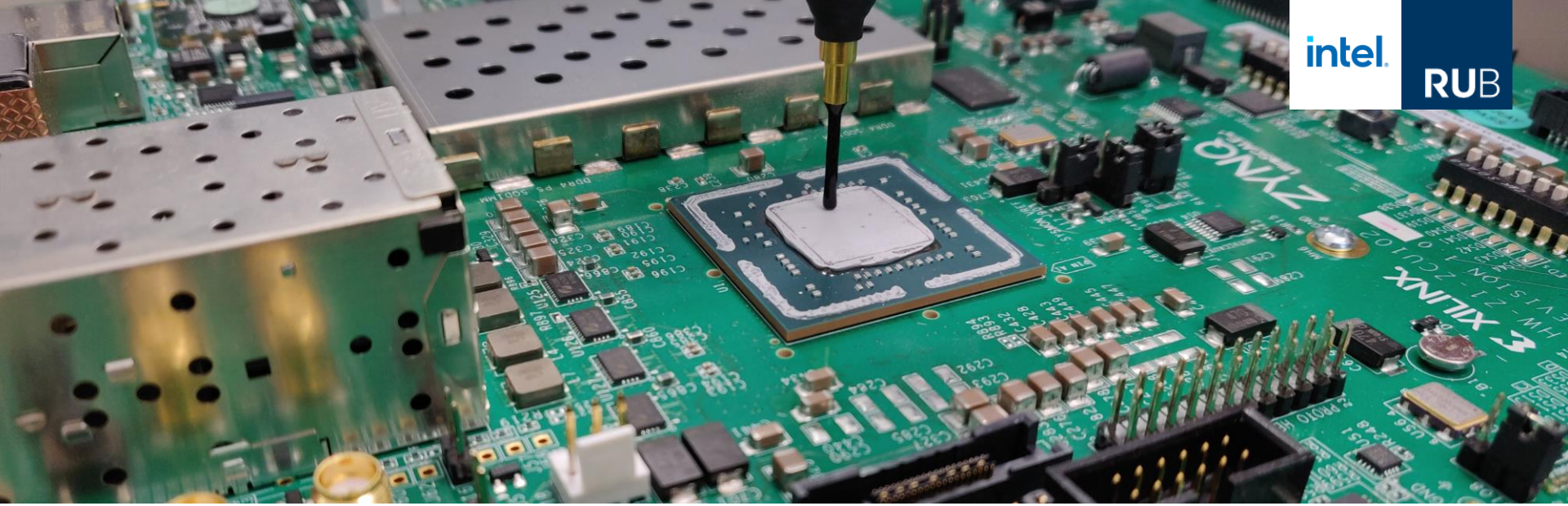


RUHR-UNIVERSITÄT BOCHUM

HADES – Automated HArdware Design Exploration for Efficient and Secure Cryptographic Primitives

Fabian Buschkowski, Georg Land, Niklas Höher, Jan-Richter Brockmann, Pascal Sasdrich, Tim Güneysu

CHES 2025
September 15, 2025

Chair for Security Engineering
Faculty of Computer Science
Ruhr University Bochum

SECURITY ENGINEERING

Motivation

Efficient Cryptography

- Efficiency relative to a certain performance metric (e.g., latency, area, delay, …)

Motivation

Efficient Cryptography

- Efficiency relative to a certain performance metric (e.g., latency, area, delay, …)

4-bit function **S**

Motivation

Efficient Cryptography

- Efficiency relative to a certain performance metric (e.g., latency, area, delay, …)

4-bit function **S**

8-bit state

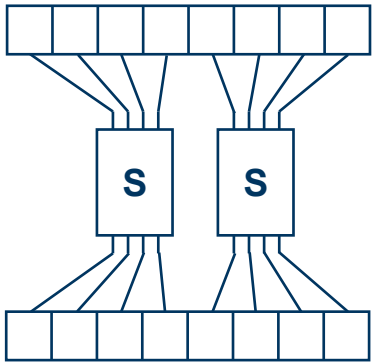
Motivation

Efficient Cryptography

- Efficiency relative to a certain performance metric (e.g., latency, area, delay, …)

4-bit function **S**

8-bit state



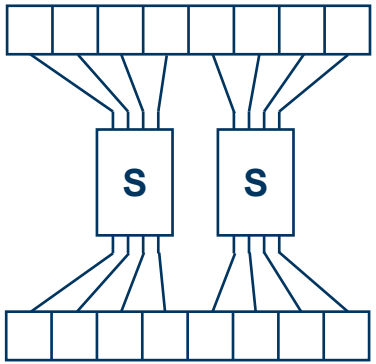
Motivation

Efficient Cryptography

- Efficiency relative to a certain performance metric (e.g., latency, area, delay, …)

4-bit function **S**

8-bit state



\+ Faster
\- More area

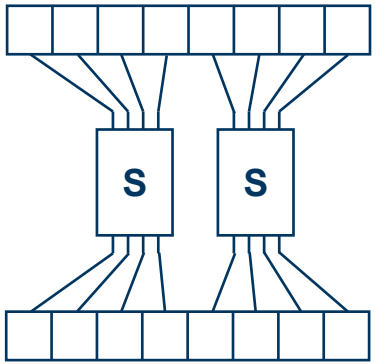
Motivation

Efficient Cryptography

- Efficiency relative to a certain performance metric (e.g., latency, area, delay, …)

4-bit function **S**

8-bit state

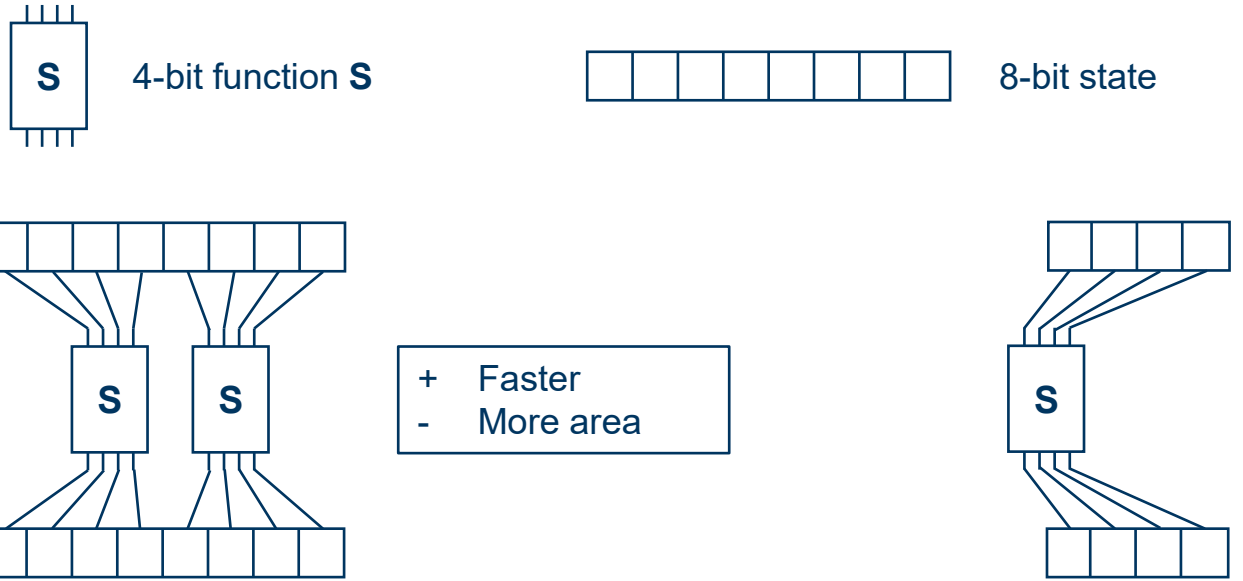


\+ Faster
\- More area

Motivation

Efficient Cryptography

- Efficiency relative to a certain performance metric (e.g., latency, area, delay, …)



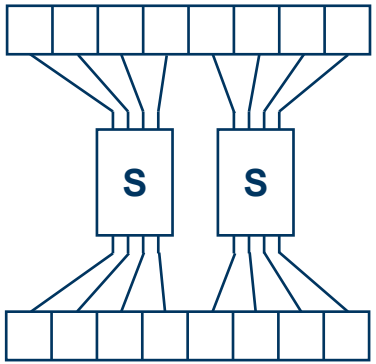
Motivation

Efficient Cryptography

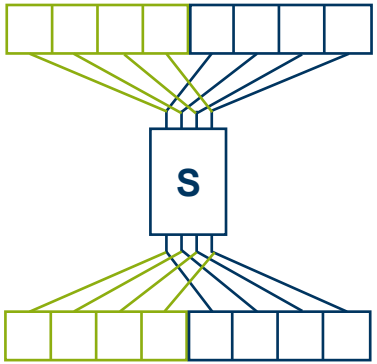
- Efficiency relative to a certain performance metric (e.g., latency, area, delay, …)

4-bit function **S**

8-bit state



\+ Faster
\- More area



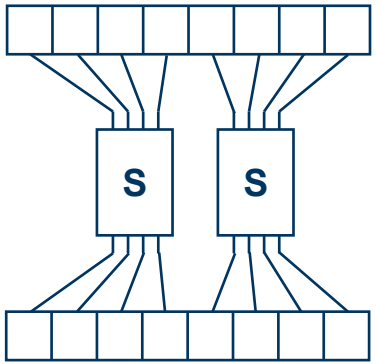
Motivation

Efficient Cryptography

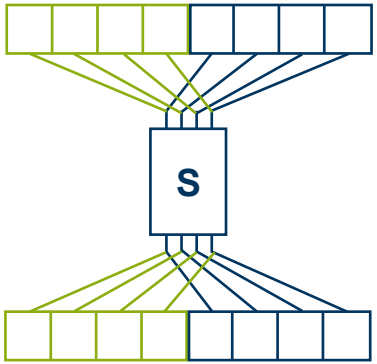
- Efficiency relative to a certain performance metric (e.g., latency, area, delay, …)

4-bit function **S**

8-bit state



+ Faster
- More area



+ Less area
- Slower

Motivation

Secure Cryptography

Secure cryptographic algorithms running on a chip are always secure!

Motivation

Secure Cryptography

Secure cryptographic al… …n a chip are always secure!

Physical Attacks

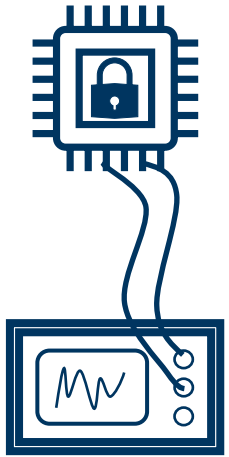
Motivation

Secure Cryptography

Secure cryptographic al… …n a chip are always secure!

Physical Attacks

Side-Channel Attacks



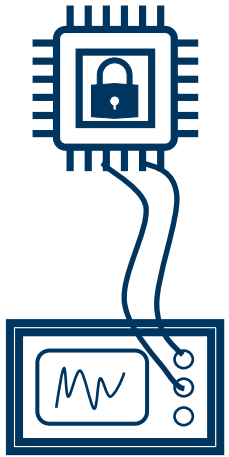
Motivation

Secure Cryptography

Secure cryptographic al… …n a chip are always secure!

Physical Attacks

Side-Channel Attacks



Masking

$$x = x_0 \oplus x_1 \oplus \ ... \oplus x_d$$

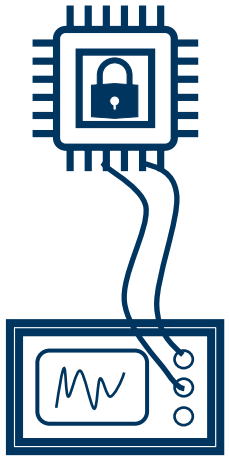
Motivation

Secure Cryptography

Secure cryptographic al… …n a chip are always secure!

Physical Attacks

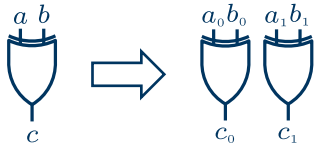
Side-Channel Attacks



Masking

$$x = x_0 \oplus x_1 \oplus \ldots \oplus x_d$$

Linear Functions



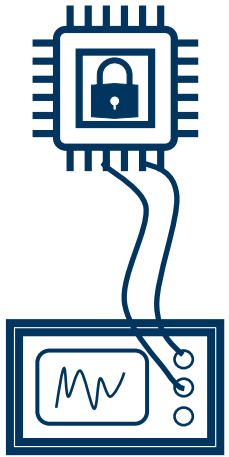
Motivation

Secure Cryptography

Secure cryptographic al… …n a chip are always secure!

Physical Attacks

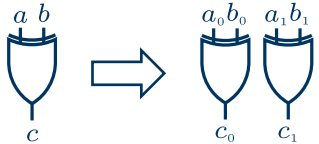
Side-Channel Attacks



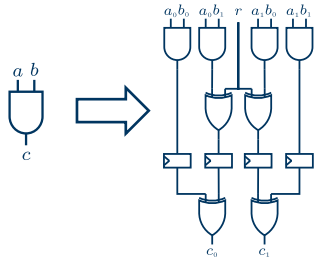
Masking

$$x = x_0 \oplus x_1 \oplus \ldots \oplus x_d$$

Linear Functions



Non-linear Functions



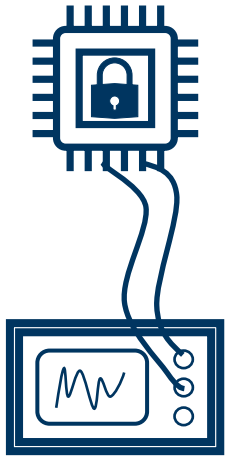
Motivation

Secure Cryptography

Secure cryptographic al… …n a chip are always secure!

Physical Attacks

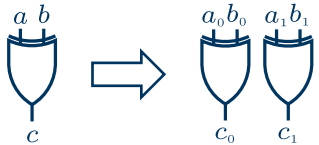
Side-Channel Attacks



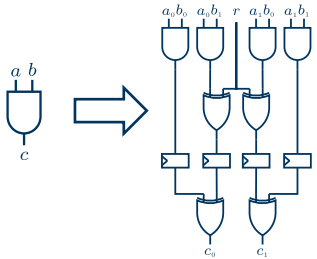
Masking

$$x = x_0 \oplus x_1 \oplus \ldots \oplus x_d$$

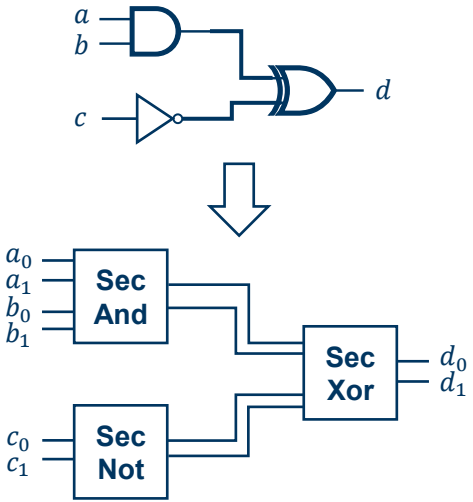
Linear Functions



Non-linear Functions



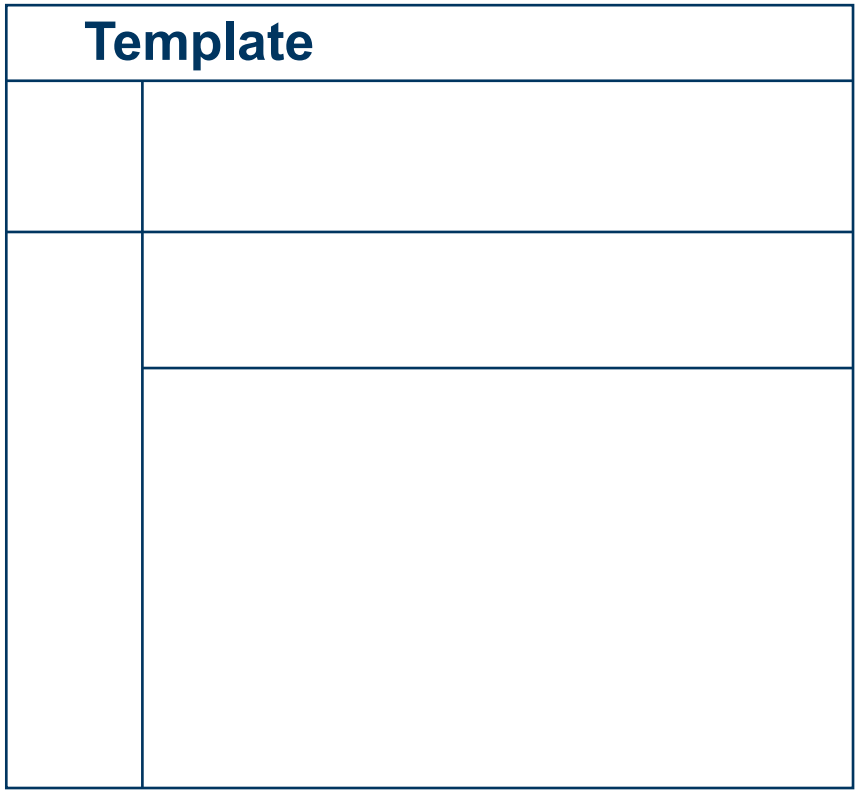
Gadget-based Masking



HADES

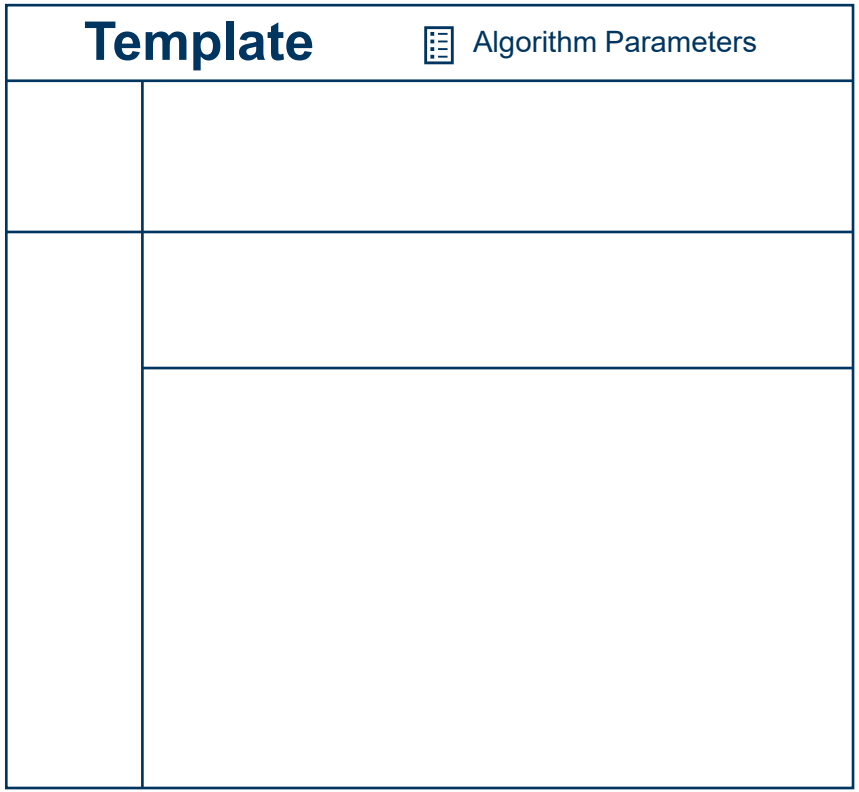
HADES

Templates



HADES

Templates

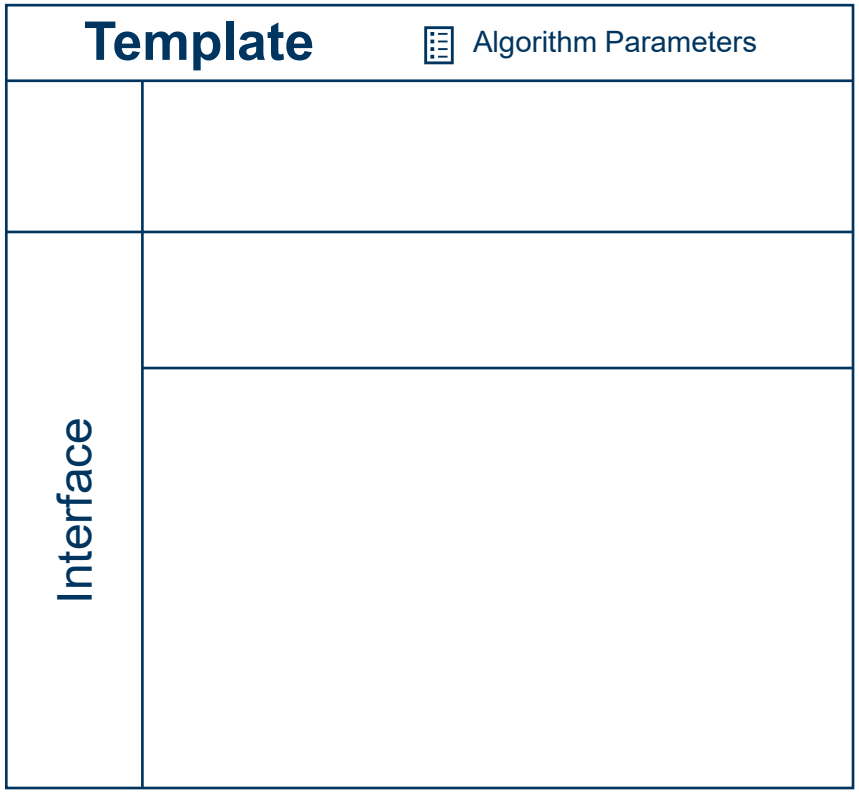


Algorithm Parameters

- Adder width
- AES key size

HADES

Templates

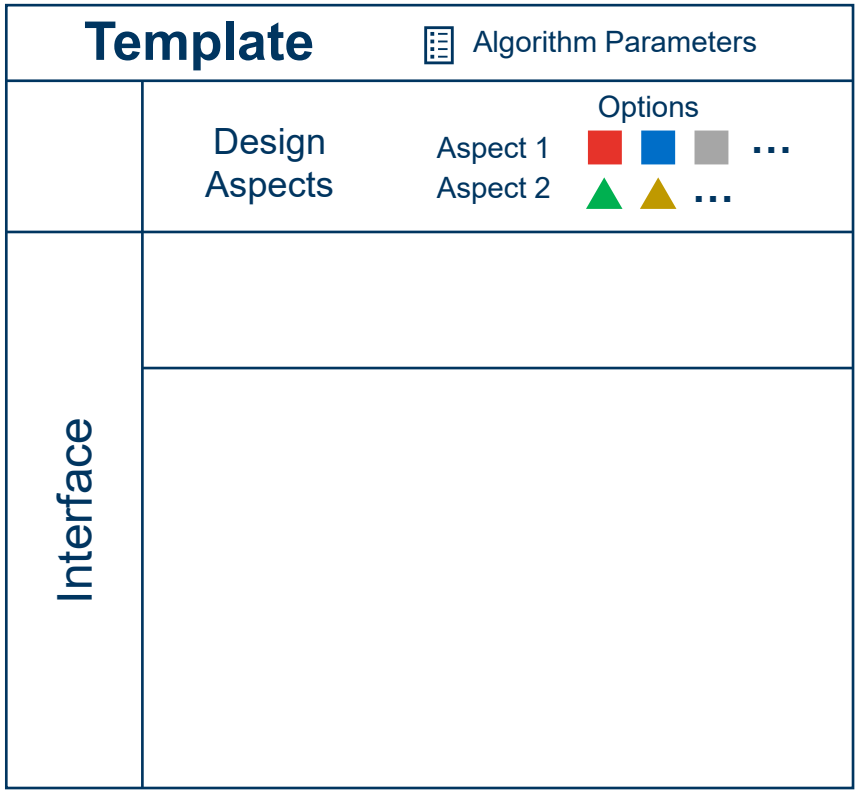


Algorithm Parameters
- Adder width
- AES key size

Interface
- Input and output signals
- Control signals

HADES

Templates

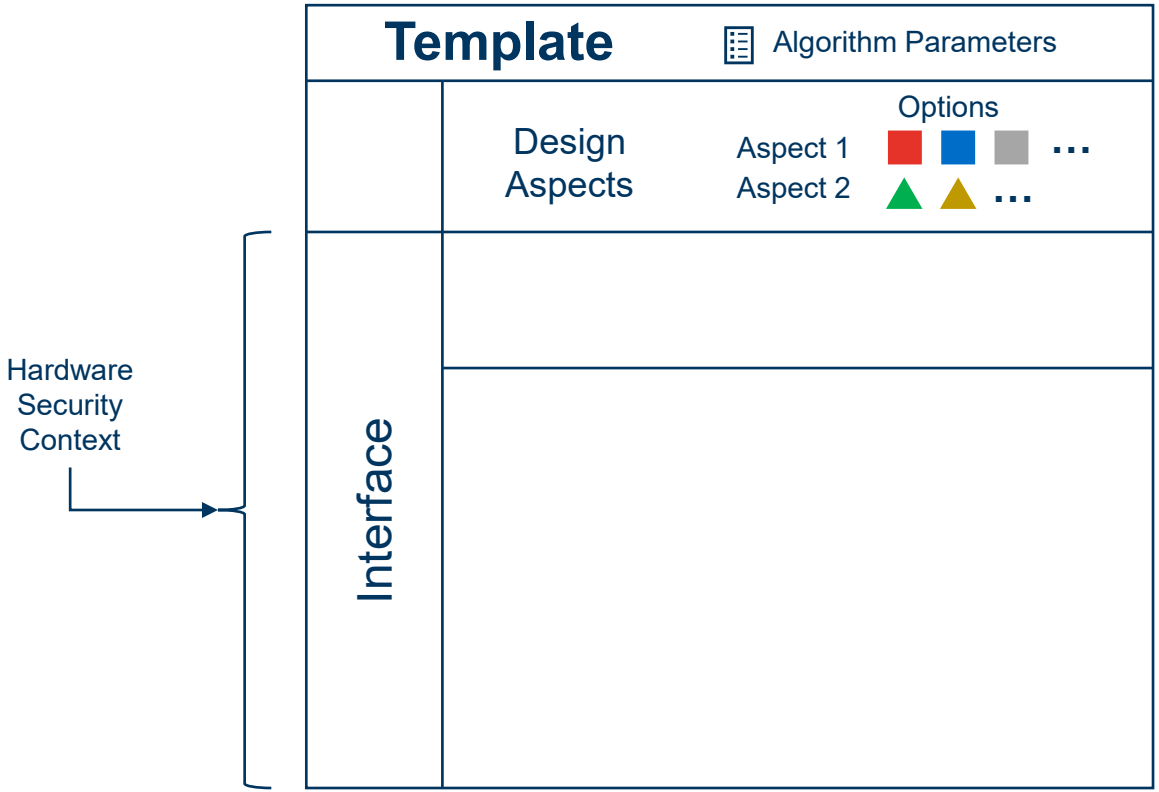


Design Aspects

- Do not change external template functionality
- Adder type
- SBox type
- Parallelism

HADES

Templates



Design Aspects

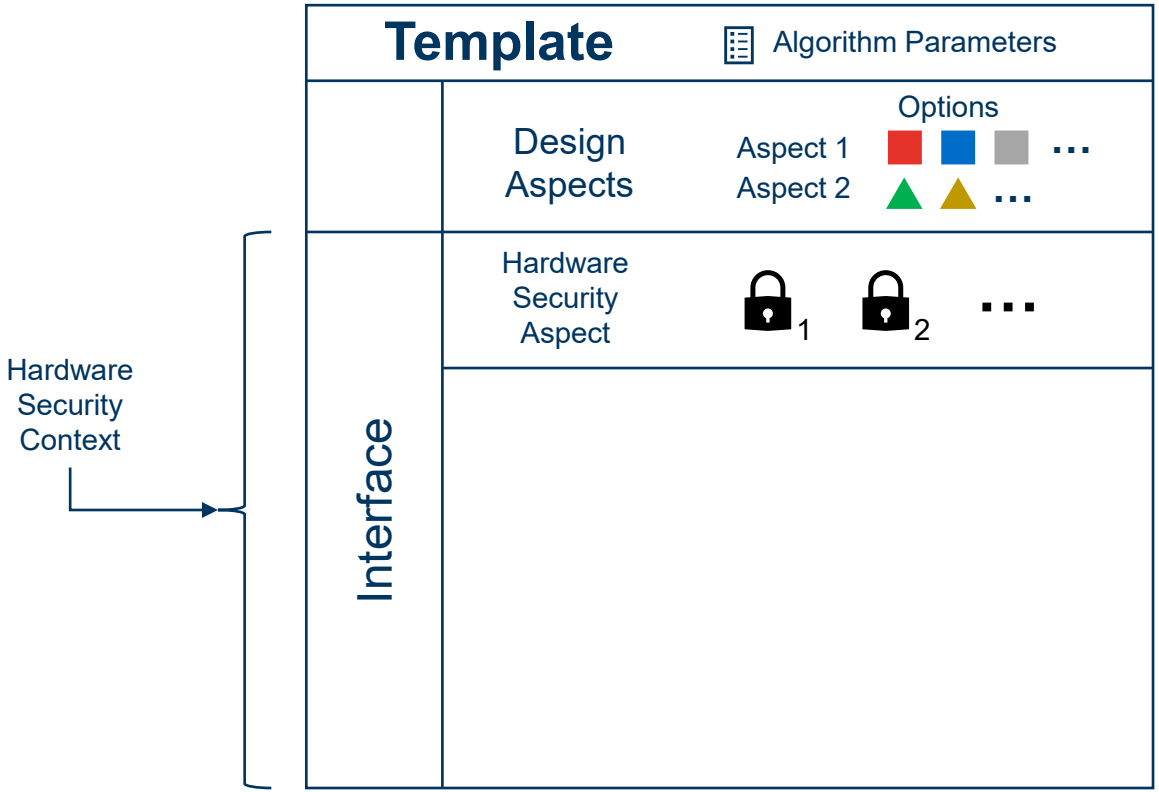
- Do not change external template functionality
- Adder type
- SBox type
- Parallelism

Hardware Security Context

- Defines security goals
- Masking degree

HADES

Templates



Design Aspects

- Do not change external template functionality
- Adder type
- SBox type
- Parallelism

Hardware Security Context

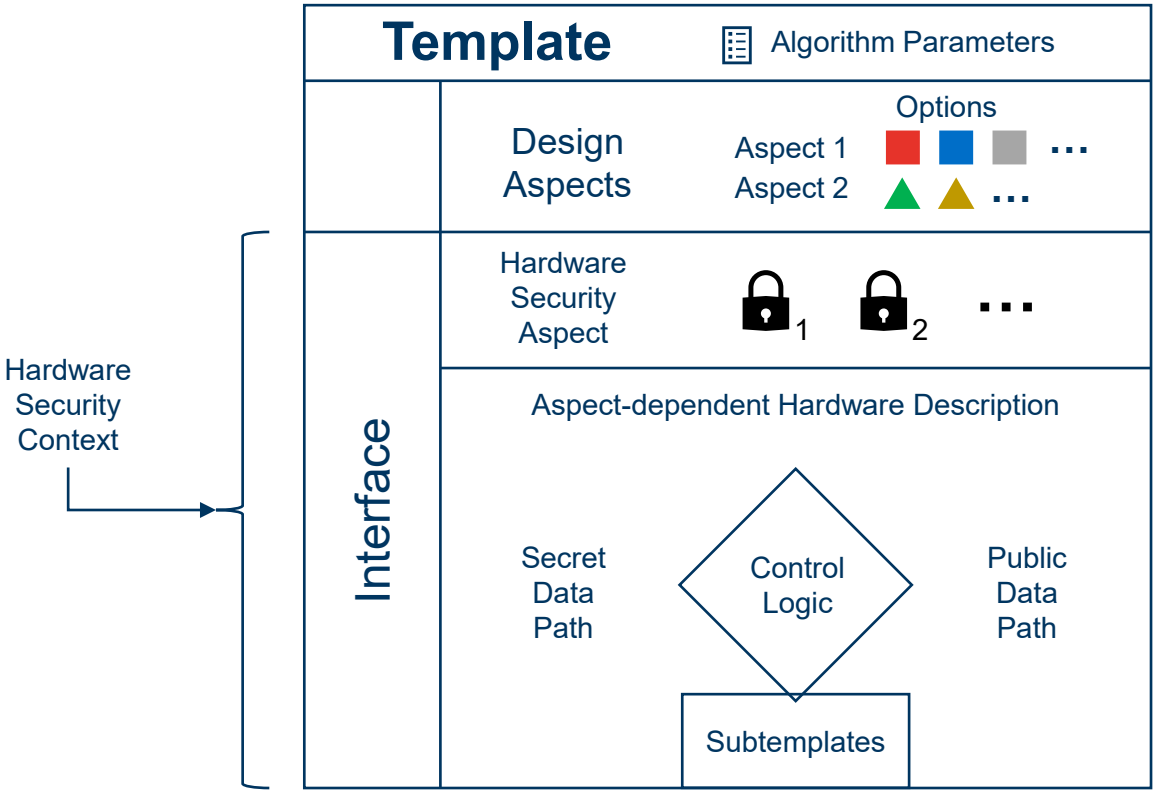
- Defines security goals
- Masking degree

Hardware Security Aspects

- Different options to achieve desired security
- Gadget type (HPC2 vs. HPC3)

HADES

Templates



Hardware Description

- Calculation of outputs based on inputs (using sub-templates)
- Relative to design aspects
- Secret data path (with protection)
- Public data path

HADES

Workflow

HADES

Workflow

HADES

Workflow

Template Library

HADES

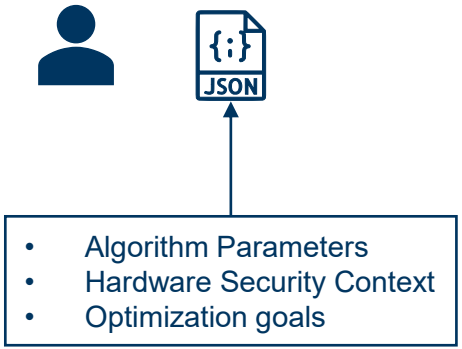
Workflow

Template Library

HADES

Workflow

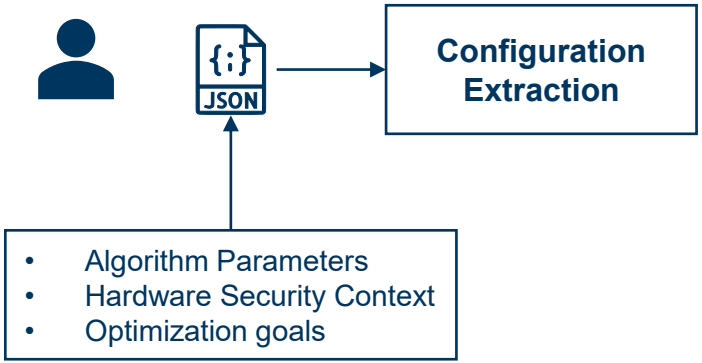
Template Library



HADES

Workflow

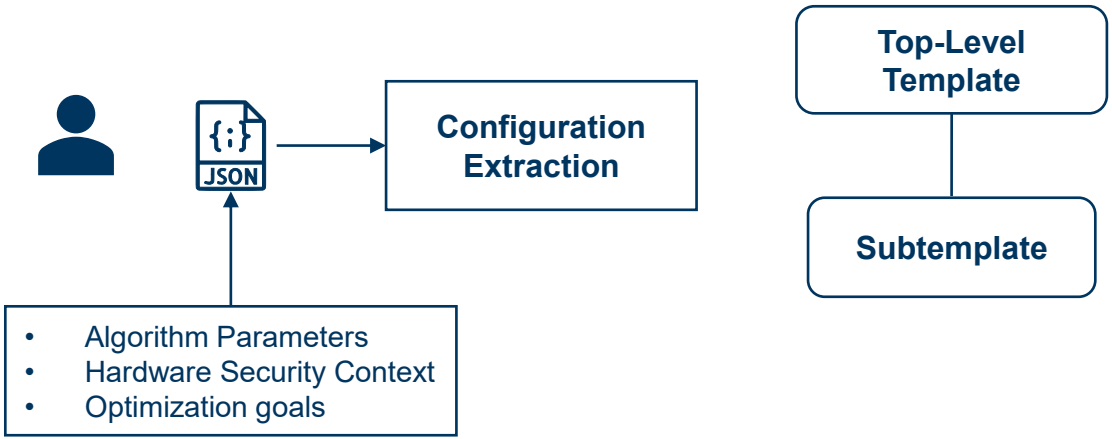
Template Library



HADES

Workflow

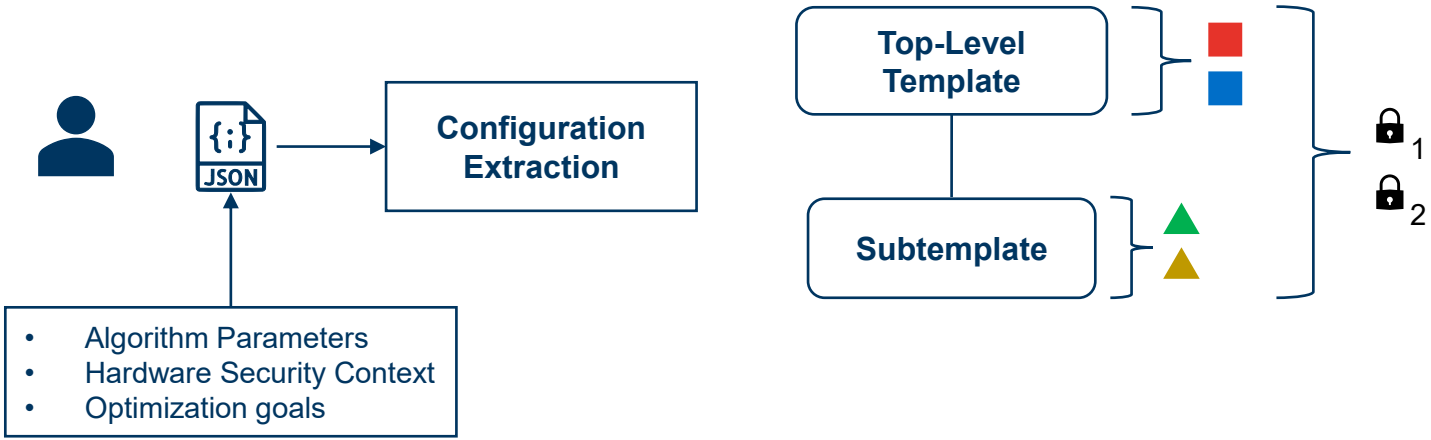
Template Library



HADES

Workflow

Template Library

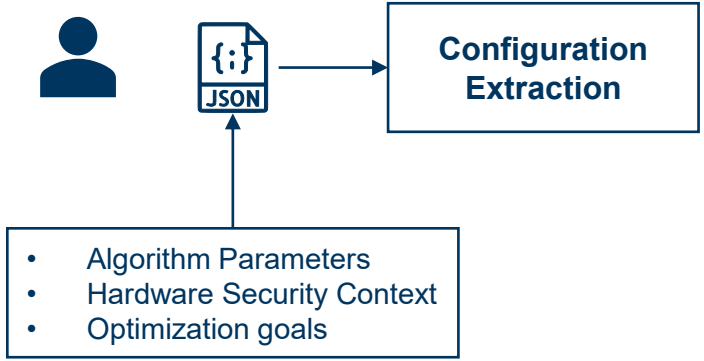


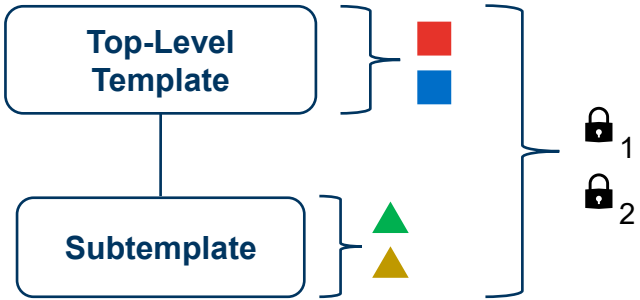
HADES

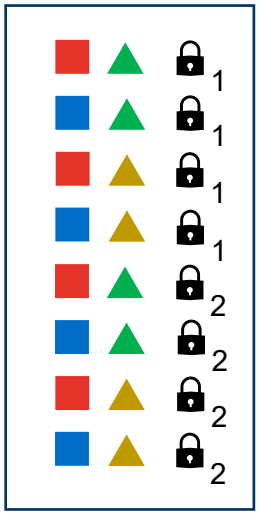
Workflow

Template Library

List of configurations



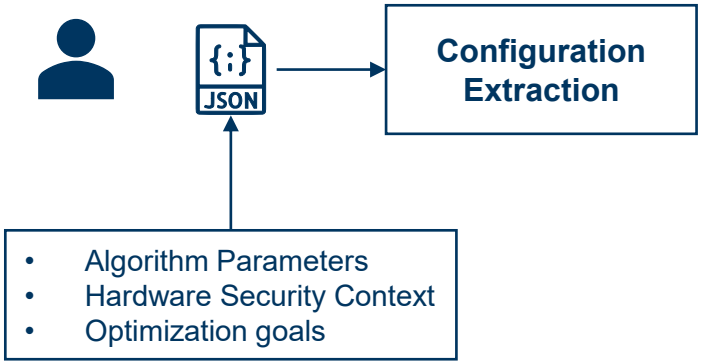




HADES

Workflow

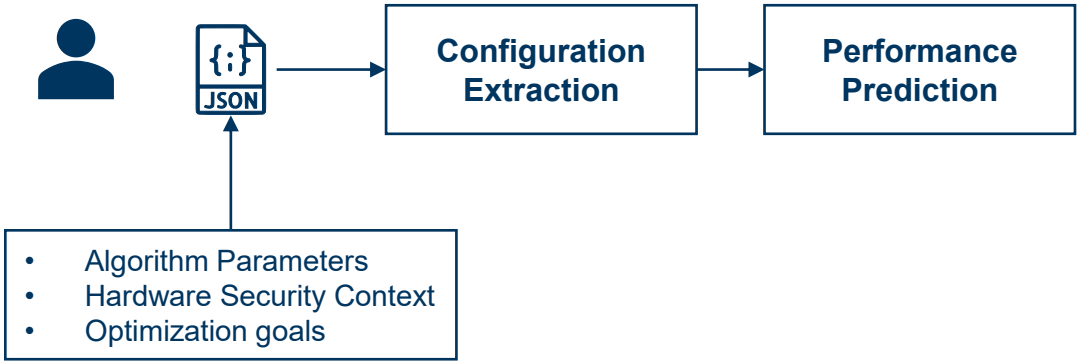
Template Library



HADES

Workflow

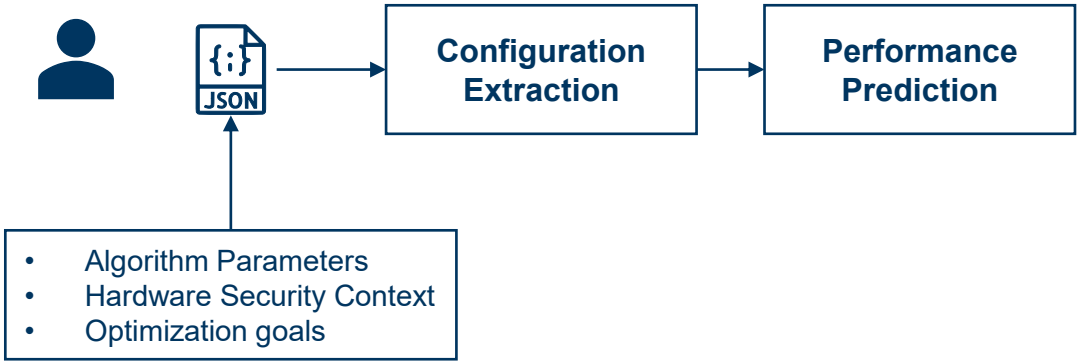
Template Library

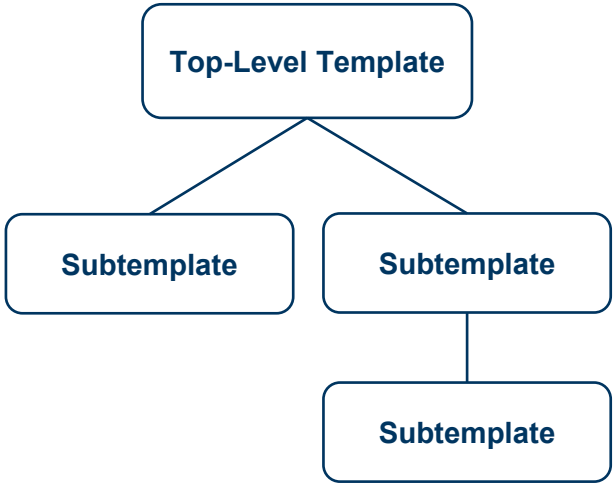


HADES

Workflow

Template Library

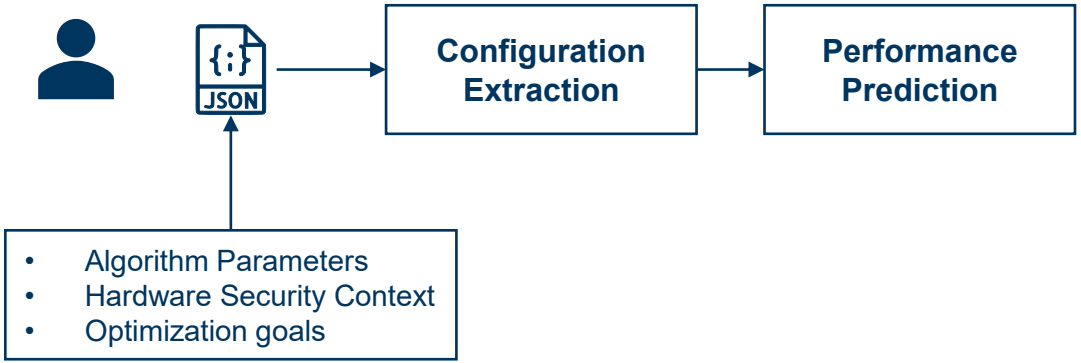


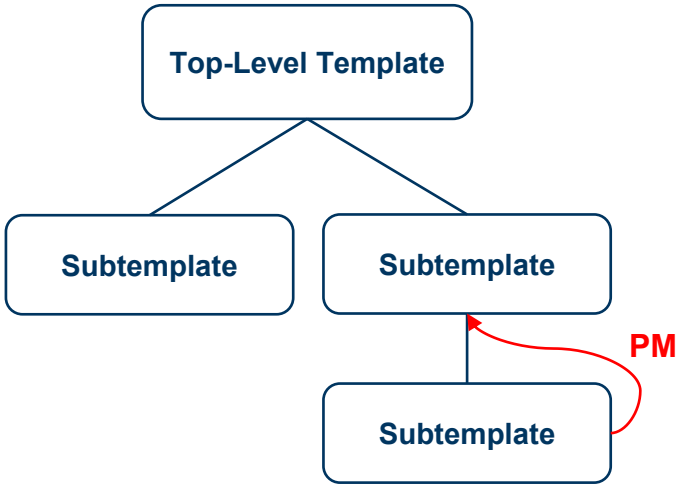


HADES

Workflow

Template Library

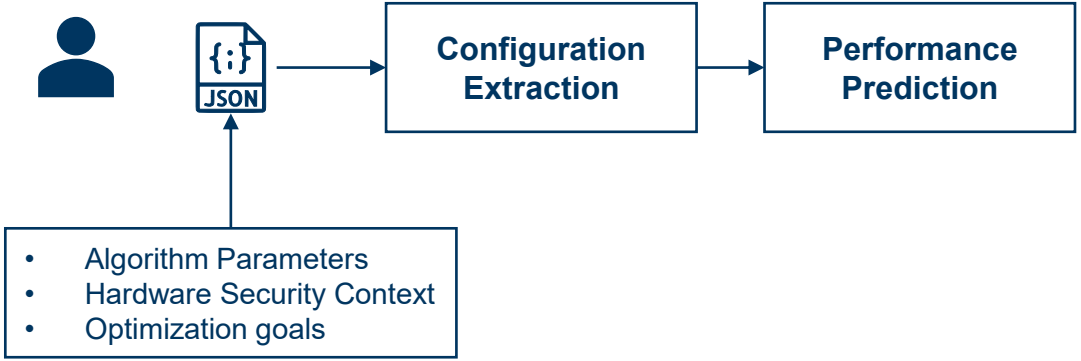


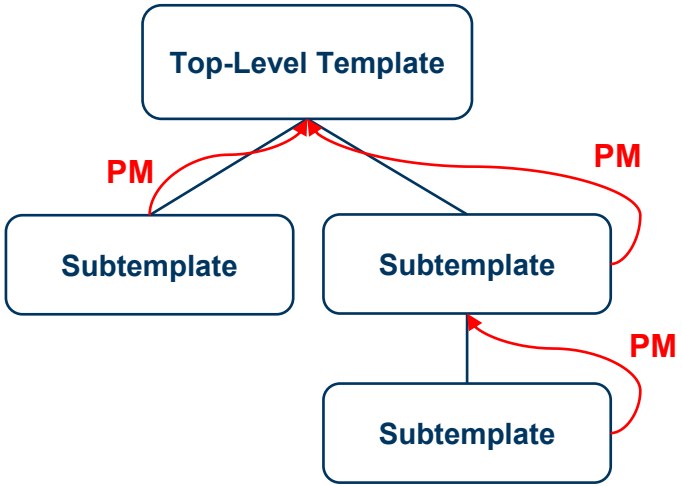


HADES

Workflow

Template Library





HADES

Workflow

Template Library

Configuration Extraction

Performance Prediction

- Algorithm Parameters
- Hardware Security Context
- Optimization goals

Overall Performance Prediction

Top-Level Template

PM

PM

Subtemplate

Subtemplate

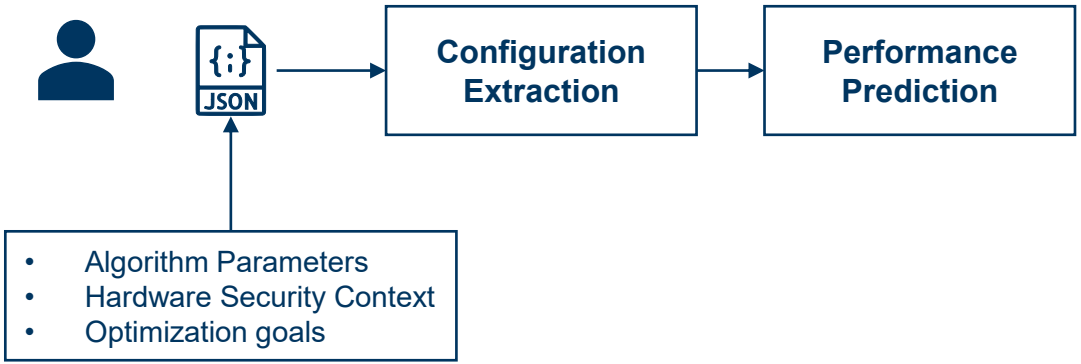
PM

Subtemplate

HADES

Workflow

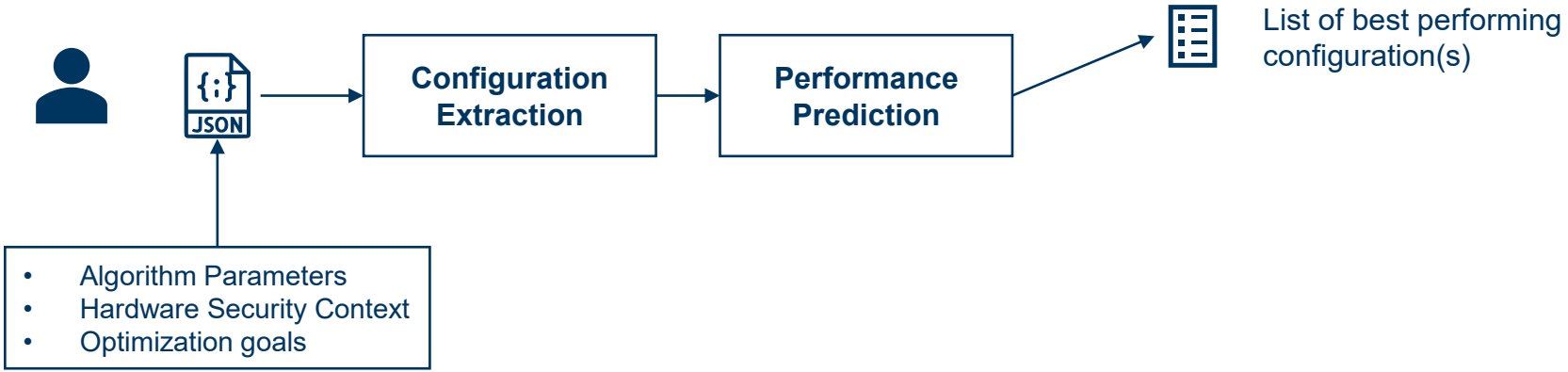
Template Library



HADES

Workflow

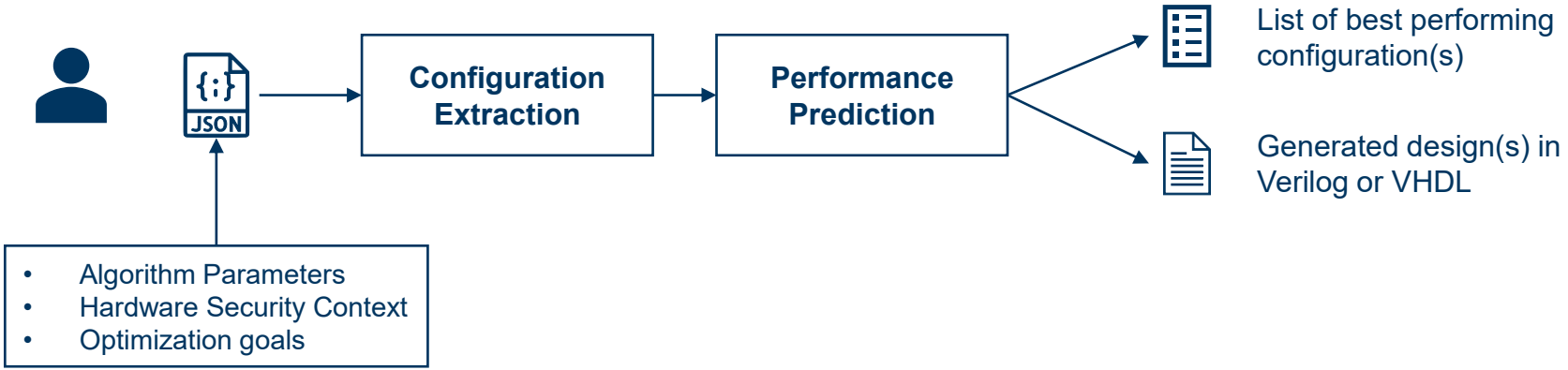
Template Library



HADES

Workflow

Template Library



Case Studies

Case Studies

Overview

Adder	Adder mod q	ChaCha20
AES	Keccak	Sparse Polynomial Multiplication
Polynomial Multiplication	Kyber-CPA	ML-KEM

Case Studies

Overview

Adder	Adder mod q	ChaCha20
AES	Keccak	Sparse Polynomial Multiplication
Polynomial Multiplication	Kyber-CPA	ML-KEM

Case Study: AES

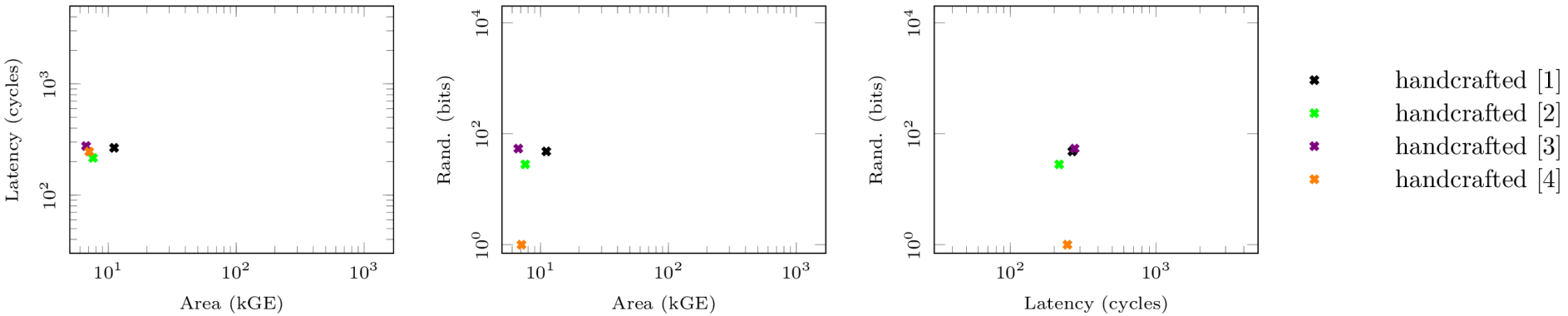
Design Space

architecture	S-box type	# of S-boxes in key schedule	# of S-boxes in round function	# of MC instances	gadget
- round-based - unrolled	- Canright - Boyar-Peralta	- 1 - 2 - 4 - 8 (AES 256 only)	- use from KS - 1 - 2 - 4 - 8 - 16	- 1 - 2 - 4	- HPC2 - HPC3

4,032 possible designs explored in ~10 seconds

Case Study: First-order Masked AES-128

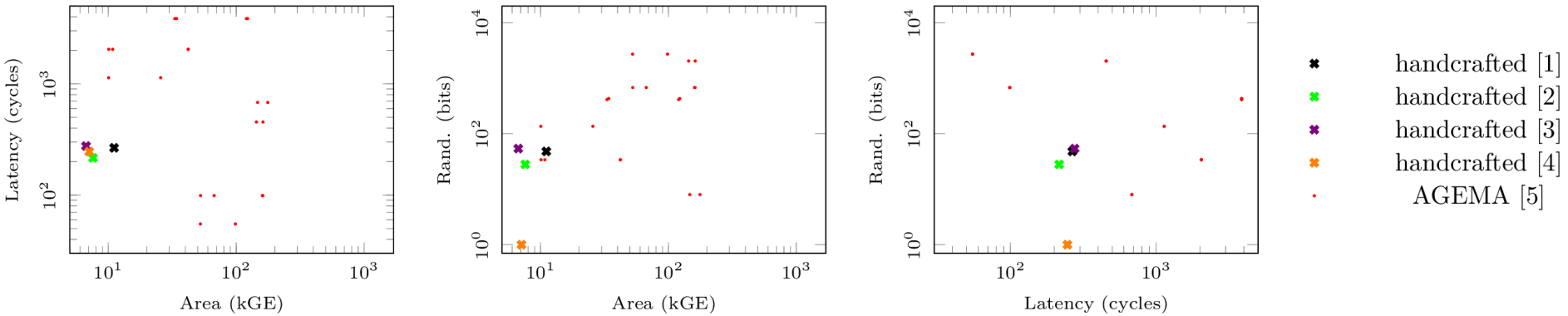
Comparison



[1] Pushing the limits: A very compact and a threshold implementation of AES. Moradi et al., Eurocrypt ‘11
[2] Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order. Groß et al., TIS ‘16
[3] Masking AES with d+1 Shares in Hardware. De Cnudde et al., CCS ‘16
[4] Re-consolidating first-order Masking Schemes. Shahmirzadi and Moradi, CHES ’21

Case Study: First-order Masked AES-128

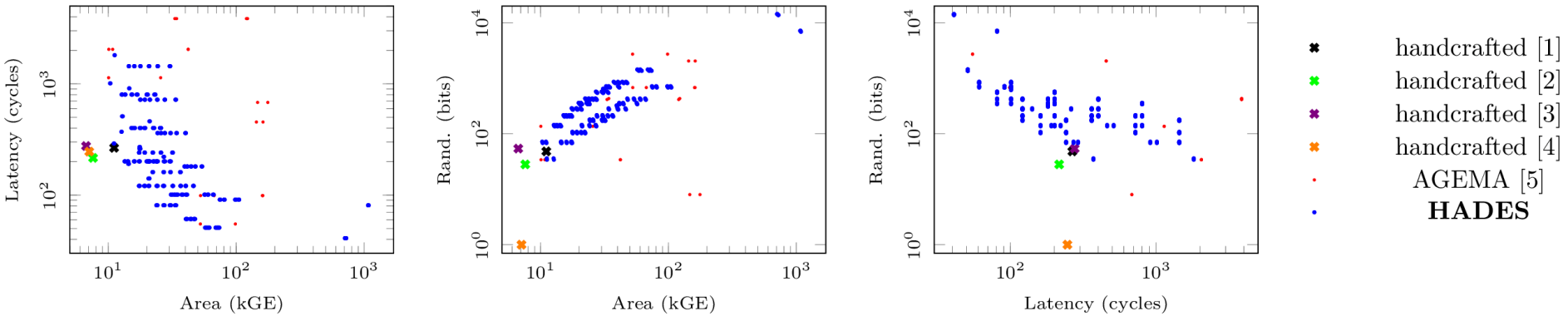
Comparison



[1] Pushing the limits: A very compact and a threshold implementation of AES. Moradi et al., Eurocrypt ‘11
[2] Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order. Groß et al., TIS ‘16
[3] Masking AES with d+1 Shares in Hardware. De Cnudde et al., CCS ‘16
[4] Re-consolidating first-order Masking Schemes. Shahmirzadi and Moradi, CHES ’21
[5] Automated Generation of Masked Hardware. Knichel et al., CHES ‘22

Case Study: First-order Masked AES-128

Comparison



[1] Pushing the limits: A very compact and a threshold implementation of AES. Moradi et al., Eurocrypt ‘11
[2] Domain-Oriented Masking: Compact Masked Hardware Implementations with Arbitrary Protection Order. Groß et al., TIS ‘16
[3] Masking AES with d+1 Shares in Hardware. De Cnudde et al., CCS ‘16
[4] Re-consolidating first-order Masking Schemes. Shahmirzadi and Moradi, CHES ’21
[5] Automated Generation of Masked Hardware. Knichel et al., CHES ‘22

Case Study: ML-KEM Decapsulation

Basic Idea

- **Boolean-masked Schoolbook multiplication to enable gadget-based masking (Land et al., CHES ‘24):**
  1) compute all potential results of public-times-secret multiplication
  2) multiplex the correct one (masked)
  3) accumulate the result to perform polynomial multiplication (masked)
- **Inverse NTT on $\hat{A}$**
- **FO transform: “decompressed comparison” idea by Bos et al., CHES ‘21**

Case Study: ML-KEM Decaps

Design Space

# of single-coefficient multipliers	adder type (3 different)	# of χ step modules in Keccak	# of Compress_1 modules	gadget
all numbers $i \in [1, 256]$ such that $\left\lceil \frac{256}{i} \right\rceil$ is unique and appears for the first time (31 total options)	- serial ripple-carry - pipelined ripple-carry - Sklansky - Kogge-Stone	- 1 - 2 - 4 - 8 - 16 - 32 - 64	all numbers $i \in [1, 256]$ such that $\left\lceil \frac{256}{i} \right\rceil$ is unique and appears for the first time (31 total options)	- HPC2 - HPC3

1,148,364 possible designs explored in ~36 hours

Case Study: ML-KEM Decaps

Results

d	Opt.	Area [est. kGE]	Area [kGE]	Rand. [bit]	Latency [cycles]
0	L	490	-	-	19,612
0	A	75	316	-	19,960,068
1	L	7,869	-	105,922	69,220
1	A	259	447	326	36,368,922
1	ALP	505	668	3,880	138,441
2	L	18,095	-	317,766	69,220
2	A	544	575	978	36,368,922
2	ALP	1,113	1,013	11,640	138,441

1,148,364 possible designs explored in ~36 hours

Case Study: ML-KEM Decaps

Results

d	Opt.	Area [est. kGE]	Area [kGE]	Rand. [bit]	Latency [cycles]
0	L	490	-	-	19,612
0	A	75	316	-	19,960,068
1	L	7,869	-	105,922	69,220
1	A	259	447	326	36,368,922
1	ALP	505	668	3,880	138,441
2	L	18,095	-	317,766	69,220
2	A	544	575	978	36,368,922
2	ALP	1,113	1,013	11,640	138,441

1,148,364 possible designs explored in ~36 hours

Case Study: ML-KEM Decaps

Results

d	Opt.	Area [est. kGE]	Area [kGE]	Rand. [bit]	Latency [cycles]
0	L	490	-	-	19,612
0	A	75	316	-	19,960,068
1	L	7,869	-	105,922	69,220
1	A	259	447	326	36,368,922
1	ALP	505	668	3,880	138,441
2	L	18,095	-	317,766	69,220
2	A	544	575	978	36,368,922
2	ALP	1,113	1,013	11,640	138,441

1,148,364 possible designs explored in ~36 hours

Case Study: ML-KEM Decaps

Results

d	Opt.	Area [est. kGE]	Area [kGE]	Rand. [bit]	Latency [cycles]
0	L	490	-	-	19,612
0	A	75	316	-	19,960,068
1	L	7,869	-	105,922	69,220
1	A	259	447	326	36,368,922
1	ALP	505	668	3,880	138,441
2	L	18,095	-	317,766	69,220
2	A	544	575	978	36,368,922
2	ALP	1,113	1,013	11,640	138,441

1,148,364 possible designs explored in ~36 hours

Case Study: ML-KEM Decaps

Results

d	Opt.	Area [est. kGE]	Area [kGE]	Rand. [bit]	Latency [cycles]
0	L	490	-	-	19,612
0	A	75	316	-	19,960,068
1	L	7,869	-	105,922	69,220
1	A	259	447	326	36,368,922
1	ALP	505	668	3,880	138,441
2	L	18,095	-	317,766	69,220
2	A	544	575	978	36,368,922
2	ALP	1,113	1,013	11,640	138,441

1,148,364 possible designs explored in ~36 hours

Case Study: ML-KEM Decaps

Discussion

Q: Why is even the area-optimized configuration so big?

A: The tool cannot enhance non-optimal templates.

- In this case: there are always two different Keccak states: an unmasked Keccak submodule and a masked Keccak module ($\geq 1600(d+1)$ registers always instantiated).
- An iNTT module is required nevertheless to perform an inverse transformation on $\widehat{\boldsymbol{A}}$.

Q: Why is there such a big discrepancy between estimated area and synthesized area?

A: In the PoC, we only estimate the secret data path, which is usually the largest factor in masked implementations.

Conclusion + Future Work

Our work:

- Novel approach to model, design, describe security-aware cryptographic hardware
- Proof-of-concept implementation of this approach
- Wide range of case studies

paper

code

Conclusion + Future Work

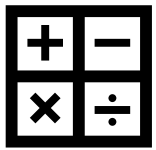
Our work:

- Novel approach to model, design, describe security-aware cryptographic hardware
- Proof-of-concept implementation of this approach
- Wide range of case studies

paper

code

Future work:



Arithmetic masking

- Conversion methods
- Area-duplicating vs. time-duplicating

Optimization

- Reduction of randomness consumption
- Gadget mixing

Formal verification

- Security
- Correctness
- Equality with a blueprint or standard

Case Studies

DSE Performance

Algorithm	# Configurations	Time
Keccak	14	0.5s
AdderModQ	42	0.7s
Sparse Polynomial Multiplication	372	1.2s
ChaCha20	1 080	3.2s
AES	1 440	5.4s
Polynomial Multiplication	1 302	7.9s
ML-KEM-CPA	40 362	196.5s
ML-KEM-CCA	1 148 364	36h

Case Study: AES

Results

d	Opt.	Area [est. kGE]	Area [kGE]	Rand. [bit]	Latency [cycles]
0	L	8.6	9.8	-	11
0	A	2.1	4.2	-	656
1	L	1057.4	707.9	14,400	41
1	A	9.5	10.3	72	1,011
1	ALP	28.9	23.7	360	81
2	L	2375.6	1426.1	43,200	41
2	A	20.9	17.2	204	1,011
2	ALP	64.7	44.6	1,080	81

4,032 possible designs explored in ~10 seconds

Case Study: AES

Results

d	Opt.	Area [est. kGE]	Area [kGE]	Rand. [bit]	Latency [cycles]
0	L	8.6	9.8	-	11
0	A	2.1	4.2	-	656
1	L	1057.4	707.9	14,400	41
1	A	9.5	10.3	72	1,011
1	ALP	28.9	23.7	360	81
2	L	2375.6	1426.1	43,200	41
2	A	20.9	17.2	204	1,011
2	ALP	64.7	44.6	1,080	81

4,032 possible designs explored in ~10 seconds

Case Study: AES

Results

d	Opt.	Area [est. kGE]	Area [kGE]	Rand. [bit]	Latency [cycles]
0	L	8.6	9.8	-	11
0	A	2.1	4.2	-	656
1	L	1057.4	707.9	14,400	41
1	A	9.5	10.3	72	1,011
1	ALP	28.9	23.7	360	81
2	L	2375.6	1426.1	43,200	41
2	A	20.9	17.2	204	1,011
2	ALP	64.7	44.6	1,080	81

4,032 possible designs explored in ~10 seconds

Case Study: AES

Results

d	Opt.	Area [est. kGE]	Area [kGE]	Rand. [bit]	Latency [cycles]
0	L	8.6	9.8	-	11
0	A	2.1	4.2	-	656
1	L	1057.4	707.9	14,400	41
1	A	9.5	10.3	72	1,011
1	ALP	28.9	23.7	360	81
2	L	2375.6	1426.1	43,200	41
2	A	20.9	17.2	204	1,011
2	ALP	64.7	44.6	1,080	81

4,032 possible designs explored in ~10 seconds

Case Studies

Kyber with Local Optimizations

ALP

Depth	# Configurations	DSE Time	Area [est. kGE]	Rand. [bit]	Latency [Cycles]
1	110	7s (L)	504	4,420	2,385,918
2	662	35s (L)	2,142	29,240	315,401
4	1,148,364	36h (S)	505	3,880	138,441

ALRP

Depth	# Configurations	DSE Time	Area [est. kGE]	Rand. [bit]	Latency [Cycles]
1	110	7s (L)	266	163	52,707,120
2	662	35s (L)	287	273	26,492,720
4	1,148,364	36h (S)	399	162	164,874

Ref.	Opt.	d	Area [kGE]	Rand. [bit]	Lat. [cycles]	Delay [ns]	T.put [Gb/s]	Technology
[MPL+11]		0	2.6	–	226			UMCL18G212T3
[SKS12]*		0	58.4	–			1.6	180 nm
[AR18]*		0	29.4	–	46	39	13.130	65 nm
[AR18]*		0	29.4	–	46	58	8.904	65 nm
[KMMS22]		0	3.3	–	227	188	0.679	NanGate 45 nm
[KMMS22]		0	9.9	–	11	20	6.290	NanGate 45 nm
HADES	L	0	85.5	–	11	23	61.000	NanGate 45 nm
HADES	A	0	4.2	–	211	597	0.214	NanGate 45 nm
HADES	ALP	0	9.8	–	11	25	5.195	NanGate 45 nm
[MPL+11]		1	11.1	48	266			UMCL18G212T3
[GMK16]		1	7.6	28	216			UMC 180 nm
[CRB+16]		1	6.7	54	276			NanGate 45 nm
[SM21]		1	7.1	1	246	1537	0.083	UMC 180 nm
[SM21]		1	7.7	0	246	1537	0.083	UMC 180 nm
[KMMS22]		1	33.1	414	3859	9879	0.013	NanGate 45 nm
[KMMS22]		1	10.0	34	2043	4310	0.030	NanGate 45 nm
[KMMS22]		1	52.6	680	99	202	0.634	NanGate 45 nm
HADES	L	1	707.9	14400	41	60	87.070	NanGate 45 nm
HADES	A	1	10.3	72	1011	1446	0.088	NanGate 45 nm
HADES	R	1	11.2	34	1811	2608	0.049	NanGate 45 nm
HADES	ALRP	1	12.7	34	371	575	0.226	NanGate 45 nm
HADES	ALP	1	23.7	360	81	151	0.847	NanGate 45 nm
[CBR+15]		2	18.6	126	276			NanGate 45 nm
[GMK16]		2	12.8	84	216			UMC 180 nm
[KMMS22]		2	17.6	102	2043	5434	0.023	NanGate 45 nm
[KMMS22]		2	131.6	2040	99	237	0.540	NanGate 45 nm
HADES	L	2	1426.1	43200	41	64	82.580	NanGate 45 nm
HADES	A	2	17.2	204	1011	1526	0.084	NanGate 45 nm
HADES	R	2	19.3	102	1811	2898	0.044	NanGate 45 nm
HADES	ALRP	2	21.5	102	371	545	0.235	NanGate 45 nm
HADES	ALP	2	44.6	1080	81	134	0.955	NanGate 45 nm

∗ These papers report gate counts instead of GE.

Ref.	Opt.	d	Area [kGE]	Rand. [bit]	Lat. [cycles]	Delay [ns]	Technology
[BDN+13]		1	116.6	4	25	42.2	NanGate 45 nm
[BDN+13]		1	39.0	4	1625	2561.2	NanGate 45 nm
[GSM17]		1	18.7	0	3160	3690.7	UMC 130 µm
[GSM17]		1	22.3	0	1648	2028.8	UMC 130 µm
[GSM17]		1	108.0	0	25	28.2	UMC 130 µm
[SM21]		1	129.3	0	72	92.9	UMC 130 µm
HADES	L	1	149.4	3200	120	40.8	NanGate 45 nm
HADES	A	1	30.1	50	3144	1792.5	NanGate 45 nm
HADES	R/ALRP	1	34.6	25	4680	3744.0	NanGate 45 nm
HADES	ALP	1	61.6	800	264	198.1	NanGate 45 nm
[GSM17]		2	28.8	75	3160	3706.7	UMC 130 µm
[GSM17]		2	34.6	75	1648	1951.2	UMC 130 µm
[GSM17]		2	232.3	4800	25	29.8	UMC 130 µm
[SM21]		2	231.5	0	72	108.0	UMC 130 µm
HADES	L	2	325.9	9600	120	40.8	NanGate 45 nm
HADES	A	2	52.2	150	3144	2954.9	NanGate 45 nm
HADES	R/ALRP	2	54.1	75	4680	2901.4	NanGate 45 nm
HADES	ALP	2	117.7	2400	264	211.2	NanGate 45 nm

d	Scheme	Area [kGE]	SRAM [bit]	Rand. [bit]	Latency [cycles]	Delay [μs]
1	sNTRUp-761	201	189440	310	1870049	9034
	Kyber-512	447	28608	326	36368922	136213
	Kyber-512	454	28608	163	52707120	205887
	Kyber-512	485	28608	211	931632	3741
	Kyber-512	668	28608	3880	138441	541
	Kyber-768	433	35776	250	66103166	250391
	Kyber-768	439	35776	125	95698594	359769
	Kyber-768	469	35776	173	1557154	6254
	Kyber-768	797	35776	3956	209081	791
2	sNTRUp-761	373	246272	930	1870049	11334
	Kyber-512	575	30400	978	36368922	137241
	Kyber-512	590	30400	489	52707120	198895
	Kyber-512	635	30400	633	931632	3516
	Kyber-512	1013	30400	11640	138441	543
	Kyber-768	553	38336	750	66103166	265475
	Kyber-768	564	38336	375	95698594	359769
	Kyber-768	609	38336	519	1557154	6155
	Kyber-768	1179	38336	11868	209081	795