Update on the Performance and Mode-level Properties of ISAP

Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, **Bart Mennink**, **Robert Primas**, Thomas Unterluggauer

NIST LWC Workshop 2022

Outline

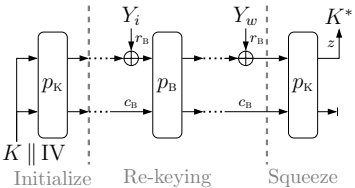
- Mode-level Properties
- Implications on Physical Attacks

Mode-level Properties

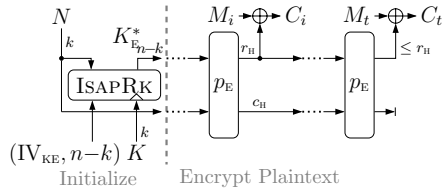
ISAP

- Sponge/duplex-based authenticated encryption mode
- Instantiation:
  - ASCON-p
  - KECCAK-p [400]
- Carefully selected capacities and rates:
  - Robustness against DPA
  - Hardening against fault attacks: DFA, SFA, SIFA

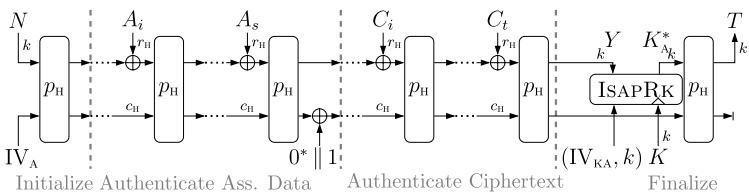
Isap



IsapRK

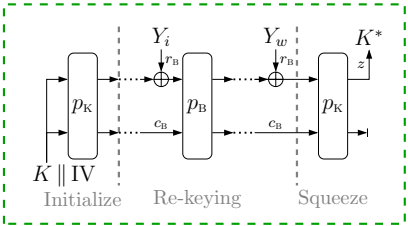


IsapEnc

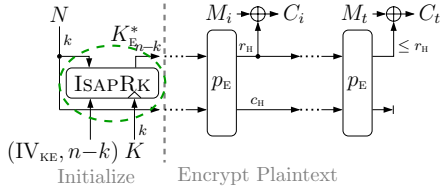


IsapMAC

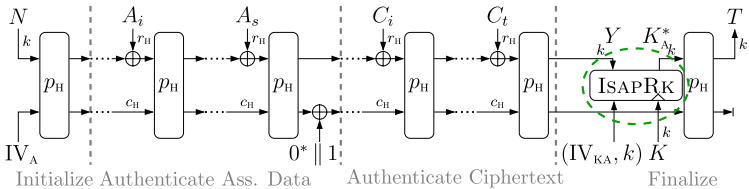
ISAP



ISAPRK

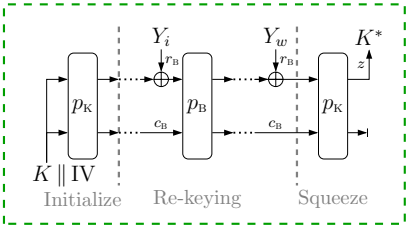


ISAPENC



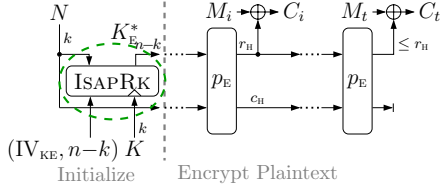
ISAPMAC

IsAP

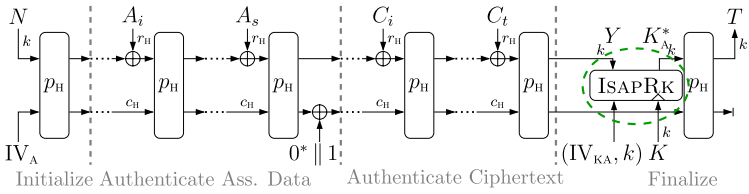


IsAPRK

keyed duplex with master key and rate 1 bit

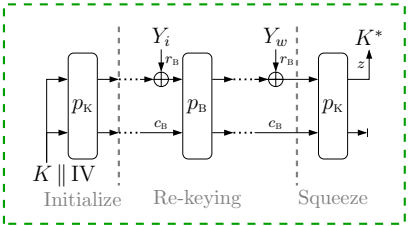


IsAPEnc

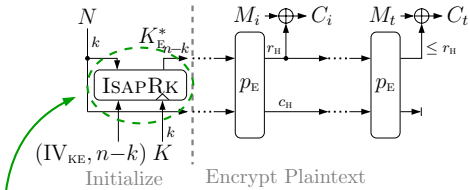


IsAPMAC

ISAP



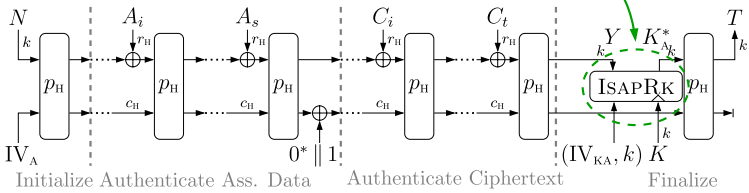
ISAPRK



ISAPENC

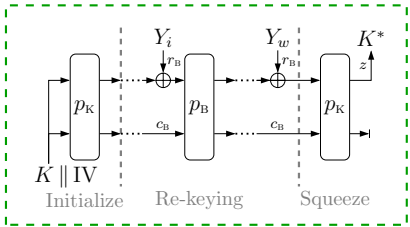
pseudorandom subkeys

keyed duplex with master key and rate 1 bit

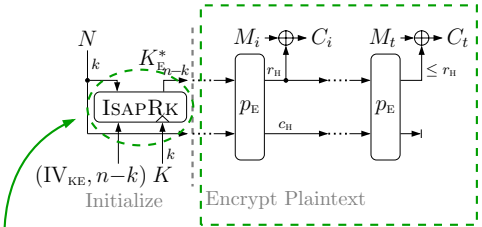


ISAPMAC

Isap



IsapRK

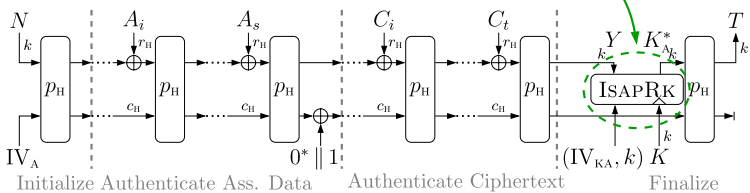


IsapEnc

keyed duplex with master key and rate 1 bit

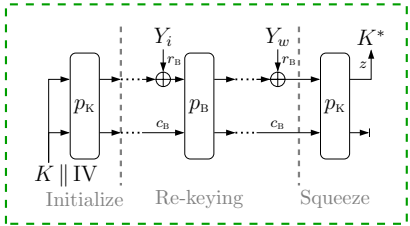
pseudorandom subkeys

keyed duplex with fresh keys and high rate

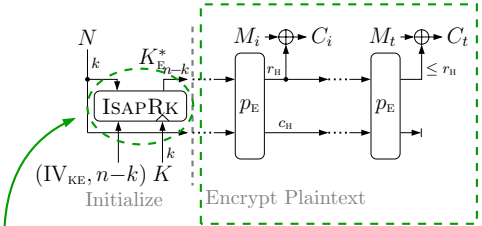


IsapMAC

Isap



IsapRK

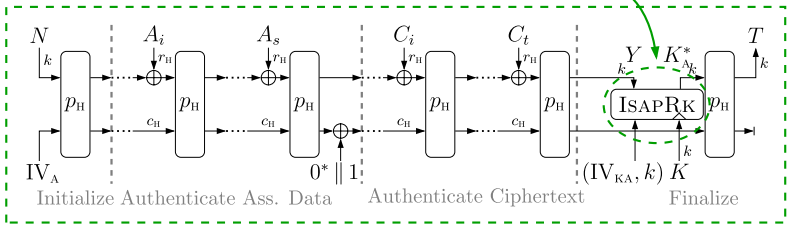


IsapEnc

keyed duplex with master key and rate 1 bit

pseudorandom subkeys

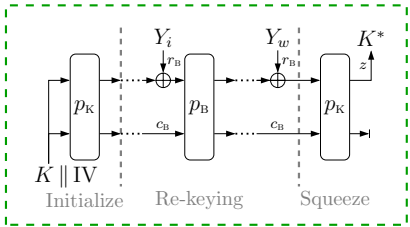
keyed duplex with fresh keys and high rate



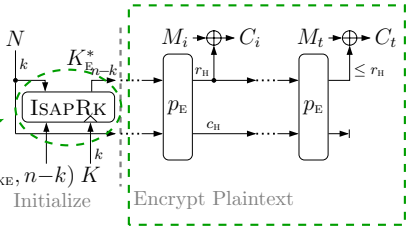
IsapMAC

SuKS hashes at high rate

Leakage Resilience of ISAP Mode



ISAPRK

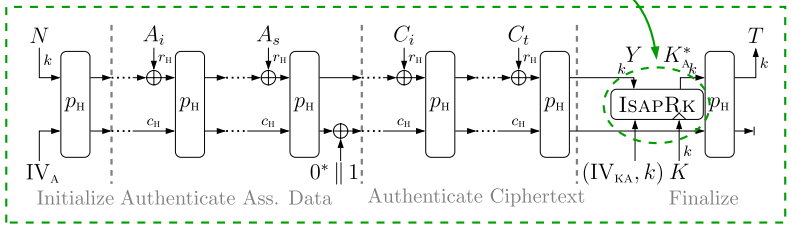


ISAPENC

pseudorandom subkeys

keyed duplex with master key and rate 1 bit

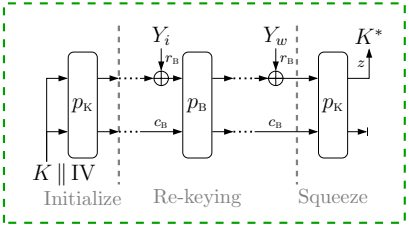
keyed duplex with fresh keys and high rate



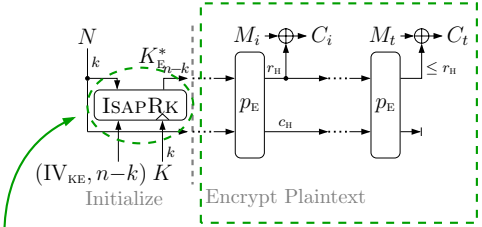
ISAPMAC

SuKS hashes at high rate

Leakage Resilience of ISAP Mode



ISAPRK

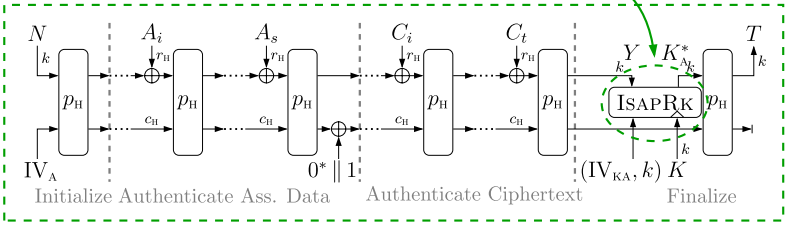


ISAPENC

pseudorandom subkeys

keyed duplex with master key and rate 1 bit ≡ LR-PRF [DM19a]

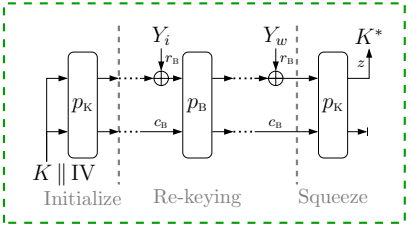
keyed duplex with fresh keys and high rate



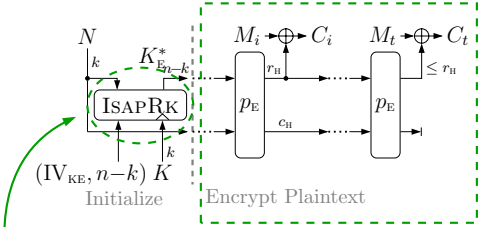
ISAPMAC

SuKS hashes at high rate

Leakage Resilience of ISAP Mode



ISAPRK

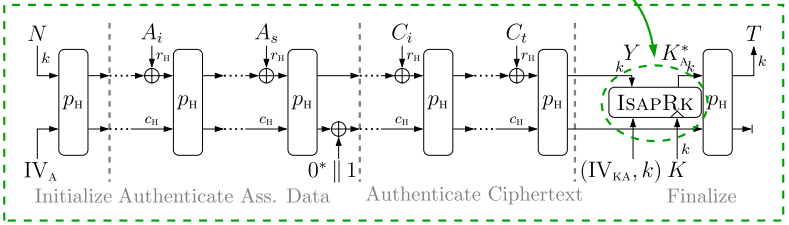


ISAPENC

sufficiently random subkeys

keyed duplex with master key and rate 1 bit ≡ LR-PRF [DM19a]

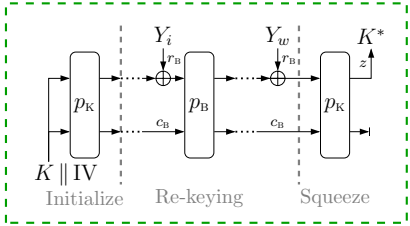
keyed duplex with fresh keys and high rate



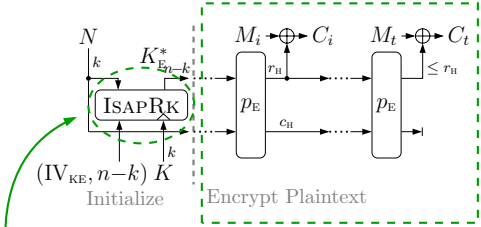
ISAPMAC

SuKS hashes at high rate

Leakage Resilience of ISAP Mode



ISAPRK

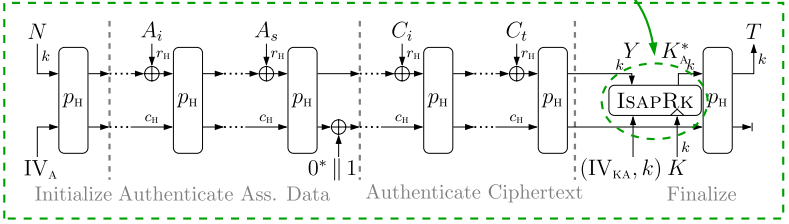


ISAPENC

sufficiently random subkeys

keyed duplex with master key and rate 1 bit ≡ LR-PRF [DM19a]

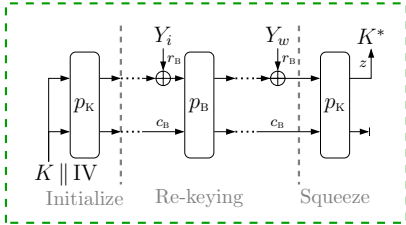
keyed duplex with fresh keys and high rate ≡ LR-PRF [DM19a]



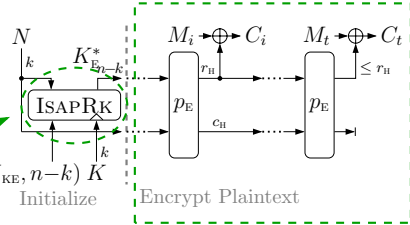
ISAPMAC

SuKS hashes at high rate

Leakage Resilience of ISAP Mode



ISAPRK

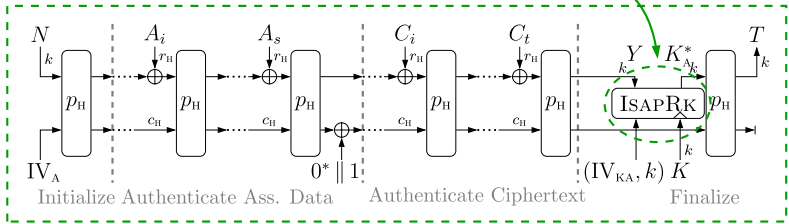


ISAPENC

sufficiently random subkeys

keyed duplex with master key and rate 1 bit ≡ LR-PRF [DM19a]

keyed duplex with fresh keys and high rate ≡ LR-PRF [DM19a]



ISAPMAC

SuKS hashes at high rate ≡ LR-PRF [DM19b]

Summary of Conventional Security

- Security of ISAP in nonce-respecting setting
  - Assumption: random permutation model, bounded leakage model
  - ISAP is a secure authenticated encryption scheme [DEM+20]
- Exotic features
  - Security of nonce-reuse: leakage-resilient authenticity preserved!
  - Release of unverified plaintext: security guaranteed due to EtM
  - Key-committing security: usage of ISAPRK within ISAPMAC

Summary of Conventional Security

- Security of ISAP in nonce-respecting setting
  - Assumption: random permutation model, bounded leakage model
  - ISAP is a secure authenticated encryption scheme [DEM+20]
- Exotic features
  - Security of nonce-reuse: leakage-resilient authenticity preserved!
  - Release of unverified plaintext: security guaranteed due to EtM
  - Key-committing security: usage of ISAPRK within ISAPMAC

Two New Leakage Resilience Results

EUROCRYPT 2021 [DM21]

Leakage Resilient Value Comparison With Application to Message Authentication

Christoph Dobraunig and Bart Mennink

- Solution to perform tag verification …
  - in a leakage resilient way
  - without extra primitives

In submission [DMP20]

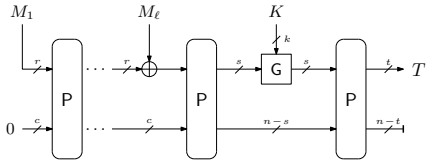
Leakage and Tamper Resilient Permutation-Based Cryptography

Christoph Dobraunig and Bart Mennink and Robert Primas

- More practical leakage resilience model
  - closer fit to actual leakage
  - also captures fault attacks

Leakage Resilient Value Comparison (LRVC)

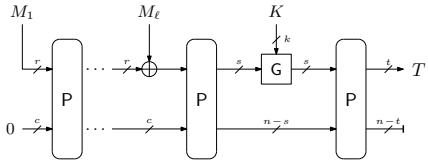
- SuKS (Suffix Keyed Sponge):



- Naive Tag Verification on input of message/tag tuple $(M, T^\star)$:
  - Compute $T = \mathsf{SuKS}(K, M)$
  - If $T^\star = T$ return 1, otherwise return 0
- Verification might leak information about T!
- This work: formal analysis of leakage resilient value comparison

Leakage Resilient Value Comparison (LRVC)

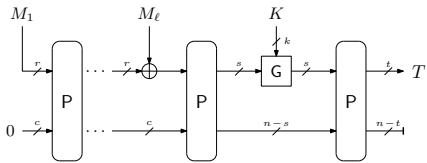
- SuKS (Suffix Keyed Sponge):



- Naive Tag Verification on input of message/tag tuple $(M, T^\star)$:
  - Compute $T = \mathsf{SuKS}(K, M)$
  - If $T^\star = T$ return 1, otherwise return 0
- Verification might leak information about T!
- This work: formal analysis of leakage resilient value comparison

Leakage Resilient Value Comparison (LRVC)

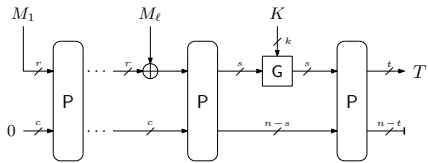
- SuKS (Suffix Keyed Sponge):



- Naive Tag Verification on input of message/tag tuple $(M, T^\star)$:
  - Compute $T = \mathsf{SuKS}(K, M)$
  - If $T^\star = T$ return 1, otherwise return 0
- Verification might leak information about T!
- This work: formal analysis of leakage resilient value comparison

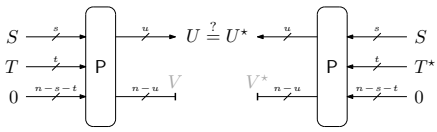
Leakage Resilient Value Comparison (LRVC)

- SuKS (Suffix Keyed Sponge):

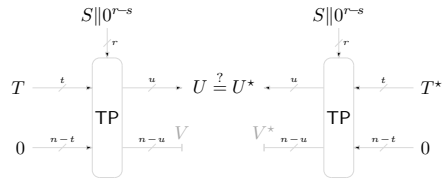


- Naive Tag Verification on input of message/tag tuple $(M, T^\star)$:
  - Compute $T = \mathsf{SuKS}(K, M)$
  - If $T^\star = T$ return 1, otherwise return 0
- Verification might leak information about T!
- This work: formal analysis of leakage resilient value comparison

LRVC: (Tweakable) Permutation-Based Value Processing



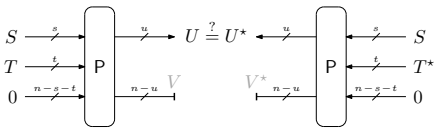
PVP



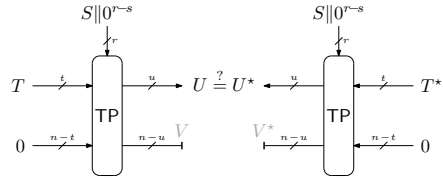
TPVP

- PVP gives leakage resilient value comparison
  - P is public or secret permutation
  - Similar to earlier suggestion of designers of ISAP [DEM+20]
- TPVP gives leakage resilient value comparison
  - TP is public or secret tweakable permutation
  - TPVP with secret tweakable permutation was used in Spook [BBB+20]

LRVC: (Tweakable) Permutation-Based Value Processing



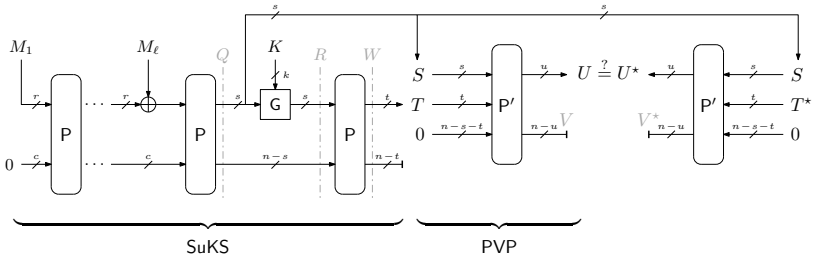
PVP



TPVP

- PVP gives leakage resilient value comparison
  - P is public or secret permutation
  - Similar to earlier suggestion of designers of ISAP [DEM+20]
- TPVP gives leakage resilient value comparison
  - TP is public or secret tweakable permutation
  - TPVP with secret tweakable permutation was used in Spook [BBB+20]

LRVC: SuKS-then-PVP (StP)



- Natural combination of SuKS and PVP
- Salt taken from keyless computation of SuKS
- Leakage resilience of StP (and thus of Isap) follows from that of SuKS and of PVP

Accumulated Interference

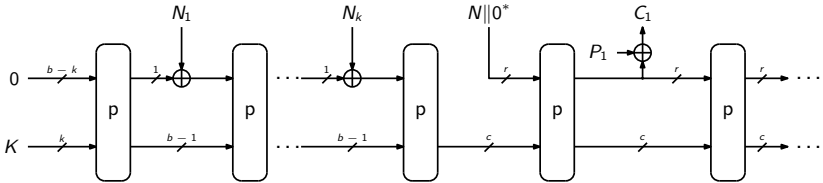
- Previous analysis in bounded leakage model
  - Adversary can choose any polynomial-time computable leakage function with bounded range [DP08]
  - Each new primitive call leaks $\leq \lambda$ bits
- This work: Accumulated Interference
  - We do not have to bound leakage per primitive call a priori
  - Rather, define it as the Accumulated Gain over time: $\mathrm{AG}(i)$
  - A priori and a posteriori, accurate bound on $\mathrm{AG}(i)$ derived through measurements
  - Additionally model effect of faults

Accumulated Interference

- Previous analysis in bounded leakage model
  - Adversary can choose any polynomial-time computable leakage function with bounded range [DP08]
  - Each new primitive call leaks $\leq \lambda$ bits
- This work: Accumulated Interference
  - We do not have to bound leakage per primitive call a priori
  - Rather, define it as the Accumulated Gain over time: $\mathrm{AG}(i)$
  - A priori and a posteriori, accurate bound on $\mathrm{AG}(i)$ derived through measurements
  - Additionally model effect of faults

Accumulated Interference: ASAKEY

- We showcase the power of Accumulated Interference in ASAKEY



- ASAKEY $\approx$ encryption part of ISAP
- ASAKEY is a LR-PRF up to (simplified) bound $\sum_{i=1}^{P} \frac{Q}{2^{b-\mathrm{AG}(i)-\tau}}$
- Naive (a priori) bounding: $\mathrm{AG}(i) \leq i \cdot \lambda$
- More accurate bounding using simulations

Implications on Physical Attacks

Mode-level Robustness Claims of ISAP

- Protection against Differential Power Analyis (DPA)
  - DPA-based key recovery $\rightarrow$ AI
  - DPA-based plaintext recovery
  - Optional: DPA-based tag recovery $\rightarrow$ STP
- Protection against Statistical (Ineffective) Fault Analysis (SFA/SIFA) $\rightarrow$ AI
- Hardening against Differenial Fault Analysis (DFA)
- Hardening against Profiling/Simple Power Analysis (SPA)

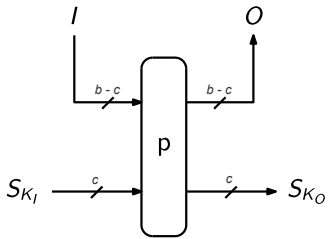
Mode-level Robustness Claims of ISAP

- Protection against Differential Power Analyis (DPA)
  - DPA-based key recovery $\rightarrow$ AI
  - DPA-based plaintext recovery
  - Optional: DPA-based tag recovery $\rightarrow$ STP
- Protection against Statistical (Ineffective) Fault Analysis (SFA/SIFA) $\rightarrow$ AI
- Hardening against Differenial Fault Analysis (DFA)
- Hardening against Profiling/Simple Power Analysis (SPA)

DPA-based Key Recovery

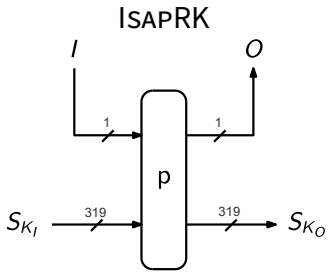
- Attack:
  - Query a keyed primitive using multiple different inputs
  - Hypothesis tests with corresponding power traces and sub-key guesses
- ISAP processes the main key only during ISAPRK
- ISAPRK limits amount of different queries to primitive with same key to 2

Accumulated Interference: Estimating $\mathrm{AG}_{\mathtt{atk}}(\boldsymbol{X}, \boldsymbol{q}, r)$



- $\mathrm{AG}_{\mathtt{atk}}(\boldsymbol{X}, \boldsymbol{q}, r)$
  - $\mathtt{atk} \in \{\mathtt{spa}, \mathtt{dpa}, \mathtt{sfa}, \ldots\}$
  - $\boldsymbol{X}$ inputs to p
  - $\boldsymbol{q}$ evaluation of p per input
  - r maximal number of X_i with the same inner part

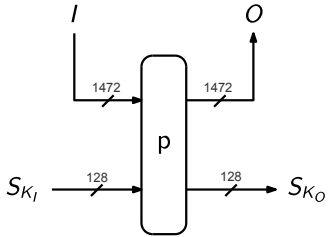
Accumulated Interference: Estimating $\mathrm{AG}_{\mathtt{atk}}(\boldsymbol{X}, \boldsymbol{q}, r)$



- $\mathrm{AG}_{\mathtt{atk}}(\boldsymbol{X}, \boldsymbol{q}, r)$
  - $\mathtt{atk} \in \{\mathtt{spa}, \mathtt{dpa}, \mathtt{sfa}, \ldots\}$
  - $\boldsymbol{X}$ inputs to p
  - $\boldsymbol{q}$ evaluation of p per input
  - r maximal number of X_i with the same inner part

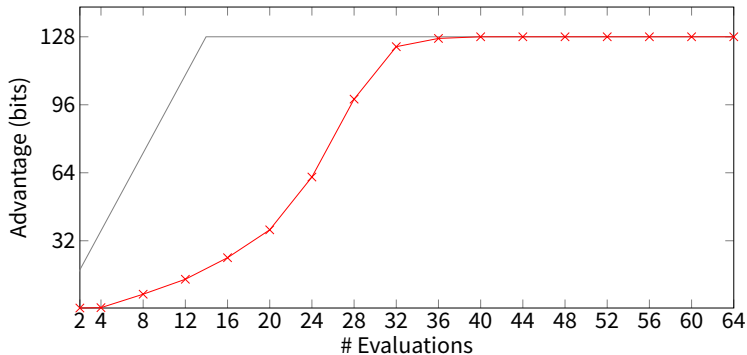
Accumulated Interference: Estimating $\mathrm{AG}_{\mathtt{atk}}(\boldsymbol{X}, \boldsymbol{q}, r)$

Our analysis [DMP20]



- $\mathrm{AG}_{\mathtt{atk}}(\boldsymbol{X}, \boldsymbol{q}, r)$
  - $\mathtt{atk} \in \{\mathtt{spa}, \mathtt{dpa}, \mathtt{sfa}, \ldots\}$
  - $\boldsymbol{X}$ inputs to p
  - $\boldsymbol{q}$ evaluation of p per input
  - r maximal number of X_i with the same inner part

Accumulated Interference: Estimating $\mathrm{AG}_{\mathrm{dpa}}(\boldsymbol{X}, \boldsymbol{q}, r)$



Evaluation Setup: Chipwhisperer-Lite with XMEGA128D4 target [DMP20]

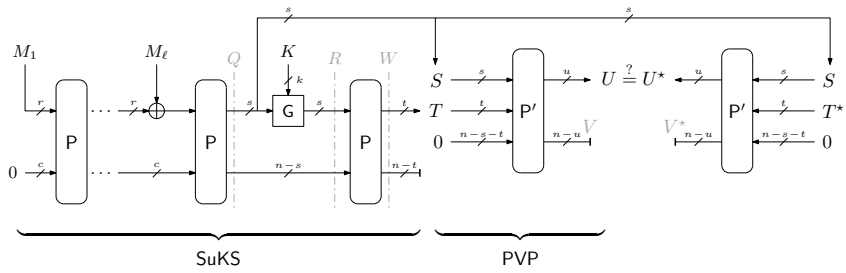
DPA-based Plaintext Recovery

- Attack:
  - Query the decryption with a constant nonce and varying ciphertexts
  - Constant key stream is combined with varying ciphertext blocks
  - Simple DPA-style attack can reveal the key stream
- Such attacks do not require direct extraction of cryptographic keys
- Firmware updates: Plaintext could carry cryptographic keys or IP
- ISAP's two-pass construction prevents this scenario
  - Tag verification before start of decryption

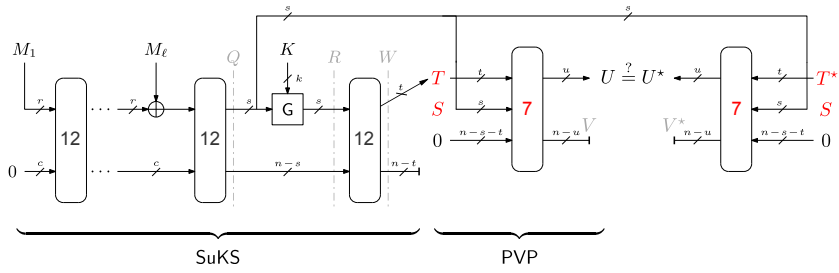
DPA-based Tag Recovery

- Attack:
  - Given a correct ciphertext C, flip some bits leading to related $C_1 \ldots C_n$, that then correspond to related messages $M_1 \ldots M_n$
  - These can then be used to forge valid tags for $C_1 \ldots C_n$ due to leaks in tag comparison [BBC+20]
- Tag comparison needs protection
  - Algorithmic masking: 127 ANDs with mult. depth of 7 (128-bit tag)
  - On mode-level: SuKS-then-PVP (STP)

SuKS-then-PVP (STP)

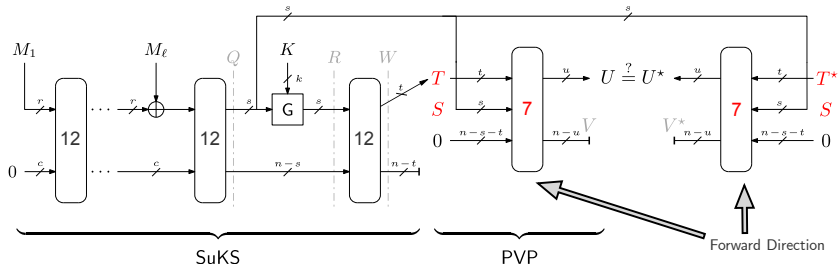


SuKS-then-PVP (StP)



- Our instantiation of StP at the end of IsapMAC (of Isap-A-128a):
  - 14 additional permutation rounds for tag comparison
  - No noteworthy increase in code-size/area
  - No changes to Isap algorithm and cryptographic properties

SuKS-then-PVP (StP)

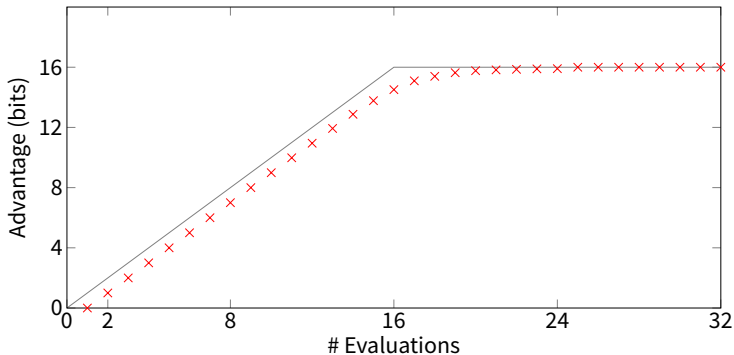


- Our instantiation of StP at the end of IsapMAC (of Isap-A-128a):
  - 14 additional permutation rounds for tag comparison
  - No noteworthy increase in code-size/area
  - No changes to Isap algorithm and cryptographic properties

Statistical (Ineffective) Fault Analysis (SFA/SIFA)

- Attack:
  - Query a keyed primitive using multiple different inputs and inject a fault in a certain location
  - Only collect those inputs where the fault did not affect the computation
  - Evaluate distributions of certain state bits using the collected inputs and partial key guesses
- ISAPRK limits amount of different queries to primitive with same key to 2

Accumulated Interference: Estimating $\mathrm{AG}_{\mathtt{sifa}}(\boldsymbol{X}, \boldsymbol{q}, r)$



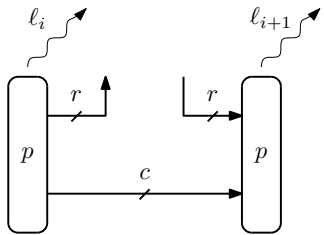
Evaluation Setup: Chipwhisperer-Lite with XMEGA128D4 target using clock glitches [DMP20]

Differential Fault Analysis

- Attack:
  - Query decryption with constant nonce
  - Inject faults at different locations before plaintext extraction
  - Use differences in plaintext (or leakage thereof) to recover the state
- In case of ISAP one only learns $K_E^\star$
- Adapted attack strategy for learning K exists [DMP20]
  - Combinations of precise faults in ISAPRK and ISAPENC during one execution
  - Quadratic amount of faulty executions

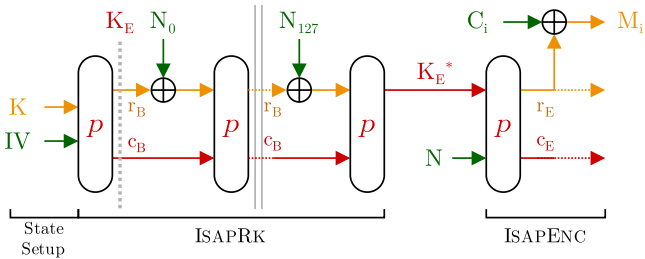
Profiling/Simple Power Analysis (SPA)

- ISAP's sponge-based mode gives SPA hardening
  - Leakage behaves similar to rate
  - Cryptographic properties ensure bounded security loss for bounded leakage
- Concrete application in [SP20]



Profiling/Simple Power Analysis (SPA)

- SPA leakage on 32-bit platform using HW acceleration for Ascon-*p* [SP20]
  - Green values are public (or leak fully)
  - Orange values create some leakage in 32-bit chunks
  - Red values result in hard to exploit leakage



Applications of ISAP Outside of LWC

- High performance AEAD in HW with protection/hardening against SCA/FI:

		MAC (0+x)				AEAD (x+0)			
Algorithm	Urol	Latency	64B	1 536B	long	Latency	64B	1 536B	long
ISAP-A-128A	1	217	5.1	1.9	1.8	378	8.5	3.0	2.8
ISAP-A-128A (StP)	1	230	5.1	1.9	1.8	392	8.5	3.0	2.8
ISAP-A-128A	2	116	2.8	1.0	1.0	199	4.6	1.8	1.6
ISAP-A-128A	4	66	1.6	0.7	0.6	109	2.7	1.1	1.1

Numbers represent cycles (for latency) or c/b (for throughput) of authentication decryption (x+0) or tag verification (0+x). Urol indicates the number of permutation rounds that are executed within one clock cycle.

Questions

Bibliography I

[BBB+20] Davide Bellizia, Francesco Berti, Olivier Bronchain, Gaëtan Cassiers, Sébastien Duval, Chun Guo, Gregor Leander, Gaëtan Leurent, Itamar Levi, Charles Momin, Olivier Pereira, Thomas Peters, François-Xavier Standaert, Balazs Udvarhelyi, and Friedrich Wiemer. **Spook: Sponge-Based Leakage-Resistant Authenticated Encryption with a Masked Tweakable Block Cipher**. IACR Trans. Symmetric Cryptol. 2020.S1 (2020), pp. 295–349. URL: https://doi.org/10.13154/tosc.v2020.iS1.295-349.

[BBC+20] Davide Bellizia, Olivier Bronchain, Gaëtan Cassiers, Vincent Grosso, Chun Guo, Charles Momin, Olivier Pereira, Thomas Peters, and François-Xavier Standaert. **Mode-Level vs. Implementation-Level Physical Security in Symmetric Cryptography - A Practical Guide Through the Leakage-Resistance Jungle**. CRYPTO (1). Vol. 12170. Lecture Notes in Computer Science. Springer, 2020, pp. 369–400.

Bibliography II

[DEM+20] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, Robert Primas, and Thomas Unterluggauer. **Isap v2.0**. IACR Transactions on Symmetric Cryptology 2020.S1 (2020), pp. 390–416. URL: https://doi.org/10.13154/tosc.v2020.iS1.390-416.

[DM19a] Christoph Dobraunig and Bart Mennink. **Leakage Resilience of the Duplex Construction**. ASIACRYPT (3). Vol. 11923. Lecture Notes in Computer Science. Springer, 2019, pp. 225–255.

[DM19b] Christoph Dobraunig and Bart Mennink. **Security of the Suffix Keyed Sponge**. IACR Transactions on Symmetric Cryptology 2019.4 (2019), pp. 223–248.

[DM21] Christoph Dobraunig and Bart Mennink. **Leakage Resilient Value Comparison with Application to Message Authentication**. EUROCRYPT (2). Vol. 12697. Lecture Notes in Computer Science. Springer, 2021, pp. 377–407.

Bibliography III

[DMP20] Christoph Dobraunig, Bart Mennink, and Robert Primas. **Leakage and Tamper Resilient Permutation-Based Cryptography**. IACR Cryptol. ePrint Arch. (2020), p. 200. URL: https://eprint.iacr.org/2020/200.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. **Leakage-Resilient Cryptography**. FOCS 2008. IEEE Computer Society, 2008, pp. 293–302. DOI: 10.1109/FOCS.2008.56. URL: https://doi.org/10.1109/FOCS.2008.56.

[SP20] Stefan Steinegger and Robert Primas. **A Fast and Compact RISC-V Accelerator for Ascon and Friends**. CARDIS. Vol. 12609. Lecture Notes in Computer Science. Springer, 2020, pp. 53–67.