Introduction to the Accordion Mode and Derived Functions

Alyssa Thompson

June 20, 2024

NIST Workshop on the Requirements for an Accordion Cipher Mode 2024
NIST intends to standardize an *accordion*, or tweakable VIL-SPRP, mode of operation.

This talk: terminology and definitions
- accordion mode
- security goals
- derived functions

More details: “Proposal of Requirements for an Accordion Mode” [2]
The term *accordion mode* may be new, but the work is not ...


- Bellare and Rogaway [1] achieved authenticated encryption using the encode-then-encipher approach on an SPRP

- Many other examples exist, these are just a few
## Parameters of the Accordion Mode

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>block size of the underlying block cipher</td>
</tr>
<tr>
<td>$k$</td>
<td>length of the secret key</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_{\text{min}}$</td>
<td>minimum allowed tweak size</td>
</tr>
<tr>
<td>$s_{\text{max}}$</td>
<td>maximum allowed tweak size</td>
</tr>
<tr>
<td>$s$</td>
<td>bit-size of a given tweak</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g$</td>
<td>granularity of allowed message sizes</td>
</tr>
<tr>
<td>$a$</td>
<td>integer: $ag$ is the minimum allowed message size</td>
</tr>
<tr>
<td>$b$</td>
<td>integer: $bg$ is the maximum allowed message size</td>
</tr>
<tr>
<td>$\ell$</td>
<td>bit-size of a given message</td>
</tr>
</tbody>
</table>

$\ell \in \{ag, (a + 1)g, \ldots, bg\}$
Accordion Mode Notation

\[ K \in \{0, 1\}^k \quad \text{secret key} \]
\[ T \in \{0, 1\}^s \quad \text{tweak} \]
\[ M \in \{0, 1\}^\ell \quad \text{message} \]
\[ C \in \{0, 1\}^\ell \quad \text{ciphertext} \]

encryption: \[ A.\text{enc}(K, T, M) = C \]

decryption: \[ A.\text{dec}(K, T, C) = M \]
Accordion Mode Definition

**enciphering mode**

- length preserving (VIL)
- message space $\mathcal{M} = \bigcup_{\ell \in \mathbb{L}} \{0, 1\}^\ell$
- $enc : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{M}$
Accordion Mode Definition

**enciphering mode**
- length preserving (VIL)
- message space $\mathcal{M} = \bigcup_{\ell \in \mathbb{L}} \{0, 1\}^\ell$
- $\text{enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$

→ **tweakable enciphering mode**
- length preserving (VIL)
- tweak space $\mathcal{T}$ (tweakable)
- $\text{enc} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$
**Accordion Mode Definition**

**Enciphering mode**
- length preserving (VIL)
- message space $\mathcal{M} = \bigcup_{\ell \in \mathcal{L}} \{0, 1\}^\ell$
- $\text{enc} : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{M}$

$\rightarrow$ **Tweakable enciphering mode**
- length preserving (VIL)
- tweak space $\mathcal{T}$ (tweakable)
- $\text{enc} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \rightarrow \mathcal{M}$

$\rightarrow$ **Accordion mode**
- length preserving (VIL)
- tweak space $\mathcal{T}$ (tweakable)
- adaptive CCA model (SPRP)
- $\text{enc} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \rightarrow \mathcal{M}$
Security Goal of the Accordion Mode

\[ K \leftarrow \{0, 1\}^k \text{ and } b \leftarrow \{0, 1\} \]

**Challenger**

- Select and remember \( \Pi_{T,|x|} \)
- \( \text{query: } \text{encrypt}(T, x) \text{ or decrypt}(T, x) \)

**Distinguisher**

- \( \text{Challenger, } b = 0 \)
  - \( \text{answer encrypt: } y \leftarrow A.\text{enc}(K, T, x) \) or \( y \leftarrow A.\text{dec}(K, T, x) \)
- \( \text{Challenger, } b = 1 \)
  - \( \text{answer encrypt: } y \leftarrow \Pi_{T,|x|}(x) \) or \( y \leftarrow \Pi_{T,|x|}^{-1}(x) \)

**Distinguisher**

- \( \text{receive: } y \)

**Distinguisher**

- \( \text{guess: } b = 0 \text{ or } b = 1 \)

*Advantage of a \( (q, \sigma, t) \)-distinguisher should be negligible.*
Accordion Mode Properties

- The accordion “shrinks” or “expands” to match the size of the input
Accordion Mode Properties

- The accordion “shrinks” or “expands” to match the size of the input.
Accordion Mode Properties

- The accordion “shrinks” or “expands” to match the size of the input

![Diagram showing Accordion Mode Properties](image)
Accordion Mode Properties

- The accordion “shrinks” or “expands” to match the size of the input

- Any change to the inputs has a randomizing effect on the outputs
Why an accordion?

An accordion ...

- can achieve better security properties and additional features compared to current NIST standards
  - robust, length preserving
  - nonce misuse resistance, short tags, key commitment

- operates on an underlying block cipher
  - simplifies security analysis
  - can take advantage of AES instructions

- has a solid base of existing research and development
  - exact match does not necessarily exist (yet)
Derived Functions

- called by an application
- encodes the inputs to the accordion mode
- no crypto
Derived Functions
Authenticated Encryption with Associated Data

Applications: anywhere AEAD is used today
  ex - TLS, IPsec, SSH

Inputs: nonce $N$, associated data $AD$

Parameters: $\tau =$ number of authentication bits

Security Goal: $\leq 2^{-\tau}$ probability of forging
Derived Functions
Authenticated Encryption with Associated Data

- encoded-then-encipher paradigm
- \( \tau \) or more bits for integrity check value
- nonce-misuse resistance and RUP security
- variable length, relatively long tweaks
Derived Functions
Tweakable Encryption

Applications: storage encryption
Inputs: tweak $T$
Parameters: same as accordion
Characteristics: supports no ciphertext expansion
$\Rightarrow$ small granularity useful
Derived Functions
Tweakable Encryption

- no authentication/integrity check
- key-dependent input is possible
- tweak will change often
Derived Functions
Deterministic Authenticated Encryption

Applications: key wrapping
Inputs: fixed (or empty) tweak
Parameters: $\tau = \text{number of authentication bits}$
Security Goal: $\leq 2^{-\tau}$ probability of forging
Derived Functions
Deterministic Authenticated Encryption

- encode-then-encipher paradigm
- $\tau$ or more bits for integrity check value
- support for empty tweak
Feedback Wanted

Accordion mode - any general comments?

Do the three derived functions cover the needed applications?

Comments on the constructions for the derived functions. Others ideas to consider?

Other comments on this content?

Proposed requirements, additional features and more to come.
References


