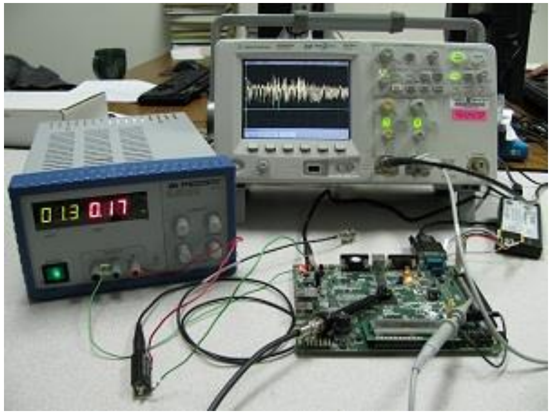
Analysing the Leakage-Resistance of some Round-2 Candidates of the NIST's Lightweight Crypto Standardization Process



François-Xavier Standaert

UCLouvain, ICTEAM, Crypto Group (Belgium)

NIST Lightweight Crypto Workshop 2019, Gaithersburg, USA

Side-channel countermeasures

	abstraction level
1999: masking countermeasure [CJRR99]	implementation
2004: hardware countermeasures [M04]	hardware
2006: shuffling countermeasure [HOS06]	implementation
2008: leakage-resilient stream cipher [DP08]	mode
2012: masking-optimized ciphers [PRC12]	primitive
2015: leakage-resilient enc. & auth. [PSV15]	mode
2017-19: leakage-res. AE [BMOS17,GPPS19]	mode

Side-channel countermeasures 1

	security target
1999: masking countermeasure [CJRR99]	key recovery
2004: hardware countermeasures [M04]	key recovery
2006: shuffling countermeasure [HOS06]	key recovery
2008: leakage-resilient stream cipher [DP08]	pseudorand.
2012: masking-optimized ciphers [PRC12]	key recovery
2015: leakage-resilient enc. & auth. [PSV15]	integ. & conf.
2017-19: leakage-res. AE [BMOS17,GPPS19]	integ. + conf.

Side-channel countermeasures

	security target
1999: masking countermeasure [CJRR99]	key recovery
2004: hardware countermeasures [M04]	key recovery
2006: shuffling countermeasure [HOS06]	key recovery
2008: leakage-resilient stream cipher [DP08]	pseudorand.
2012: masking-optimized ciphers [PRC12]	key recovery
2015: leakage-resilient enc. & auth. [PSV15]	integ. & conf.
2017-19: leakage-res. AE [BMOS17,GPPS19]	integ. + conf.

This talk: difference in primitives (*a bit*) & modes (*mostly*)

Outline

1. How to reason about (AE) leakage?
   - Specify the security target
   - Analyse the mode (& choose assumptions)
   - Evaluate the implementation (& primitive) ≈ cost needed to fulfil the assumptions

2. Case studies: NIST candidates & more
   - Level 0: no mode-level leakage-resistance
   - Level 1: re-keyed modes (including sponges)
   - Level 2: level 1 + strengthened init./final.
   - Level 3: level 2 + two-passes

Outline

1. How to reason about (AE) leakage?

- Specify the security target
- Analyse the mode (& choose assumptions)
- Evaluate the implementation (& primitive)
  ≈ cost needed to fulfil the assumptions

2. Case studies: NIST candidates & more

- Level 0: no mode-level leakage-resistance
- Level 1: re-keyed modes (including sponges)
- Level 2: level 1 + strengthened init./final.
- Level 3: level 2 + two-passes

Security targets

- Confidentiality: CPA, CCA security
- Plaintext Integrity (PI), Ciphertext Integrity (CI)
  - Composite definitions useful: confidentiality & integrity often call for $\neq$ physical assumptions

Security targets

- Confidentiality: CPA, CCA security
- Plaintext Integrity (PI), Ciphertext Integrity (CI)
  - Composite definitions useful: confidentiality & integrity often call for $\neq$ physical assumptions
- Leakage in encryption only (1) or enc./dec. (2)

Security targets

- Confidentiality: CPA, CCA security
- Plaintext Integrity (PI), Ciphertext Integrity (CI)
  - Composite definitions useful: confidentiality & integrity often call for $\neq$ physical assumptions
- Leakage in encryption only (1) or enc./dec. (2)
- Nonce misuse-resistance (M) or resilience (m)

Security targets

- Confidentiality: CPA, CCA security
- Plaintext Integrity (PI), Ciphertext Integrity (CI)
  - Composite definitions useful: confidentiality & integrity often call for $\neq$ physical assumptions
- Leakage in encryption only (1) or enc./dec. (2)
- Nonce misuse-resistance (M) or resilience (m)
- Leakage-resistance (L) or resilience (l)

Security targets

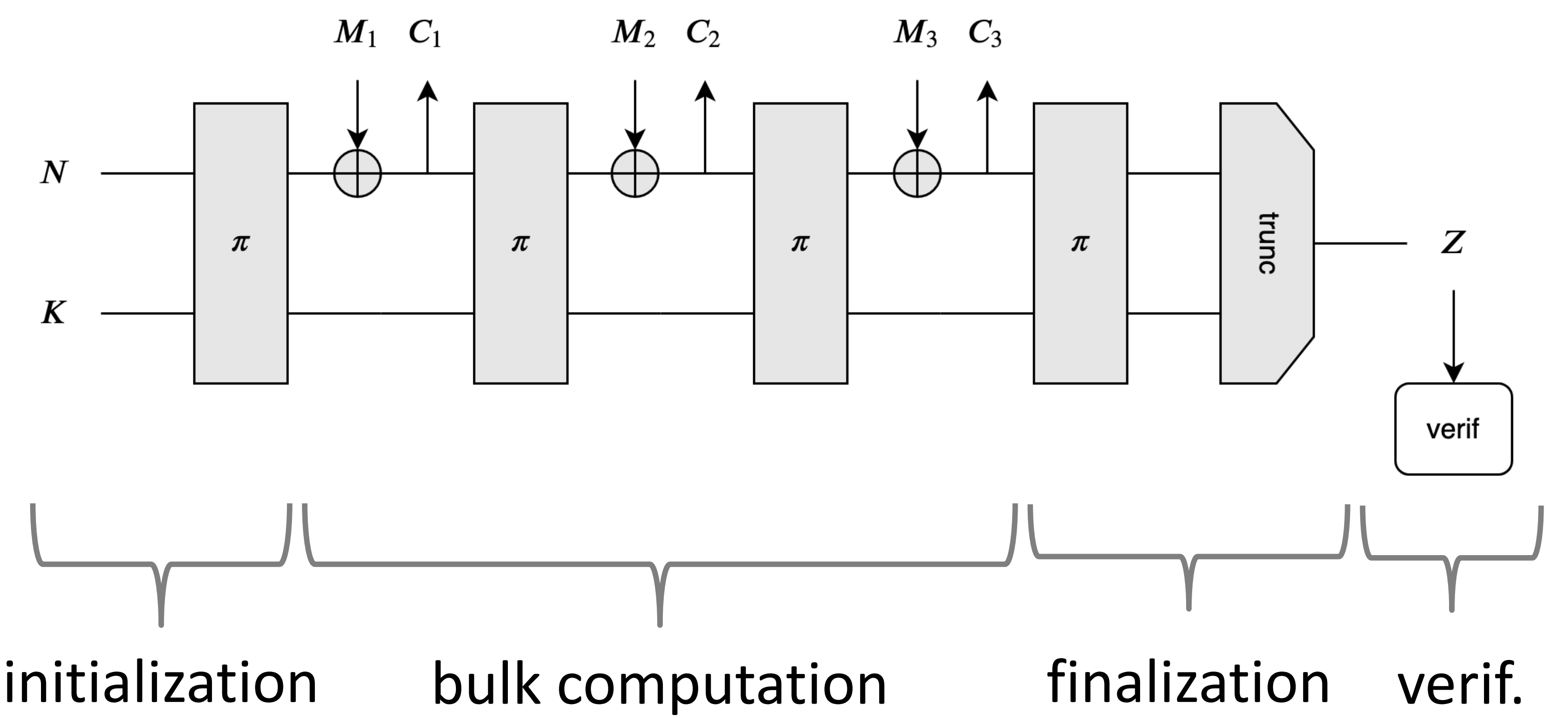
- Confidentiality: CPA, CCA security
- Plaintext Integrity (PI), Ciphertext Integrity (CI)
  - Composite definitions useful: confidentiality & integrity often call for $\neq$ physical assumptions
- Leakage in encryption only (1) or enc./dec. (2)
- Nonce misuse-resistance (M) or resilience (m)
- Leakage-resistance (L) or resilience (l)
- Single/multi-user (beyond birthday?) security

Security targets

- Confidentiality: CPA, CCA security
- Plaintext Integrity (PI), Ciphertext Integrity (CI)
  - Composite definitions useful: confidentiality & integrity often call for $\neq$ physical assumptions
- Leakage in encryption only (1) or enc./dec. (2)
- Nonce misuse-resistance (M) or resilience (m)
- Leakage-resistance (L) or resilience (l)
- Single/multi-user (beyond birthday?) security
- Selection depends on applications (e.g., software updates / control in hostile environment $\Rightarrow$ CIML2)

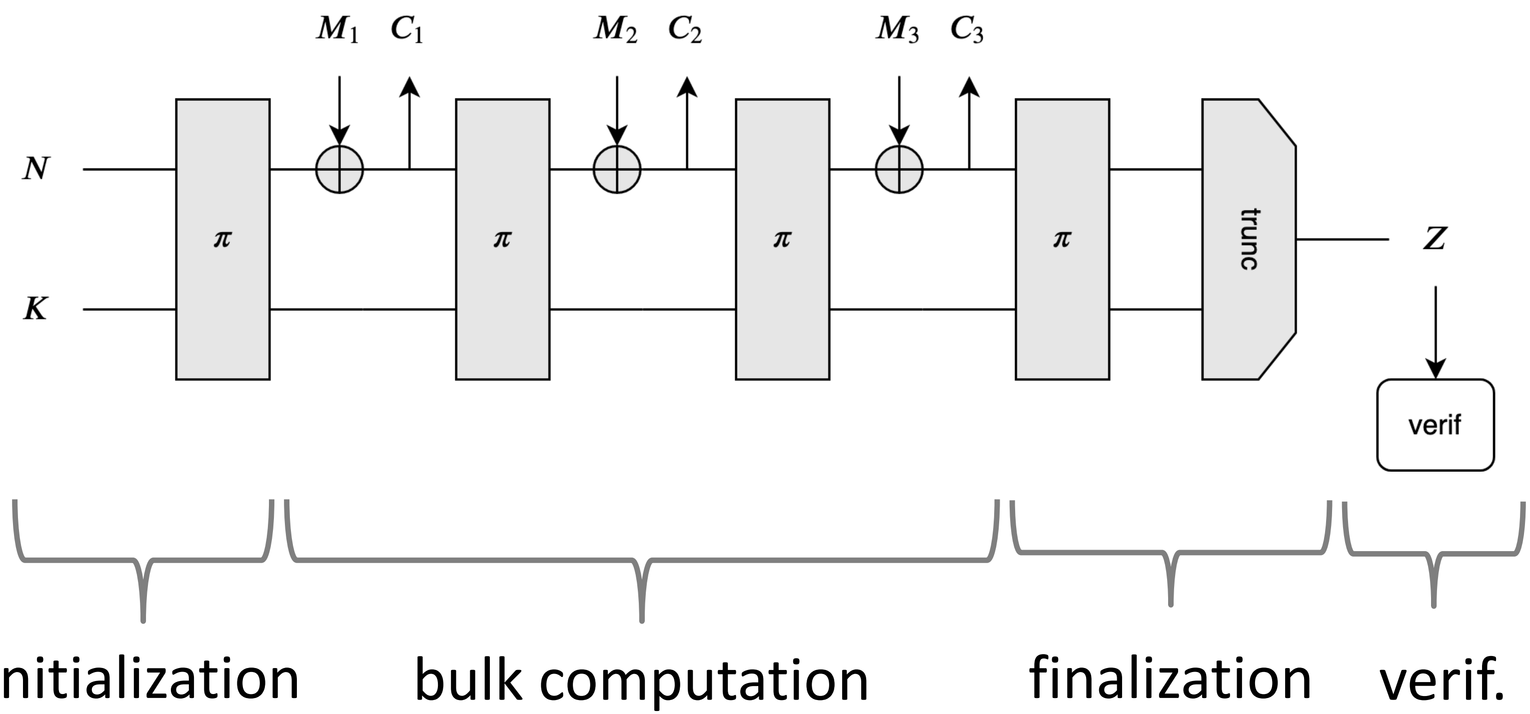
Mode analysis (I)

- Identify main steps, e.g., inner keyed sponge



Mode analysis (I)

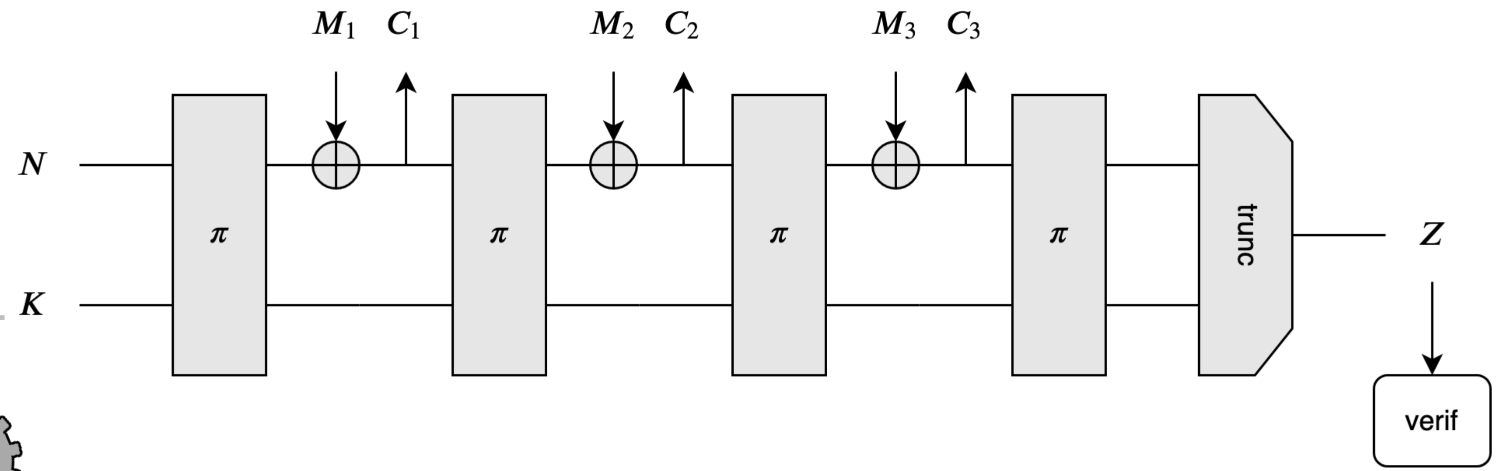
- Identify main steps, e.g., inner keyed sponge

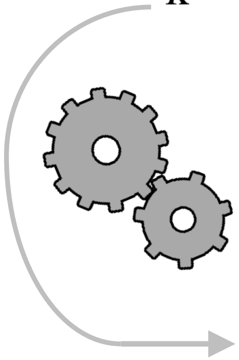


- (Some steps empty for some modes, ignoring AD)

Mode analysis (II)

- Reduce the mode to (weak) assumptions (tightly)





only computation leaks

leak-free components   bounded leakage

strong unpredictability with leakage

simulatable leakages   hard-to-invert leakages

oracle-free leakages   [...]

Practical evaluation (I)

- Translate assumptions into necessary design goals

	init./final.	bulk comp.	tag verif.
conf.	DPA (key recovery)	DPA (key recovery) SPA (key recovery) *1-block conf.*	∅
int.	DPA (key recovery)	DPA (key recovery) SPA (key recovery) unbounded leakages	DPA (tag recovery) unbounded leakages

- Set the target security level (2^m leakages, 2^t time)
- *Evaluate implementation cost & performances*

Practical evaluation (II)

- Approximate performance overheads

	init./final.	bulk comp.	tag verif.
conf.	x 5 – 10 – 100	x 5 – 10 – 100 x 1 – 5 *1-block conf.*	∅
int.	x 5 – 10 – 100	x 5 – 10 – 100 x 1 – 5 x 1	x 5 – 10 – 100 x 1

- DPA security: high-order masking, shuffling, …
- SPA security: parallel implementations, noise, …

Practical evaluation (II)

- Approximate performance overheads

	init./final.	bulk comp.	tag verif.
conf.	x 5 – 10 – 100	x 5 – 10 – 100 x 1 – 5 *1-block conf.*	∅
int.	x 5 – 10 – 100	x 5 – 10 – 100 x 1 – 5 x 1	x 5 – 10 – 100 x 1

- Beware of too simple evaluation strategies!
  - T-test negative with >100k traces, attack in <2000 traces

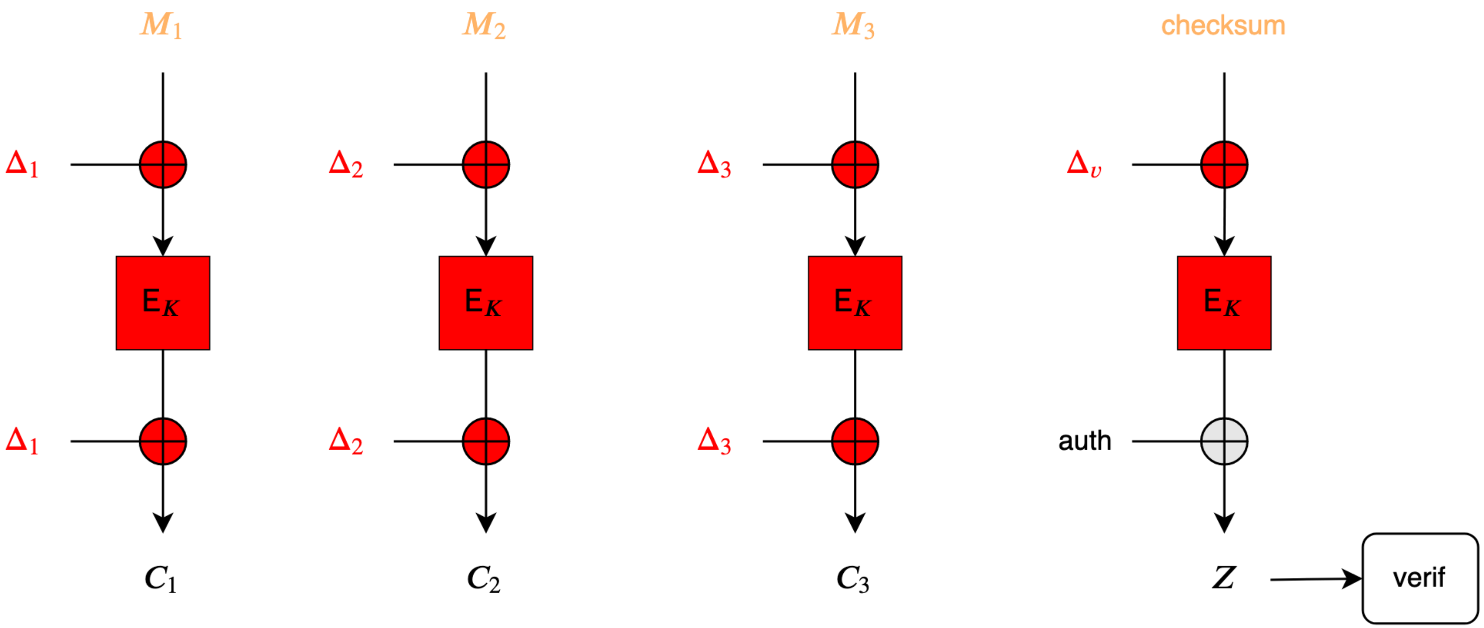
Outline

1. How to reason about (AE) leakage?
   - Specify the security target
   - Analyse the mode (& choose assumptions)
   - Evaluate the implementation (& primitive)
     $\approx$ cost needed to fulfil the assumptions

2. **Case studies: NIST candidates & more**
   - **Level 0: no mode-level leakage-resistance**
   - Level 1: re-keyed modes (including sponges)
   - Level 2: level 1 + strengthened init./final.
   - Level 3: level 2 + two-passes

OCB-Pyjamask

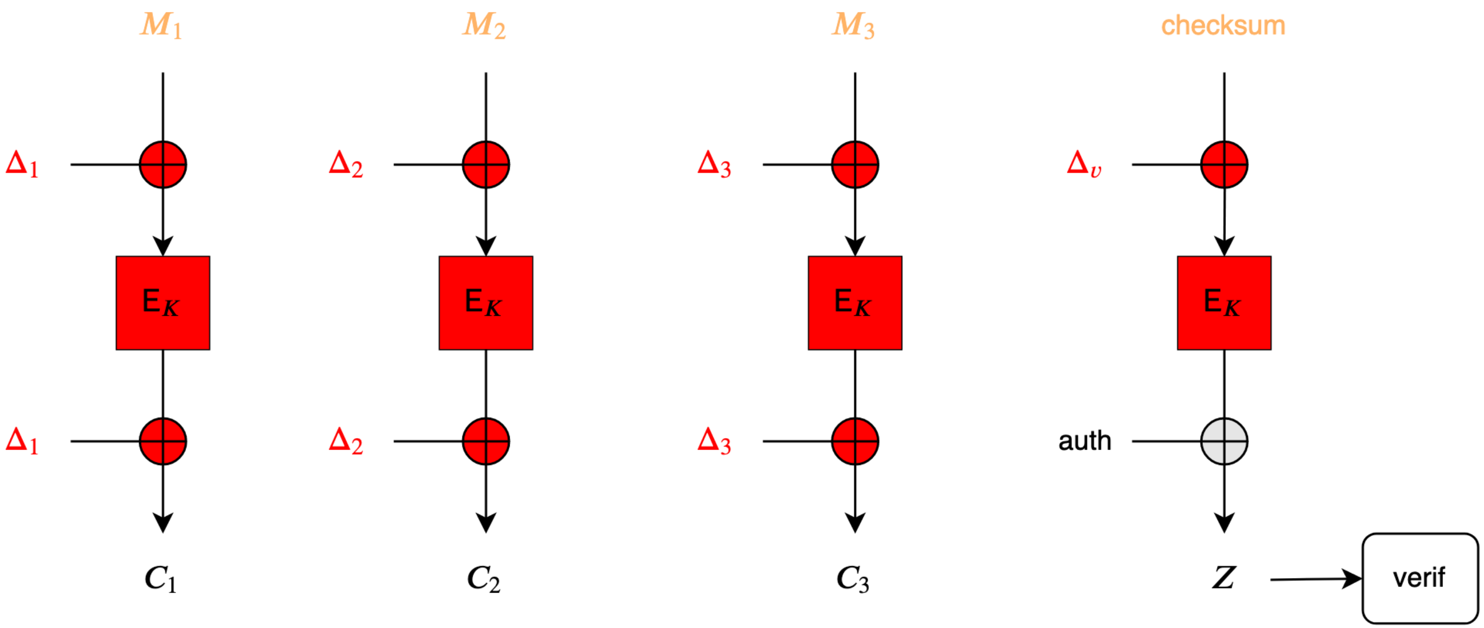
- Target: CCAL1, CIL1 (L in enc only, no misuse)



- Needs DPA resistance for all E_K blocks
  - Primitive/implementation SCA security only

OCB-Pyjamask

- Target: CCAL1, CIL1 (L in enc only, no misuse)



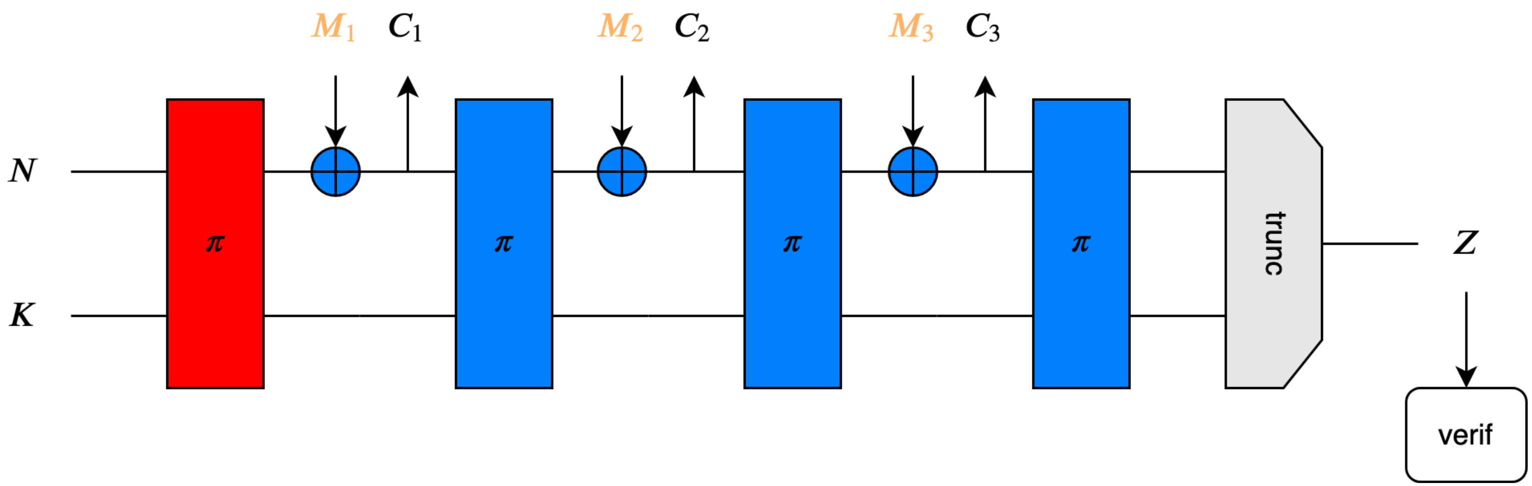
- Needs DPA resistance for all E_K blocks
  - Primitive/implementation SCA security only
- Others: SKINNY-AEAD, SUNDAE-GIFT, OCB-AES, …

Outline

1. How to reason about (AE) leakage?
   - Specify the security target
   - Analyse the mode (& choose assumptions)
   - Evaluate the implementation (& primitive)
     ≈ cost needed to fulfil the assumptions

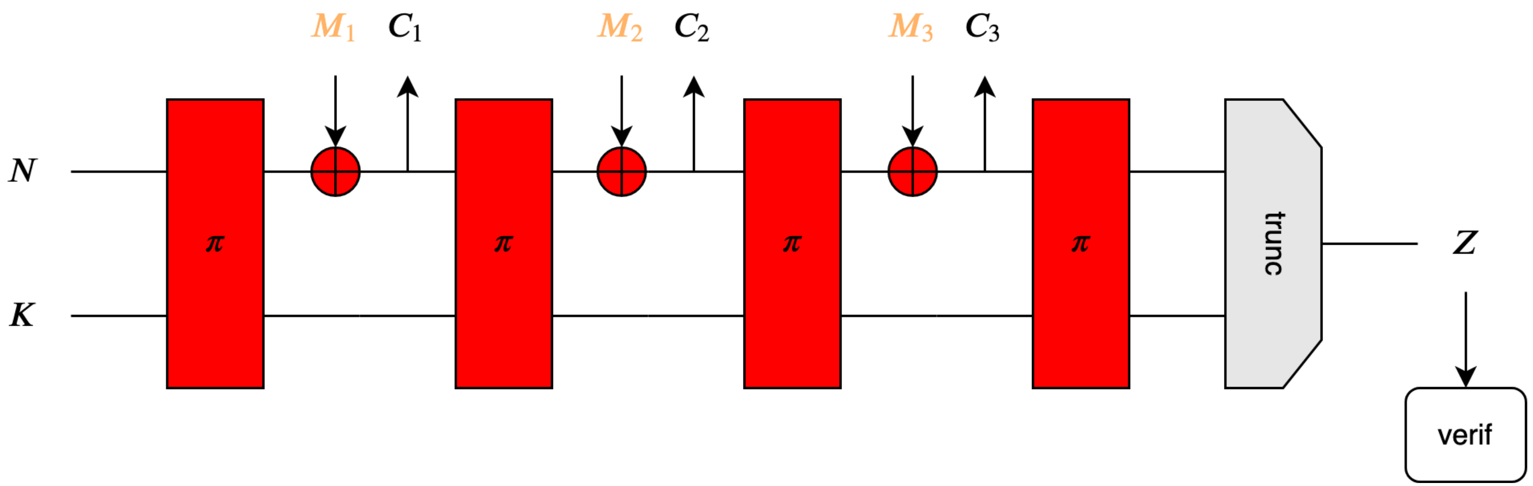
2. **Case studies: NIST candidates & more**
   - Level 0: no mode-level leakage-resistance
   - **Level 1: re-keyed modes (including sponges)**
   - Level 2: level 1 + strengthened init./final.
   - Level 3: level 2 + two-passes

- Target: CCAL1, CIL1 (L in enc only, no misuse)



- Bulk computation only requires SPA security
  - Light blue: no averaging is possible (fresh states)
- Calling for so-called leveled implementations
  - Energy gains thanks to 2 different implementations

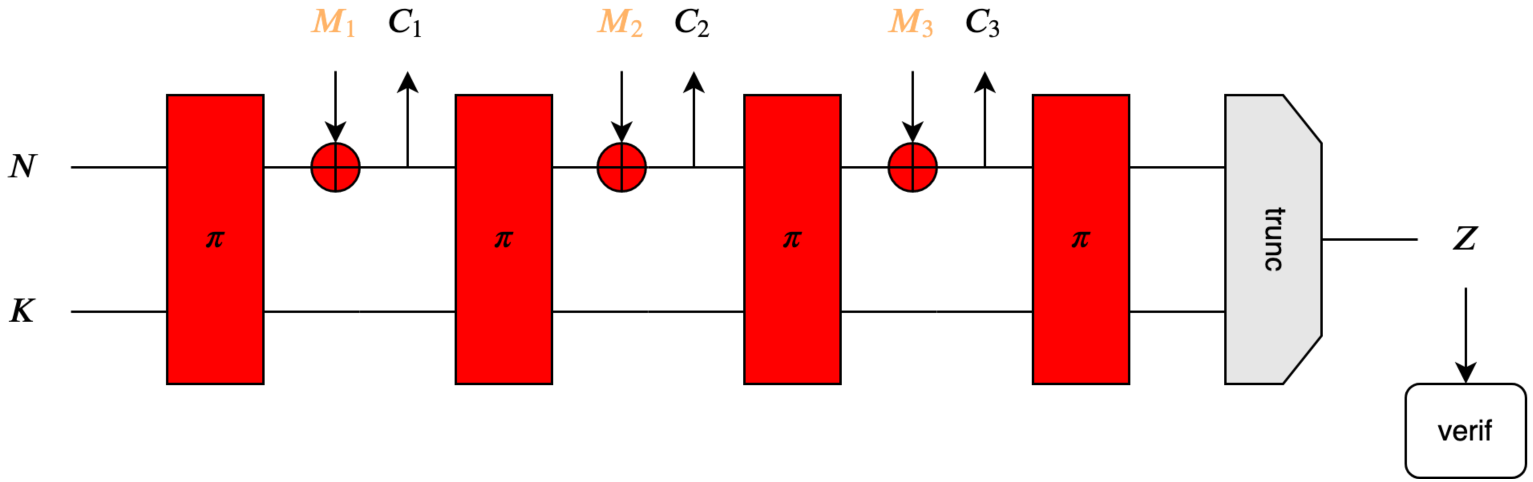
- Target: CCAmL1, CIML1 (L in enc only, misuse)



- DPA security needed everywhere with nonce misuse (idem with decryption leakages)

PHOTON-Beetle

- Target: CCAmL1, CIML1 (L in enc only, misuse)



- DPA security needed everywhere with nonce misuse (idem with decryption leakages)
- Others: Gimli, Ketje, Oribatida, …
  - (Roughly applies to all inner-keyed sponges)

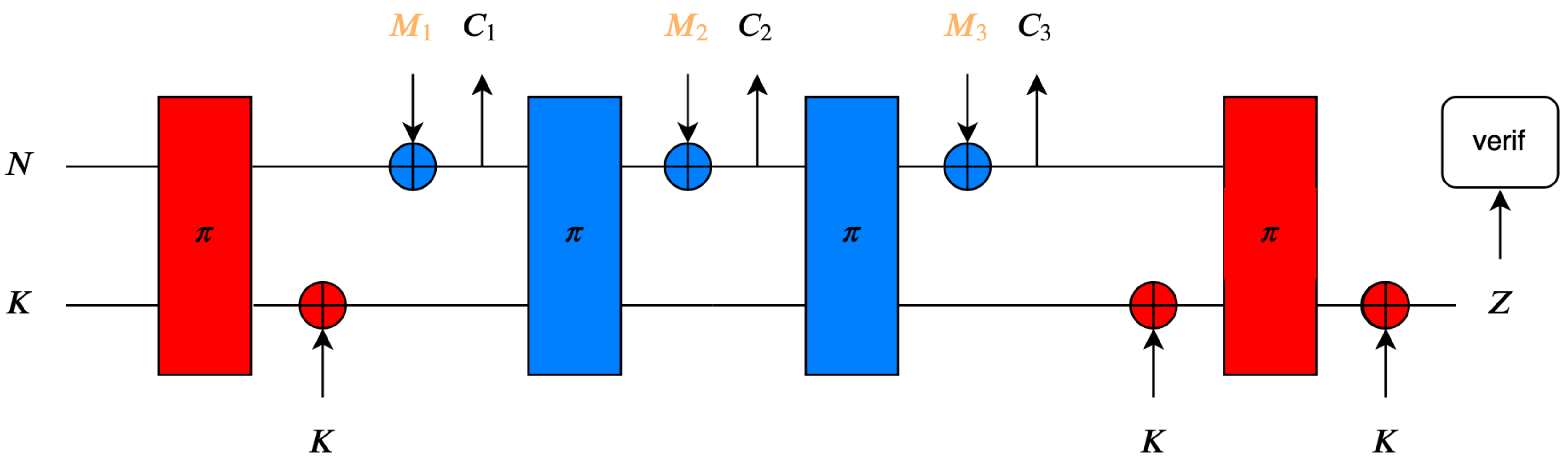
Outline

1. How to reason about (AE) leakage?
   - Specify the security target
   - Analyse the mode (& choose assumptions)
   - Evaluate the implementation (& primitive)

     ≈ cost needed to fulfil the assumptions

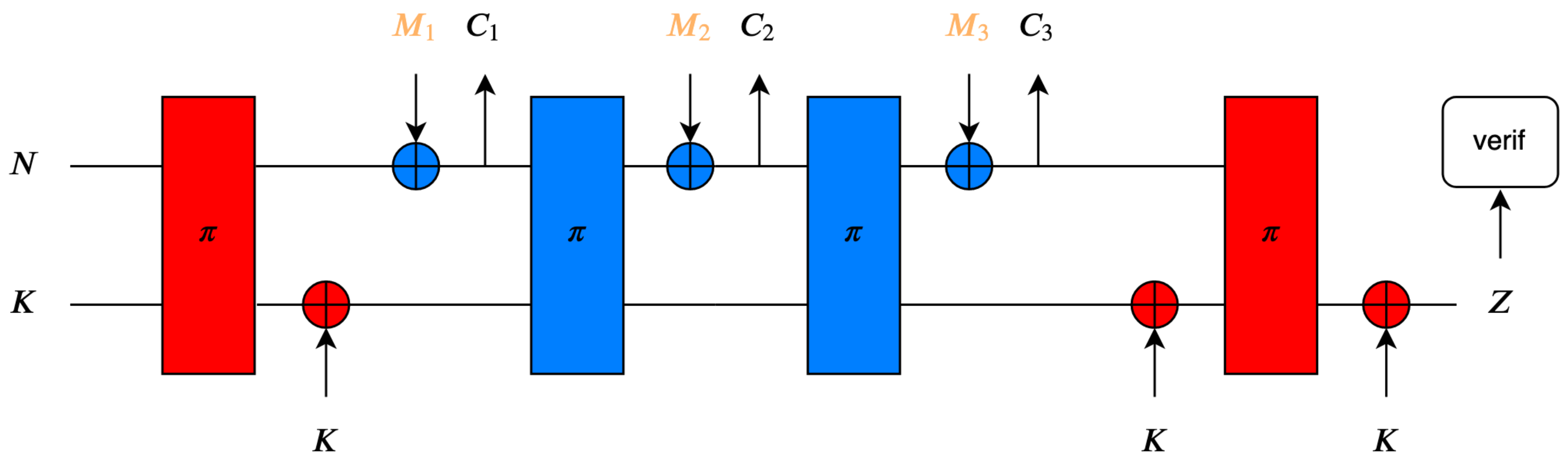
2. **Case studies: NIST candidates & more**
   - Level 0: no mode-level leakage-resistance
   - Level 1: re-keyed modes (including sponges)
   - **Level 2: level 1 + strengthened init./final.**
   - Level 3: level 2 + two-passes

- Target: CCAL1 (L in enc only, no misuse)



- Similar to inner-keyed sponges

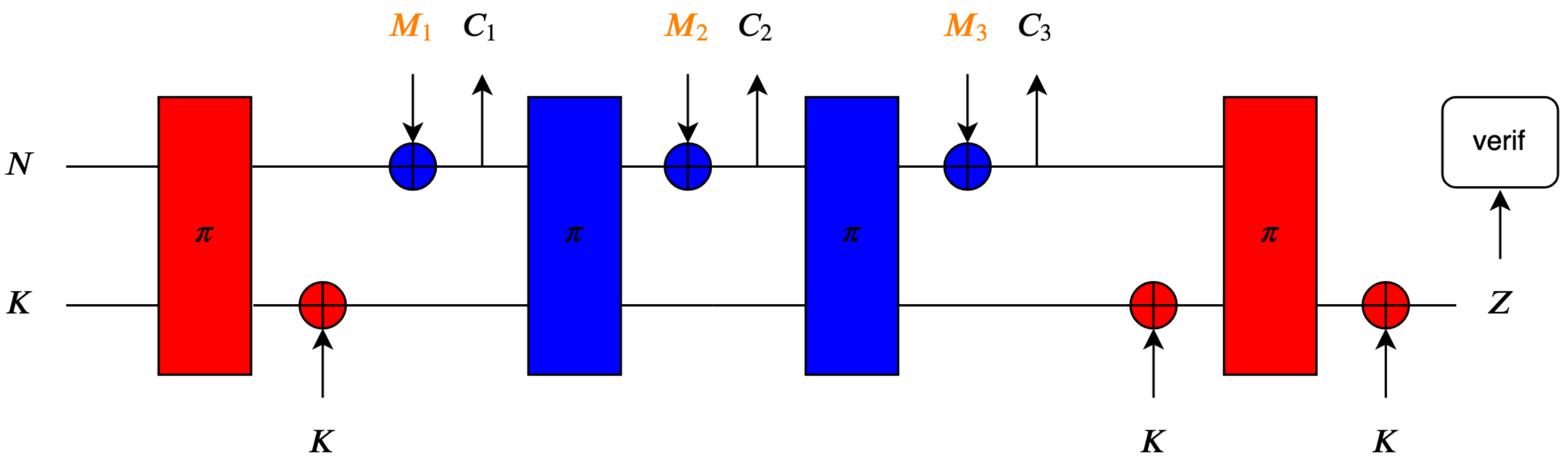
- Target: CCAmL1 (L in enc only, misuse-resilience)



- Strengthened init./final. steps maintain the SPA resistance requirement for the bulk computation with nonce misuse and encryption leakages

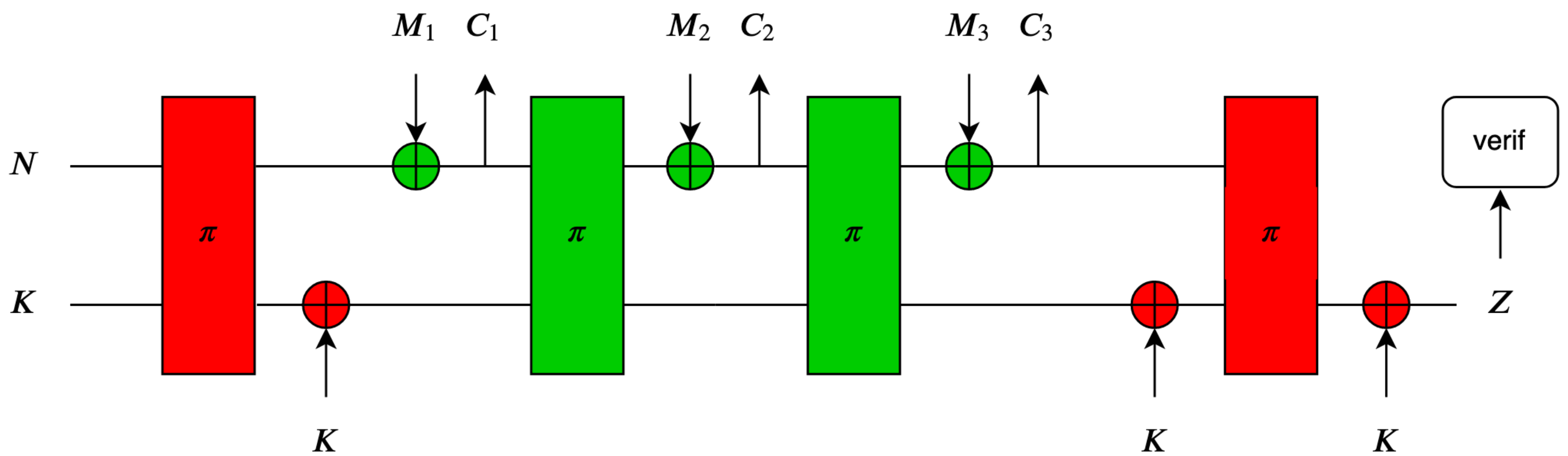
Ascon (confidentiality)

- Target: CCAmL2 (L in enc/dec, misuse-resilience)



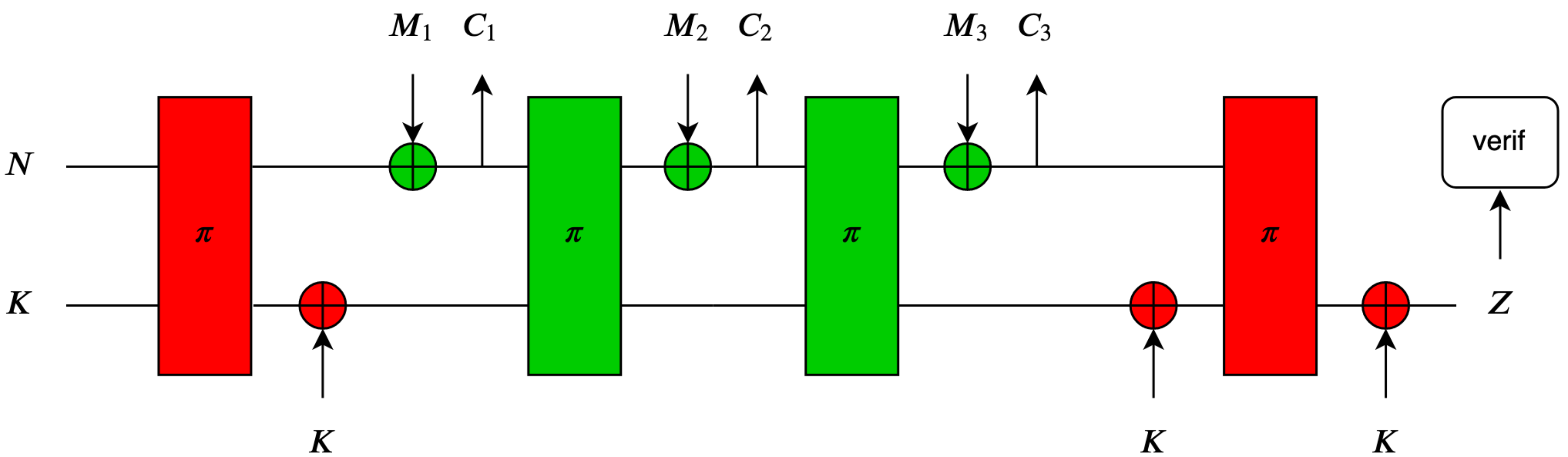
- Limited confidentiality with decryption leakages
- Dark orange/blue: message decrypted before verification ⇒ the same state can be repeatedly measured, allowing SPA with averaged leakage

- Target: CIL1 (L in enc only, no misuse)



- Bulk computation leakage can be unbounded
- Shows interest of composite definitions!

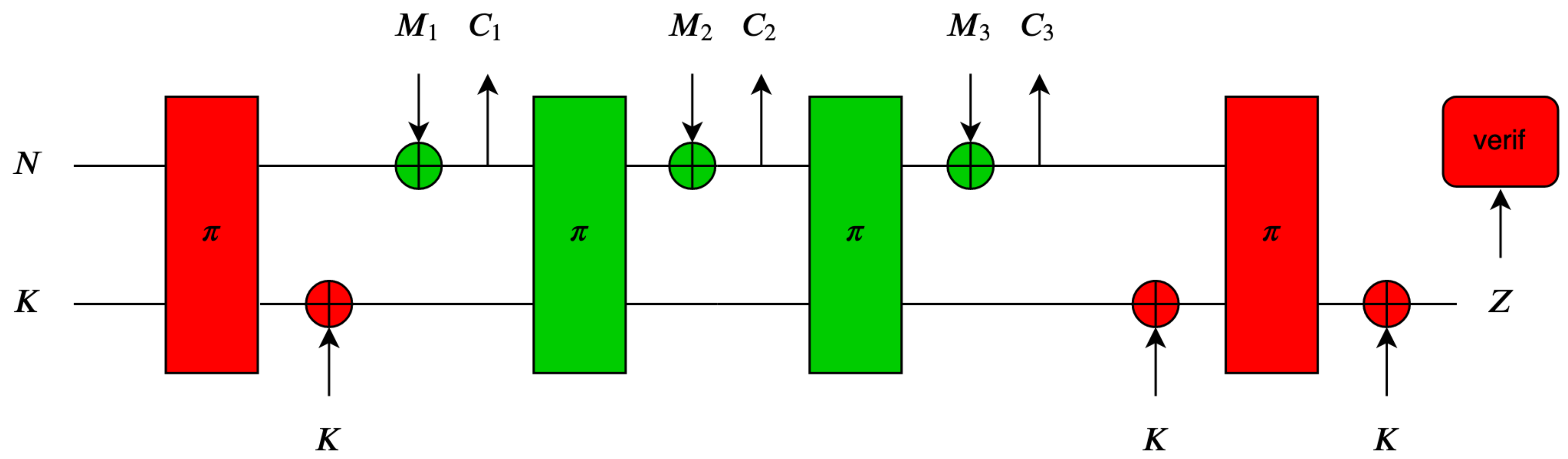
- Target: CIML1 (L in enc only, misuse-resistance)



- Same feature (unbounded leakages for the bulk)

Ascon (integrity)

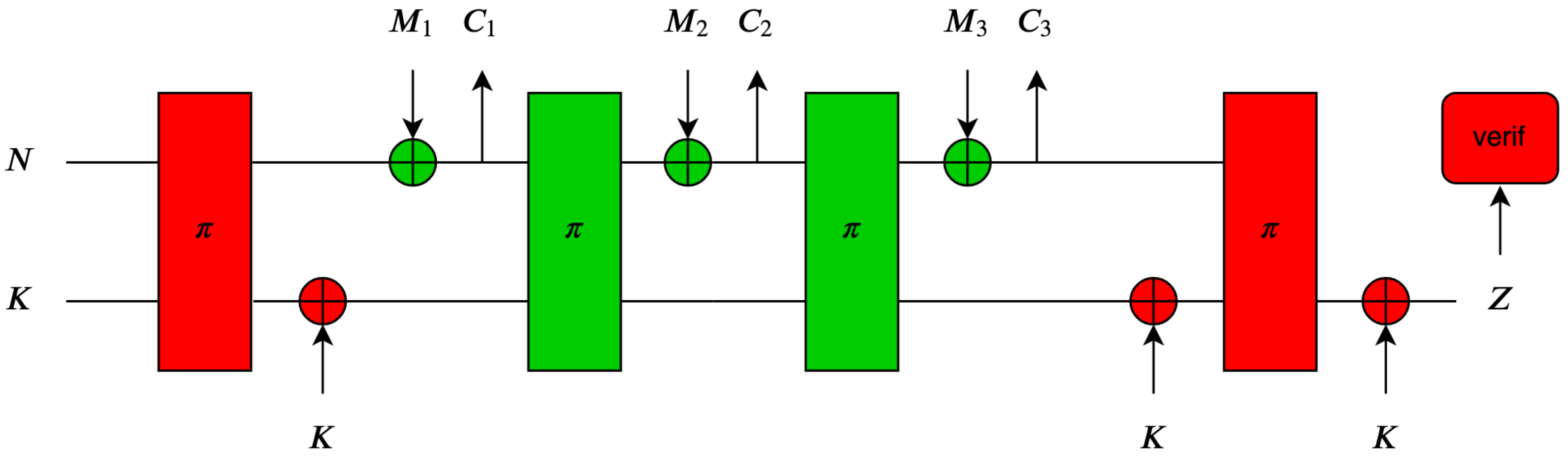
- Target: CIML2 (L in enc/dec, misuse-resistance)



- Tag verification must be protected against DPA
- Shows key recovery security is not enough!

Ascon (integrity)

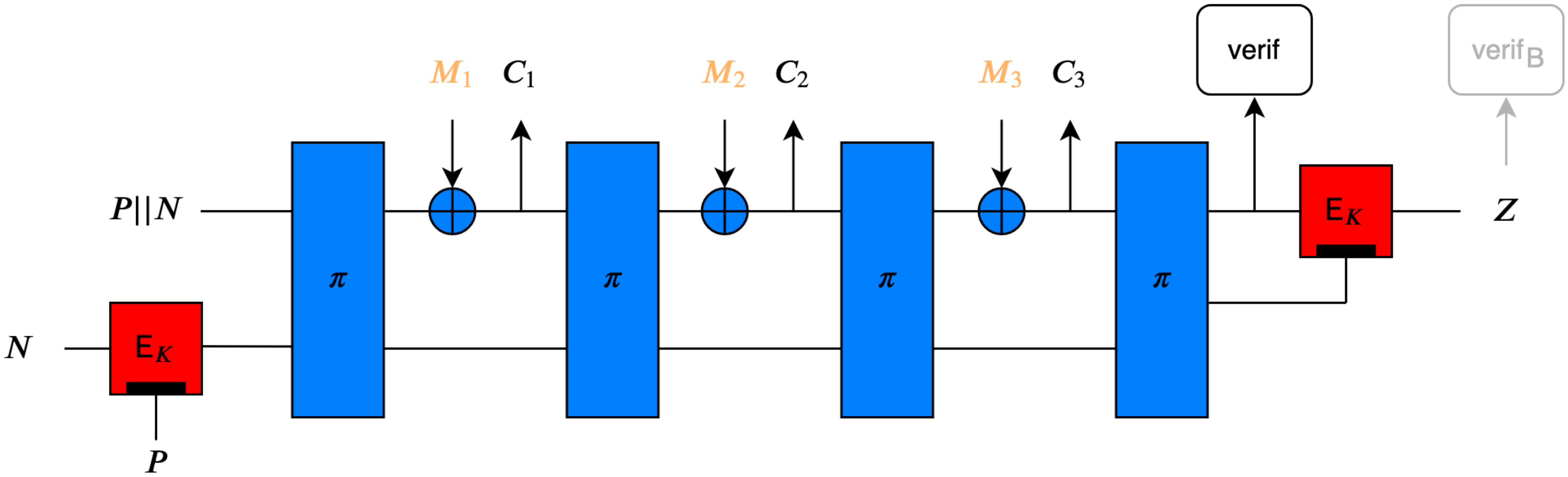
- Target: CIML2 (L in enc/dec, misuse-resistance)



- Tag verification must be protected against DPA
- Shows key recovery security is not enough!

- Others: ACE, GIBBON, Spix, WAGE, …

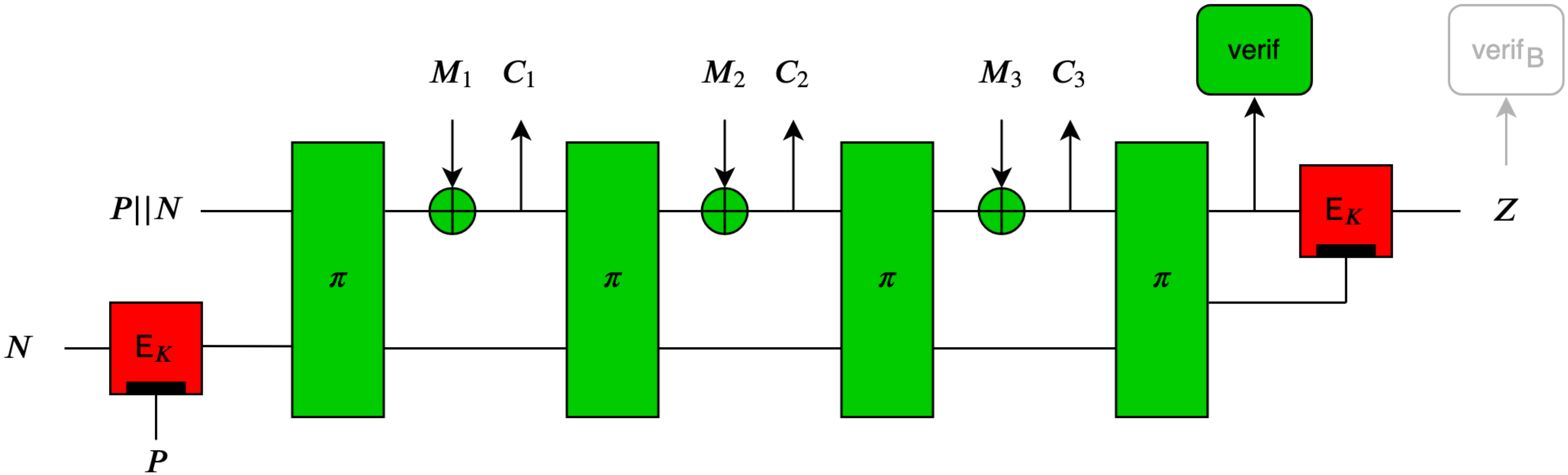
- CCAL1, CCAmL1, CCAmL2, CIL1, CIML1



≈ further exploiting the leveled implementation concept

- Similar to ASCON (but smaller masked state)

- CIML2 (L in enc/dec, misuse-resistance)



- Tag verification tolerates unbounded leakages
- (Inverse-free DPA resistant tag verif. also possible)
- Others: TBC-only variant (TET)

Outline

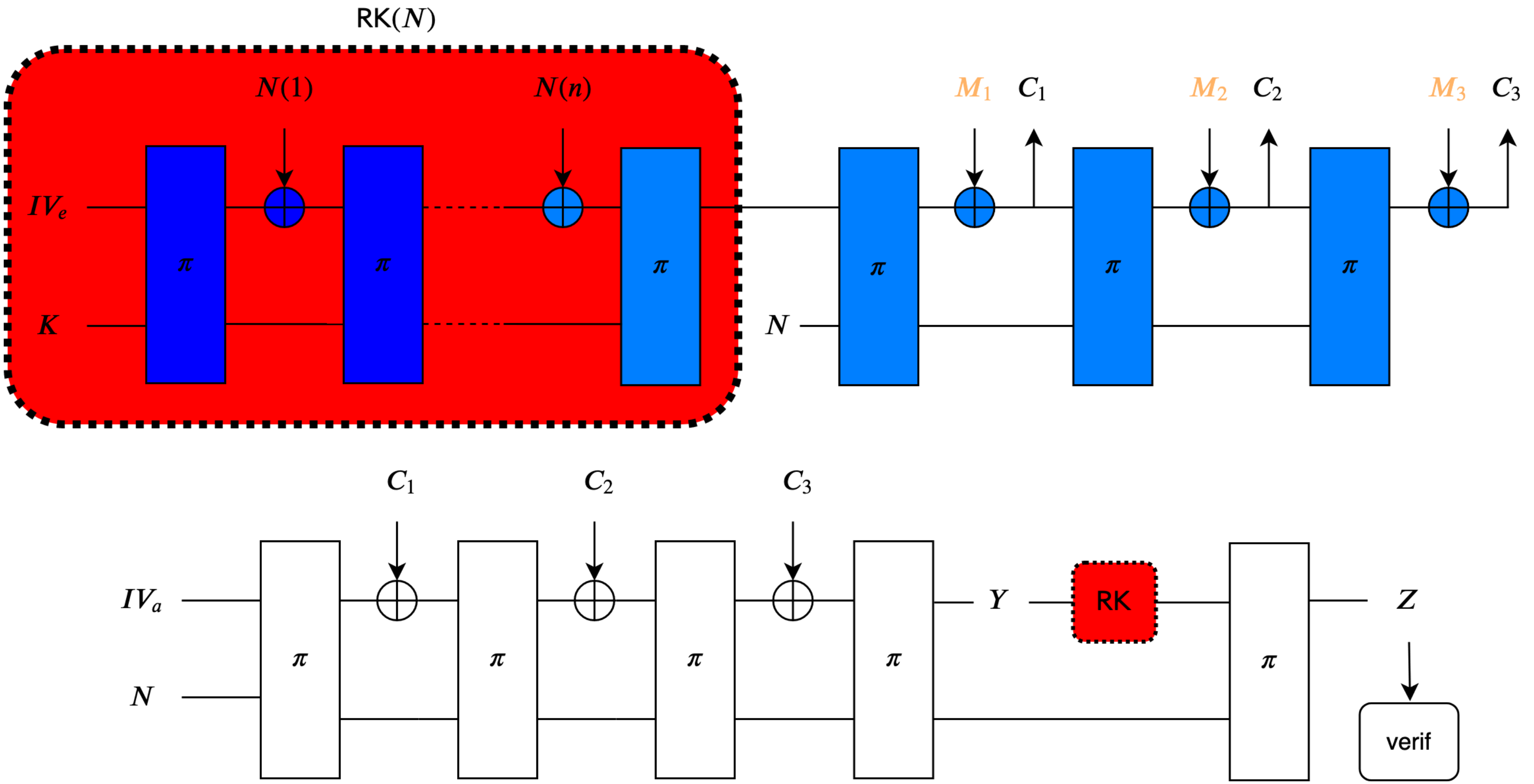
1. How to reason about (AE) leakage?
   - Specify the security target
   - Analyse the mode (& choose assumptions)
   - Evaluate the implementation (& primitive)

     ≈ cost needed to fulfil the assumptions

2. **Case studies: NIST candidates & more**
   - Level 0: no mode-level leakage-resistance
   - Level 1: re-keyed modes (including sponges)
   - Level 2: level 1 + strengthened init./final.
   - **Level 3: level 2 + two-passes**

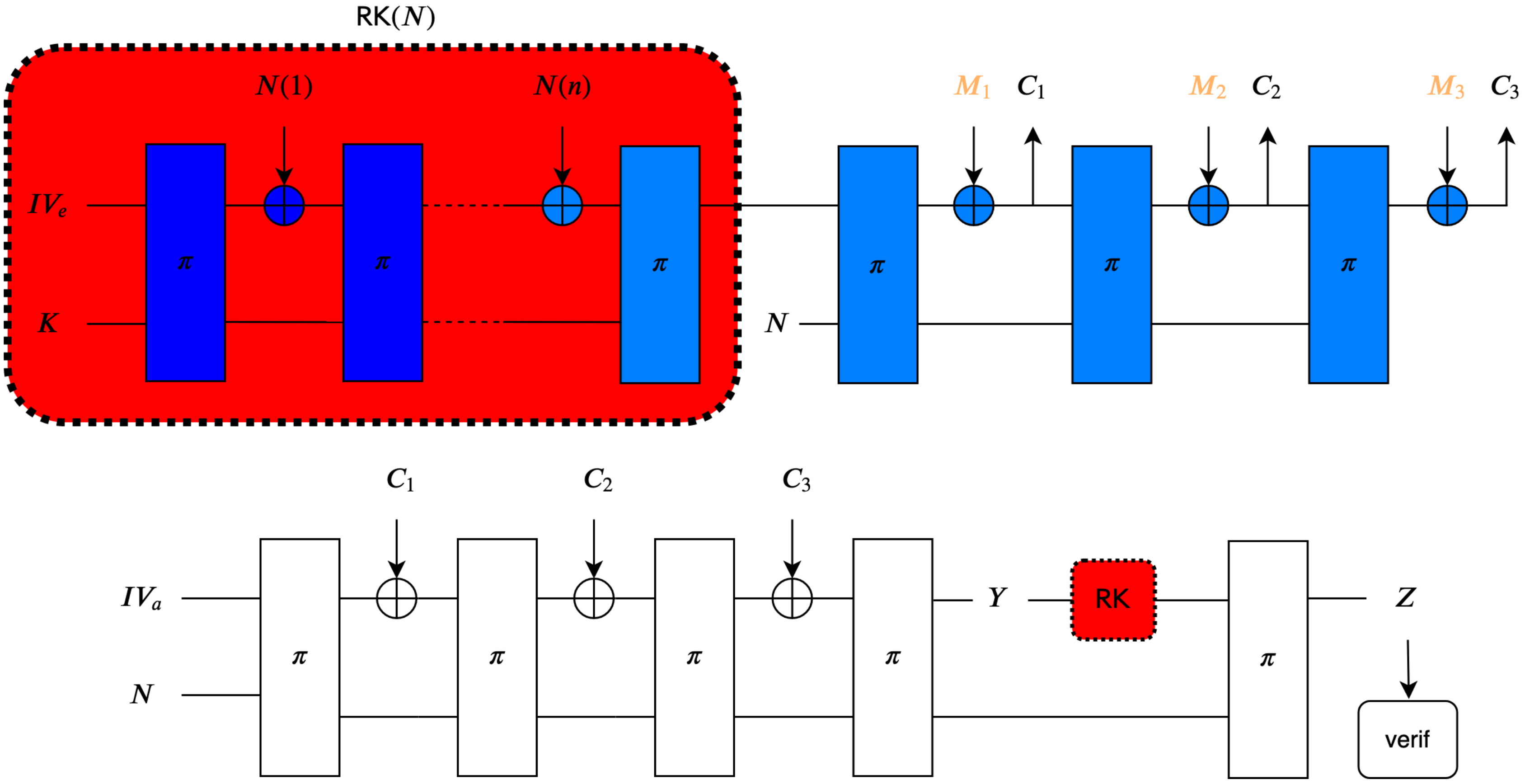
ISAP (confidentiality)

- CCAL1 (L in enc only, no misuse)



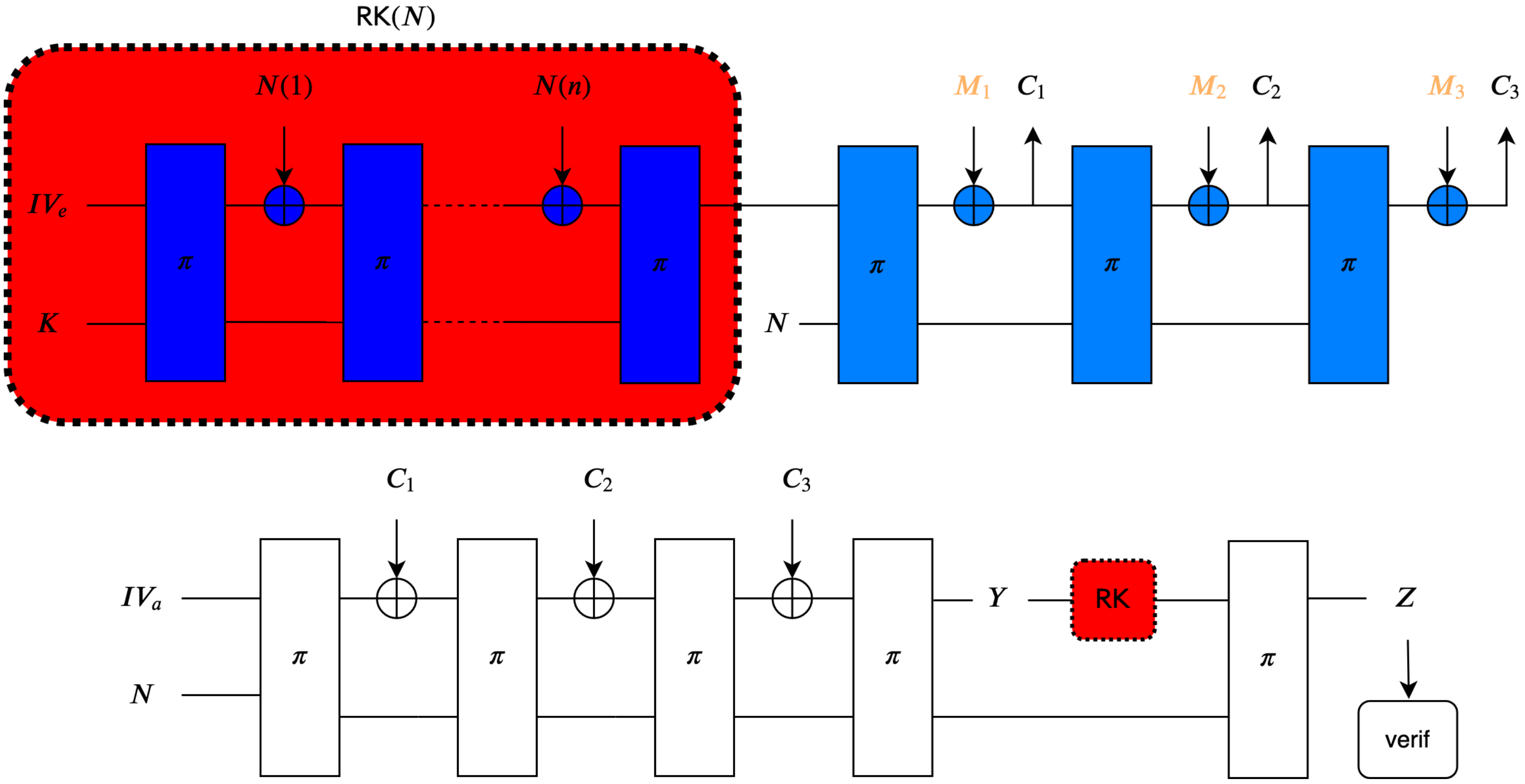
- *DPA resistance via SPA-resistance (with averaging)?*

- CCAL1 (L in enc only, no misuse)



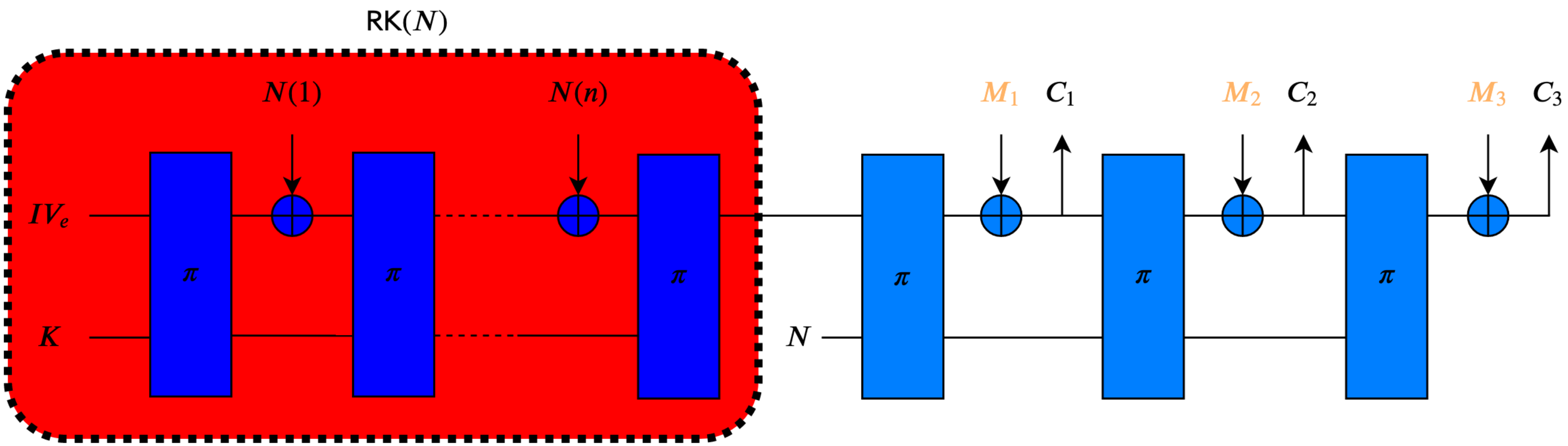
- If DPA-resistant RK, then similar to Ascon/Spook

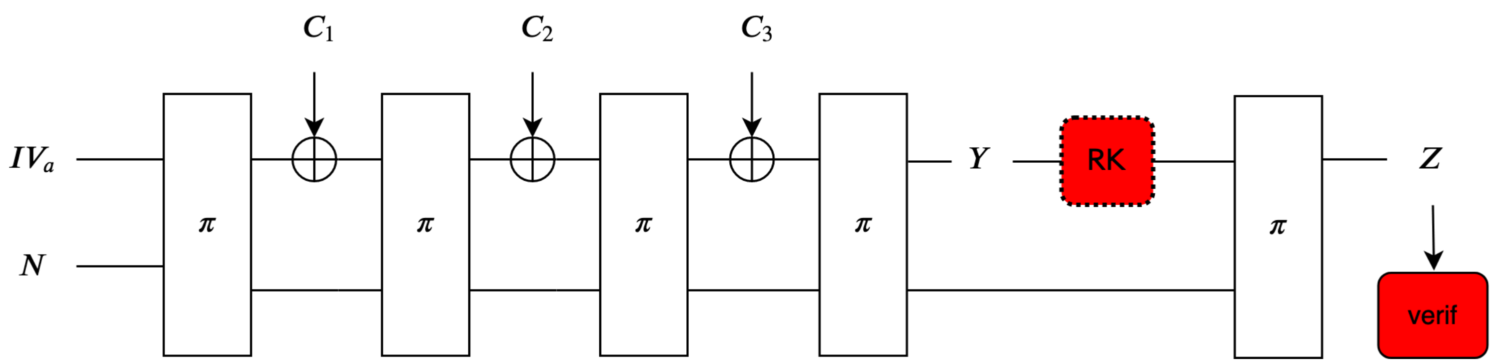
- CCAmL1 (L in enc only, misuse-resilience)



- Not much change (averaging everywhere in RK)

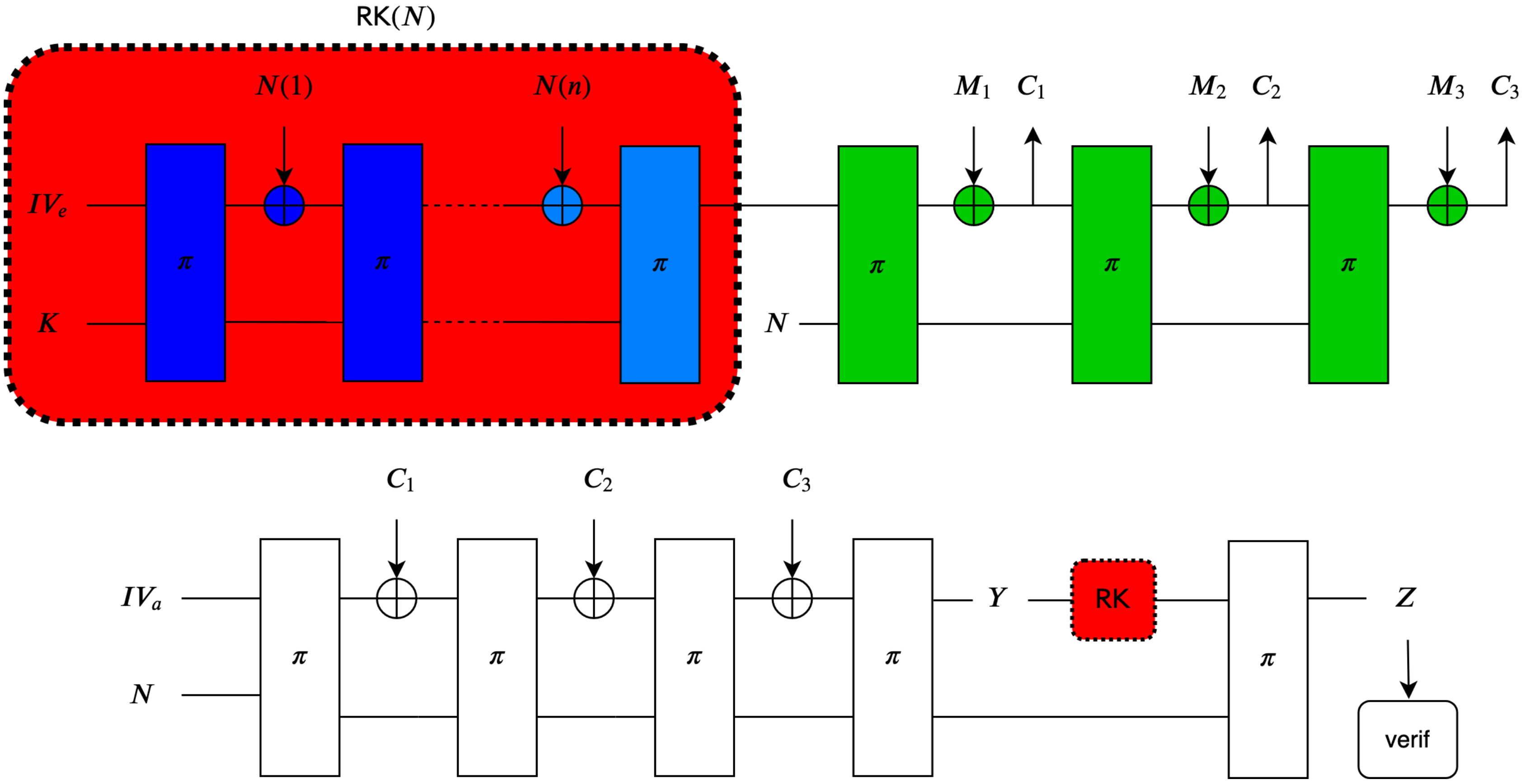
- CCAmL2 (L in enc/dec, misuse-resilience)





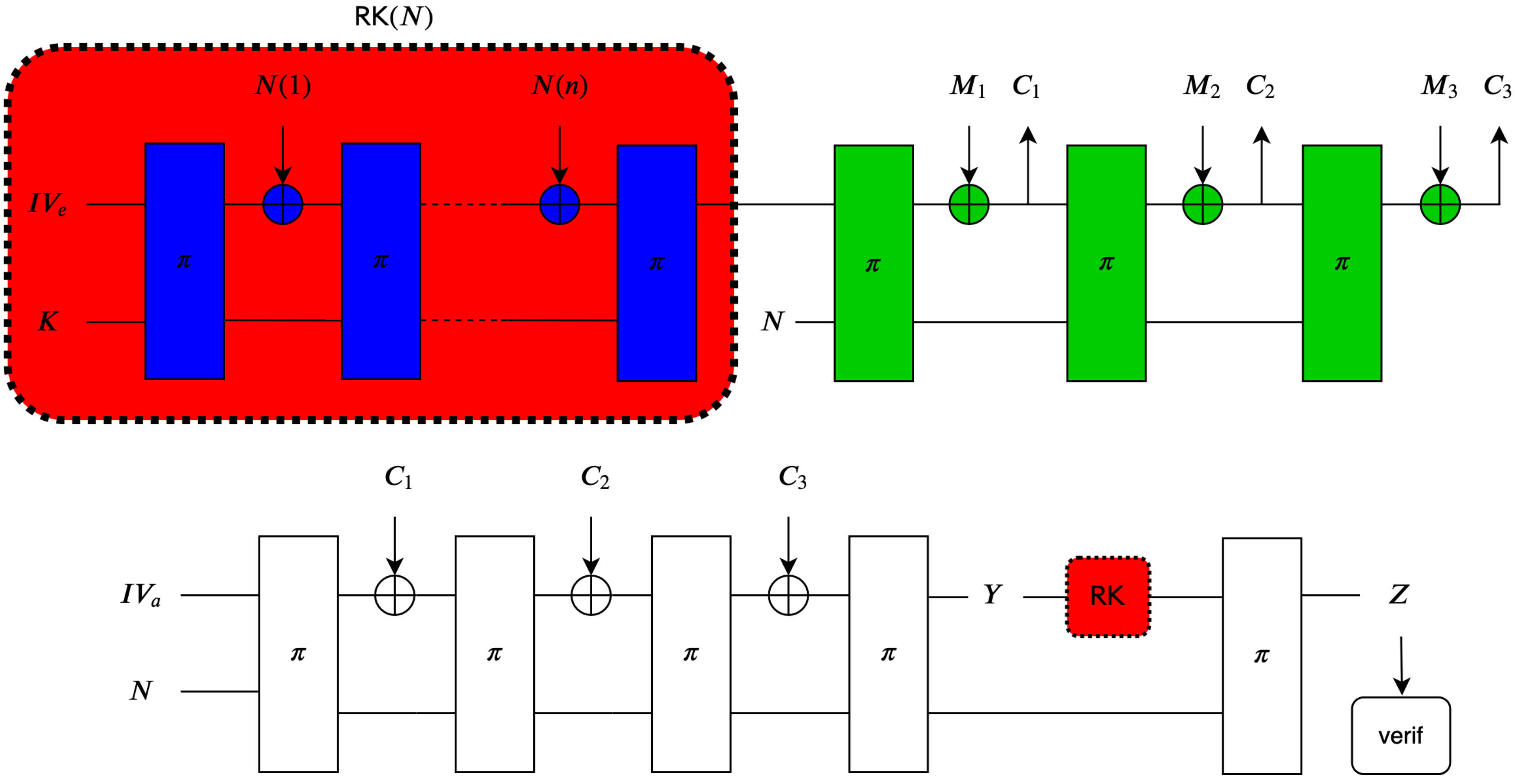
- 2 pass ⇒ confidentiality in dec. if DPA-resistant verif.

- CIL1 (L in enc only, no misuse)



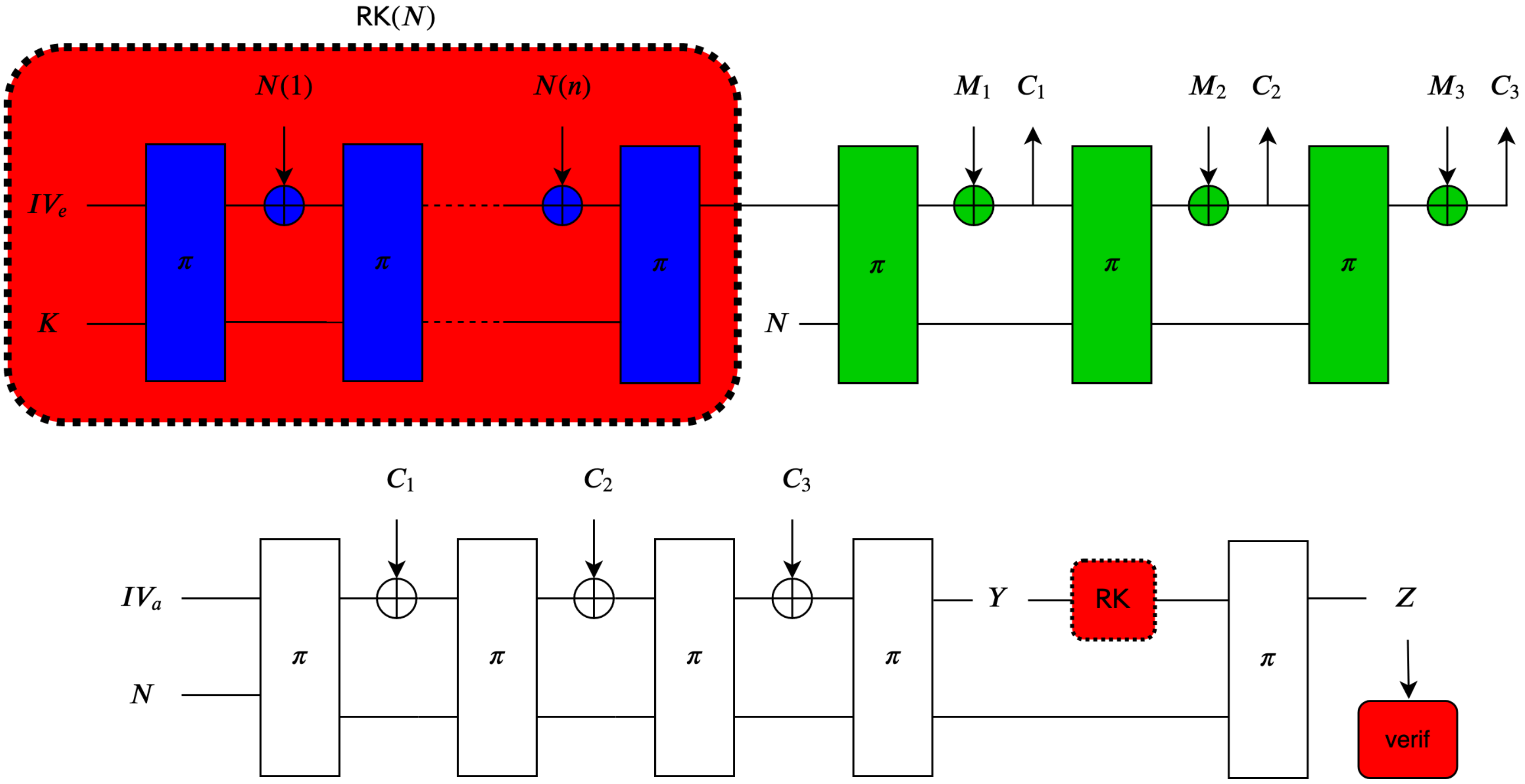
- Similar to ASCON/Spook (with ≠ init./final.)

- CIML1 (L in enc only, misuse-resistance)



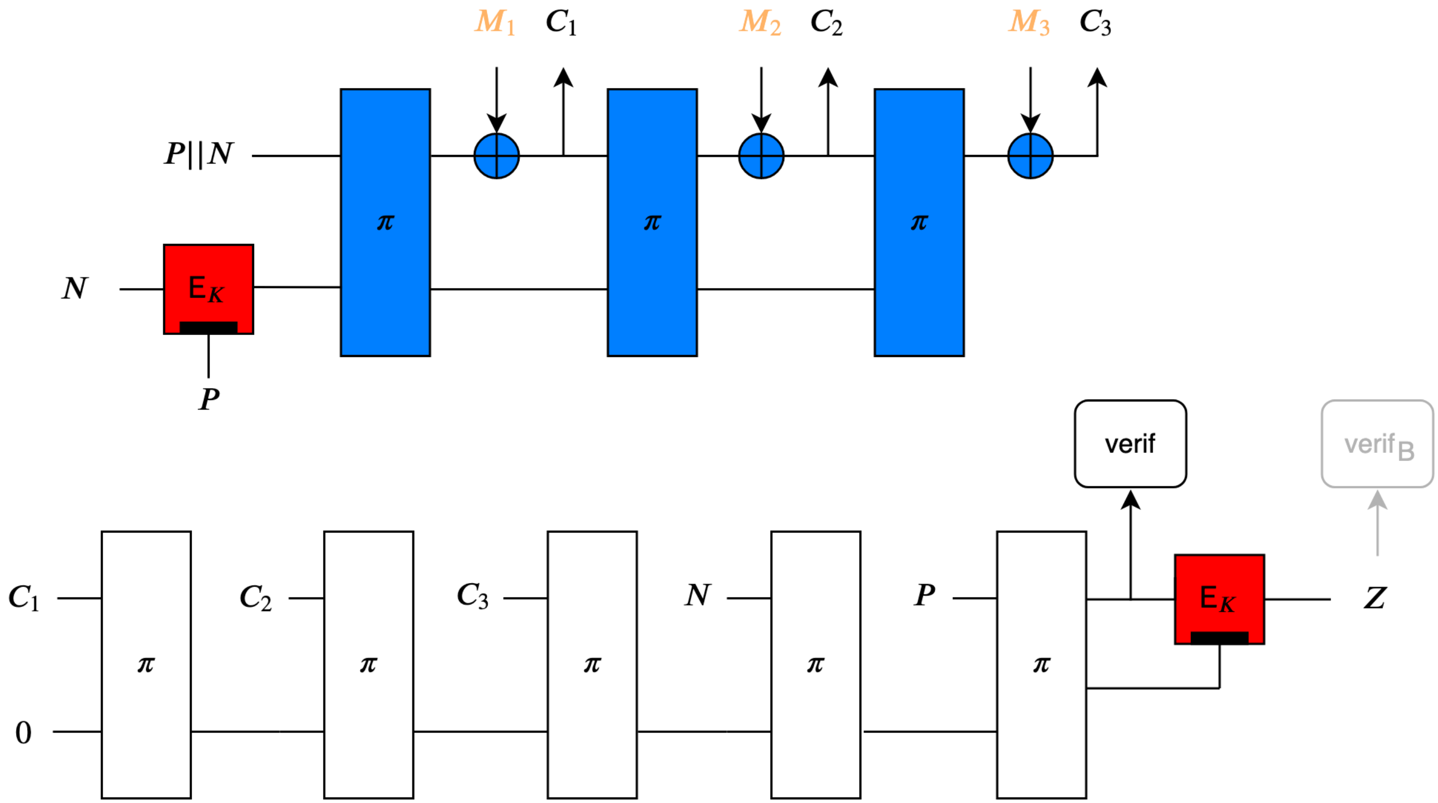
- Similar to ASCON/Spook (with ≠ init./final.)

- CIML2 (L in enc/dec, misuse-resistance)



- Similar to ASCON (need DPA-resistant tag verif.)

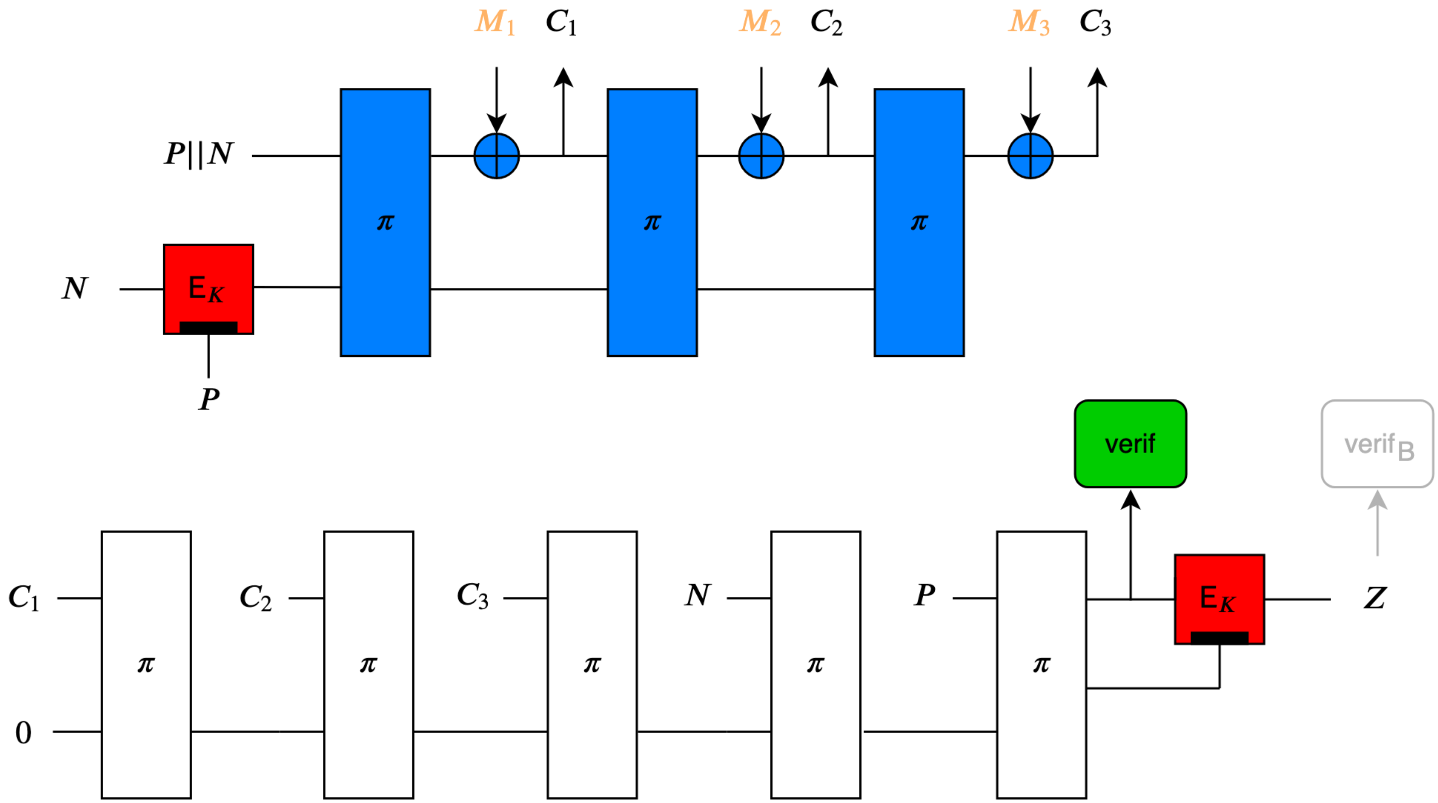
- CCAL1, CCAmL1



- Similar to ISAP with masked init./final.

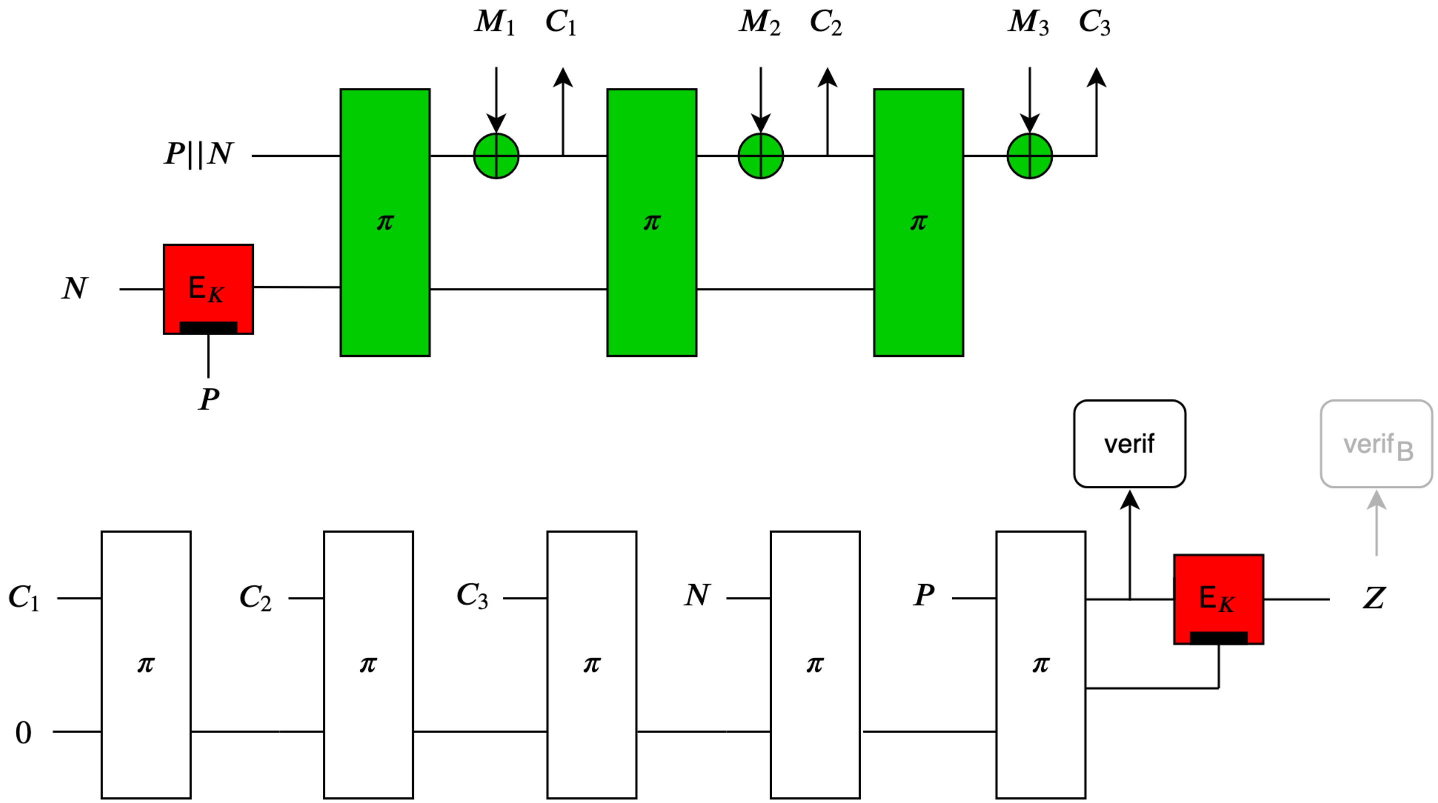
TEDT/TEDTSponge (confidentiality)

- CCAmL2 (L in enc/dec, misuse-resilience)



- Tag verification with unbounded leakages

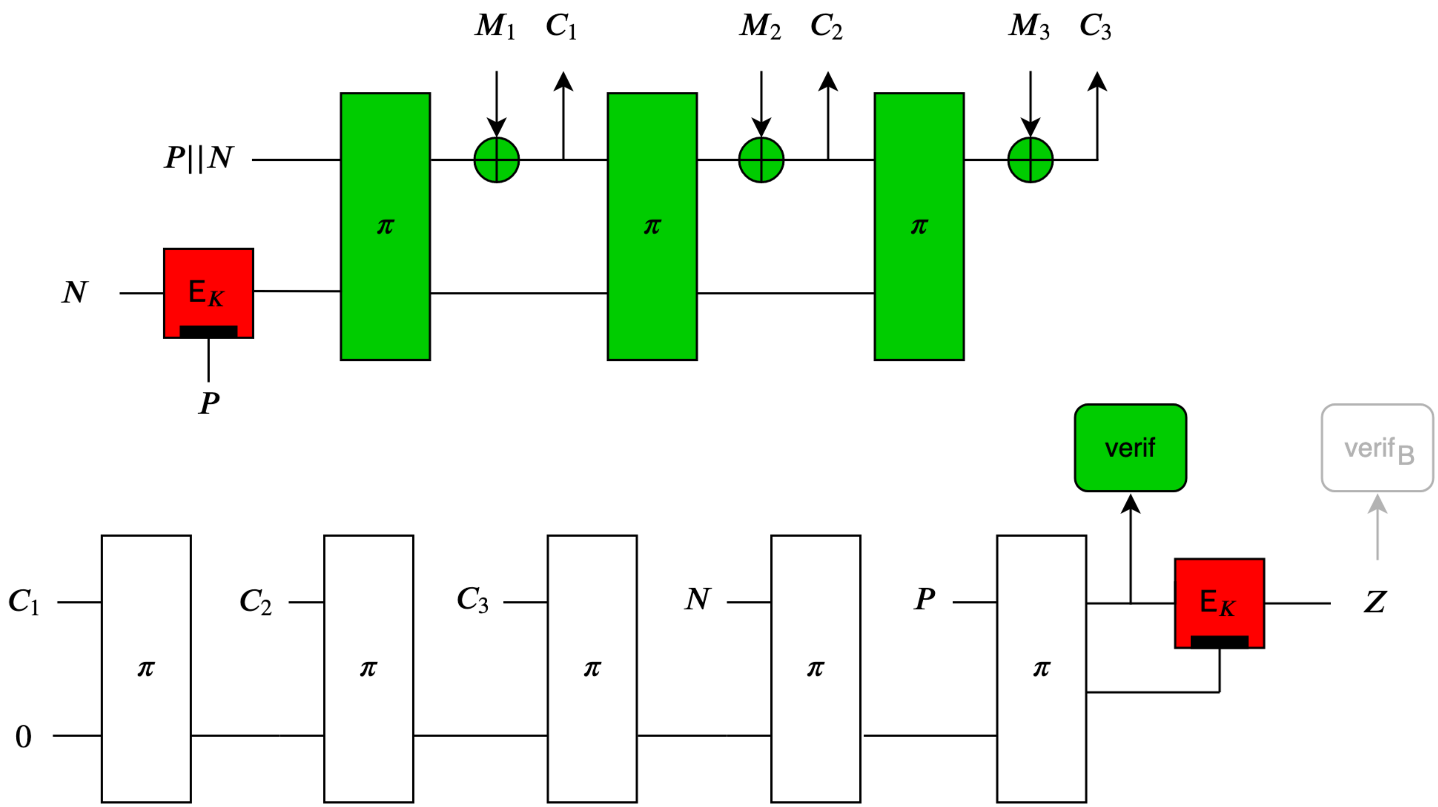
- CIL1, CIML1



- Similar to ISAP with masked init./final.

TEDT/TEDTSponge (integrity)

- CIML2 (L in enc/dec, misuse-resistance)



- Tag verification with unbounded leakages

Conclusion (I)

- **What this discussion shows**
  - $\exists$ a tradeoff between mode-level and implementation leakage-resistance
  - As the security target and level increase, mode-level leakage-resistance gains more interest

Conclusion (I)

- **What this discussion shows**
  - ∃ a tradeoff between mode-level and implementation leakage-resistance
  - As the security target and level increase, mode-level leakage-resistance gains more interest

- **What this discussion suggests**
  - Using a strengthened init./final. for duplex sponges

Conclusion (I)

- **What this discussion shows**
  - ∃ a tradeoff between mode-level and implementation leakage-resistance
  - As the security target and level increase, mode-level leakage-resistance gains more interest
- **What this discussion suggests**
  - Using a strengthened init./final. for duplex sponges
- **What this discussion does not show (yet)**
  - Which candidate is best in which context?
    - *Security evaluations & implementation results*
  - Primitives matter (e.g., OCB-Pyjamask vs. OCB-AES)

Conclusion (II)

- **What this discussion cannot show**
  - There is also a "simplicity vs. flexibility" tradeoff
    - e.g., ISAP's default implementation
      + offers some SCA security without masking
      – is affected by primitive-based overheads
        - Always there (even if SCAs are not a concern)
    - e.g., Spook's secure implementation
      – requires masking for the highly protected block
      + is affected by implementation overheads
        - Can be modulated (in function of the needs)

- **What this discussion cannot show**
  - There is also a "simplicity vs. flexibility" tradeoff
    - e.g., ISAP's default implementation

      + offers some SCA security without masking

      – is affected by primitive-based overheads
      - Always there (even if SCAs are not a concern)
    - e.g., Spook's secure implementation

      – requires masking for the highly protected block

      + is affected by implementation overheads
      - Can be modulated (in function of the needs)

+ Happy to discuss missing candidates (just contact me)

+ More discussion: https://www.youtube.com/watch?v=KdhrsuJT1sE

		CCAL1	CCAmL1	CCAmL2	CIL1	CIML1	CIML2
OCB-Pyjamask	init/final						
	bulk						
	verif.			1-pass			
	msg.						
PHOTON-Beetle	init/final						
	bulk						
	verif.			1-pass			
	msg.						
Ascon	init/final						
	bulk						
	verif.			1-pass			
	msg.						
Spook	init/final						
	bulk						
	verif.			1-pass			
	msg.						
ISAP	init/final	?	?	?	?	?	?
	bulk						
	verif.						
	msg.						
TEDT Sponge	init/final						
	bulk						
	verif.						
	msg.						

Bibliography

Side-channel countermeasures:

- Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, Pankaj Rohatgi: *Towards Sound Approaches to Counteract Power-Analysis Attacks*. CRYPTO 1999: 398-412
- Stefan Mangard: *Hardware Countermeasures against DPA ? A Statistical Analysis of Their Effectiveness*. CT-RSA 2004: 222-235
- Christoph Herbst, Elisabeth Oswald, Stefan Mangard: *An AES Smart Card Implementation Resistant to Power Analysis Attacks*. ACNS 2006: 239-252
- Stefan Dziembowski, Krzysztof Pietrzak: *Leakage-Resilient Cryptography*. FOCS 2008: 293-302
- Gilles Piret, Thomas Roche, Claude Carlet: *PICARO - A Block Cipher Allowing Efficient Higher-Order Side-Channel Resistance*. ACNS 2012: 311-328
- Olivier Pereira, François-Xavier Standaert, Srinivas Vivek: *Leakage-Resilient Authentication and Encryption from Symmetric Cryptographic Primitives*. ACM Conference on Computer and Communications Security 2015: 96-108

Side-channel countermeasures & security definitions:

- Guy Barwell, Daniel P. Martin, Elisabeth Oswald, Martijn Stam: *Authenticated Encryption in the Face of Protocol and Side Channel Leakage.* ASIACRYPT (1) 2017: 693-723
- Chun Guo, Olivier Pereira, Thomas Peters, François-Xavier Standaert: *Authenticated Encryption with Nonce Misuse and Physical Leakage: Definitions, Separation Results and First Construction - (Extended Abstract)*. LATINCRYPT 2019: 150-172

Bibliography

Countermeasures' overheads:

- Dahmun Goudarzi, Matthieu Rivain: *How Fast Can Higher-Order Masking Be in Software?* EUROCRYPT (1) 2017: 567-597
- Hannes Groß, Stefan Mangard, Thomas Korak: *An Efficient Side-Channel Protected AES Implementation with Arbitrary Protection Order*. CT-RSA 2017: 95-112
- O. Bronchain, F.-X. Standaert, *Side-Channel Countermeasures' Dissection and the Limits of Closed Source Security Evaluations*, cryptology e-Print archive, report 2019/1008

Leakage analyses:

- Francesco Berti, Chun Guo, Olivier Pereira, Thomas Peters, François-Xavier Standaert: *TEDT, a Leakage-Resilient AEAD mode for High (Physical) Security Applications*. IACR Cryptology ePrint Archive 2019: 137 (2019)
- Chun Guo, Olivier Pereira, Thomas Peters, François-Xavier Standaert: *Towards Low-Energy Leakage-Resistant Authenticated Encryption from the Duplex Sponge Construction.* IACR Cryptology ePrint Archive 2019: 193 (2019)
- Christoph Dobraunig, Bart Mennink: *Leakage Resilience of the Duplex Construction*. IACR Cryptology ePrint Archive 2019: 225 (2019)
- Jean Paul Degabriele, Christian Janson, Patrick Struck: *Sponges Resist Leakage: The Case of Authenticated Encryption*. IACR Cryptology ePrint Archive 2019: 1034 (2019)

Physical assumptions (I)

- Physical assumptions for the **initialization**
  - Leak-Free components (LF) [PSV15]
  - Strong Unpredictability with Leakage (SUL) [BGPPS19]

- Physical assumptions for the **bulk computation**
  - Leak-Free components (LF) [PSV15]
  - Oracle-Free + Hard-to-Invert Leakages (OFL+HIL) [YSPY10]
    - (HIL can be replaced by bounded leakage [DP08])
  - Simulatable Leakages (SimL) [SPY13]
  - *Only Computation Leaks* (OCL) [DP08]
  - Unbounded Leakages (UnbL) [BKPPS18]

Physical assumptions (II)

- Physical assumptions for the **finalization**
  - Leak-Free components (LF) [PSV15]
  - Strong Unpredictability with Leakage (SUL) [BGPPS19]
- Physical assumptions for the **tag verification**
  - Leak-Free components (LF) [PSV15]
    - (*HIL is probably enough for this part*)
  - Unbounded Leakages (UnbL) [BKPPS18]
- (*For leakage-resistant confidentiality, security can only be reduced to the message confidentiality of a single block*)

Aappendix bibliography

Physical assumptions (for symmetric cryptography):

- Stefan Dziembowski, Krzysztof Pietrzak: *Leakage-Resilient Cryptography*. FOCS 2008: 293-302
- Yu Yu, François-Xavier Standaert, Olivier Pereira, Moti Yung: *Practical leakage-resilient pseudorandom generators*. ACM Conference on Computer and Communications Security 2010: 141-151
- François-Xavier Standaert, Olivier Pereira, Yu Yu: *Leakage-Resilient Symmetric Cryptography under Empirically Verifiable Assumptions*. CRYPTO (1) 2013: 335-352
- Olivier Pereira, François-Xavier Standaert, Srinivas Vivek: *Leakage-Resilient Authentication and Encryption from Symmetric Cryptographic Primitives*. ACM Conference on Computer and Communications Security 2015: 96-108
- Francesco Berti, François Koeune, Olivier Pereira, Thomas Peters, François-Xavier Standaert: *Ciphertext Integrity with Misuse and Leakage: Definition and Efficient Constructions with Symmetric Primitives*. AsiaCCS 2018: 37-50
- Francesco Berti, Chun Guo, Olivier Pereira, Thomas Peters, François-Xavier Standaert: *Strong Authenticity with Leakage under Weak and Falsifiable Physical Assumptions*, Inscrypt 2019