

| BEANIE

A 32-bit Cipher for Cryptographic Mitigations against Software Attacks

Simon Gerhalter Samir Hodžić Marcel Medwed Marcel Nageler
Artur Folwarczny Ventzi Nikov Jan Hoogerbrugge Tobias Schneider
Gary McConville Maria Eichlseder

FSE 2026 - Singapore

- Modern CPUs include various features to mitigate software attacks
 - Logical isolation, memory tagging, or shadow stacks

- Modern CPUs include various features to mitigate software attacks
 - Logical isolation, memory tagging, or shadow stacks
- Basing these features on **cryptographic isolation** has advantages
 - Lower memory overhead
 - Robustness against misconfiguration

- Modern CPUs include various features to mitigate software attacks
 - Logical isolation, memory tagging, or shadow stacks
- Basing these features on **cryptographic isolation** has advantages
 - Lower memory overhead
 - Robustness against misconfiguration
- Our goal: **memory encryption**
 - Reduces frequency or introduces stall cycles
 - Latency optimized cipher needed

Motivation for new cipher

- Targets are 32-bit microcontrollers
 - **32-bit input**
 - **128-bit tweak** (Context such as address, etc.)
 - Optimized for hardware implementation

Motivation for new cipher

- Targets are 32-bit microcontrollers
 - **32-bit input**
 - **128-bit tweak** (Context such as address, etc.)
 - Optimized for hardware implementation
- ChiLow-32 [Bel+25]
 - Latency optimized for decryption only
 - Tiny number of queries per tweak

Motivation for new cipher

- Targets are 32-bit microcontrollers
 - **32-bit input**
 - **128-bit tweak** (Context such as address, etc.)
 - Optimized for hardware implementation
- ChiLow-32 [Bel+25]
 - Latency optimized for decryption only
 - Tiny number of queries per tweak
- Consider latency of instructions

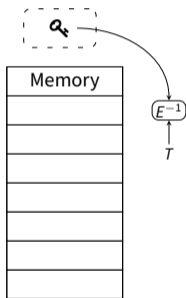
Motivation for new cipher

- Targets are 32-bit microcontrollers
 - **32-bit input**
 - **128-bit tweak** (Context such as address, etc.)
 - Optimized for hardware implementation
- ChiLow-32 [Bel+25]
 - Latency optimized for decryption only
 - Tiny number of queries per tweak
- Consider latency of instructions
- Take **attack scenario** into consideration

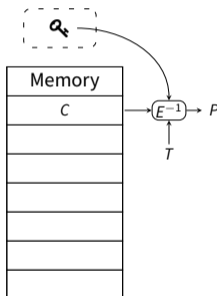
- **Separate tweak-key schedule** has advantages

Latency Considerations

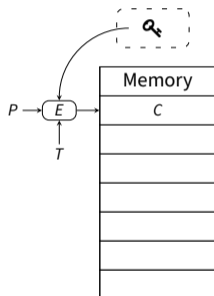
- **Separate tweak-key schedule** has advantages
- Memory reads
 - **Address** present at **first cycle**
 - Possible to precompute round keys



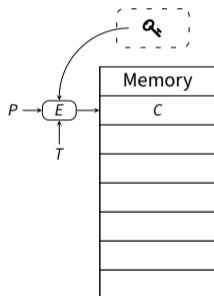
- **Separate tweak-key schedule** has advantages
- Memory reads
 - **Address** present at **first cycle**
 - Possible to precompute round keys
 - **Data** returned from memory in **second cycle**
 - Only need datapath decryption



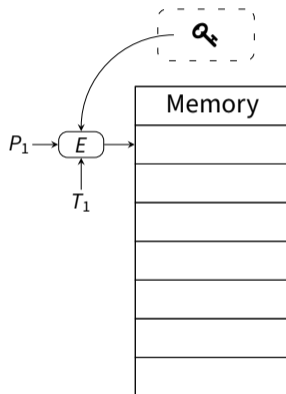
- **Separate tweak-key schedule** has advantages
- Memory reads
 - **Address** present at **first cycle**
 - Possible to precompute round keys
 - **Data** returned from memory in **second cycle**
 - Only need datapath decryption
- Memory writes
 - Sequential tweak-key schedule and encryption



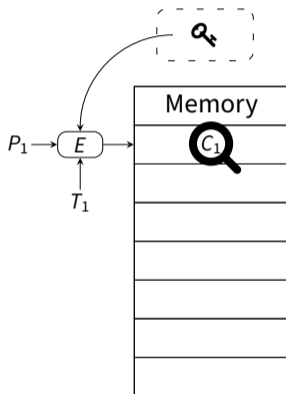
- **Separate tweak-key schedule** has advantages
- Memory reads
 - **Address** present at **first cycle**
 - Possible to precompute round keys
 - **Data** returned from memory in **second cycle**
 - Only need datapath decryption
- Memory writes
 - Sequential tweak-key schedule and encryption
 - However, significantly less writes than reads
 - Benchmark shows only 2.5% cycle increase



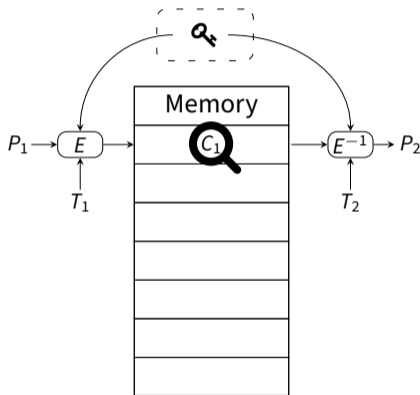
- Encrypt memory content



- Encrypt memory content
- Attack Scenarios
 - Attacker has physical access
 - Access encrypted memory content
 - For small node sizes this becomes infeasible

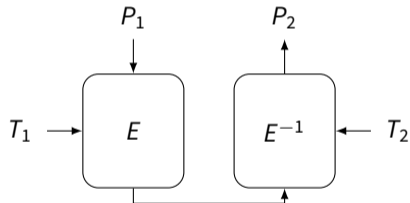


- Encrypt memory content
- Attack Scenarios
 - Attacker has physical access
 - Access encrypted memory content
 - For small node sizes this becomes infeasible
 - Attacker overcomes logical isolation
 - Access memory content decrypted with **different tweak**
 - Our attacker model



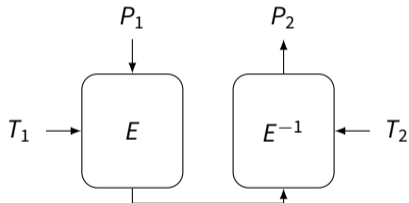
U-Shape Attack Setting

- First described for cipher SCARF [Can+23]
 - Cache-index randomization



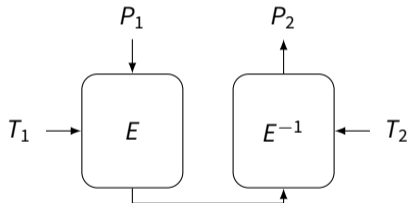
U-Shape Attack Setting

- First described for cipher SCARF [Can+23]
 - Cache-index randomization
- ⊕ Allows us to substantially **reduce rounds**



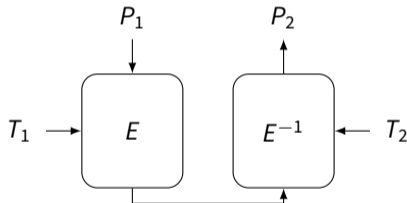
U-Shape Attack Setting

- First described for cipher SCARF [Can+23]
 - Cache-index randomization
- ⊕ Allows us to substantially **reduce rounds**
- ⚠ Can not simply half number of rounds
 - E^{-1} cancels some diffusion of E



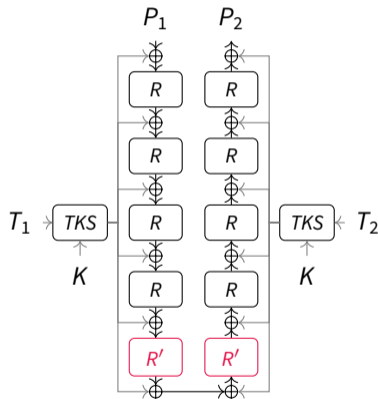
U-Shape Attack Setting

- First described for cipher SCARF [Can+23]
 - Cache-index randomization
- ⊕ Allows us to substantially **reduce rounds**
- ⚠ Can not simply half number of rounds
 - E^{-1} cancels some diffusion of E
- ➡ Cipher has to be carefully analyzed



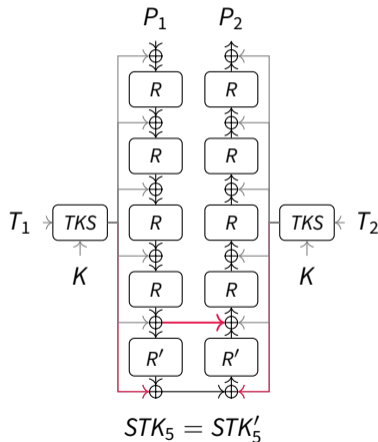
U-Shape Cipher Design Considerations

- The linear layer in the last round R' can be omitted



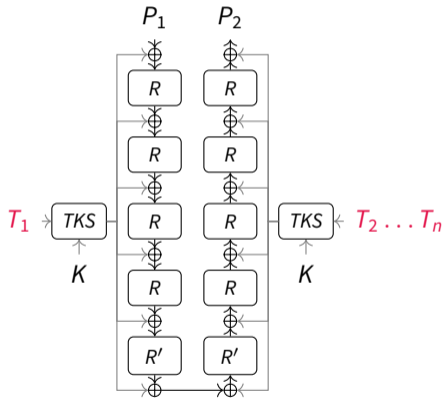
U-Shape Cipher Design Considerations

- The linear layer in the last round R' can be omitted
- Colliding subkeys can **cancel rounds**
 - Reduction in rounds may outweigh increased data complexity
 - Partial collisions possible
 - Require complex tweakkey schedule



U-Shape Cipher Design Considerations

- The linear layer in the last round R' can be omitted
- Colliding subkeys can **cancel rounds**
 - Reduction in rounds may outweigh increased data complexity
 - Partial collisions possible
 - Require complex tweakey schedule
- Only first subtweakey recovery is expensive



- Tweak-key schedule
 - Processes 128-bit tweak and 128-bit key
 - Output gets combined to round keys

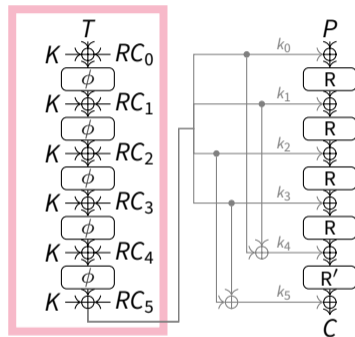


Figure: Connection between tweak-key schedule and datapath.

- Tweak-key schedule
 - Processes 128-bit tweak and 128-bit key
 - Output gets combined to round keys
- Datapath
 - 32-bit state size

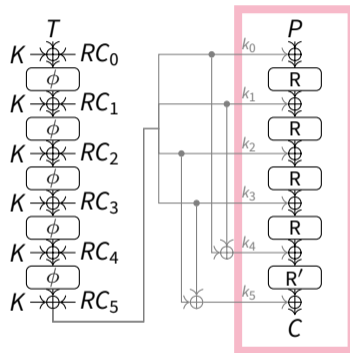


Figure: Connection between tweak-key schedule and datapath.

- Tweak-key schedule
 - Processes 128-bit tweak and 128-bit key
 - Output gets combined to round keys
- Datapath
 - 32-bit state size
- 2^{80} time and 2^{40} data security claim
 - 5 round tweak-key schedule and datapath

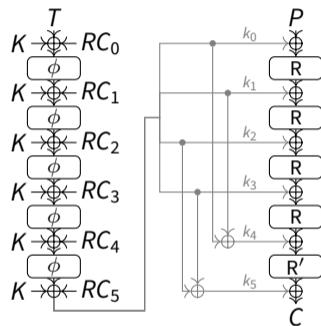


Figure: Connection between tweak-key schedule and datapath.

- AES like rounds

$$R_i = MC \circ SR \circ SC \circ ART_i, \quad R_n = ART_r \circ SR \circ SC \circ ART_{r-1}$$

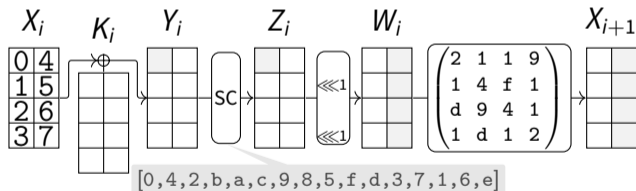


Figure: One round of BEANIE.

BEANIE Datapath

- AES like rounds

$$R_i = MC \circ SR \circ SC \circ ART_i, \quad R_n = ART_r \circ SR \circ SC \circ ART_{r-1}$$

- G7 S-Box [Saa11] and $M_{4,i,4}$ [Jea+17] MDS matrix
 - Chosen by weighing security against latency

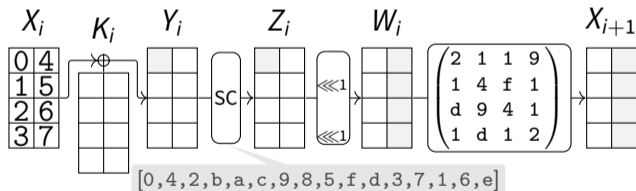


Figure: One round of BEANIE.

Tweak-key schedule

- Two parallel PRINCE like rounds combined with Feistel Type-2 layer
 - Full diffusion after 3 rounds

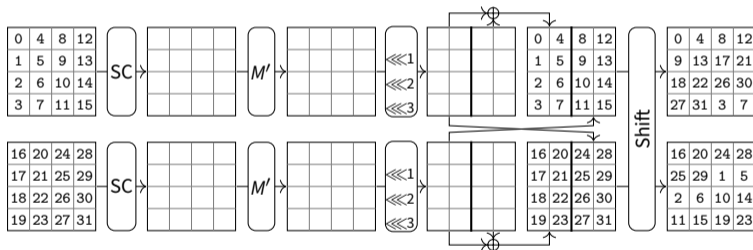


Figure: One round of the tweak-key schedule of BEANIE without AddRoundKey.

BEANIE Cryptanalysis

- Tight security margin
 - Comprehensive cryptanalysis provided

- Tight security margin
 - Comprehensive cryptanalysis provided

Type	R	R Dist.	T	D	M
Boomerang	4+4	2+4	2^{62}	2^{29}	2^{30}
Impossible diff.	4+4	4+2	2^{73}	2^{11}	-
Impossible diff. partial coll.	5+5	4+3	2^{112}	2^{30}	-
Integral	4+4	4+2	2^{67}	2^{18}	-
DS-MitM	5+5	4+3	2^{112}	2^{16}	2^{109}

- Tight security margin
 - Comprehensive cryptanalysis provided

Type	R	R Dist.	T	D	M
Boomerang	4+4	2+4	2^{62}	2^{29}	2^{30}
Impossible diff.	4+4	4+2	2^{73}	2^{11}	-
Impossible diff. partial coll.	5+5	4+3	2^{112}	2^{30}	-
Integral	4+4	4+2	2^{67}	2^{18}	-
DS-MitM	5+5	4+3	2^{112}	2^{16}	2^{109}

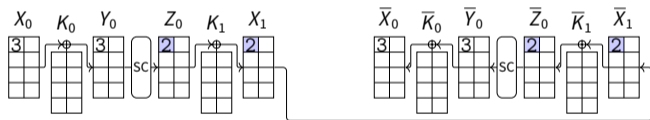
- Tight security margin
 - Comprehensive cryptanalysis provided

Type	R	R Dist.	T	D	M
Boomerang	4+4	2+4	2^{62}	2^{29}	2^{30}
Impossible diff.	4+4	4+2	2^{73}	2^{11}	-
Impossible diff. partial coll.	5+5	4+3	2^{112}	2^{30}	-
Integral	4+4	4+2	2^{67}	2^{18}	-
DS-MitM	5+5	4+3	2^{112}	2^{16}	2^{109}

- The attacks can be extended by one round via **generic last round collisions**
 - Increase time and data complexity by a factor of 2^{32}

- Probability of differential higher than of single differential characteristic
 - Especially pronounced in U-shape attack setting

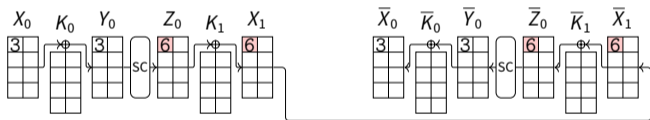
- Probability of differential higher than of single differential characteristic
 - Especially pronounced in U-shape attack setting
 - Example: Last round



0	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
3	.	.	4	.	.	2	4	2	.	2	.	2

$$\Pr[\Delta 3 \rightarrow_2 \Delta 3] = \frac{4}{16} \cdot \frac{4}{16} = \frac{1}{16}$$

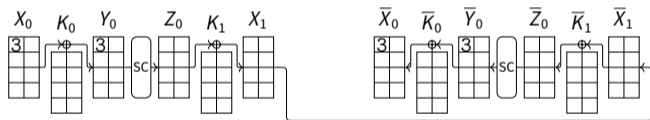
- Probability of differential higher than of single differential characteristic
 - Especially pronounced in U-shape attack setting
 - Example: Last round



0	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
3	.	.	4	.	.	2	4	2	.	2	.	2

$$\Pr[\Delta 3 \rightarrow_6 \Delta 3] = \frac{4}{16} \cdot \frac{4}{16} = \frac{1}{16}$$

- Probability of differential higher than of single differential characteristic
 - Especially pronounced in U-shape attack setting
 - Example: Last round



0	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
3	.	.	4	.	.	2	4	2	.	2	.	2

$$\Pr[\Delta 3 \rightarrow_{2,6} \Delta 3] = \frac{1}{16} + \frac{1}{16} = \frac{1}{8}$$

Partial Roundkey Collision in Impossible Differential

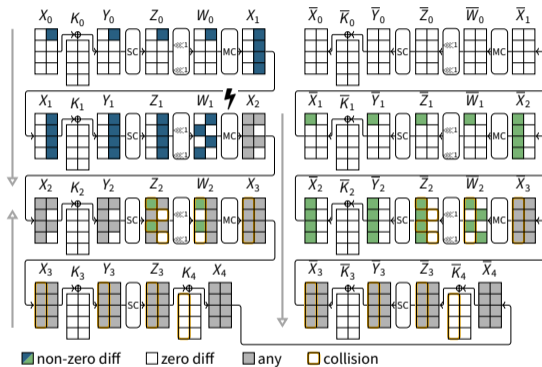


Figure: 4+3 round impossible differential distinguisher.

BEANIE Latency

Cipher	Block-Size (<i>bit</i>)	Time (<i>ps</i>)
SCARF	10	217
BipBip	24	327
ChiLow-32	32	283
BEANIE	32	203
PRINCE	64	376
QARMAv2-64- σ_0	64	305
BEANIE TKS	128	212
Orthros	128	356
QARMAv2-128-128	128	620

Table: Latency on NanGate 15nm technology.

BEANIE Latency

Cipher	Block-Size (<i>bit</i>)	Time (<i>ps</i>)
SCARF	10	217
BipBip	24	327
ChiLow-32	32	283
BEANIE	32	203
PRINCE	64	376
QARMAv2-64- σ_0	64	305
BEANIE TKS	128	212
Orthros	128	356
QARMAv2-128-128	128	620

- CoreMark Benchmark
 - Cycle count increase of about 2.5%

Table: Latency on NanGate 15nm technology.

BEANIE Latency

Cipher	Block-Size (<i>bit</i>)	Time (<i>ps</i>)
SCARF	10	217
BipBip	24	327
ChiLow-32	32	283
BEANIE	32	203
PRINCE	64	376
QARMAv2-64- σ_0	64	305
BEANIE TKS	128	212
Orthros	128	356
QARMAv2-128-128	128	620

Table: Latency on NanGate 15nm technology.

- CoreMark Benchmark
 - Cycle count increase of about 2.5%
 - About 16% drop in maximum clock frequency

Bonnetain et al. "Yoyo tricks with a BEANIE" [Bon+25]

Bonnetain et al. "Yoyo tricks with a BEANIE" [Bon+25]

- Combine
 - Birthday bound collisions of round keys
 - Query $2^{n/2}$ different tweaks
 - Among these we expect a roundkey collision in the last round

Bonnetain et al. "Yoyo tricks with a BEANIE" [Bon+25]

- Combine
 - Birthday bound collisions of round keys
 - Query $2^{n/2}$ different tweaks
 - Among these we expect a roundkey collision in the last round
 - Yoyo cryptanalysis [RBH17]

Bonnetain et al. "Yoyo tricks with a BEANIE" [Bon+25]

- Combine
 - Birthday bound collisions of round keys
 - Query $2^{n/2}$ different tweaks
 - Among these we expect a roundkey collision in the last round
 - Yoyo cryptanalysis [RBH17]
- Break tight security of the 5 round version of BEANIE

- New tweakable blockcipher BEANIE
 - Mitigating software attack with cryptography

- New tweakable blockcipher BEANIE
 - Mitigating software attack with cryptography
 - Uniquely seperated tweak-key schedule

- New tweakable blockcipher BEANIE
 - Mitigating software attack with cryptography
 - Uniquely seperated tweak-key schedule
 - Comprehensive cryptanalysis

- New tweakable blockcipher BEANIE
 - Mitigating software attack with cryptography
 - Uniquely separated tweak-key schedule
 - Comprehensive cryptanalysis
- Original security recommendation too optimistic

Conclusion

- New tweakable blockcipher BEANIE
 - Mitigating software attack with cryptography
 - Uniquely seperated tweak-key schedule
 - Comprehensive cryptanalysis
- Original security recommendation too optimistic
 - 6 rounds should still provide competative latency

- New tweakable blockcipher BEANIE
 - Mitigating software attack with cryptography
 - Uniquely seperated tweak-key schedule
 - Comprehensive cryptanalysis
- Original security recommendation too optimistic
 - 6 rounds should still provide competative latency
 - We invite you to conduct some further cryptanalysis



simon.gerhalter@tugraz.at

`</>` https://github.com/isec-tugraz/beanie_cipher/

Acknowledgments

This research was co-funded by the Austrian Science Fund (FWF) SFB project SPyCoDe (10.55776/F85), by the European Union (ERC Starting Grant KEYLESS, #101165216), and by the SINFONIA project, which has received funding from the Recovery and Resilience Facility (RRF) as the centrepiece of NextGenerationEU via the Austrian Research Promotion Agency (FFG) and Austria Wirtschaftsservice Gesellschaft mbH (aws) in the frame of the IPCEI ME/CT – Important Project of Common European Interest on Microelectronic and Communication Technologies, under FFG project No. 917423 and AWS project No. P2431566.



European Research Council
Established by the European Commission

- [Bel+25] Yanis Belkheyar et al. **ChiLow and ChiChi: New Constructions for Code Encryption**. EUROCRYPT 2025. Vol. 15601. LNCS. Springer, 2025, pp. 212–243. DOI: [10.1007/978-3-031-91107-1_8](https://doi.org/10.1007/978-3-031-91107-1_8).
- [Bon+25] Xavier Bonnetain et al. **Yoyo tricks with a BEANIE**. IACR Cryptology ePrint Archive, Paper 2025/2297. 2025. URL: <https://eprint.iacr.org/2025/2297>.
- [Can+23] Federico Canale et al. **SCARF – A Low-Latency Block Cipher for Secure Cache-Randomization**. 32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023. USENIX Association, 2023, pp. 1937–1954. URL: <https://www.usenix.org/conference/usenixsecurity23/presentation/canale>.
- [Jea+17] Jérémy Jean et al. **Optimizing Implementations of Lightweight Building Blocks**. *IACR Transactions on Symmetric Cryptology* 2017.4 (2017), pp. 130–168. DOI: [10.13154/tosc.v2017.i4.130-168](https://doi.org/10.13154/tosc.v2017.i4.130-168).

- [RBH17] Sondre Rønjom, Navid Ghaedi Bardeh, and Tor Helleseeth. **Yoyo Tricks with AES**. ASIACRYPT 2017. Vol. 10624. LNCS. Springer, 2017, pp. 217–243. doi: [10.1007/978-3-319-70694-8_8](https://doi.org/10.1007/978-3-319-70694-8_8).
- [Saa11] Markku-Juhani O. Saarinen. **Cryptographic Analysis of All 4×4 -Bit S-Boxes**. SAC 2011. Vol. 7118. LNCS. Springer, 2011, pp. 118–133. doi: [10.1007/978-3-642-28496-0_7](https://doi.org/10.1007/978-3-642-28496-0_7).