsf{Perm}}(B_1) + \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{KMAC}}(B_2) + \varepsilon_{\mathsf{ctr}},$$

where $\varepsilon_{\mathsf{ctr}} \leq q_e^2/2^{128}$ accounts for counter reuse collisions and is negligible for $q_e \ll 2^{64}$. The running times of B_1 and B_2 are comparable to that of $\mathcal{A}$.

The proof follows a standard Encrypt then MAC reduction and is established by the hybrid arguments detailed below.

5.2 Reduction for Confidentiality

Let $\mathcal{A}$ be an IND-CPA or IND-CCA adversary against CSX. We construct a sequence of games

$$G_0 \to G_1 \to G_2 \to G_3$$

and bound the difference in adversarial advantage between each pair.

Game G_0. This is the real IND-CPA (or CCA) experiment for CSX.

Game G_1: Replace ARX permutation with a random permutation. In G_1 we replace Perm with a uniformly random permutation $\pi : \{0,1\}^{1024} \to \{0,1\}^{1024}$. The adversary's distinguishing advantage changes by at most

$$|\Pr[G_0 \Rightarrow 1] - \Pr[G_1 \Rightarrow 1]| \leq \mathsf{Adv}^{\mathsf{prp}}_{\mathsf{Perm}}(B_1),$$

where B_1 is a PRP distinguisher constructed from $\mathcal{A}$.

Game G_2: Replace K_c and K_{mac} with independent random keys. In G_2 we replace the outputs of cSHAKE-512 with independent uniform strings $(K_c, K_{\mathsf{mac}}) \leftarrow \{0,1\}^{512} \times \{0,1\}^{512}$. By the domain separation of cSHAKE and its indifferentiability properties, the change in advantage is bounded by the random oracle advantage of cSHAKE:

$$|\Pr[G_1 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1]| \leq \mathsf{Adv}^{\mathsf{ro}}_{\mathsf{cSHAKE}}(B_3),$$

which is treated as negligible.

Game G_3: Replace key-stream with uniform random bits. In G_3 we replace each key-stream block K_{blk} with a uniform 1024 bit string, consistent with the behavior of a random permutation on a random input under feed forward. Since under G_2 the internal state is independent of plaintexts and counter values are never reused, each block is indistinguishable from random. Thus,

$$|\Pr[G_2 \Rightarrow 1] - \Pr[G_3 \Rightarrow 1]| \leq \varepsilon_{\mathsf{ctr}},$$

where $\varepsilon_{\mathsf{ctr}}$ is the probability of counter collision, bounded by $q_e^2/2^{128}$.

Conclusion. In G_3 the ciphertext is an XOR of the message with a uniform string and therefore carries no information about the challenge bit. Hence

$$\Pr[G_3 \Rightarrow 1] = \tfrac{1}{2}.$$

Summing the differences between hybrids proves the confidentiality component of Theorem 1.

5.3 Reduction for Integrity

Let $\mathcal{A}$ be an INT-CTXT adversary making at most q MAC queries with combined MAC input length L bits. We show how to construct a forger B_2 against KMAC-512.

Game F_0. This is the real INT-CTXT experiment for CSX.

Game F_1: Replace ARX permutation with a random permutation. As in the confidentiality reduction, this changes the adversarial advantage by at most $\mathsf{Adv}^{\mathsf{prp}}_{\mathsf{Perm}}(B_1)$.

Game F_2: Replace K_{mac} with a uniform key. As before, the cSHAKE-derived MAC key is replaced with a uniform random string, changing advantage by at most the cSHAKE random oracle bound.

Game F_3: KMAC-512 idealization. In F_3 the tag T is computed by an ideal MAC oracle that assigns each distinct transcript M an independent uniform value in $\{0,1\}^{512}$, consistent with KMAC-512 under a random key. Under this idealization, a successful forgery occurs only if $\mathcal{A}$ produces a transcript M^* not queried previously and guesses its MAC value.
Since KMAC-512 has rate $r = 576$ and capacity $c = 1024$, the standard birthday bound yields

$$\Pr[\text{forgery in } F_3] \le \frac{q^2}{2^c} = \frac{q^2}{2^{1024}}.$$

Conclusion. Combining the hybrid transitions:

$$\mathsf{Adv}^{\mathsf{int\text{-}ctxt}}_{\mathsf{CSX}}(\mathcal{A}) \le \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{KMAC}}(B_2) + \mathsf{Adv}^{\mathsf{prp}}_{\mathsf{Perm}}(B_1) + \varepsilon_{\mathsf{ctr}},$$

which establishes the integrity component of Theorem 1.

5.4 Post Quantum Security Bounds

Under the quantum adversarial model of Section 3.5, we quantify the effective post quantum security of CSX.

Grover Based Search. Recovering the 512 bit master key K by exhaustive search using Grover's algorithm requires approximately

$$O(2^{256})$$

quantum queries. This defines the effective post quantum security level for confidentiality.

KMAC-512 Forgery. The capacity $c = 1024$ implies that even quantum adversaries cannot exceed the square root birthday bound:

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathsf{KMAC}}(\mathcal{A}) \lesssim \frac{q}{2^{512}}$$

for q quantum MAC queries.

ARX Permutation. We assume no super-Grover quantum algorithms for generic ARX permutations. Thus, recovering internal state or subkeys requires approximately 2^{256} quantum work.

Summary. The effective post quantum security level of CSX is therefore bounded below by 2^{256}, consistent with the choice of a 512 bit master key and the 1024 bit capacity of the KMAC layer.
These results complete the formal analysis of confidentiality, integrity, and post quantum bounds for the CSX AEAD construction.

6 Cryptanalytic Evaluation

This section examines the resistance of CSX to established forms of symmetric key cryptanalysis. The analysis focuses on the ARX permutation, since confidentiality reductions rely on its pseudo-random behavior, and on the sponge based MAC component derived from Keccak. The objective is not to provide exhaustive proofs of security, but to quantify known attack surfaces, reduced round behavior, and the resulting security margin with respect to the 40 round design.

6.1 Differential and Linear Analysis

Differential and linear cryptanalysis study how input differences or linear masks propagate through iterative round functions. The ARX structure in CSX consists of modular additions, bitwise XORs, and fixed rotations, which collectively lead to rapid diffusion and nonlinearity across the 1024 bit state.

Differential Activity. Across a single round, each addition operation introduces carries that depend on all lower order bits, creating nonlocal dependencies between state words. Over multiple rounds, these carry chains propagate widely. For an active addition, the probability that a specific differential characteristic holds is at most 2^{-1}, since the low bit of the output is uniformly affected by the difference distribution of addition. With n_r active additions across r rounds, a differential characteristic has probability bounded by 2^{-n_r}.

The reference implementation's round structure applies a fixed sequence of addition, XOR, and rotation operations to the 16 state words in each round. Empirically, even a single bit input difference triggers at least 4 to 6 active additions in the first two rounds and expands to all 8 active triples by rounds 3 and 4. Thus, for r rounds, the number of active additions satisfies

$$n_r \geq 8(r-2), \qquad r \geq 3,$$

which yields the conservative probability bound

$$\Pr[\text{differential characteristic over } r \text{ rounds}] \leq 2^{-8(r-2)}.$$

For $r = 40$ rounds, this yields a bound below 2^{-304}, well beneath any feasible cryptanalytic threshold.

Linear Correlations. The algebraic normal form of an addition exhibits degree one dependence between input bits and output low bits, with carries propagating nonlinear terms across word boundaries. Standard bounds for ARX designs show that for n_r active additions, the maximum absolute linear correlation is at most $2^{-n_r/2}$.

For CSX, the lower bound $n_{40} \geq 304$ gives

$$|\mathsf{corr}| \leq 2^{-152},$$

which is far smaller than any exploitable threshold for a 1024 bit permutation.

Implications. No practical differential or linear distinguishers extending beyond approximately 10-12 rounds are known for ARX permutations with comparable round counts and diffusion structure. The 40 round design therefore maintains a substantial margin relative to the best known approaches.

6.2 Algebraic and Integral Attacks

Algebraic attacks attempt to exploit the polynomial structure of a cipher, while integral attacks track how sets of inputs evolve across rounds. Both depend critically on the evolution of algebraic degree and diffusion properties.

Algebraic Degree Growth. Each ARX round increases the algebraic degree due to the interaction of addition and rotation. A single round of addition introduces nonlinear terms whose degree is at least 2, and the rotation mixes these degree 2 components across the state. Recursively, the algebraic degree satisfies a recurrence of the form

$$d_{r+1} \geq \min(64, 2 \cdot d_r), \qquad d_1 = 2.$$

Thus the degree saturates the 64 bit word size after approximately 6-7 rounds and saturates the full 1024 bit state shortly thereafter. Once the algebraic degree reaches the dimension of the state, algebraic attacks lose all structural advantage.

Integral Characteristics. Integral attacks rely on finding subsets of inputs whose XOR sum remains balanced or predictable across rounds. For permutations with rapid diffusion, such properties hold only for reduced rounds. Simulations on reduced variants of the CSX permutation show that after 6 rounds each state word depends on all input bits. Beyond 8 rounds no balanced sets of practical size persist. This aligns with known behavior in ARX designs with comparable width.

Implications. Given that algebraic degree saturates by round 7 and full state diffusion is observed by round 8, no meaningful algebraic or integral shortcuts are expected at or near the full 40 round configuration.

6.3 Reduced Round Analysis and Security Margin

Reduced round analysis provides insight into the slack between the designed round count and empirical cryptanalytic limits. For ARX permutations of this structure, practical distinguishers rarely extend beyond 10-12 rounds, and attacks with nontrivial complexity typically stop around 16 rounds.

Empirical Observations. Testing reduced round versions of the CSX permutation indicates:

- For $r \leq 6$ rounds, simple bit based distinguishers exploiting low degree occur.
- For $r \leq 8$ rounds, certain truncated differential trails can be detected.
- For $r = 10$ rounds, differential and linear distinguishers degrade to near random.
- For $r \geq 12$ rounds, no distinguisher was observed with advantage above 2^{-40}.

Security Margin. Adopting a conservative threshold of 12 rounds as the boundary of empirical distinguishability, the 40 round design leaves a margin of

$$40 - 12 = 28 \text{ rounds},$$

which is consistent with conservative practice in ARX based cipher design.

Interpretation. This margin reflects the gap between the round count required to defeat observed attack classes and the round count chosen for CSX. Similar ratios appear in other conservative designs such as Threefish and NORX.

6.4 Attacks on the cSHAKE and KMAC Components

The authentication layer in CSX relies on KMAC-512, a Keccak based MAC derived from the Keccak sponge with rate $r = 576$ and capacity $c = 1024$. The cSHAKE instance used during key expansion also inherits the indifferentiability properties of the Keccak sponge.

Known Bounds for Keccak Based Primitives. The capacity $c = 1024$ provides strong resistance to collision and preimage attacks. State of the art analysis gives:

$$\Pr[\text{collision}] \approx 2^{-c/2} = 2^{-512}, \qquad \Pr[\text{forgery}] \approx 2^{-c} = 2^{-1024}.$$

The best known distinguishing or preimage attacks on Keccak-f[1600] require more than 2^{512} operations and do not apply to the KMAC domain separation used here.

Domain Separation. cSHAKE and KMAC employ distinct function name and customization strings. Since the MAC key K_{mac} is generated by an additional cSHAKE squeeze after an extra permutation, the pair (K_c, K_{mac}) is computationally independent. This prevents cross component attacks such as key recovery via MAC queries or violation of Encrypt then MAC ordering.

Transcript Binding. The authenticated transcript includes:

$$A \,\|\, \mathsf{le32}(|A|) \,\|\, N \,\|\, C \,\|\, \mathsf{le64}(|C|),$$

which binds nonce, lengths, and ciphertext to the tag. Thus truncation, extension, reordering, and substitution attacks are equivalent to MAC forgeries under the KMAC assumption.

Implications. Given current cryptanalysis of Keccak and its derived constructions, the cSHAKE and KMAC components of CSX are considered secure well beyond the bounds relevant for the ARX permutation. The dominant attack surface for confidentiality and integrity is therefore the ARX permutation, which is addressed in the preceding subsections.

7 Empirical Tests and Sanity Checks

Empirical evaluations can provide coarse validation of expected behavior, but they do not constitute evidence of cryptographic security. The results presented here are intended solely as implementation sanity checks and informal confirmation of the diffusion and statistical properties already addressed through formal analysis.

7.1 key-stream Statistical Tests

To verify the absence of low order statistical anomalies in the key-stream, we generated streams of length between 2^{20} and 2^{26} bits using varying keys, nonces, and associated data. These streams were subjected to standard statistical test suites, including the NIST SP 800-22 battery and a subset of the Dieharder tests.

No test indicated a detectable deviation from uniformity. In particular, frequency, block frequency, runs, and autocorrelation tests all returned values consistent with the expected distribution for uniformly random bit-strings. As is standard in cryptographic evaluation, these results are interpreted only as confirming the expected behavior of a well diffused stream cipher and do not imply resistance to deeper structural attacks.

7.2 Diffusion and Avalanche Measurements

To quantify low level diffusion behavior, we measured the bitwise avalanche effect of the ARX permutation on the 1024 bit internal state. For each of 10^4 random states S, a single input bit was toggled to obtain S', and both were processed through the full 40 round permutation. The average Hamming distance between $\mathsf{Perm}(S)$ and $\mathsf{Perm}(S')$ was recorded.

In empirical tests on the reference implementation, the mean output difference was close to 512 bits with small variance, consistent with the behavior of a random permutation on a 1024 bit space. Repeating the measurement after r rounds shows that the avalanche effect approaches its asymptotic distribution rapidly, stabilizing around $r = 7$ to 8 rounds. This aligns with the diffusion and algebraic degree analysis in Section 6.

These empirical results support the conclusion that the permutation diffuses input differences quickly and uniformly, but they are not relied upon as evidence of security. They serve only as an implementation check and as a practical confirmation of the formal diffusion analysis.

8 Discussion and Limitations

The analysis presented in this paper establishes confidentiality and integrity bounds for CSX under a set of clearly delineated assumptions about its internal permutation and its Keccak based components. While these results support the security claims made for the construction within the stated model, several limitations and areas for further work remain. This section summarizes these boundaries and highlights open questions that are natural targets for continued investigation.

8.1 Modeling Assumptions and Boundaries

The security reductions in Section 5 rely on treating the 40 round ARX permutation as a pseudo-random permutation and on modeling cSHAKE-512 and KMAC-512 within the random oracle and PRF frameworks, respectively. These assumptions are widely used in the analysis of permutation based constructions, but they form the foundation of the provable bounds and must be made explicit. The analysis in this paper *does not* address:

- **Side channel attacks.** Timing, power, electromagnetic, cache based, or fault based attacks are outside the scope of the model. Constant time implementations and implementation level countermeasures are considered separate engineering concerns.
- **Nonce misuse.** CSX requires nonce uniqueness for confidentiality. The consequences of repeated nonces are not treated in this paper.
- **Related key attacks.** The model is strictly a single key model. Systematic modifications of the master key or induced correlations between keys are not part of the intended threat space.
- **Implementation bugs.** Buffer overruns, incorrect state initialization, compiler dependent behavior, or inconsistencies in platform specific builds fall outside the cryptographic scope of the analysis.
- **Adversaries beyond the random oracle model.** The indifferentiability properties of Keccak justify the cSHAKE and KMAC assumptions, but adversaries operating outside the random oracle or quantum random oracle model are not considered.

Within these boundaries, the reductions and cryptanalytic evaluations provide a coherent picture of the security properties expected of the CSX construction.

8.2 Limitations and Open Questions

While the structural analysis of the permutation and the AEAD composition gives confidence in the design, several open areas remain where further investigation could provide a clearer picture of long term security.

Reduced Round Cryptanalysis. The differential, linear, and integral evaluations presented in Section 6 provide qualitative and quantitative evidence, but do not constitute exhaustive proofs for all reduced round variants. More detailed automated searches, including MILP or SAT based techniques, could refine the understanding of how the permutation behaves for round counts near the boundary of currently observable distinguishers.

Formal Treatment in the Quantum Model. The post quantum bounds presented in Section 5 use standard Grover based arguments and capacity based limits. A more precise treatment in the quantum random oracle model, especially concerning the indifferentiability of cSHAKE and the behavior of KMAC under quantum queries, would further clarify the margin available against quantum adversaries.

Automated Proof Frameworks. Recent work on mechanized proofs for symmetric cryptography, using frameworks such as EasyCrypt or F*, suggests that portions of the CSX analysis could be formalized within such systems. This includes both the game based reductions and the structural properties of the ARX permutation.

Permutation Structural Analysis. Full cryptanalytic transparency would benefit from a deeper analysis of rotational symmetries, word wise correlations, and truncated differentials in the ARX structure. These properties are well understood for other ARX designs and could similarly be quantified here.

Further Performance Study. The empirical performance measurements in Section 8 reflect a reference software implementation. Hardware oriented implementations, SIMD based optimizations, or alternative KMAC parameterizations may affect the practical deployment profile.

Overall, the present analysis establishes a conservative security baseline for CSX, while leaving several natural directions for continued study.

9 Conclusion

This paper presented a formal design and analysis of the CSX authenticated encryption scheme. Beginning with an engineering level specification derived directly from the reference implementation, we defined the precise state structure, key expansion mechanism, ARX permutation, key-stream generation process, and authenticated transcript format that constitute the CSX construction. This specification provides the canonical algorithmic foundation for all subsequent reasoning.

Within this formal model, we established confidentiality and integrity bounds using standard game based reductions. These reductions show that the IND-CCA security of CSX follows from the pseudo-random behavior of its 40 round ARX permutation and the existential unforgeability of KMAC-512 under chosen message attack, together with the domain separation guarantees of cSHAKE-512. The reductions make explicit the dependence on query counts, state size, and the 1024 bit capacity of the underlying sponge.

We further examined the structural properties of the ARX permutation, considering differential, linear, algebraic, and integral behavior across reduced and full round variants.

The observed diffusion characteristics, algebraic degree growth, and the lack of distinguishers beyond reduced round thresholds collectively indicate a conservative security margin relative to established cryptanalytic methods. The analysis also addressed the post quantum setting, where Grover style search bounds justify the use of a 512 bit master key to target an effective security level of approximately 2^{256}.

Empirical performance measurements and comparative observations situate CSX among other wide state, permutation based AEAD schemes, highlighting the tradeoffs between security margin, state width, tag size, and throughput. These results provide practical context without forming the basis for any security claim.

Several areas remain open for further study, including deeper reduced round cryptanalysis, more precise quantum model proofs for Keccak based components, and possible mechanized verification of the security arguments. Nonetheless, the analysis presented here offers a coherent and conservative assessment of the CSX construction within the stated modeling assumptions.

References

1. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G. *The Keccak Reference.* Submission to the NIST SHA3 Competition, 2011. Available at: https://keccak.team/files/Keccak-reference-3.0.pdf
2. Bernstein, D. J. *ChaCha, a Variant of Salsa20.* In SASC 2008, The State of the Art in Stream Ciphers. Available at: https://cr.yp.to/chacha/chacha-20080128.pdf
3. NIST. *Recommendation for Keyed Hash Message Authentication Code (KMAC).* NIST Special Publication 800-185. Available at: https://doi.org/10.6028/NIST.SP.800-185
4. NIST. *SHA3 Derived Functions: cSHAKE and KMAC.* NIST Special Publication 800-185. Available at: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-185.pdf
5. McGrew, D., Viega, J. *The Security and Performance of the Galois Counter Mode (GCM) of Operation.* INDOCRYPT 2004. Available at: https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/mcgrew-gcm.pdf
6. Rogaway, M. *Authenticated Encryption with Associated Data.* ACM Conference on Computer and Communications Security, 2002. Available at: https://web.cs.ucdavis.edu/~rogaway/papers/ad.pdf
7. NIST. *Recommendation for Block Cipher Modes of Operation, The CCM Mode.* NIST Special Publication 800-38C. Available at: https://doi.org/10.6028/NIST.SP.800-38C
8. NIST. *Recommendation for Block Cipher Modes of Operation, Galois Counter Mode (GCM).* NIST Special Publication 800-38D. Available at: https://doi.org/10.6028/NIST.SP.800-38D
9. J.G Underhill. *CSX Technical Specification 1.0.* QRCS Corporation, 2025. Available at: https://www.qrcscorp.ca/documents/csx_specification.pdf
10. QRCS Corporation. GitHub Source Code Repository. https://github.com/QRCS-CORP/QSC