Toward a New Block Cipher Mode Standard: Reasoning about Requirements

Featuring the NECST Framework

Nicky Mouha

June 20, 2024

Announcements

- NIST Workshop on Formal Methods within Certification Programs (FMCP 2024)
  - July 23-25, This Room, $205

Announcements

- NIST Workshop on Formal Methods within Certification Programs (FMCP 2024)
  - July 23-25, This Room, $205
- Conference for Failed Approaches and Insightful Losses in Cryptology (CFAIL 2024)
  - August 17, Santa Barbara, CA, $150 (Students: $80) if also attending CRYPTO 2024

Announcements

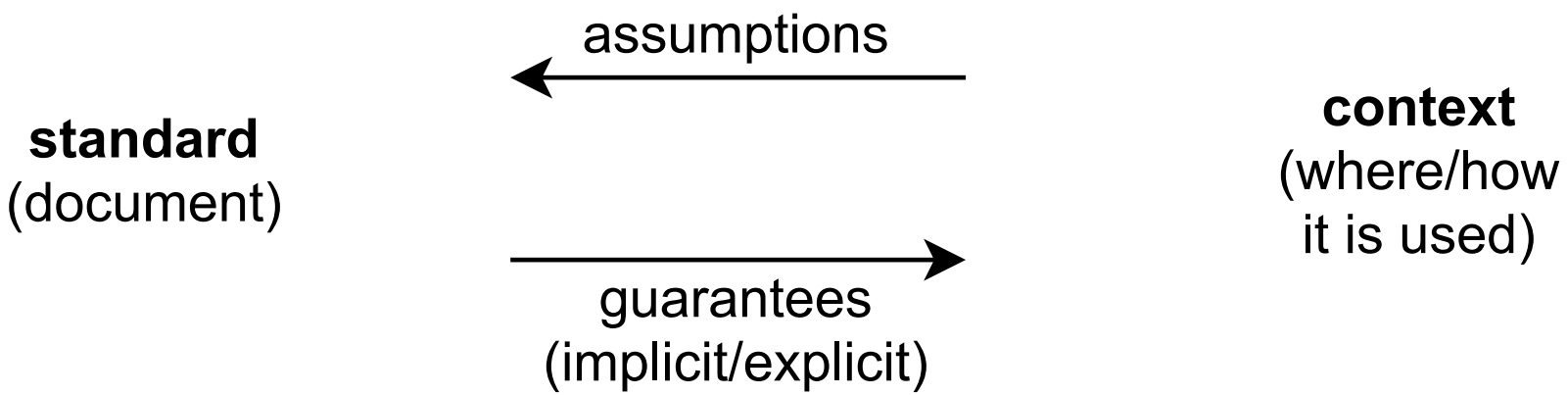
- NIST Workshop on Formal Methods within Certification Programs (FMCP 2024)
  - July 23-25, This Room, $205
- Conference for Failed Approaches and Insightful Losses in Cryptology (CFAIL 2024)
  - August 17, Santa Barbara, CA, $150 (Students: $80) if also attending CRYPTO 2024
- Information Security Conference (ISC 2024)
  - October 23-25, Arlington, VA

Standards Review

- NIST commitment:
  - Periodical review of standards
- First review: AES (FIPS 197)
  - NISTIR 8319: Review of the AES (July 2021)
- Next: Modes of operation (SP 800-38 Series)
  - Draft NISTIR 8459: Modes Report (March 2023)

Methodology

- Where is the standard used?
- What security properties are required there?
- Does failure of security properties lead to **attacks**?



Security Models

- E.g., known/chosen plaintext attacks, but...
  - Purpose of encryption if attacker already knows the plaintext?
  - How can an attacker choose the plaintext?

Security Models

- E.g., known/chosen plaintext attacks, but...
  - Purpose of encryption if attacker already knows the plaintext?
  - How can an attacker choose the plaintext?

- Again, focus on **attacks**...
  - Exhaustive search is infeasible, but...
    ...side-channel attacks recover key in minutes?
  - Block cipher secure against related-key attacks, or...
    ...disallow generating keys with known relations?
  - Vulnerabilities (CVE numbers)?

NIST SP 800-38 Series

- NIST SP 800-38A + Add: ECB, CBC, CFB, OFB, CTR
- NIST SP 800-38B: CMAC
- NIST SP 800-38C: CCM
- NIST SP 800-38D: GCM
- NIST SP 800-38E: XTS-AES
- NIST SP 800-38F: KW, KWP, TKW
- NIST SP 800-38G: FF1, FF3

NIST SP 800-38 Series

- NIST SP 800-38A + Add: ECB, CBC, CFB, OFB, CTR
  - CPA security (except ECB)
- NIST SP 800-38B: CMAC
  - MAC security
- NIST SP 800-38C: CCM
  - CPA + MAC = CCA security
- NIST SP 800-38D: GCM
  - CPA + MAC = CCA security
- NIST SP 800-38E: XTS-AES
  - CCA up-to-block (without MAC)
- NIST SP 800-38F: KW, KWP, TKW
  - CCA (without MAC)
- NIST SP 800-38G: FF1, FF3
  - CCA (without MAC)

NIST SP 800-38 Series

- NIST SP 800-38A + Add: ECB, CBC, CFB, OFB, CTR
  - CPA security (except ECB), IV (except ECB)
- NIST SP 800-38B: CMAC
  - MAC security
- NIST SP 800-38C: CCM
  - CPA + MAC = CCA security, nonce
- NIST SP 800-38D: GCM
  - CPA + MAC = CCA security, nonce
- NIST SP 800-38E: XTS-AES
  - CCA up-to-block (without MAC), tweak
- NIST SP 800-38F: KW, KWP, TKW
  - CCA (without MAC), no tweak
- NIST SP 800-38G: FF1, FF3
  - CCA (without MAC), tweak

Where Are 38A Modes Used?

- Length-preserving encryption
  - XTS (38E): only for storage devices
    (but 38A modes "continue to be approved for such devices")
  - Applications exist where ciphertext cannot be expanded...
- Building block for AEAD
  - Authentication-only mode: CMAC (38B), HMAC (FIPS 198-1)
  - Generic AEAD: e.g., CBC + HMAC
  - CCM (38C) and GCM (38D): AEAD based on CTR!

Always Use AEAD Modes?

- GCM (38C) and CCM (38D)
  - Based on **CTR**
- Nonce reuse:
  - Deduce plaintext from ciphertext difference!
- Short tag / no tag:
  - Control plaintext through ciphertext difference!

Alternatives?

- Nonce reuse: KW/KWP/TKW (38F) or FF1/FF3 (38G)
  - "Misuse resistant" AEAD
  - Most suitable for key wrapping (38F) or formatted data (38G)
  - Very slow for general use...
- No tag: XTS (38E)
  - ECB-like mode: independently encrypts every block...
  - Only for storage devices
  - CBC (38A): sometimes preferable?

New Encryption Primitive?

- "Conventional" Authenticated Encryption
  - Use GCM or CCM
  - Secure in commonly-used protocols such as TLS

New Encryption Primitive?

- "Conventional" Authenticated Encryption
  - Use GCM or CCM
  - Secure in commonly-used protocols such as TLS
- Other Encryption Applications?
  - Low-latency encryption?
  - Lightweight encryption?
  - SHA-3-based encryption?
  - Insecure uses of CBC, GCM, CCM?
  - Other?

New Encryption Primitive?

- "Conventional" Authenticated Encryption
  - Use GCM or CCM
  - Secure in commonly-used protocols such as TLS

- Other Encryption Applications?
  - ~~Low-latency encryption?~~
  - ~~Lightweight encryption?~~
  - ~~SHA-3-based encryption?~~
  - Insecure uses of CBC, GCM, CCM? ← Last Workshop!
  - ~~Other?~~

New Encryption Primitive?

- "Conventional" Authenticated Encryption
  - Use GCM or CCM
  - Secure in commonly-used protocols such as TLS
- Other Encryption Applications?
  - ~~Low-latency encryption?~~
  - ~~Lightweight encryption?~~
  - ~~SHA-3-based encryption?~~
  - Insecure uses of CBC, GCM, CCM? ← Last Workshop!
  - ~~Other?~~
- Why?
  - IV reuse, short or no tags, no key commitment, release of unverified plaintext,...
  - Many applications: disk encryption, packet encryption, message franking,...

Where We Are Now

- Third Block Cipher Modes Workshop (BCM3)
  - "NIST is particularly interested in discussing the possibility of standardizing a tweakable wide block encryption technique that could support a large range of input lengths"
  - "Discuss how NIST can best address the limitations of the [...] NIST Special Publication 800-38 series" [MD23]

Where We Are Now

- Third Block Cipher Modes Workshop (BCM3)
  - "NIST is particularly interested in discussing the possibility of standardizing a tweakable wide block encryption technique that could support a large range of input lengths"
  - "Discuss how NIST can best address the limitations of the [...] NIST Special Publication 800-38 series" [MD23]

- This Workshop
  - Mode with provable security based on underlying block cipher

Where We Are Now

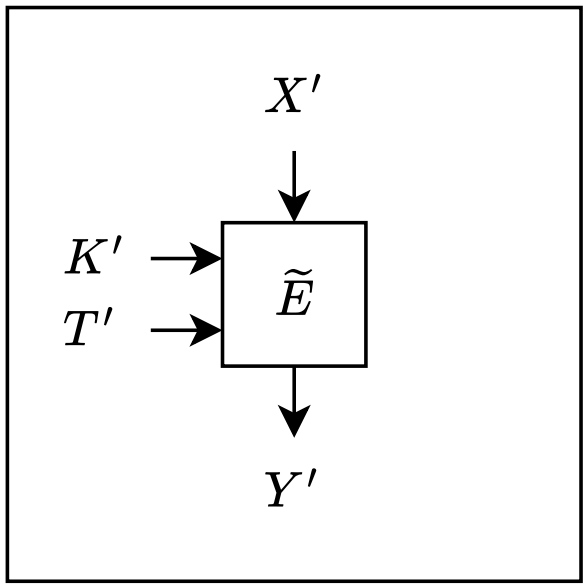
- Third Block Cipher Modes Workshop (BCM3)
  - "NIST is particularly interested in discussing the possibility of standardizing a tweakable wide block encryption technique that could support a large range of input lengths"
  - "Discuss how NIST can best address the limitations of the [...] NIST Special Publication 800-38 series" [MD23]
- This Workshop
  - Mode with provable security based on underlying block cipher
- Next
  - Requirements for mode of operation?
  - Block cipher: AES or Rijndael? Related keys? Tweak input?

Hierarchy

AEAD, Key Wrapping, Disk Encryption, etc. (derived functions) ← only encoding here

Accordion Cipher (mode of operation) ← all crypto here

(Tweakable) Block Cipher ← AES/Rijndael here



NECST Framework

- New features / future-proofing
- Efficiency / performance
- Compatibility (can it replace existing standards?)
- Security
- Technical reason (something doesn't work / can't be done)

Rogaway at BCM3

- Rogaway's Observation on SP 800-38 Series (BCM3)
  - SP 800-38 Series: 17 AES-based modes
  - Starting from scratch: maybe we would just have one?
  - "A mode that is like a wide block cipher"

Rogaway at BCM3

- Rogaway's Observation on SP 800-38 Series (BCM3)
  - SP 800-38 Series: 17 AES-based modes
  - Starting from scratch: maybe we would just have one?
  - "A mode that is like a wide block cipher"
- Requirements
  - (C) Can replace SP 800-38 Series modes
  - (T) Except 38G (FF1, FF3) (e.g., radix causes headaches)

Accordion Cipher

- Accordion Cipher: $Y = E_{K,T}(X)$
  - Key $K \in \{0,1\}^k$
  - Tweak $T \in \{0,1\}^*$
  - Input X, Output $Y \in \{0,1\}^{m+\ell g}$, $\ell \in \{0,1,2,\ldots\}$
  - Minimum input size m
  - Granularity g
  - Invertible

- CTR, OFB

  ($\ell \geq 0$)

- CFB

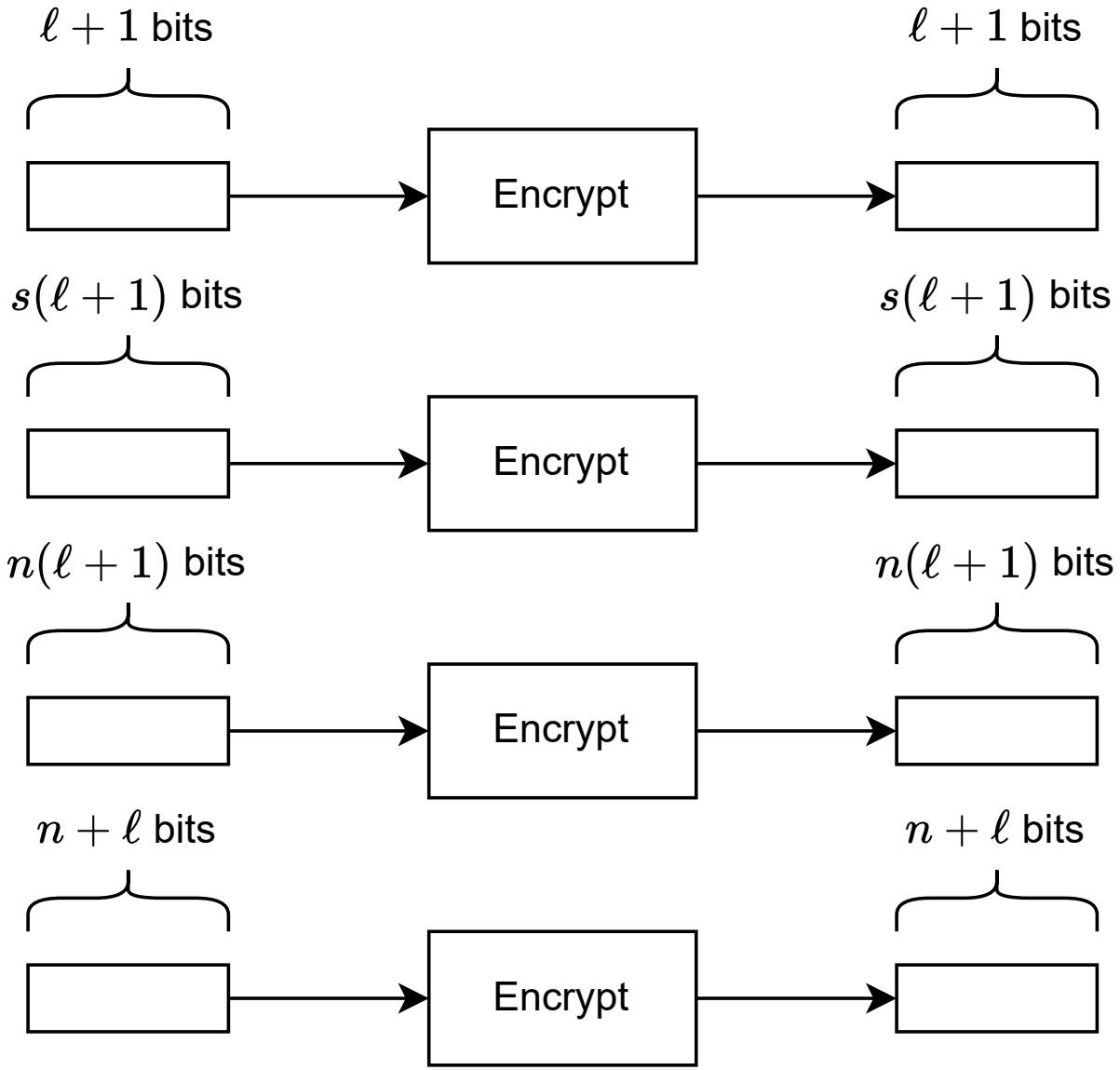
  (s = size of segment)

- ECB, CBC

  (n = block size)

- CBC-CS

  (CS1, CS2, CS3)



Minimum Input Size For 38A Compatibility

- Minimum Input Size m
  - (C) $m = 1$ bit (CTR, OFB, CFB with $s = 1$)

Minimum Input Size For 38A Compatibility

- Minimum Input Size m
  - (C) $m = 1$ bit (CTR, OFB, CFB with $s = 1$)

or

  - (C) $m = n$ bits (ECB, CBC, CBC-CS)
  - (S) Some minimum domain size against guessing attacks (38G)
  - (E,T) No FFX-style construction needed (38G)

Minimum Input Size For 38A Compatibility

- Minimum Input Size m
  - (C) $m = 1$ bit (CTR, OFB, CFB with $s = 1$)

or

  - (C) $m = n$ bits (ECB, CBC, CBC-CS)
  - (S) Some minimum domain size against guessing attacks (38G)
  - (E,T) No FFX-style construction needed (38G)

or

  - (T) $m = 2n$ bits may simplify designs (e.g., AEZ v5 [HKR17])

Granularity For 38A Compatibility

- Granularity g
  - (C) $g = 1$ bit: CTR, OFB, CFB with $s = 1$, CBC-CS

Granularity For 38A Compatibility

- Granularity g
  - (C) $g = 1$ bit: CTR, OFB, CFB with $s = 1$, CBC-CS

or

  - (T) $g = 8$ bits: CAESAR, NIST LWC call: only byte strings

Granularity For 38A Compatibility

- Granularity g
  - (C) $g = 1$ bit: CTR, OFB, CFB with $s = 1$, CBC-CS

or

  - (T) $g = 8$ bits: CAESAR, NIST LWC call: only byte strings

or

  - (S) large g helps to hide plaintext length [GJLN22]
    (but more secure: length-hiding padding)
  - (C,N) $g = 8$ bits sometimes needed: e.g., 509-byte for Tor

Maximum Input Size For 38-Series Compatibility

- Maximum Input Size?
  - (C) No limit specified in 38A for CBC, OFB, CFB
    (CTR limit can be small in practice, e.g.: 64 KiB [RW03])
  - (S) Stay well below $2^{n/2}$ blocks [BL16]
  - (N) BCM3: consider $n = 256$ (512?) to avoid limits in practice
    (Note: NIST LWC call required at least $2^{50} - 1$ bytes)

Maximum Input Size For 38-Series Compatibility

- Maximum Input Size?
  - (C) No limit specified in 38A for CBC, OFB, CFB
    (CTR limit can be small in practice, e.g.: 64 KiB [RW03])
  - (S) Stay well below $2^{n/2}$ blocks [BL16]
  - (N) BCM3: consider $n = 256$ (512?) to avoid limits in practice
    (Note: NIST LWC call required at least $2^{50} - 1$ bytes)
- (E) Note: Increasing n is much more efficient than mode with beyond-birthday-bound security

Key Size for 38-Series Compatibility

- Key Size k
  - (C,S) 128, 192, 256 bits
  - (E) Fastest: 128 bits
  - (N) 512 bits? (seems not necessary: [Mou21])

Key Size for 38-Series Compatibility

- Key Size k
  - (C,S) 128, 192, 256 bits
  - (E) Fastest: 128 bits
  - (N) 512 bits? (seems not necessary: [Mou21])
- Bit Strength
  - NIST LWC call: 128 bit length, 112 bit strength
  - (E) 256 bit length, 128 bit strength?

Tweak Size for 38-Series Compatibility

- Tweak Size
  - (C) empty for Key Wrapping (38F),
    $1 \ldots 2^{64} - 1$ bits for GCM IV,
    $0 \ldots$ *maxTlen* bytes for FF1 (no restrictions on *maxTlen*)
  - (S) Stay well below $2^{n/2}$ blocks
  - (T) Only byte strings
  - (N) Avoid limits in practice

(N) Extra Requirements?

- Nonce-Hiding, Length-Hiding, Short-Tag, RUP-Secure MRAE?
  - Handle at higher level

(N) Extra Requirements?

- Nonce-Hiding, Length-Hiding, Short-Tag, RUP-Secure MRAE?
  - Handle at higher level
- AEAD with Key/Context Commitment?
  - BCM3: Important in practice

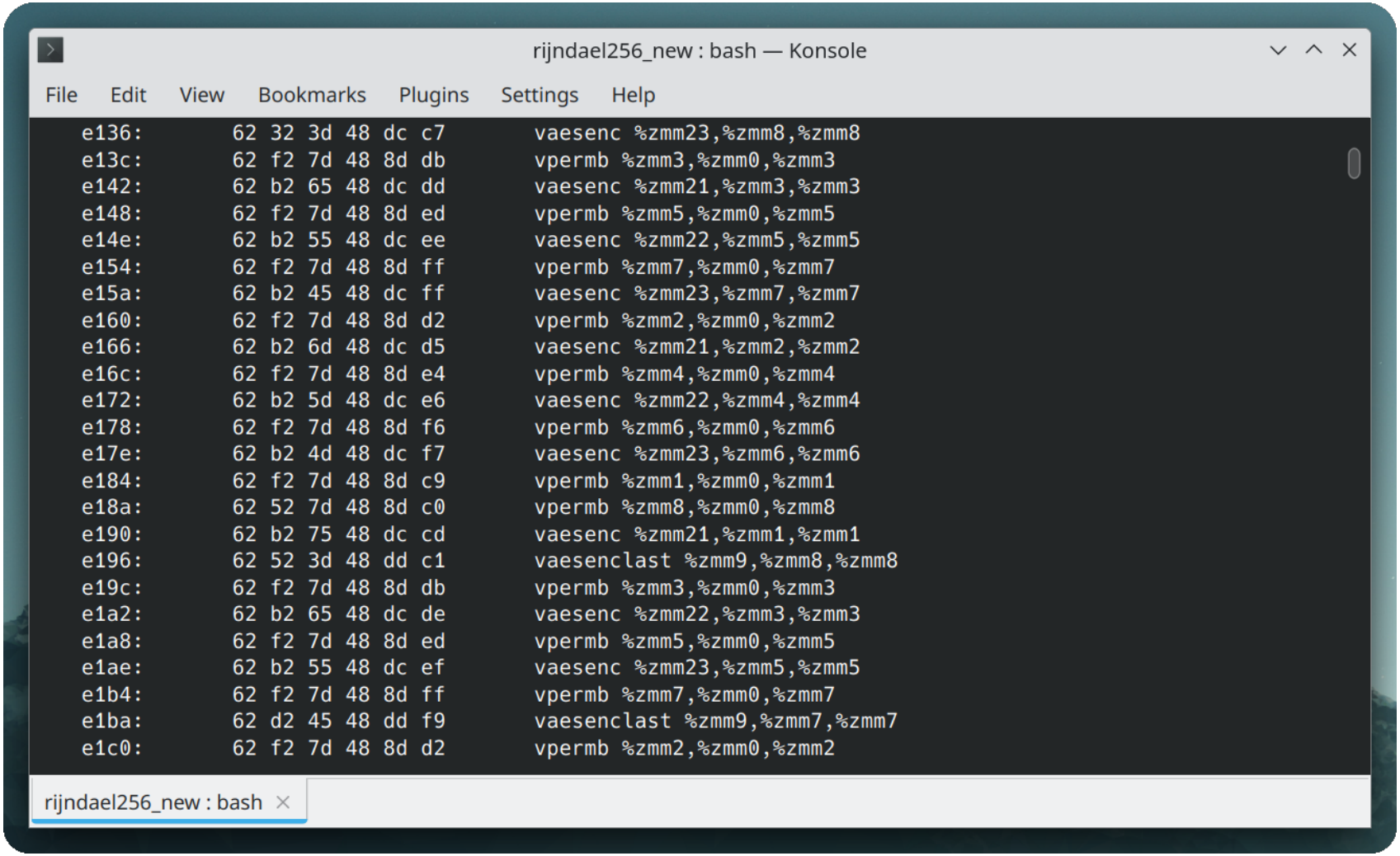
(N) Extra Requirements?

- Nonce-Hiding, Length-Hiding, Short-Tag, RUP-Secure MRAE?
  - Handle at higher level
- AEAD with Key/Context Commitment?
  - BCM3: Important in practice
- Key-Dependent Message Security?
  - Was requirement for XTS-AES (38E) [MD23]

(N) Extra Requirements?

- Nonce-Hiding, Length-Hiding, Short-Tag, RUP-Secure MRAE?
  - Handle at higher level
- AEAD with Key/Context Commitment?
  - BCM3: Important in practice
- Key-Dependent Message Security?
  - Was requirement for XTS-AES (38E) [MD23]
- Inverse-Free?
  - CMAC, GCM, CCM, FF1, etc. do not use inverse block cipher

(E) Block Cipher Requirements



(E) Block Cipher Requirements

- AES-256: one AES round = one clock cycle throughput since Sandy Bridge (2011)
  - $\frac{14 \text{ rounds} \times 1 \text{ cycle/round}}{16 \text{ bytes/round}} \approx 0.88$ cycles/byte

(E) Block Cipher Requirements

- AES-256: one AES round = one clock cycle throughput since Sandy Bridge (2011)
  - $\frac{14 \text{ rounds} \times 1 \text{ cycle/round}}{16 \text{ bytes/round}} \approx 0.88$ cycles/byte
- AES-256: throughput $4\times$ speedup since Ice Lake (2018)
  - $\frac{14 \text{ rounds} \times 1 \text{ cycle/round}}{4 \times 16 \text{ bytes/round}} \approx 0.22$ cycles/byte

(E) Block Cipher Requirements

- AES-256: one AES round = one clock cycle throughput since Sandy Bridge (2011)
  - $\frac{14 \text{ rounds} \times 1 \text{ cycle/round}}{16 \text{ bytes/round}} \approx 0.88$ cycles/byte
- AES-256: throughput $4\times$ speedup since Ice Lake (2018)
  - $\frac{14 \text{ rounds} \times 1 \text{ cycle/round}}{4 \times 16 \text{ bytes/round}} \approx 0.22$ cycles/byte
- Rijndael ($n = 256$, also $n = 512$) on same platform?
  - Same as AES would be: 0.22 cycles/byte
  - Rijndael benchmarks: 0.25 cycles/byte
  - Benchmark speed independent of key schedule!
    (S) fix related-key attacks and (N) possibly add tweak input?
    (E) ... but many keys/tweaks will have impact on performance

(E) Block Cipher Requirements

- AES-256: one AES round = one clock cycle throughput since Sandy Bridge (2011)
  - $\frac{14 \text{ rounds} \times 1 \text{ cycle/round}}{16 \text{ bytes/round}} \approx 0.88$ cycles/byte
- AES-256: throughput 4× speedup since Ice Lake (2018)
  - $\frac{14 \text{ rounds} \times 1 \text{ cycle/round}}{4 \times 16 \text{ bytes/round}} \approx 0.22$ cycles/byte
- Rijndael ($n = 256$, also $n = 512$) on same platform?
  - Same as AES would be: 0.22 cycles/byte
  - Rijndael benchmarks: 0.25 cycles/byte
  - Benchmark speed independent of key schedule!
    (S) fix related-key attacks and (N) possibly add tweak input?
    (E) ... but many keys/tweaks will have impact on performance
- (E) Note: New mode needs to be parallel (so that bottleneck is throughput, not latency)

Conclusion and Next Steps

- Separate Block Cipher and Mode
  - AES (Rijndael? Related keys? Tweak input?)
  - Provably secure mode

Conclusion and Next Steps

- Separate Block Cipher and Mode
  - AES (Rijndael? Related keys? Tweak input?)
  - Provably secure mode
- Properties?
  - Try to justify (NECST framework)

Conclusion and Next Steps

- Separate Block Cipher and Mode
  - AES (Rijndael? Related keys? Tweak input?)
  - Provably secure mode
- Properties?
  - Try to justify (NECST framework)
- Questions?