

ATNA-CIPHER, LLC.(ACL)  
*Accordion Cipher-mode  
Preferable Features*



**ATNA-CIPHER, LLC.**  
COEVAL AUTHENTICATED ENCRYPTION

Tushar Patel  
*Lead Architect/Owner*  
ATNA-CIPHER, LLC.  
(408)242-5016  
[si@atnacipher.com](mailto:si@atnacipher.com)  
P/O. Box 2130, Sunnyvale, CA 94087

# Introduction

- ATNA-CIPHER, LLC. (aka ACL) is the incubation entity for the development of an accordion and tweakable style encryption cipher(+/-mode.) , namely, “atnaCM.”
- As NIST is not seeking a full proposal submission at this time; ACL would like to highlight some key features in the following 5 broad are categories as recommendations for an Accordion Mode.
  1. **Enciphering Properties** like Key Sizes, Block Cipher Sizes and Tweakable Block sizes
  2. **Parallelism** relating to large input sizes
  3. **Confirmation** of keys like encryption, integrity, etc.
  4. **Fast Drop Tags** for Authenticated Encryption
  5. **Padding Attacks Prevention** for all tweaks and enciphering properties
- Subsequent slides present each ones of these topics in some light detail.
- Note: In the slides, messages like payloads/packets are 1-unit of encryption from a possible larger set.

# Enciphering Properties (1)

- Cryptography is consistently changing; however, future adaptability has been tedious and difficult,
  - e.g., PQC finalization, Legacy RSA deprecation, SHA-1 deprecation or Next-gen Cryptographic Adaptation of ECDHE ECDSA.
- Currently, Cipher Blocks are integral multiples of 128-bits, AES-128 or Rijndael-256-256 (i.e., 2\*128-bit.)
- Cipher Key Sizes and generally multiples of 16-bytes, 24-bytes or 32-bytes, i.e., AES-(128/192/256).
- Recommended hash sizes as per CNSA 2.0 are (SHA-384) 48-bytes, (SHA-512) 64-bytes or SHA-3 Hashes as integral multiples of 128-bits.
- Disk or Container Block Sizes are in Powers of Two, i.e.,  $2^n$  where,  $n \in \{9,10, \dots, (n-1), n\}$  (multiples of 128-bits)
  - (note: blocks can be smaller than 512 in smaller systems)
- Due to interoperability between HW registers (32/64/128/256)-bit and Encryption units,
  - Architectures define blocks in multiples of Cipher Block Sizes (i.e., 128-bits), e.g., AES-NI/ARM: AES-128, AES-256, AES-512, etc.

# Enciphering Properties (2)

- Most implementations are limited to 4096 cipher blocks per request, (e.g., TLS, Missing Jumbo in MACSec)
- Given these factors, it would be highly preferable for Accordion modes to,
  - Define Accordion Cipher-Blocks, Hashes and Keys as multiples of 128-bits. (90% cases)
  - For other sizes, padding rules must be defined and security validated/assured. (10% cases.)
  - Update number of cipher blocks to 16384 (i.e.,  $2^{14}$ ) from 4096 (i.e.,  $2^{12}$ )
  - with Rijndael-256-256, it has the necessary future proof single payload size support.
- This RISC style approach permits Accordion HW designs to be compatible, efficient and within safe manufacturing bounds.

# Parallelism

- Current architectures support **three** levels of pipelined Parallelism.
  - **Key Schedule Parallelism** (like current AES key schedules)
  - **Encryption Stage Parallelism** like the 14 stages of AES (128-bit block, 256-bit keys)
  - **Tweakable Block** (like AES-NI encryption in multiples of 128-bit)
- New designs should introduce **two** more multiprocessing methods
  - **Per Message Parallelism**: For large packets, implementations can support multi-processing across multiple AES units.
  - **Integrity or MAC parallelism**: Most implementations miss multi-processing MACs or integrity checks.
- This initiative should prove out as the most important performance enhancement.
- These **5-levels of parallelism** are recommended for Accordion modes.

# Confirmations

- The penalties of missing/failing a decryption in a pipeline (e.g., HW) are
  - Costly in proportion to speed.
  - Most HW complete ops. in two passes. (HW packet recirculation is common.)
  - Using Key Confirmations prevent such penalties and utmost important in AEAD.
- All messages should be confirming both Integrity and Enciphering Keys, however, some applications may keep them optional.
  - Persistent or resident encryption like disk and storage may not need this.
- Implementations can include other confirmations like enciphering or domain parameters for better application related security assurances.
- This will be a highly preferable feature set for the Accordion mode.

# Fast Drop Tagging

- Fast drop tags is a method to achieve message parallel cross-compatibility and confirmations (e.g., keys.)
- Many current modes push items like Flow Control and Attack Prevention to the upper layers.
  - These layers may allow DoS attacks if not using AEAD ciphers over upper-layer headers,
    1. Inline – DoS as packets can get queued in the stack until flow control processing.
    2. No upper bound – An attacker can DoS replay in leaps shortening the window.
- In compromised and mirrored hypervisors, VMs and containers, it may be possible to mirror ciphertext based on protocol knowledge, e.g., SPI, RTSP headers to a compromised unit with an upper layer side-channel in JavaScript, e.g., attacks on multicast groups.
  - While subject to bad implementations, such attacks have been known to occur in the past.
  - Not all systems can incur the costs of Enclaves, or Cloud/Global TPMs and HSMs.
- Fast Drop Tags implementing bounds on flow control and providing service segmentation adaptation and assembly in cipher-modes to thwart or eliminate such attacks.
- It is highly preferable for Accordion designs to include this functionality.



# Prevention of Padding Attacks (1)

- Previous cipher-modes have had some issues with the first-two blocks (CTR/CBC) (due to implementation error) leading to
  - The introduction of authenticated encryption like AES-GCM.
  - Repetition Padding Attacks due to Chosen Plaintext and Chosen Ciphertexts + padding.
  - Attacks from late-stage verification of cryptographic algorithms (a FIPS 140-2/3 release is usually 6 months after an initial release.)
- Padding and enciphering must support, both bits and byte modes.
  - Bit-Mode is for IoT and other stream applications like MPEG bit-fields.
  - Variable Bit-padded cipher blocks are more difficult to crack than Byte-padded cipher blocks.
  - Currently less than 96-bits can be brute-forced. This applies to certain fields securely encrypted, however, the field itself can be brute-forced or rainbow tabled.



# Prevention of Padding Attacks (2)

- Implementations must use different padding bits for Integrity Calculations and Enciphering Padding Schemes.
  - Additionally necessary to support integrity checking at intermediate nodes in transport without decryption.
- It should be a **must** to provide the proofs and results alongside submissions
  - Theoretical Proofs of IND CCA1, CCA2 and IND-CPA, IND-CPA2
  - Formalized testing of the same and provided under the new FIPS ACTS (i.e., CAVP test).
- This is already a required property of an Accordion mode.

# Accordion Compliance

## Section 3

- Modes must support selecting parameters to comply with the 3 types mentioned in Section 3. of the Accordion requirements.
  - **ACL postpones the discussion of its atnaCM details until the final requirements of Accordion mode** making some adjustments if necessary.
- Approaches of Accordion Mode should support
  - **Segmentation** – Allowing or preventing access to sub-segments of ciphertext.
    - Such support should **allow random-access to ciphertext sub-segments**.
  - As kernel sk-buffs (Linux), mbufs (BSD) only **allow a tail increment of (36/40)-bytes**.
    - Current Verification Tags or MAC(s) must be within this limit to prevent performance loss due to fragmenting an sk-buff in two.
    - Also simplifies message exchange across the OS kernel to user space interfaces.
  - Should **support extendibility and adaptability** methods due to diverse application needs.
  - Should facilitate backdoor free, **data search in the encrypted form and law enforcement**.

# Accordion Compliance (Misc.)

- Approaches of Accordion Mode should support
  - **Segmentation** – Allowing or preventing access to sub-segments of ciphertext.
    - Such support should **allow random-access to ciphertext sub-segments**.
  - As kernel sk-buffs (Linux), mbufs (BSD) only **allow a tail increment of (36/40)-bytes**.
    - Current Verification Tags or MAC(s) must be within this limit to prevent performance loss due to fragmenting an sk-buff in two.
    - Also simplifies message exchange across the OS kernel to user space interfaces.
  - Should **support extendibility and adaptability** methods due to diverse application needs.
  - Should facilitate backdoor free, **data search in the encrypted form** and **law enforcement**.

# Accordion Feedback

- Parameter Lengths – Key ( $n * 128$ -bit), Tweak ( $n * \text{cipher-block-length}$ ), data-input (any size with efficient lengths (90% cases) and inefficient lengths (10% cases))
- 256-bit cipher blocks – Yes, in short, any multiple of 128-bits.
- Security Goals – many, in general – known attack free at 256 bit. min strength, PRF: Keyed, min 384-bits (~strength 192 bits or more)
- AEAD – Yes, in general: support all three operational cases in Accordion Requirements, however, there may be algorithms that a preferred in each case.
- Potential Design Strategies – SW/HW/FW Codesign, Facilitate fast path applications
- Performance Targets – At optimal rates, it should be faster than comparable AES-GCM.
- Please review the document “*ATNA Accordion Cipher-mode Proposal Summary*” submitted alongside this presentation for some additional information.

# References

1. NIST SP800-38(A/B/C/D/E/F/G), SP800-90Ar1, NIST SP800-90B, NIST PQC Round 4., NIST SP800-131Ar2 Proposal of Requirements for an Accordion Mode
2. FIPS 140-2/3, 180-1,180-2,180-3,180-4, FIPS 198-1, SP800-108., ACTS, CAVP Algorithm Tests, CMVP and other FIPS and Common Criteria Specifications.
3. A) Thomas Leighton – Morgan Kaufmann Publishers, Parallel Algorithm Architectures, 1992. b) Dr. Donald Knuth – Addison Wesley Publishers, The Art of Computer Programming, Vol. 1 through 4B.c) William Stallings – Cryptography and Networking Security
4. The Crossed Cube Architecture for Parallel Computation – Kemal Efe, Transactions on Parallel and Distributed Systems, Vol. 3, 0.5, Sept. 1992.
5. GCM Multiple: a) The Galois/Counter Mode of Operation (GCM), D. McGrew, J Viega, b) Development of the Advanced Encryption Standard Aug 16, 2021, c) Authentication Weakness in AES-GCM, Neil Ferguson, 2005, d) Authentication Failures in NIST version of GCM, Antoine Joux, e) On the Construction of Variable-Input-Length Ciphers, M. Bellare, P. Rogaway, f) Padding Oracle Attacks on CBC-mode Encryption with Secret and Random IVs- Arnold K. L. Yau? , Kenneth G. Paterson and Chris J. Mitchell, g) A Tweakable Encryption Mode, S. Halevi and P. Rogaway, h) Proposals for Standardization of Encryption Schemes John Preuß Mattsson, Ben Smeets, Erik Thormarker *Ericsson*, i) Different Types of Attacks on Block Ciphers - Wageda Ibrahim Al Sobky, Hala Saeed Omar, j) Cycling Attacks on GCM, GHASH and Other Polynomial MACs and Hashes- Markku-Juhani O. Saarinen, k) Partitioning Oracle Attacks Julia Len Paul Grubbs Thomas Ristenpart, l) Authentication Key Recovery on Galois/Counter Mode (GCM) - John Mattsson and Magnus Westerlund, m) Adiantum: length-preserving encryption for entry-level processors - Paul Crowley and Eric Biggers
6. A) Bit Twiddling Hacks, Sean Eron Anderson, Stanford., f) Digital Design – Nicholas L. Pappas. g) Arithmetic Tutorial Collection, Douglas W. Jones University of IOWA – 2001 h) Operating Systems Concepts – 10<sup>th</sup> Edition, Silberschatz, Galvin, Gagne – Wiley 2018.
7. PQC a) Shor’s Algorithm: <https://www.nccoe.nist.gov/sites/default/files/2023-12/pqc-migration-nist-sp-1800-38b-preliminary-draft.pdf>, b) Grover’s Algorithm : <https://csrc.nist.gov/csrc/media/Events/2024/fifth-pqc-standardization-conference/documents/papers/on-practical-cost-of-grover.pdf>
8. There are many other books, papers, presentation and documents over the years and hence, a full list is not possible to reproduce here.

# Acknowledgements

- ACL hopes this presentation is helpful towards the final requirements for the Accordion Mode.
- *Sincere appreciation to NIST for organizing this event and allowing ACL to present.*
- *Sincere appreciation to all the Seers, Attendees, Mentors, Colleagues, Well-wishers, ACL members, and Family for your time, dedication, suggestions and support for the “atnaCM” solution.*
- *Details, Questions, Concerns? Info*
  - [si@atnacipher.com](mailto:si@atnacipher.com)
  - <https://www.atnacipher.com>

