NIST Interagency Report
NIST IR 8552

Requirements for Cryptographic Accordions

Yu Long Chen
Michael Davidson
Morris Dworkin
John Kelsey
Yu Sasaki
Meltem Sönmez Turan
Donghoon Chang
Nicky Mouha
Alyssa Thompson

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8552

NATIONAL INSTITUTE OF
STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Interagency Report
NIST IR 8552

Requirements for Cryptographic Accordions

Yu Long Chen
Michael Davidson
Morris Dworkin
John Kelsey
Yu Sasaki
Meltem Sönmez Turan
Computer Security Division
Information Technology Laboratory

Donghoon Chang
Strativia
Largo, MD

Nicky Mouha
FWI
Fairfax, VA

Alyssa Thompson
National Security Agency
Guest Researcher, Computer Security Division
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8552

April 2025

U.S. Department of Commerce
Howard Lutnick, Secretary

National Institute of Standards and Technology
Craig Burkhardt, Acting Under Secretary of Commerce for Standards and Technology and Acting NIST Director

NIST Technical Series Policies

Publication History
Approved by the NIST Editorial Review Board on 2025-04-02

How to cite this NIST Technical Series Publication
Yu Long Chen, Michael Davidson, Morris Dworkin, John Kelsey, Yu Sasaki, Meltem Sönmez Turan, Donghoon Chang, Nicky Mouha, Alyssa Thompson (2025) Requirements for Cryptographic Accordions. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) NIST IR 8552. https://doi.org/10.6028/NIST.IR.8552

Author ORCID iDs
Yu Long Chen: 0000-0002-6281-6430
Michael Davidson: 0000-0002-4862-5697
Morris Dworkin: 0000-0003-2745-9906
John Kelsey: 0000-0002-3427-1744
Yu Sasaki: 0000-0002-1273-2394
Meltem Sönmez Turan: 0000-0002-1950-7130
Donghoon Chang: 0009-0000-8496-703X
Nicky Mouha: 0000-0001-8861-782X
Alyssa Thompson: 0009-0009-8137-2589

Contact Information
ciphermodes@nist.gov

Additional Information
Additional information about this publication is available at https://csrc.nist.gov/pubs/ir/8552/final, including related content, potential updates, and document history.

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

This report introduces the *cryptographic accordion* as a tweakable, variable-input-length strong pseudorandom permutation (VIL-SPRP) that is constructed from an underlying block cipher. An accordion facilitates the cryptographic processing of messages of various sizes while offering enhanced security compared to the approved block cipher modes of operation that are specified in the NIST SP 800-38 series. This report introduces associated terminology, outlines design requirements for accordions, and describes three categories of applications for them.

Keywords

accordion; authenticated encryption; disk encryption; encode-then-encipher; key wrapping; length-preserving encryption.

Reports on Computer Systems Technology

Table of Contents

Acknowledgments

While he was a guest researcher at NIST, Jinkeon Kang was a co-author of the discussion draft of this report [1] that was prepared for the NIST Worskhop on the Requirements for an Accordion Cipher Mode 2024. The participants in that workshop and the *Third Workshop on Block Cipher Modes of Operation 2023* provided valuable feedback for the development of this publication.

The authors also appreciate the support of colleagues in NIST's Computer Security Division, especially Andrew Regenscheid, Lily Chen, Ray Perlner, Daniel Smith-Tone, Elaine Barker, James Foti, and Sara Kerman.

1. Introduction

A mode of operation is a method that defines how a block cipher cryptographically processes (e.g., encrypts and/or authenticates) data beyond a single block length. The National Institute of Standards and Technology (NIST) Special Publication (SP) 800-38 series [2–10] specifies several modes that were submitted for NIST's consideration from several different sources over many years to address a variety of security requirements and use cases.

NIST introduces the term *cryptographic accordion* — or simply *accordion* — to describe a technique constructed from an underlying block cipher that itself acts like a cipher on inputs of varying sizes. Thus, an accordion is simultaneously a cipher and a mode of the underlying block cipher. Formally, an accordion is defined as a tweakable, variable-input-length-strong pseudorandom permutation (VIL-SPRP) that is constructed from an underlying block cipher. NIST expects to develop one or more accordions with a generic proof of security.

The limitations of the current NIST modes were identified in NIST Internal Report (IR) 8459 [11] as part of NIST's review process under the auspices of the NIST Crypto Publication Review Board.[1] NIST hosted the Third Workshop on Block Cipher Modes of Operation 2023 [2] to discuss how to improve NIST's modes portfolio, including desirable additional security features and a variety of proposed techniques. As a follow up, NIST hosted the NIST Workshop on the Requirements for an Accordion Cipher Mode 2024 [12] to 1) propose the development of an accordion as a comprehensive solution for a variety of cryptographic functionalities and 2) solicit feedback on the design requirements.[3]

In this report, the cryptographic functions that may be constructed from an accordion are called *derived functions*. These derived functions can use the properties of the accordion to offer improved security, usage bounds, and functionality over existing modes. For example, a derived function for authenticated encryption with associated data (AEAD) could provide additional security properties compared to Galois/Counter Mode (GCM) [6], such as nonce-misuse resistance, support for short tags, nonce hiding, and key commitment. These features could make the accordion-based derived function a more robust choice than GCM for many cryptographic applications.

Another benefit is that an accordion facilitates the conceptual separation of the enciphering from the functional roles of different types of inputs (e.g., confidential data, associated data, randomization values, and nonces). This separation should promote easier, faster, and simpler deployment compared to the ad hoc development of dedicated modes for specific functionalities and security features.

[1] The Crypto Publication Review Board within the Computer Security Division identifies a publication for review based on its original publishing date and any relevant issues raised since it was published. See https://csrc.nist.gov/projects/crypto-publication-review-project.

[2] See https://csrc.nist.gov/Events/2023/third-workshop-on-block-cipher-modes-of-operation.

[3] Prior to the workshop, NIST released a discussion draft this report [1].

1.1. Historical Background

The notion of an accordion as a tweakable VIL-SPRP evolved from related terminology in the academic literature. The term *strong pseudorandom permutation* (SPRP) formally captures the idea that changing any bit of the input randomizes every bit of output for both the permutation and its inverse. The term *enciphering mode* describes a block cipher mode of operation that provides length-preserving encryption for variable input length (VIL) messages. The term *tweakable enciphering mode* indicates that the enciphering mode would take an additional input called a *tweak*, which was first introduced only in the context of a *tweakable block cipher* [13] (i.e., with fixed-length input).

Some early works are CMC [14], EME [15], and the Hasty Pudding Cipher [16]. Several related design approaches have since emerged, such as AEZ [17], HCTR2 [18], Glevian and Vigornian [19], and Docked Double Decker [20]. There are also constructions based on other primitives (e.g., XChacha [21]) and tweakable block ciphers (e.g., Adiantum [22] and ZCZ [23]). Bellare and Rogaway's encode-then-encipher encryption [24] is an example of a technique that could be applied to an accordion to achieve authenticated encryption. Another category of techniques that support variable input lengths is format-preserving encryption (FPE), though the input domain is usually small in this case. Moreover, NIST's standard FF1 [10] for FPE, although AES-based, is expected to perform much less efficiently than a well-designed accordion.

While none of the above designs may meet the exact requirements that are outlined in this document, they provide valuable background and reference material.

1.2. Layers of Components

An accordion is one layer of a larger process for achieving a cryptographic functionality, as illustrated in Fig. 1. In particular, the underlying block cipher is the innermost layer, which is called by the accordion. The accordion is called by a derived function that defines how the accordion will be used but does not necessarily provide additional cryptographic processing of the inputs. For example, a derived function could provide AEAD functionality by encoding its inputs and providing them to the accordion. The derived function would then be called by an application, such as the Transport Layer Security (TLS) protocol for secure communication on the internet. One motivation for the development of an accordion is that it can support a variety of functionalities, which are currently met by ad hoc or dedicated modes of operation.

The derived function layer will specify an encoding for the inputs to the accordion, and the accordion will specify the inputs for the block cipher. For an encryption application, the outer layer takes in the raw data to be encrypted and applies an encoding (e.g., by adding some integrity check bits). The shared secret or master key provided to the outer layer may be used directly as the encryption key for the accordion or to specify some new accordion key. Using any other inputs provided, the derived function layer defines a tweak for the

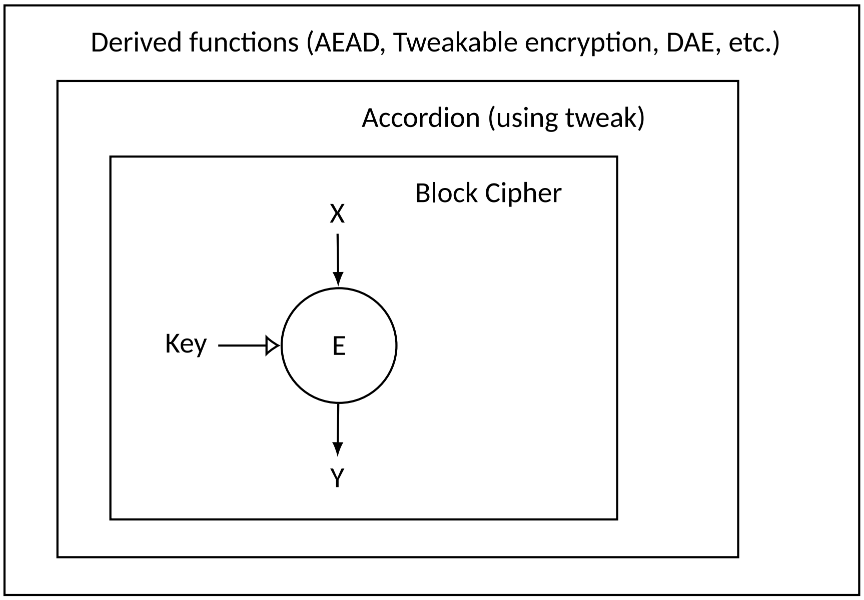

Figure 1. Layered structure of the accordion and derived functions

accordion. The accordion tweak, encoded message, and accordion key are used to call the accordion encryption function. Within the accordion encryption, a block cipher key, which may be the same as the accordion key, is used to call the block cipher. The encoded message blocks are input to the block cipher encryption, and ciphertext blocks are returned. The accordion collects the ciphertext blocks into a single ciphertext, which is returned to the derived function.

When the layer is clear from the context, the terminology can be simplified. For example, within the accordion level, the terms *accordion key* and *encoded message* can be shortened to the *key* and the *message*.

NIST intends to develop and standardize an accordion that can support derived functions for at least three applications: AEAD, tweakable encryption, and deterministic authenticated encryption.

2. The Accordion

This section elaborates on the notation, definitions, and properties envisioned for approved accordions.

2.1. Notation

An accordion will have several parameters that are fixed within a given instance. The eventual accordion standard may support multiple parameter sets. The following table provides the notation for elements of an accordion.

Symbol	**Definition**
a	Integer such that ag is the minimum allowed message size in bits
b	Integer such that bg is the maximum allowed message size in bits
C	Ciphertext
E	Underlying block cipher
g	Granularity of message sizes
K	Secret key
k	Length of the secret key K in bits
ℓ	Length of the message M in bits
M	Message
n	Block size of block cipher E
T	Tweak
s	Length of tweak in bits
s_{min}	Minimum allowed bit size for tweak T
s_{max}	Maximum allowed bit size for tweak T

An accordion consists of the encryption algorithm `A.enc` and the decryption algorithm `A.dec`.[4] The three inputs to `A.enc` are a secret key $K \in \{0,1\}^k$; a tweak $T \in \{0,1\}^s$, where $s_{min} \leq s \leq s_{max}$; and a message $M \in \{0,1\}^\ell$, where $\ell \in \{ag, (a+1)g, \dots, bg\}$. The output is a ciphertext $C \in \{0,1\}^\ell$:

$$\texttt{A.enc}(K, T, M) = C. \tag{1}$$

For any fixed values of K and T, `A.enc` is a permutation, and the decryption algorithm `A.dec` is its inverse so that

$$\texttt{A.dec}(K, T, C) = M. \tag{2}$$

[4] Although the names of these functions and the term *ciphertext* suggest a focus on confidentiality applications, an accordion may also support applications that do not require confidentiality (e.g., authentication only). Similarly, the term *message* can refer to any kind of data input to the accordion.

2.2. Security Targets

An accordion design should support each of the following security goals to the highest extent reasonable, and a specification for the design should include analysis of the security goals and identify the level of security achieved for each.

2.2.1. Formal Goal

A (q,σ,t)-distinguisher against an accordion is an algorithm $\mathcal{D}$ making at most q oracle queries with the total number of queried blocks being at most σ, running in time at most t, and outputting a single bit b. The following game defines the security that is expected from this construction:

1. At the beginning of the game, the challenger generates a bit b and a key K uniformly at random.
2. The distinguisher $\mathcal{D}$ is allowed to make up to q queries to the challenger of the form encrypt(T,x) or decrypt(T,x).

   (a) If $b = 0$, the challenger answers these queries as follows:

   $$\begin{aligned}\text{encrypt}(T,x) &= \mathtt{A.enc}(K,T,x)\\ \text{decrypt}(T,x) &= \mathtt{A.dec}(K,T,x).\end{aligned}$$

   (b) If $b = 1$, then for each distinct choice of $(T,|x|)$, where $|x|$ denotes the length of x in bits, the challenger selects and remembers a new random permutation on $2^{|x|}$ elements, $\Pi_{T,|x|}$. It answers queries using $\Pi_{T,|x|}, \Pi^{-1}_{T,|x|}$ as follows:

   $$\begin{aligned}\text{encrypt}(T,x) &= \Pi_{T,|x|}(x)\\ \text{decrypt}(T,x) &= \Pi^{-1}_{T,|x|}(x).\end{aligned}$$

3. After making q such queries, the distinguisher $\mathcal{D}$ must guess b.

An accordion is required to ensure that for any (q, σ, t)-distinguisher $\mathcal{D}$, the advantage of winning this game is negligibly small for relevant values of q, σ, and t. The relevant values for q, σ, and t will depend on the block size, the key size, the accordion, and the computation model used for the adversary. For example, the time t for a purely classical adversary will always be bounded above by $2^{|K|}$. Additionally, NIST may consider various approaches to defining the time parameter t with respect to parallelization, memory access costs, communications overhead, and other estimated costs of an attack and may relate relevant values of t to the security strength categories defined in the NIST PQC standardization process [25] and used in NIST's transition plan to PQC [26].

2.2.2. Beyond-Birthday-Bound Security

The usage bounds of many cryptographic techniques are limited by the "birthday bound," where collisions (i.e., repetitions among a set of randomly generated data blocks) can be expected to occur. A technique that is specifically designed to provide "beyond-birthday-bound" (BBB) security against generic cryptanalytic attacks would support less restrictive usage bounds on the amount of data that is processed per key.

More formally, for an accordion to support BBB security, the advantage of winning the game defined in Sec. 2.2.1 against any (q, σ, t)-distinguisher $\mathscr{D}$ should be negligibly small, even when $2^{n/2}$ or more blocks of input are processed. Moreover, NIST expects the BBB security to hold for all queries, including multiple queries with the same tweak.

2.2.3. Multi-User Security

Security bounds for primitives (e.g., an accordion or derived function) are typically given for a single user. Multi-user security considers how the security of these primitives changes as the number of independent users of the primitive gets larger. There are generic bounds on multi-user security, but a complete analysis requires consideration of how a given construction of a primitive affects its multi-user security (e.g., [27–36]).

Evaluating how security degrades as the number of users grows is a challenging technical problem. While the generic bounds are acceptable, NIST prefers to standardize an accordion with better multi-user security than is provided by generic bounds.

2.2.4. Key-Dependent-Input Security

Some applications involve encrypting inputs that include the key or are derived from the key in some way. For example, when the operating system swaps the contents of some memory page to disk, the storage encryption mode being used could encrypt a copy of its own key. The security proofs of most chaining modes do not cover this use case; the property required to ensure security is called *security against key-dependent inputs* (KDI security) [37–39]. As shown in [39], it is impossible to construct a KDI-secure deterministic construction (e.g., the accordion) if there is no restriction on generating KDIs.

However, it is expected that the typical application of an accordion for disk encryption will not have practical attacks in the KDI setting, as the attacks that were applicable to the initial drafts of the IEEE 1619 standard [11] are unlikely to apply to the setting where an accordion encrypts plaintexts whose size corresponds to an entire disk sector.

2.2.5. Post-Quantum Security

NIST is interested in both analysis and security proofs of the accordion in a quantum setting following the **Q1** model [40–42], which assumes that a quantum adversary can only interact with the target primitive (i.e., make queries to Encrypt(T,x) and Decrypt(T,x)) through classical communication.

2.3. Performance Targets

While performance is not as critical as security, the better an accordion performs, the more useful it is likely to be. The most important platform is likely to be relatively powerful processors in desktop or laptop computers and those used in cloud environments with hardware AES support. Performance in dedicated hardware, field programmable gate arrays, and constrained devices is also worthwhile but is not NIST's focus for an accordion.

The security requirements for an accordion require multiple passes over the message. Thus, it is almost certain to be slower than many existing one-pass block cipher modes, like GCM. However, an accordion will ideally not be significantly more expensive than twice the cost (e.g., in time, block cipher calls, gates) of GCM over the same input size.

An accordion should allow substantial parallelism for large input sizes. Modern CPUs often support multiple AES instructions or other operations in parallel, and an accordion should allow the user to take advantage of this.

An accordion should not impose too much additional overhead on small input sizes. An accordion that does a substantial amount of setup work before processing an input block will generally perform poorly on small inputs. Some applications require efficient processing of even relatively short messages.

3. Requirements for Accordion Parameters

This section proposes the ranges of sizes/lengths that should be required or supported for accordion parameters.

3.1. Block Size

NIST has proposed to approve a block cipher with 256-bit blocks [43] and, therefore, expects to develop an accordion with $n = 256$. This main accordion, denoted by Acc256, would provide very high assurance against generic cryptanalytic attacks with relatively little performance overhead.

NIST also expects to develop two accordions with $n = 128$ for compatibility with existing implementations of the AES block cipher [44]. One option, denoted by Acc128, would have an analogous design to Acc256. The second option, denoted by BBBAcc, would provide BBB security against generic cryptanalytic attacks, as discussed in Sec. 2.2.2.

3.2. Key Size

NIST requires any approved accordion to support $k = 256$ and may also allow $k = 128$ or $k = 192$ for Acc128 or BBBAcc.

3.3. Tweak Size

NIST requires all three accordions to support variable-length tweaks with some minimum bit length s_{min} and up to some maximum bit length s_{max}. For some applications, NIST expects that the accordion will need to support tweaks with a large maximum length. For example, in order to efficiently construct an AEAD derived function from an accordion, the tweak will likely contain the AEAD's associated data. The value for s_{min} should be at most that of the block size n, and the value of s_{max} should be at least 2^{48} bits. While some applications (e.g., storage encryption) can take advantage of more efficient processing for tweaks that are short and of a fixed length, supporting fixed-length tweaks alone is insufficient. An accordion must be capable of handling the full range of required tweak lengths, even if it optimizes for specific, fixed-length tweaks to enhance performance in certain use cases. The ability to process variable-length tweaks remains essential to ensure flexibility and compatibility across diverse applications.

3.4. Message Lengths

Given the granularity g, the minimum message bit length ag, and the maximum message bit length bg, an accordion supports a total of $(b - a + 1)$ allowed message lengths $\ell \in \{ag, (a+1)g, ..., bg\}$. An accordion with a granularity of one bit ($g = 1$) can process *any* message between its minimum and maximum length, whereas a scheme with a granularity of 128 bits can only process messages whose length is a multiple of 128 bits between its minimum and maximum lengths.

NIST requires a granularity of $g \leq 8$ and the minimum message length ag to be at most $2n$ bits for any accordion. NIST also requires the maximum message length to be at least 2^{96} bits for Acc256, at least 2^{48} bits for Acc128, and at least 2^{64} bits for BBBAcc.

4. Derived Functions and Applications

This section describes the derived functions of an accordion for the following three categories of applications: AEAD, tweakable encryption, and deterministic authenticated encryption (DAE). Each of these functions will have some application-specific inputs along with the message and a key. The particular use case of the function motivates a set of properties that should be achieved. For example, AEAD by definition includes a means of data authentication, while tweakable encryption does not. The following subsections provide 1) some guidelines for each of the derived functions, 2) an indication of how the functions could be constructed out of an accordion, and 3) discussion of additional security considerations.

4.1. Authenticated Encryption with Associated Data (AEAD)

An AEAD scheme uses a key, a nonce, and optional associated data to encrypt a plaintext into a ciphertext while authenticating the plaintext and associated data. One possible construction for authenticated encryption from an accordion is shown in Fig. 2. In this example, the accordion message consists of the plaintext padded with τ fixed bits (zeros) to support authentication[5], which is verified upon decryption (not depicted). This construction encodes the nonce and any associated data into the accordion tweak.

In practice, it can be difficult to ensure that nonces are not repeated. Therefore, nonce misuse resistance is an important property for an AEAD scheme. If the nonce is encoded as part of the tweak, then the security definition of the accordion ensures that repetition of the tweak[6] imposes the smallest possible security loss. Accordions are deterministic, so any invocations of the accordion on the same input — including the same tweak — will result in the same output. Otherwise, each new invocation will appear to be randomly selected from the set of not-yet-seen outputs of the correct length from the accordion function.

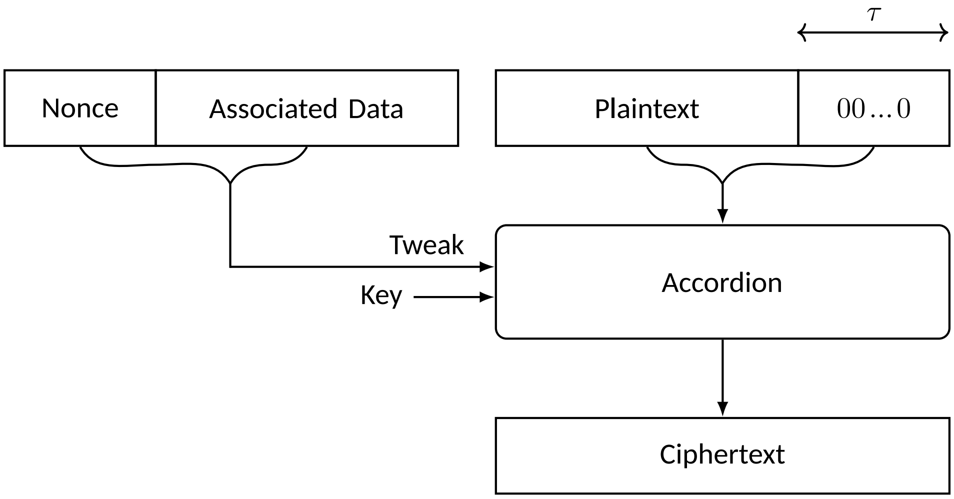

Figure 2. Accordion-based authenticated encryption using an encode-then-encipher approach

[5] It is possible to specify a message encoding that also hides the length of the plaintext.

[6] This repetition constitutes "nonce misuse" in the context of the AEAD scheme, even though tweaks are permitted to repeat at the accordion level.

4.2. Tweakable Encryption

A tweakable encryption derived function includes a tweak input to the accordion to encrypt a message. Unlike the AEAD case, an integrity check value is not encoded into the message, so message authentication is not provided. Although padding may still need to be applied to the message to reach an allowed message size, a small granularity could enable encryption without ciphertext expansion, making this derived function useful for storage encryption.

Figure 3 shows one possible derived function for tweakable encryption for storage devices. In this example, each data unit is encoded as a separate accordion message for which the accordion tweak is an encoding of the data unit's index. For an accordion to be practical for this application, changing the tweak should be very efficient.

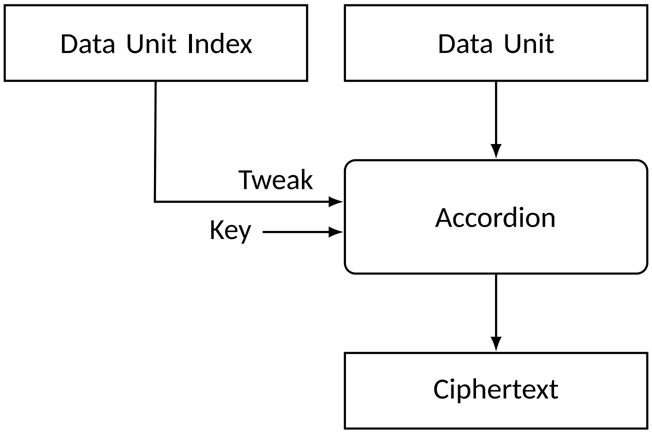

Figure 3. Accordion-based tweakable encryption for storage devices

4.3. Deterministic Authenticated Encryption

A derived function for deterministic authenticated encryption (DAE) provides authentication without a tweak. A common use case for this derived function is key wrapping. As DAE does not involve a tweak, any constant minimal length tweak could be chosen to specify the mode.

Figure 4 shows one possible construction for a derived function for key wrapping. In this example, the empty string is furnished as the accordion tweak. The plaintext input — an encoding of the key to be wrapped — is also padded with τ fixed bits (zeros) as the accordion message. Upon decryption (not depicted), the τ padding bits are verified for correctness to authenticate the message.

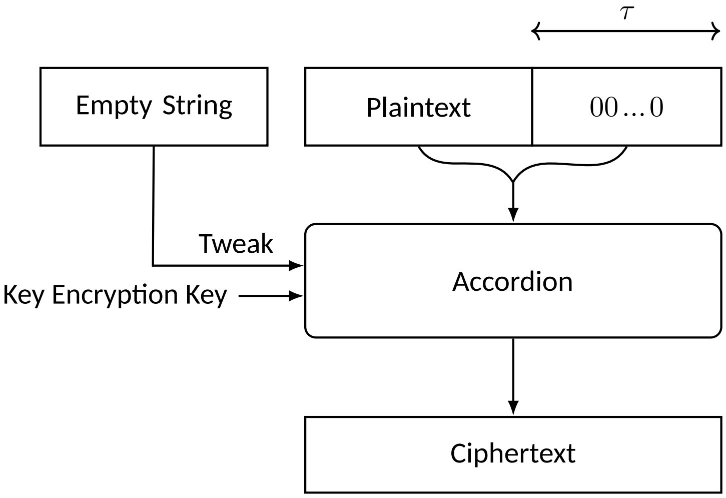

Figure 4. Accordion-based DAE for key wrapping

4.4. Security Properties of the Derived Functions

The derived functions should support each of the security properties described below to the highest extent reasonable. A specification of the derived functions should include analysis of the security goals and identify the level of security achieved for each.

4.4.1. Authentication

The AEAD and DAE derived functions both support authentication of the input data. The number of bits of authentication security is τ, which means that $2^{-\tau}$ is the maximum allowable probability that any given invalid ciphertext will be accepted as valid. The value of τ also indicates the minimum number of bits of ciphertext expansion. A derived function that supports authentication will specify the set of values that are allowed for τ, which may include options for small ciphertext expansion.

4.4.2. Key and Context Commitment

In many applications of cryptographic functions with authentication, it is important to have assurance that a given output can only be successfully authenticated when the key and other context (e.g., nonce and associated data) are correct (i.e., unchanged). In other words, it should be infeasible to find two different tuples of inputs that are cryptographically processed to the same output. This property is called *key commitment* when the key varies but the other context does not and *context commitment* when the key and other context may all vary.

Formally, the "committing security" notions by Bellare and Hoang [45] are adapted: let $C = F(K, P, U)$ be a derived function that processes key K, (plaintext) message P, and additional context U to produce (ciphertext) output C. Key and context commitment are properties for which it should be computationally infeasible to find K_1, P_1, U_1, K_2, P_2, and U_2 such that $F(K_1, P_1, U_1) = F(K_2, P_2, U_2)$. The key commitment property requires that $K_1 \neq K_2$, while the context commitment property requires that $(K_1, P_1, U_1) \neq (K_2, P_2, U_2)$. For the AEAD derived function, this notion of key (context) commitment aligns with the definition in [45] for CMT-1 (CMT-4). Analogous definitions would apply for DAE and any other derived function with authentication.

Because key commitment is important in many real-world applications, NIST expects to require this property in the derived functions that support authentication. Although context commitment would also be useful, it seems less essential and possibly more difficult to achieve. For this reason, NIST does not require context commitment but would consider designs that support this property.

4.4.3. Nonce Hiding

In some applications that involve a cryptographic function with a nonce, the nonce may reveal private data and should not be disclosed. A function that protects the contents of the nonce from disclosure is said to be *nonce-hiding* [46]. It is straightforward to construct derived functions for AEAD with this property.

4.4.4. Release of Unverified Plaintext

Cryptographic functions that provide authentication are subject to a particular kind of implementation failure: they can reveal plaintext before it has been authenticated. This is often difficult to avoid, especially when very large messages are being processed. Security under the release of unverified plaintext (RUP) is a property of a cryptographic function with authentication that ensures that the adversary does not gain an advantage if authentication fails, even if the unauthenticated plaintext is released. Several variations of this notion exist [47–50], the strongest of which requires leakage to be indistinguishable from random data. Using an encode-then-encipher approach for authenticated encryption over the accordion provides assurance of RUP security.

5. Next Steps

NIST intends to lead a collaborative, transparent process with the cryptographic community to develop accordions that are suitable for most general-purpose cryptographic applications, especially those that require confidentiality and/or authentication.

NIST expects to release a series of increasingly specific proposals for public feedback, beginning with a recommendation for a high-level design approach based on the current state of the art and ending with a full specification that meets the requirements in this publication.

The ciphermodes-forum@list.nist.gov emailing list has been established for dialogue regarding NIST's Block Cipher Modes project. To subscribe to the mailing list, visit https://groups.google.com/a/list.nist.gov/g/ciphermodes-forum.

References

[1] Chen YL, Davidson M, Dworkin M, Kang J, Kelsey J, Sasaki Y, Sönmez Turan M, Chang D, Mouha N, Thompson A (2024) Proposal of Requirements for an Accordion Mode – Discussion Draft for the NIST Accordion Mode Workshop 2024, National Institute of Standards and Technology, Workshop Discussion Draft. Available at https://csrc.nist.gov/files/pubs/other/2024/04/10/proposal-of-requirements-for-an-accordion-mode-dis/iprd/docs/proposal-of-requirements-for-an-accordion-mode-discussion-draft.pdf.

[2] Dworkin M (2001) Recommendation for Block Cipher Modes of Operation: Methods and Techniques, NIST SP 800-38A. https://doi.org/10.6028/NIST.SP.800-38A.

[3] Dworkin M (2010) Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode, Addendum to NIST SP 800-38A. https://doi.org/10.6028/NIST.SP.800-38A-Add.

[4] Dworkin M (2005) Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication, (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-38B. DOI:10.6028/NIST.SP.800-38B.

[5] Dworkin M (2004) Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, NIST SP 800-38C. https://doi.org/10.6028/NIST.SP.800-38C.

[6] Dworkin M (2007) Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, NIST SP 800-38D. https://doi.org/10.6028/NIST.SP.800-38D.

[7] Dworkin M (2010) Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices, NIST SP 800-38E. https://doi.org/10.6028/NIST.SP.800-38E.

[8] Dworkin M (2012) Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping, NIST SP 800-38F. https://doi.org/10.6028/NIST.SP.800-38F.

[9] Dworkin M (2016) Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption, NIST SP 800-38G. https://doi.org/10.6028/NIST.SP.800-38G.

[10] Dworkin M, Mouha N (2025) Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption, NIST SP 800-38G Revision 1 Second Public Draft. https://doi.org/10.6028/NIST.SP.800-38Gr1.2pd.

[11] Mouha N, Dworkin M (2024) Report on the Block Cipher Modes of Operation in the NIST SP 800-38 Series, NISTIR 8459. https://doi.org/10.6028/NIST.IR.8459.

[12] Thompson A, Sönmez Turan M (2024) NIST Workshop on the Requirements for an Accordion Cipher Mode 2024: Workshop Report, NISTIR 8537. https://doi.org/10.6028/NIST.IR.8537.

[13] Liskov MD, Rivest RL, Wagner DA (2002) Tweakable block ciphers. *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*, ed Yung M (Springer), *Lecture Notes*

in Computer Science, Vol. 2442, pp 31–46. DOI:10.1007/3-540-45708-9_3. Available at https://doi.org/10.1007/3-540-45708-9_3

[14] Halevi S, Rogaway P (2003) A Tweakable Enciphering Mode. *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, ed Boneh D (Springer), *Lecture Notes in Computer Science*, Vol. 2729, pp 482–499. Available at https://doi.org/10.1007/978-3-540-45146-4_28.

[15] Halevi S, Rogaway P (2004) A Parallelizable Enciphering Mode. *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, ed Okamoto T (Springer), *Lecture Notes in Computer Science*, Vol. 2964, pp 292–304. Available at https://doi.org/10.1007/978-3-540-24660-2_23.

[16] Schroeppel R (1998) Hasty Pudding Cipher Specification. Available at http://richard.schroeppel.name:8015/hpc/hpc-spec.

[17] Hoang VT, Krovetz T, Rogaway P (2017) AEZ v5: Authenticated encryption by enciphering. https://www.cs.ucdavis.edu/~rogaway/aez/aez.pdf.

[18] Crowley P, Huckleberry N, Biggers E (2023) Length-preserving encryption with HCTR2, The Third NIST Workshop on Block Cipher Modes of Operation 2023. https://csrc.nist.gov/csrc/media/Events/2023/third-workshop-on-block-cipher-modes-of-operation/documents/accepted-papers/Length%20Preserving%20Encryption.pdf.

[19] Campbell P (2023) GLEVIAN and VIGORNIAN: Robust beyond-birthday AEAD modes, Cryptology ePrint Archive, Paper 2023/1379. Available at https://eprint.iacr.org/2023/1379.

[20] Dobraunig C, Matusiewicz K, Mennink B, Tereschenko A (2024) Efficient Instances of Docked Double Decker With AES, and Application to Authenticated Encryption, NIST Workshop on the Requirements for an Accordion Cipher Mode 2024. https://csrc.nist.gov/csrc/media/Events/2024/accordion-cipher-mode-workshop-2024/documents/papers/efficient-instances-docked-double-decker.pdf.

[21] Arciszewski S (2018) XChaCha: eXtended-nonce ChaCha and AEAD_XChaCha20_Poly1305, IETF Internet-Draft. https://datatracker.ietf.org/doc/html/draft-arciszewski-xchacha-02.

[22] Crowley P, Biggers E (2018) Adiantum: length-preserving encryption for entry-level processors. *IACR Trans Symmetric Cryptol* 2018(4):39–61. Available at https://doi.org/10.13154/tosc.v2018.i4.39-61.

[23] Bhaumik R, List E, Nandi M (2018) ZCZ - Achieving n-bit SPRP Security with a Minimal Number of Tweakable-Block-Cipher Calls. *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I*, eds Peyrin T, Galbraith SD (Springer), *Lecture Notes in Computer Science*, Vol. 11272, pp 336–366. Available at https://doi.org/10.1007/978-3-030-03326-2_12.

[24] Bellare M, Rogaway P (2000) Encode-Then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography. *Advances in Cryptology - ASI-*

ACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings, ed Okamoto T (Springer), *Lecture Notes in Computer Science*, Vol. 1976, pp 317–330. Available at https://doi.org/10.1007/3-540-44448-3_24.

[25] National Institute of Standards and Technology (2016) Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process. Available at https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[26] Moody D, Perlner R, Regenscheid A, Robinson A, Cooper D (2024) Transition to Post-Quantum Cryptography Standards, NISTIR 8547. https://doi.org/10.6028/NIST.IR.8547.ipd.

[27] Mouha N, Luykx A (2015) Multi-key Security: The Even-Mansour Construction Revisited. *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, eds Gennaro R, Robshaw M (Springer), *Lecture Notes in Computer Science*, Vol. 9215, pp 209–223. Available at https://doi.org/10.1007/978-3-662-47989-6_10.

[28] Hoang VT, Tessaro S (2016) Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, eds Robshaw M, Katz J (Springer), *Lecture Notes in Computer Science*, Vol. 9814, pp 3–32. Available at https://doi.org/10.1007/978-3-662-53018-4_1.

[29] Hoang VT, Tessaro S (2017) The multi-user security of double encryption. *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, eds Coron J, Nielsen JB, *Lecture Notes in Computer Science*, Vol. 10211, pp 381–411. Available at https://doi.org/10.1007/978-3-319-56614-6_13.

[30] Bose P, Hoang VT, Tessaro S (2018) Revisiting AES-GCM-SIV: multi-user security, faster key derivation, and better bounds. *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*, eds Nielsen JB, Rijmen V (Springer), *Lecture Notes in Computer Science*, Vol. 10820, pp 468–499. Available at https://doi.org/10.1007/978-3-319-78381-9_18.

[31] Hoang VT, Tessaro S, Thiruvengadam A (2018) The multi-user security of gcm, revisited: Tight bounds for nonce randomization. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, eds Lie D, Mannan M, Backes M, Wang X (ACM), pp 1429–1440. Available at https://doi.org/10.1145/3243734.3243816.

[32] Chen YL, Tessaro S (2021) Better security-efficiency trade-offs in permutation-based two-party computation. *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part II*, eds Tibouchi M, Wang H

(Springer), *Lecture Notes in Computer Science*, Vol. 13091, pp 275–304. Available at https://doi.org/10.1007/978-3-030-92075-3_10.

[33] Chen YL (2022) A modular approach to the security analysis of two-permutation constructions. *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part I*, eds Agrawal S, Lin D (Springer), *Lecture Notes in Computer Science*, Vol. 13791, pp 379–409. Available at https://doi.org/10.1007/978-3-031-22963-3_13.

[34] Bhattacharya S, Nandi M (2021) Luby-rackoff backwards with more users and more security. *Advances in Cryptology - ASIACRYPT 2021 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part III*, eds Tibouchi M, Wang H (Springer), *Lecture Notes in Computer Science*, Vol. 13092, pp 345–375. Available at https://doi.org/10.1007/978-3-030-92078-4_12.

[35] Choi W, Kim H, Lee J, Lee Y (2022) Multi-user security of the sum of truncated random permutations. *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part II*, eds Agrawal S, Lin D (Springer), *Lecture Notes in Computer Science*, Vol. 13792, pp 682–710. Available at https://doi.org/10.1007/978-3-031-22966-4_23.

[36] Chen YL, Choi W, Lee C (2023) Improved multi-user security using the squared-ratio method. *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part II*, eds Handschuh H, Lysyanskaya A (Springer), *Lecture Notes in Computer Science*, Vol. 14082, pp 694–724. Available at https://doi.org/10.1007/978-3-031-38545-2_23.

[37] Ball MV (2008) NIST's consideration of XTS-AES as standardized by IEEE Std 1619-2007, https://csrc.nist.gov/CSRC/media/Projects/Block-Cipher-Techniques/documents/BCM/Comments/XTS/XTS_comments-Ball.pdf.

[38] Black J, Rogaway P, Shrimpton T (2002) Encryption-Scheme Security in the Presence of Key-Dependent Messages. *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002. Revised Papers*, eds Nyberg K, Heys HM (Springer), *Lecture Notes in Computer Science*, Vol. 2595, pp 62–75. Available at https://doi.org/10.1007/3-540-36492-7_6.

[39] Halevi S, Krawczyk H (2007) Security under key-dependent inputs. *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*, eds Ning P, di Vimercati SDC, Syverson PF (ACM), pp 466–475. Available at https://doi.org/10.1145/1315245.1315303.

[40] Zhandry M (2012) How to construct quantum random functions. *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012* (IEEE Computer Society), pp 679–687. DOI:10.1109/FOCS.2012.37. Available at https://doi.org/10.1109/FOCS.2012.37

[41] Kaplan M, Leurent G, Leverrier A, Naya-Plasencia M (2016) Quantum differential and linear cryptanalysis. *IACR Trans Symmetric Cryptol* 2016(1):71–94. DOI:10.13154/TOSC.V2016.I1.71-94. Available at https://doi.org/10.13154/tosc.v2016.i1.71-94

[42] Bonnetain X, Hosoyamada A, Naya-Plasencia M, Sasaki Y, Schrottenloher A (2019) Quantum attacks without superposition queries: The offline simon's algorithm. *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I*, eds Galbraith SD, Moriai S (Springer), *Lecture Notes in Computer Science*, Vol. 11921, pp 552–583. DOI:10.1007/978-3-030-34578-5_20. Available at https://doi.org/10.1007/978-3-030-34578-5_20

[43] National Institute of Standards and Technology (2024) NIST Proposes to Standardize a Wider Variant of AES. https://csrc.nist.gov/news/2024/nist-proposes-to-standardize-wider-variant-of-aes.

[44] National Institute of Standards and Technology (Published 2001; Updated 2023) Advanced Encryption Standard (AES), FIPS 197. https://doi.org/10.6028/NIST.FIPS.197-upd1.

[45] Bellare M, Hoang VT (2022) Efficient schemes for committing authenticated encryption. *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part II*, eds Dunkelman O, Dziembowski S (Springer), *Lecture Notes in Computer Science*, Vol. 13276, pp 845–875. DOI:10.1007/978-3-031-07085-3_29. Available at https://doi.org/10.1007/978-3-031-07085-3_29

[46] Bellare M, Ng R, Tackmann B (2019) Nonces are noticed: AEAD revisited. *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, eds Boldyreva A, Micciancio D (Springer), *Lecture Notes in Computer Science*, Vol. 11692, pp 235–265. Available at https://doi.org/10.1007/978-3-030-26948-7_9.

[47] Andreeva E, Bogdanov A, Luykx A, Mennink B, Mouha N, Yasuda K (2014) How to securely release unverified plaintext in authenticated encryption. *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, eds Sarkar P, Iwata T (Springer), *Lecture Notes in Computer Science*, Vol. 8873, pp 105–125. Available at https://doi.org/10.1007/978-3-662-45611-8_6.

[48] Hoang VT, Krovetz T, Rogaway P (2015) Robust Authenticated-Encryption AEZ and the Problem That It Solves. *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, eds Oswald E, Fischlin M (Springer), *Lecture Notes in Computer Science*, Vol. 9056, pp 15–44. Available at https://doi.org/10.1007/978-3-662-46800-5_2.

[49] Barwell G, Page D, Stam M (2015) Rogue decryption failures: Reconciling ae robustness notions. *IMACC 2015: Proceedings of the 15th IMA International Conference on*

Cryptography and Coding, ed Groth J (Springer), *Lecture Notes in Computer Science*, Vol. 9496, pp 94–111. Available at https://doi.org/10.1007/978-3-319-27239-9_6.

[50] Ashur T, Dunkelman O, Luykx A (2017) Boosting authenticated encryption robustness with minimal modifications. *Annual International Cryptology Conference* (Springer), pp 3–33.