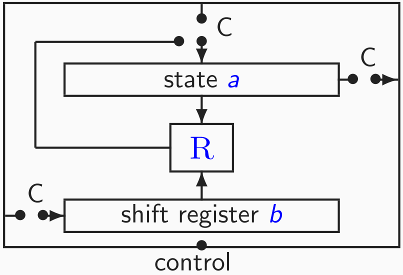
Radbound University

The Subterranean 2.0 Cipher Suite

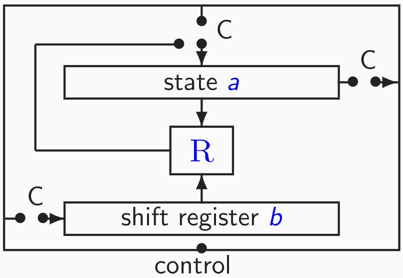
Joan Daemen[1], Pedro Maat Costa Massolino[3], Alireza Mehrdad[1], Yann Rotella[2]
[1]Radboud University NL, [3]PQShield UK, [2]UVSQ, LMV, Université Paris-Saclay FR

Fast Software Encryption Workshop
November 9, 2020

ESCADA

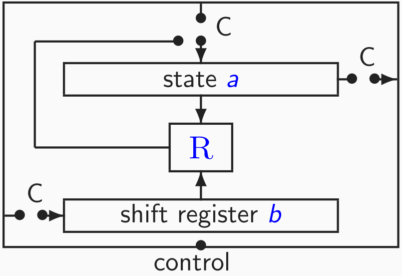

C
C
state a
R
C
shift register b
control

SUBTERRANEAN [JDA 1992]: a stream/hash module



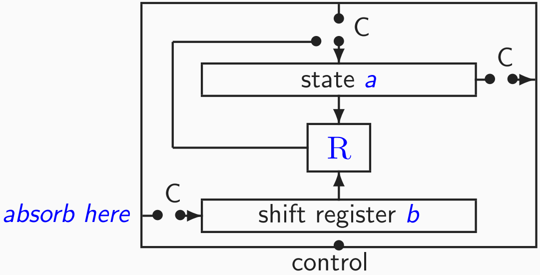
Subhash: $M \rightarrow h$

SUBTERRANEAN [JDA 1992]: a stream/hash module

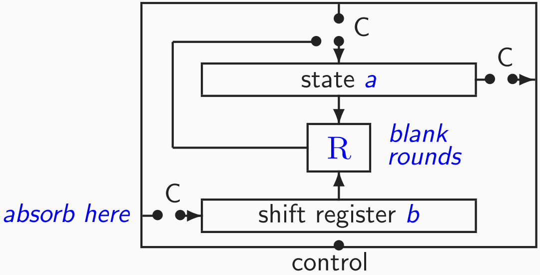


Subhash:	M	$\rightarrow$	h
Substream:	$(K; D)$	$\rightarrow$	Z

SUBTERRANEAN [JDA 1992]: a stream/hash module



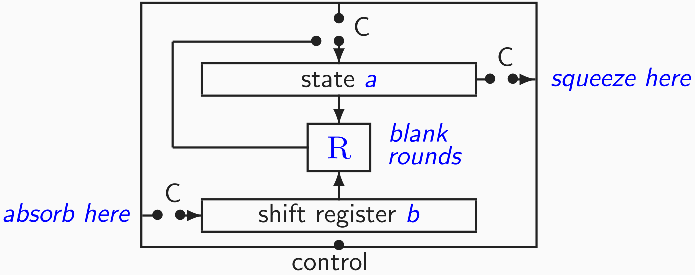
Subhash:	M	$\rightarrow$	h
Substream:	$(K; D)$	$\rightarrow$	Z



Subhash: $M \rightarrow h$

Substream: $(K; D) \rightarrow Z$

SUBTERRANEAN [JDA 1992]: a stream/hash module

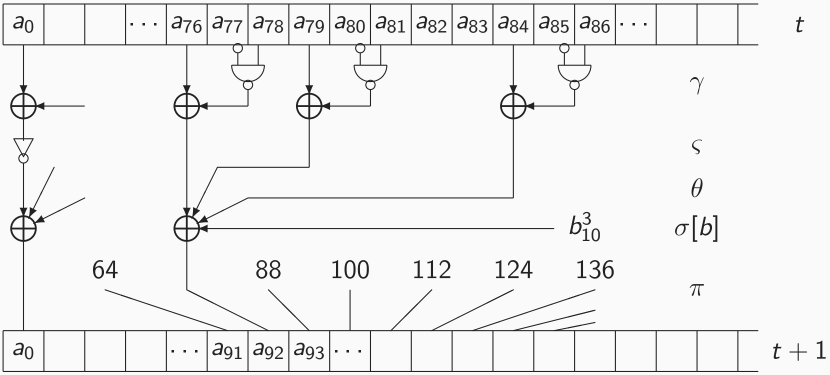


Subhash:	M	$\rightarrow$	h
Substream:	$(K;D)$	$\rightarrow$	Z

b: 256-bit shift register with 32-bit stages

b: 256-bit shift register with 32-bit stages

a: 257-bit state: $a \leftarrow \mathrm{R}(a, b)$



Could SUBTERRANEAN 1992 compete in the lightweight competition 2020?

Could SUBTERRANEAN 1992 compete in the lightweight competition 2020?

- In 1992 it was not intended as *lightweight*
  - 257-bit CV (the state)
  - compare with 128-bit CVs in MD4 and MD5

Could SUBTERRANEAN 1992 compete in the lightweight competition 2020?

- In 1992 it was not intended as *lightweight*
  - 257-bit CV (the state)
  - compare with 128-bit CVs in MD4 and MD5
- R is hardware-oriented and unsuitable for software

Could SUBTERRANEAN 1992 compete in the lightweight competition 2020?

- In 1992 it was not intended as *lightweight*
  - 257-bit CV (the state)
  - compare with 128-bit CVs in MD4 and MD5
- R is hardware-oriented and unsuitable for software
  - but we would go for *low energy* and that implies ASIC anyway

Could SUBTERRANEAN 1992 compete in the lightweight competition 2020?

- In 1992 it was not intended as *lightweight*
  - 257-bit CV (the state)
  - compare with 128-bit CVs in MD4 and MD5
- R is hardware-oriented and unsuitable for software
  - but we would go for *low energy* and that implies ASIC anyway
- Low energy?
  - R takes 4 XOR, 1 NAND, 1 NOT per bit and is *shallow*
  - absorbing: 32 bits per round $\rightarrow$ 32 XOR, 8 NAND, 8 NOT per bit
  - squeezing: 16 bits per round $\rightarrow$ 64 XOR, 16 NAND, 16 NOT per bit

Could Subterranean 1992 compete in the lightweight competition 2020?

- In 1992 it was not intended as *lightweight*
  - 257-bit CV (the state)
  - compare with 128-bit CVs in MD4 and MD5
- R is hardware-oriented and unsuitable for software
  - but we would go for *low energy* and that implies ASIC anyway
- Low energy?
  - R takes 4 XOR, 1 NAND, 1 NOT per bit and is *shallow*
  - absorbing: 32 bits per round $\rightarrow$ 32 XOR, 8 NAND, 8 NOT per bit
  - squeezing: 16 bits per round $\rightarrow$ 64 XOR, 16 NAND, 16 NOT per bit
- Not bad, so let's give it a shot!

Three primitives

XOF: unkeyed hashing with arbitrary-length output & input strings
Deck: keyed function with arbitrary-length output & input strings
SAE: session-supporting nonce-based authentication encryption

Three primitives

XOF: unkeyed hashing with arbitrary-length output & input strings

Deck: keyed function with arbitrary-length output & input strings

SAE: session-supporting nonce-based authentication encryption

Refactoring into two levels

Subterranean 2.0 is Subterranean 1992 refurbished

Three primitives

XOF: unkeyed hashing with arbitrary-length output & input strings
Deck: keyed function with arbitrary-length output & input strings
SAE: session-supporting nonce-based authentication encryption

Refactoring into two levels

- Duplex

- Mode

Subterranean 2.0 is Subterranean 1992 refurbished

Three primitives

XOF: unkeyed hashing with arbitrary-length output & input strings

Deck: keyed function with arbitrary-length output & input strings

SAE: session-supporting nonce-based authentication encryption

Refactoring into two levels

- Duplex
  - $r = 32$ in squeezing and keyed absorbing
- Mode

SUBTERRANEAN 2.0 is SUBTERRANEAN 1992 refurbished

Three primitives

XOF: unkeyed hashing with arbitrary-length output & input strings
Deck: keyed function with arbitrary-length output & input strings
SAE: session-supporting nonce-based authentication encryption

Refactoring into two levels

- Duplex
  - $r = 32$ in squeezing and keyed absorbing
  - $r = 8$ per 2 rounds in unkeyed absorbing (for 112 bits of security)
- Mode

SUBTERRANEAN 2.0 is SUBTERRANEAN 1992 refurbished

Three primitives

XOF: unkeyed hashing with arbitrary-length output & input strings
Deck: keyed function with arbitrary-length output & input strings
SAE: session-supporting nonce-based authentication encryption

Refactoring into two levels

- Duplex
  - $r = 32$ in squeezing and keyed absorbing
  - $r = 8$ per 2 rounds in unkeyed absorbing (for 112 bits of security)
  - delete shift register b and just absorb in, and squeeze from a
- Mode

SUBTERRANEAN 2.0 is SUBTERRANEAN 1992 refurbished

Three primitives

XOF: unkeyed hashing with arbitrary-length output & input strings
Deck: keyed function with arbitrary-length output & input strings
SAE: session-supporting nonce-based authentication encryption

Refactoring into two levels

- Duplex
  - $r = 32$ in squeezing and keyed absorbing
  - $r = 8$ per 2 rounds in unkeyed absorbing (for 112 bits of security)
  - delete shift register b and just absorb in, and squeeze from a
- Mode
  - 8 *blank* rounds between absorbing and squeezing

Subterranean 2.0 is Subterranean 1992 refurbished

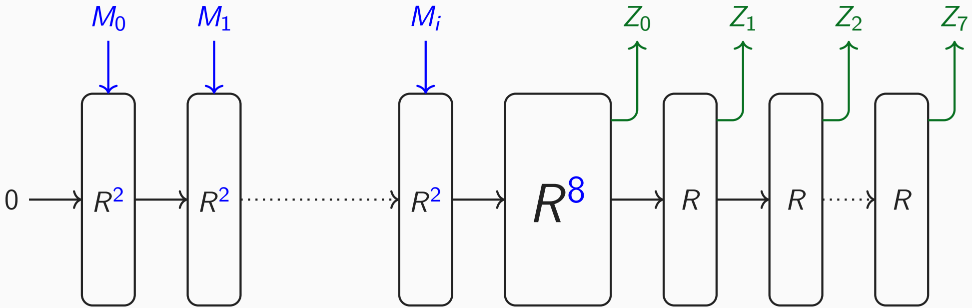
Three primitives

XOF: unkeyed hashing with arbitrary-length output & input strings
Deck: keyed function with arbitrary-length output & input strings
SAE: session-supporting nonce-based authentication encryption

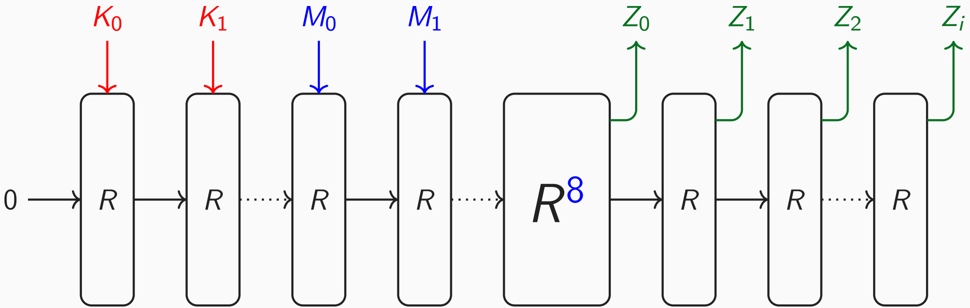
Refactoring into two levels

- Duplex
  - $r = 32$ in squeezing and keyed absorbing
  - $r = 8$ per 2 rounds in unkeyed absorbing (for 112 bits of security)
  - delete shift register b and just absorb in, and squeeze from a
- Mode
  - 8 *blank* rounds between absorbing and squeezing
  - except for encryption/decryption in SAE that relies on nonce uniqueness

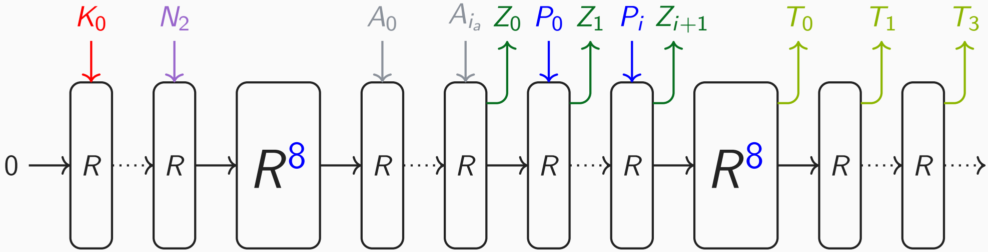
And now to SUBTERRANEAN 2.0 and its rationale in more detail!



- $|M_j|$: **one byte**
- $|Z_j|$: **4 bytes**

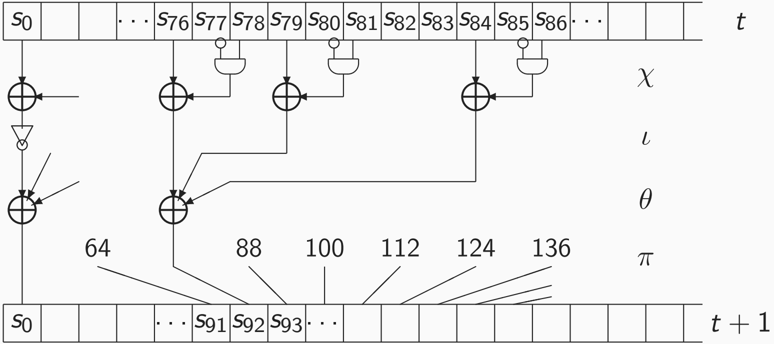


- $|M_j|$, $|Z_j|$, $|K_j|$: **4 bytes**



- $|K_j|$, $|N_j|$, $|A_j|$, $|Z_j|$, $|P_j|$, $|T_j|$: **4 bytes**

The SUBTERRANEAN 2.0 round function



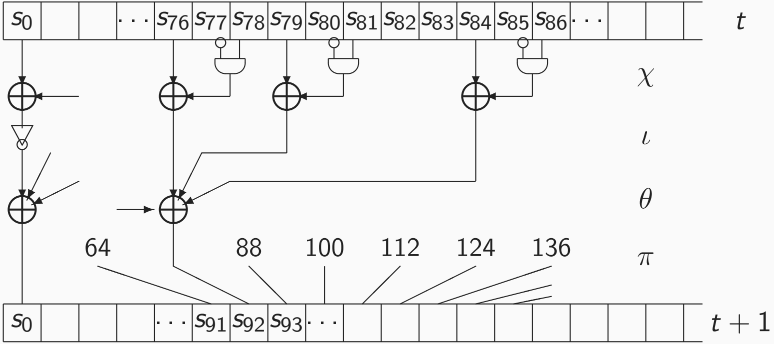
$$\chi: \quad s_i \leftarrow \quad s_i + (s_{i+1} + 1)s_{i+2}$$

$$\iota: \quad s_i \leftarrow \quad s_i + \delta_i$$

$$\theta: \quad s_i \leftarrow \quad s_i + s_{i+3} + s_{i+8}$$

$$\pi: \quad s_i \leftarrow \quad s_{12i}$$

Absorb and Squeeze



$$12^4 = 176$$

$$\mathcal{G}_{64} = \{1, 176, 136, \ldots, 92\} \prec \mathbb{Z}/257\mathbb{Z}^*$$

$$z_i = s_{176^i} + s_{176^{-i}}$$

$$s_{176^i} = s_{176^i} + p_i$$

The choice of $\mathcal{G}_{64}$:

- non-consecutive bits (State-Recovery attacks on Ketje Jr [Fuhr, Naya-Plasencia, Rotella, ToSC 2018])
- consistent with π dispersion

Design Rationale in a nutshell

The choice of $\mathcal{G}_{64}$:

- non-consecutive bits (State-Recovery attacks on Ketje Jr [Fuhr, Naya-Plasencia, Rotella, ToSC 2018])
- consistent with π dispersion

The number of rounds:

- Separator: 8 blank rounds
- Unkeyed mode: 2 rounds ($8 + 1$ bits absorbed)
- Keyed mode: 1 round ($32 + 1$ bits absorbed)

Third Party Cryptanalysis

Fukang Liu, Takanori Isobe and Willi Meier, Cube-Based Cryptanalysis of Subterranean-SAE, ToSC 2020

- key recovery from Subterranean-SAE in nonce-misuse scenario
- reduced-round scenario: 4 blank rounds out of 8

Third Party Cryptanalysis

Fukang Liu, Takanori Isobe and Willi Meier, Cube-Based Cryptanalysis of SUBTERRANEAN-SAE, ToSC 2020

- key recovery from SUBTERRANEAN-SAE in nonce-misuse scenario
- reduced-round scenario: 4 blank rounds out of 8

Ling Song, Yi Tu, Danping Shi and Lei Hu, Security Analysis of SUBTERRANEAN 2.0, eprint 2020, report 1133

- size-reduced versions
- no observable biases
- nonce-misuse scenario

Third Party Cryptanalysis

Fukang Liu, Takanori Isobe and Willi Meier, Cube-Based Cryptanalysis of SUBTERRANEAN-SAE, ToSC 2020

- key recovery from SUBTERRANEAN-SAE in nonce-misuse scenario
- reduced-round scenario: 4 blank rounds out of 8

Ling Song, Yi Tu, Danping Shi and Lei Hu, Security Analysis of SUBTERRANEAN 2.0, eprint 2020, report 1133

- size-reduced versions
- no observable biases
- nonce-misuse scenario

More work is welcome

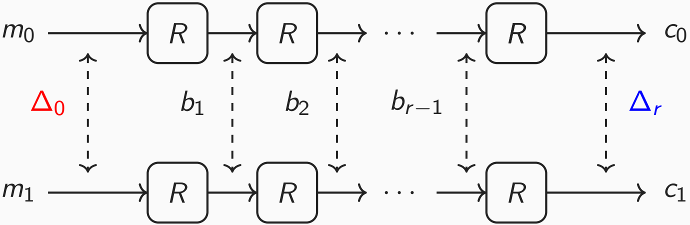
Difference propagation

- Security: $\max \mathrm{DP}(\Delta_0 \to \Delta_r)$

Difference propagation

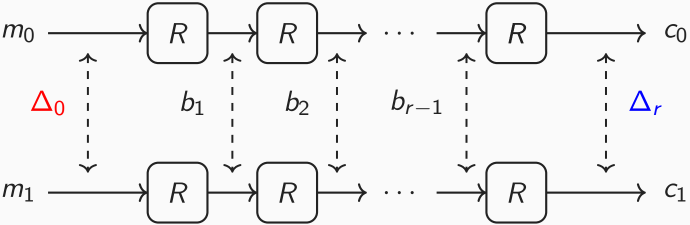
- Security: $\max \mathrm{DP}(\Delta_0 \to \Delta_r)$
  It is hard to determine

Difference propagation



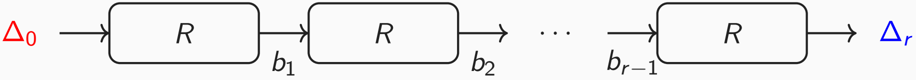
- Security: $\max \mathrm{DP}(\Delta_0 \to \Delta_r)$
  It is hard to determine
- $\max \mathrm{DP}(\Delta_0 \to \Delta_r) \approx \max_{Q_r} \mathrm{DP}(Q_r)$
  - Q_r is a differential trail
  - $\Delta_0 \to b_1 \to b_2 \to \cdots \to b_{r-1} \to \Delta_r$

Difference propagation

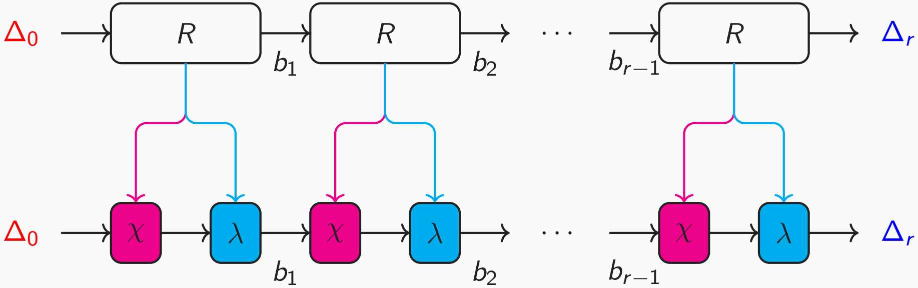


- Security: $\max \mathrm{DP}(\Delta_0 \to \Delta_r)$
  It is hard to determine
- $\max \mathrm{DP}(\Delta_0 \to \Delta_r) \approx \max_{Q_r} \mathrm{DP}(Q_r)$
  - Q_r is a differential trail
  - $\Delta_0 \to b_1 \to b_2 \to \cdots \to b_{r-1} \to \Delta_r$
- Trail weight: $w(Q) = -\log_2(\mathrm{DP})$

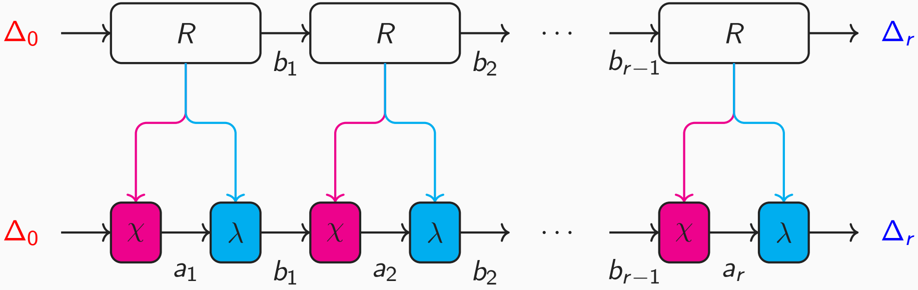
Differential trail core



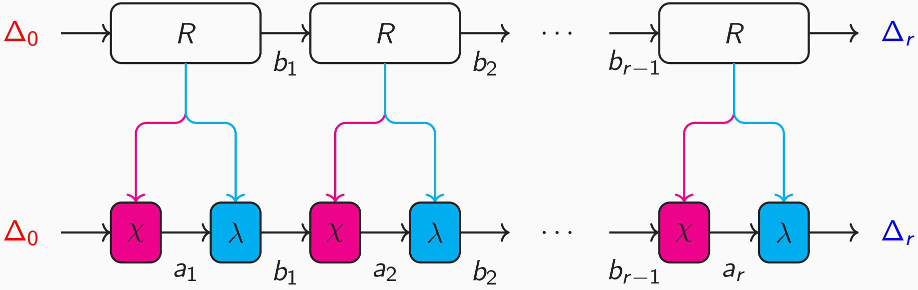
Differential trail core



Differential trail core

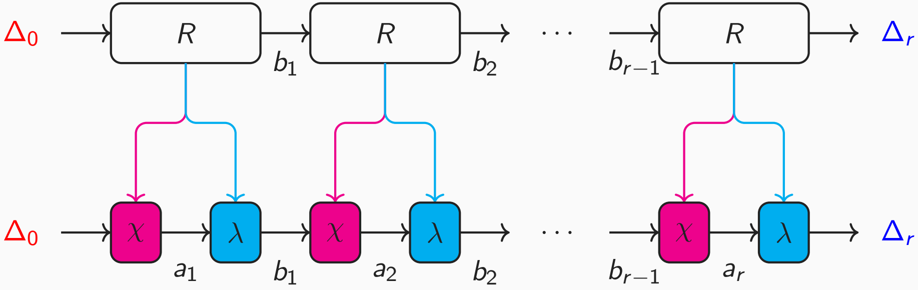


Differential trail core



$$w(Q_r) = w(\Delta_0 \to a_1) + \sum_{i=1}^{r-1} w(b_i \to a_{i+1})$$

Differential trail core



$$w(Q_r) = w(\Delta_0 \to a_1) + \sum_{i=1}^{r-1} w(b_i \to a_{i+1}) = \min w^{-1}(a_1) + \sum_{i=1}^{r-1} w(b_i)$$

Lower bound on the weight of differential trail cores

# rounds:	1	2	3	4	5	6	7	8
lower bound:	?	?	?	?	?	?	?	?

Lower bound on the weight of differential trail cores

# rounds:	1	2	3	4	5	6	7	8
lower bound:	2	8	?	?	?	?	?	?

Lower bound on the weight of differential trail cores

# rounds:	1	2	3	4	5	6	7	8
lower bound:	2	8	?	?	?	?	?	?

- We generated all 3-round trails cores up to weight 39
  The same method as introduced in [Mella, Daemen, Van Assche, ToSC 2016]

Lower bound on the weight of differential trail cores

# rounds:	1	2	3	4	5	6	7	8
lower bound:	2	8	25	?	?	?	?	?

- We generated all 3-round trails cores up to weight 39
  The same method as introduced in [Mella, Daemen, Van Assche, ToSC 2016]

weight	25	28	29	30	32	33	34	35	36	37	38	39
# trail cores (mod *rotation*)	1	1	2	3	2	1	5	6	4	9	12	17

Lower bound on the weight of differential trail cores

# rounds:	1	2	3	4	5	6	7	8
lower bound:	2	8	25	?	?	?	?	?

- We generated all 3-round trails cores up to weight 39
  The same method as introduced in [Mella, Daemen, Van Assche, ToSC 2016]

weight	25	28	29	30	32	33	34	35	36	37	38	39
# trail cores (mod *rotation*)	1	1	2	3	2	1	5	6	4	9	12	17

- 3-round trail core with the lowest weight

state	**weight**	**# active bits**	**active bit positions**
a_1	2	1	$\{0\}$
b_1	6	3	$\{0, 64, 85\}$
b_2	17	9	$\{0, 64, 85, 91, 155, 157, 176, 221, 242\}$

Lower bound for 4-round differential trail cores

- We searched the space of all 4-round trail cores up to weight 48

Lower bound for 4-round differential trail cores

- We searched the space of all 4-round trail cores up to weight 48
  - there are no trail cores with weight 48 or less

Lower bound for 4-round differential trail cores

- We searched the space of all 4-round trail cores up to weight 48
  - there are no trail cores with weight 48 or less
  - we did find 4-round trail core with weight 58

Lower bound for 4-round differential trail cores

- We searched the space of all 4-round trail cores up to weight 48
  - there are no trail cores with weight 48 or less
  - we did find 4-round trail core with weight 58
  - so $49 \leq \min w(Q_4) \leq 58$

Lower bound for 4-round differential trail cores

- We searched the space of all 4-round trail cores up to weight 48
  - there are no trail cores with weight 48 or less
  - we did find 4-round trail core with weight 58
  - so $49 \leq \min w(Q_4) \leq 58$
- The 4-round trail core with weight 58:

state	weight	# active bits	active bit positions
a_1	12	9	$\{0, 5, 8, 10, 12, 15, 16, 18, 21\}$
b_1	7	5	$\{65, 66, 85, 86, 87\}$
b_2	11	6	$\{7, 28, 134, 198, 200, 219\}$
b_3	28	15	$\{16, 18, 22, 39, 54, 86, 88, 107, 118, 139, 152, 173, 188, 211, 252\}$

Lower bounds on differential trails

# rounds:	1	2	3	4	5	6	7	8
lower bound:	2	8	25	$[49, 58]$	?	?	?	?

Lower bounds on differential trails

# rounds:	1	2	3	4	5	6	7	8
lower bound:	2	8	25	$[49, 58]$	?	?	?	?

- An 8-round trail Q_8 can be divided into two 4-round trails $Q_4 \mid Q'_4$

Lower bounds on differential trails

# rounds:	1	2	3	4	5	6	7	8
lower bound:	2	8	25	$[49, 58]$	?	?	?	?

- An 8-round trail Q_8 can be divided into two 4-round trails $Q_4 \mid Q'_4$
- If $w(Q_8) \leq (2 \times 48) + 1 = 97$ then $w(Q_4) \leq 48$ or $w(Q'_4) \leq 48$

Lower bounds on differential trails

# rounds:	1	2	3	4	5	6	7	8
lower bound:	2	8	25	$[49, 58]$	?	?	?	≥ 98

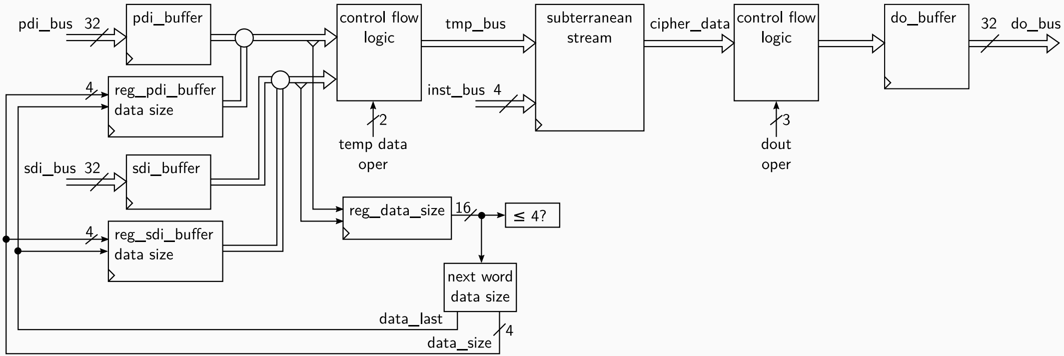
- An 8-round trail Q_8 can be divided into two 4-round trails $Q_4 \mid Q'_4$
- If $w(Q_8) \leq (2 \times 48) + 1 = 97$ then $w(Q_4) \leq 48$ or $w(Q'_4) \leq 48$

Lower bounds on differential trails

# rounds:	1	2	3	4	5	6	7	8
lower bound:	2	8	25	$[49, 58]$	≥ 54	≥ 65	≥ 70	≥ 98

- An 8-round trail Q_8 can be divided into two 4-round trails $Q_4 \mid Q'_4$
- If $w(Q_8) \leq (2 \times 48) + 1 = 97$ then $w(Q_4) \leq 48$ or $w(Q'_4) \leq 48$
- Different methods to find the lower bound on the weight of other trails

Hardware LWC architecture



- Streaming based architecture - high throughput
- Separate buffers for public and secret data in (PDI/SDI)
- Flow controlled by main state machine

FPGA Results

Mohajerani et al. "FPGA Benchmarking of Round 2 Candidates in the NIST Lightweight Cryptography Standardization Process: Methodology, Metrics, Tools, and Results". https://eprint.iacr.org/2020/1207

- 1st AEAD throughput for messages of 64 bytes or more in Artix 7
- 6th Hash throughput for long messages in Artix 7

AEAD	Throughput	LUT
Subterranean 2.0	6 Gbps	915
Xoodyak	3 Gbps	2040

Hash	Throughput	LUT
Gimli	1.9 Gbps	1900
Xoodyak	1.8 Gbps	2040
Saturnin	1.6 Gbps	2414
DryGascon	1.5 Gbps	2074
Ascon	987 Mbps	1723
Subterranean 2.0	744 Mbps	915

ASIC Results

Khairallah et al. "Preliminary Hardware Benchmarking of a Group of Round 2 NIST Lightweight AEAD Candidates".
`https://github.com/mustafam001/lwc-aead-rtl`

- AEAD for ASIC cells TSMC TSBN 65nm 9-track
- 1st in Throughput and Energy
- Results for 64 bytes messages:

AEAD	Throughput	Area (GE)	Energy (pJ)	Clock period (ns)
Subterranean 2.0	17 Gbps	7050	16	0.47
Romulus	8 Gbps	14218	44	0.88
Xoodyak	12 Gbps	17898	51	0.50

Conclusion

Subterranean 2.0 in a nutshell:

Conclusion

SUBTERRANEAN 2.0 in a nutshell:

- Target security strength
  - 128 bits for keyed modes: Deck and SAE
  - 112 bits for unkeyed mode: XOF

Conclusion

SUBTERRANEAN 2.0 in a nutshell:

- Target security strength
  - 128 bits for keyed modes: Deck and SAE
  - 112 bits for unkeyed mode: XOF
- Safety margin is comfortable, per our analysis and two 3rd-party papers

Conclusion

SUBTERRANEAN 2.0 in a nutshell:

- Target security strength
  - 128 bits for keyed modes: Deck and SAE
  - 112 bits for unkeyed mode: XOF
- Safety margin is comfortable, per our analysis and two 3rd-party papers
  - more 3rd party cryptanalysis is welcome!

Conclusion

SUBTERRANEAN 2.0 in a nutshell:

- Target security strength
  - 128 bits for keyed modes: Deck and SAE
  - 112 bits for unkeyed mode: XOF
- Safety margin is comfortable, per our analysis and two 3rd-party papers
  - more 3rd party cryptanalysis is welcome!
- Lightweight

Conclusion

SUBTERRANEAN 2.0 in a nutshell:

- Target security strength
  - 128 bits for keyed modes: Deck and SAE
  - 112 bits for unkeyed mode: XOF
- Safety margin is comfortable, per our analysis and two 3rd-party papers
  - more 3rd party cryptanalysis is welcome!
- Lightweight
  - total storage in SAE and XOF: 257-bit state and some 32-bit I/O buffers

Conclusion

SUBTERRANEAN 2.0 in a nutshell:

- Target security strength
  - 128 bits for keyed modes: Deck and SAE
  - 112 bits for unkeyed mode: XOF
- Safety margin is comfortable, per our analysis and two 3rd-party papers
  - more 3rd party cryptanalysis is welcome!
- Lightweight
  - total storage in SAE and XOF: 257-bit state and some 32-bit I/O buffers
  - # operations per absorbed/squeezed bit very low

Conclusion

SUBTERRANEAN 2.0 in a nutshell:

- Target security strength
  - 128 bits for keyed modes: Deck and SAE
  - 112 bits for unkeyed mode: XOF
- Safety margin is comfortable, per our analysis and two 3rd-party papers
  - more 3rd party cryptanalysis is welcome!
- Lightweight
  - total storage in SAE and XOF: 257-bit state and some 32-bit I/O buffers
  - # operations per absorbed/squeezed bit very low
  - especially non-linear operations → suitable for masking

Conclusion

SUBTERRANEAN 2.0 in a nutshell:

- Target security strength
  - 128 bits for keyed modes: Deck and SAE
  - 112 bits for unkeyed mode: XOF
- Safety margin is comfortable, per our analysis and two 3rd-party papers
  - more 3rd party cryptanalysis is welcome!
- Lightweight
  - total storage in SAE and XOF: 257-bit state and some 32-bit I/O buffers
  - # operations per absorbed/squeezed bit very low
  - especially non-linear operations $\rightarrow$ suitable for masking
  - confirmed by benchmarks

Conclusion

SUBTERRANEAN 2.0 in a nutshell:

- Target security strength
  - 128 bits for keyed modes: Deck and SAE
  - 112 bits for unkeyed mode: XOF
- Safety margin is comfortable, per our analysis and two 3rd-party papers
  - more 3rd party cryptanalysis is welcome!
- Lightweight
  - total storage in SAE and XOF: 257-bit state and some 32-bit I/O buffers
  - # operations per absorbed/squeezed bit very low
  - especially non-linear operations → suitable for masking
  - confirmed by benchmarks

Thanks for your attention!