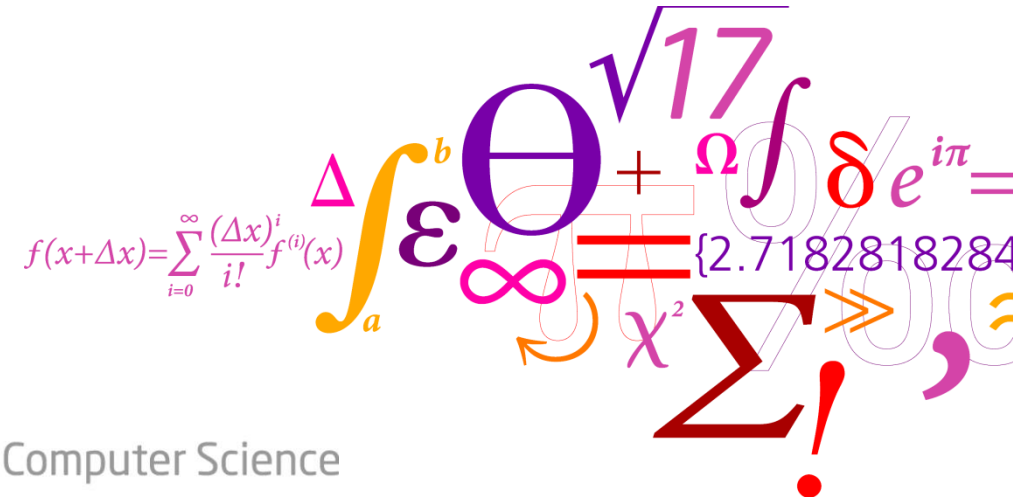
Lightweight Crypto on a Full Circle: From industry to academia and back

Christian Rechberger

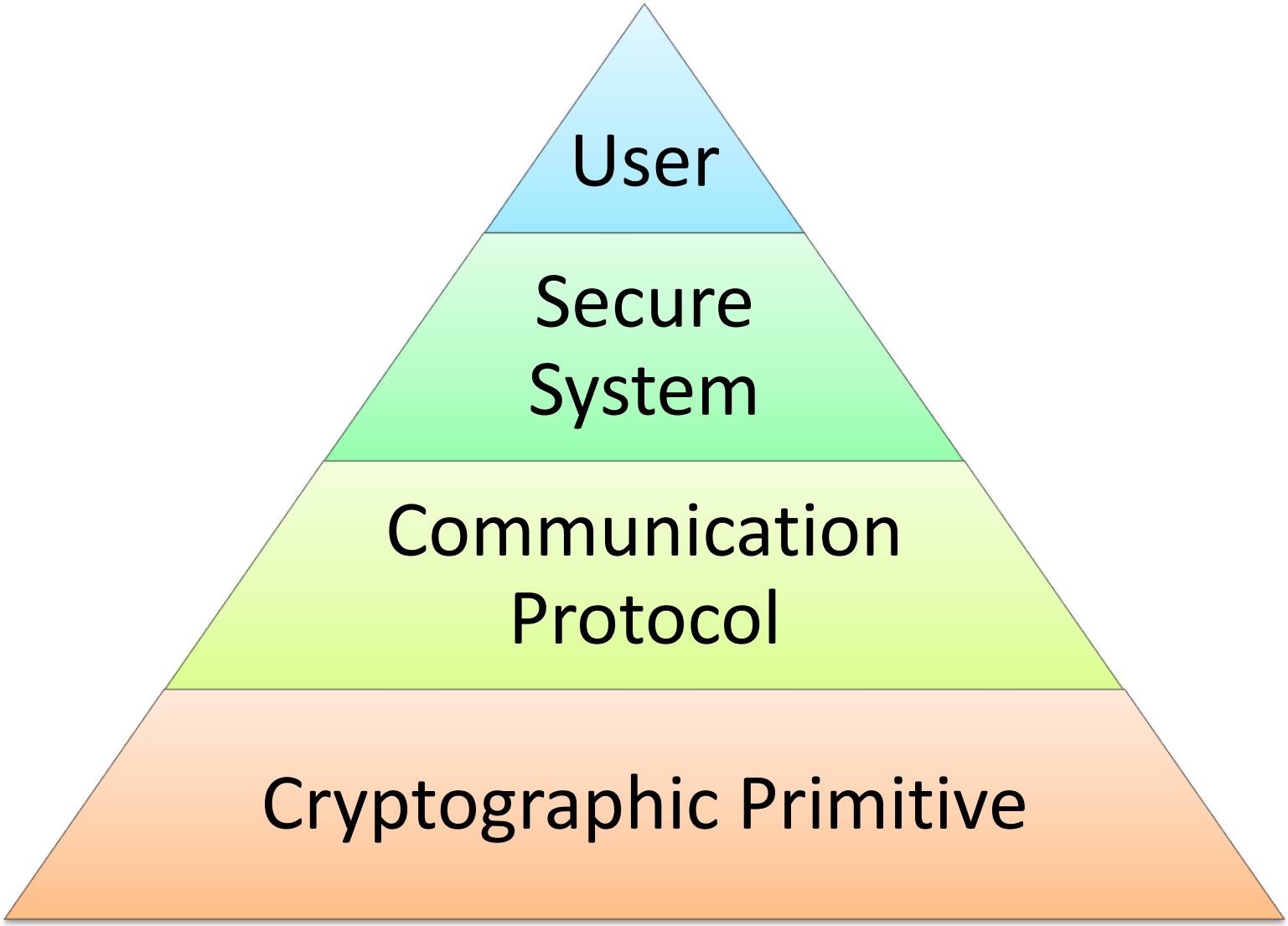


DTU Compute
Department of Applied Mathematics and Computer Science

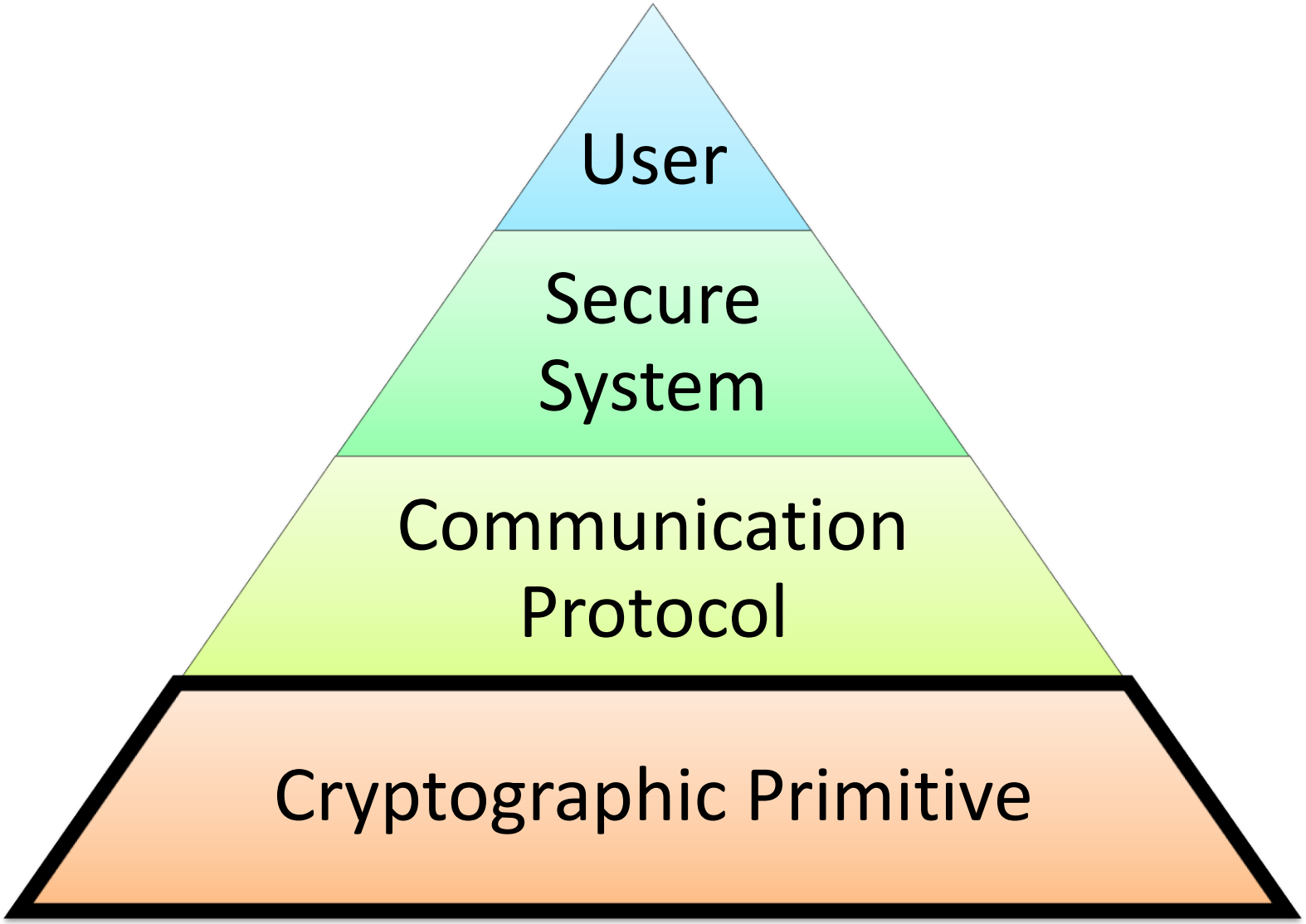
An often quoted myth

"Crypto algorithms are
never the weakest link in a system"

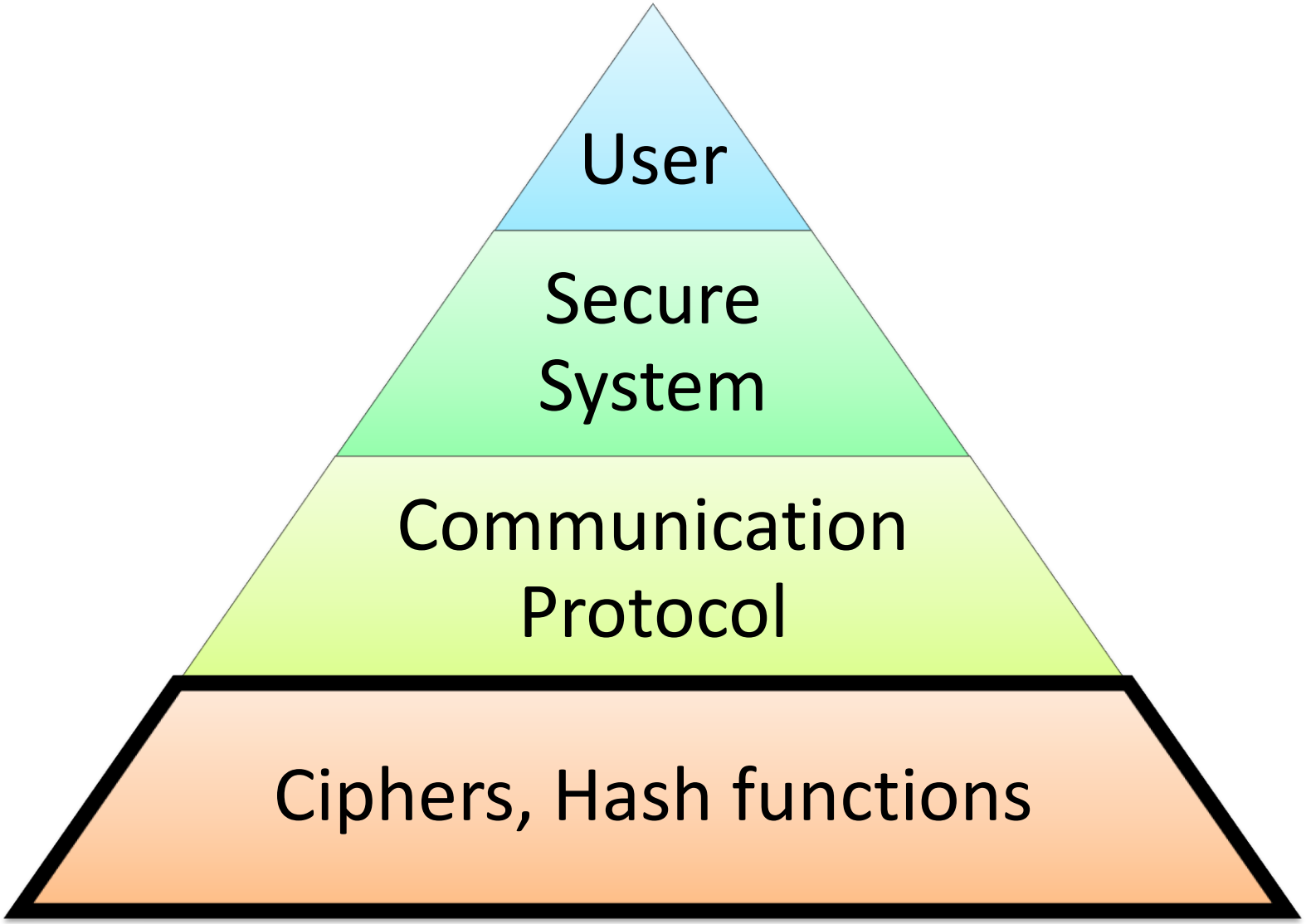
Security of modern IT Systems



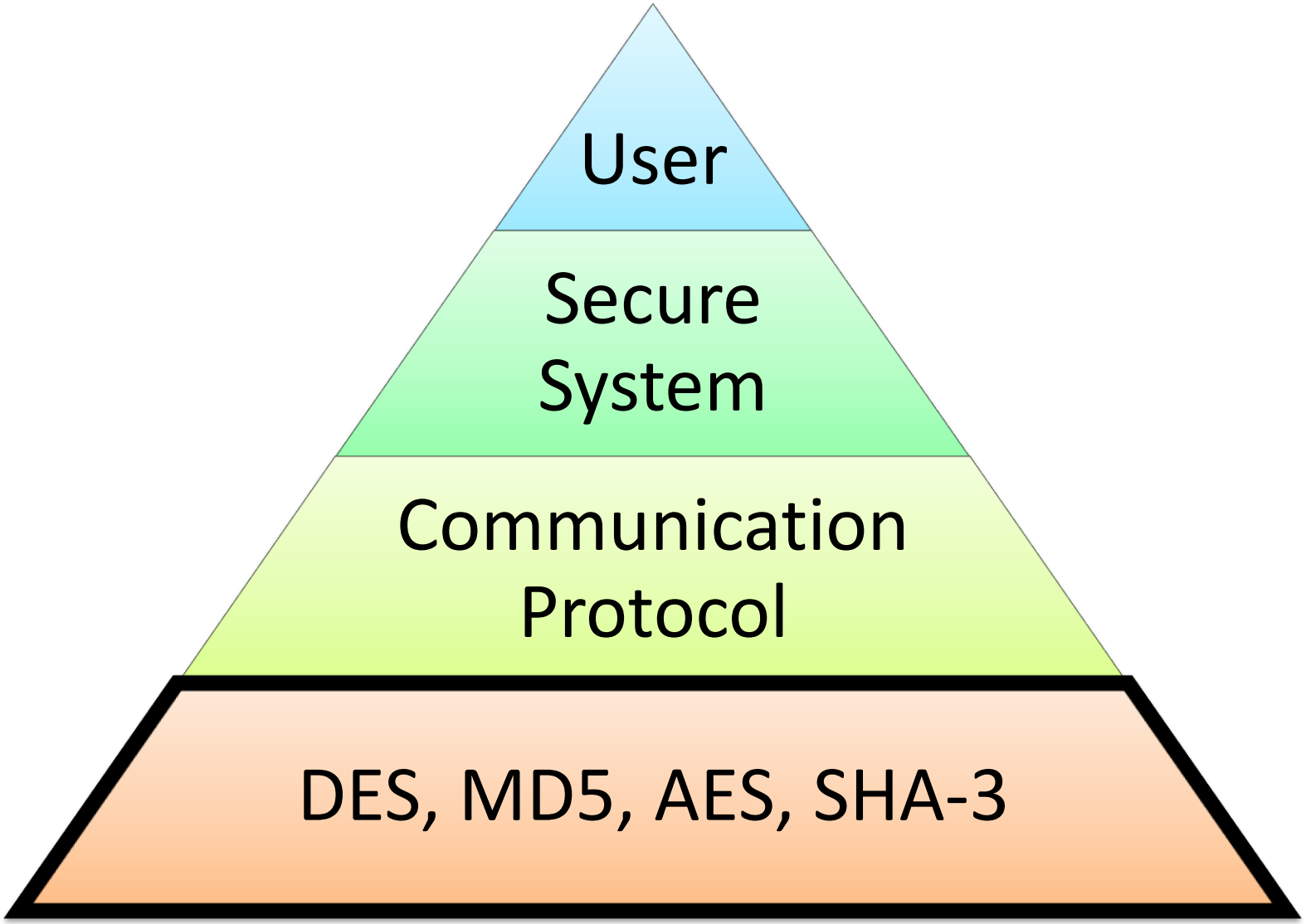
Security of modern IT Systems



Security of modern IT Systems



Security of modern IT Systems



MD5 cryptanalysis

- Widely used cryptographic hash function
- Chosen-prefix differential collision attacks since 2007
- Rogue certificates
- Malware “Flame”

RC4 cryptanalysis

- Widely used stream cipher
- Practical attack on WEP
- Attack on WPA/TKIP
- Attack on TLS

Mifare (classic) and attacks

- Contactless chipcard, product line by market leader NXP
  - 2 billion cards sold, 25 million readers
  - Based on **proprietary** cipher/protocol “Crypto-1”
  - Very resource constrained

- Public reverse engineering in 2007, attacks since 2008
  - Cloning of card in 10 seconds with 300 queries
  - Lots of bad press,
    direct financial impact not clear

Keeloq attacks

- Cipher design in 1985
- Sold to Microchip Technologies Inc. (10M$)
- Widely used for car immobilizer and in garage doors

- Badly broken since mid 2000s

Megamos cipher

- Used in car-immobilizers
- Once code was discovered in public, attack was found
- Disclosure of attack blocked by UK court

Many more examples

- DST cipher, attacks on payment and car immobilizer systems
- A5/1, A5/2 as used in GSM communication
- DECT, GMR, …

DES – the first lightweight cipher

- First public block cipher
- Designed in mid 70s by IBM
- NSA intervened: key-space only 56 bits

- From mid 90s:
  easy to break by
  brute-force

Advanced Encryption Standard

- Designed as „Rijndael“ in 1997 by Joan Daemen and Vincent Rijmen
- Selected to be the AES in 2001
  - Open, public competition
  - Participation from Academia, Industry
  - Successor of DES
- Key sizes: 128, 192, and 256 bit

Is AES a lightweight cipher?

- Perhaps yes: It can be implemented with less gates than ciphers standardized by ISO in the lightweight category (ISO/IEC 29192-2:2012)

Why was AES not used?

- AES is only around since 2001
- AES is a general purpose cipher, very versatile within limits
- Too slow, too large, in very constrained environments

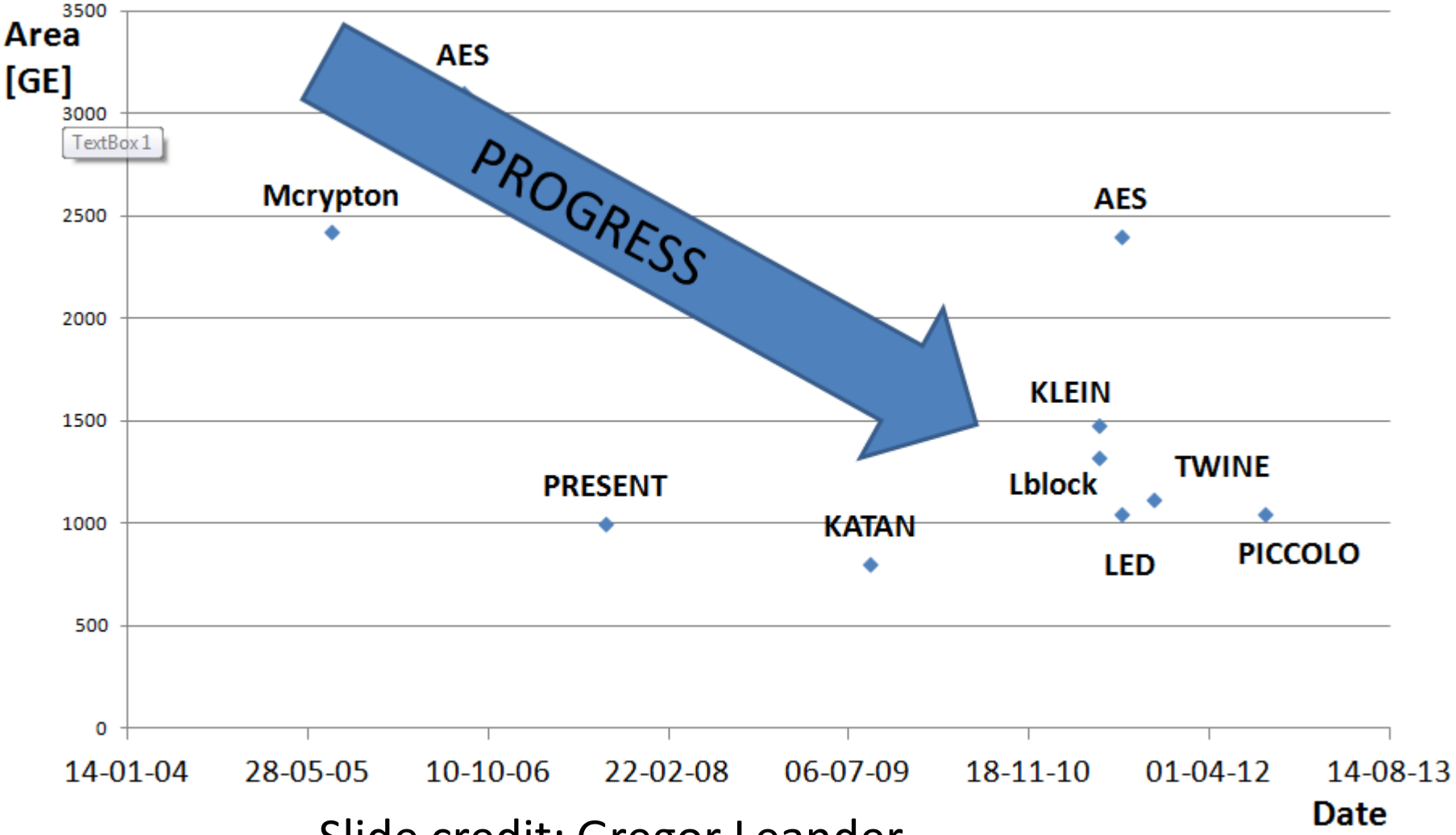
Long term trends in computing

- Past: Crypto was expensive
- Now: Crypto is cheap
- Future: Crypto will be expensive (energy)

Why is data protection getting harder?

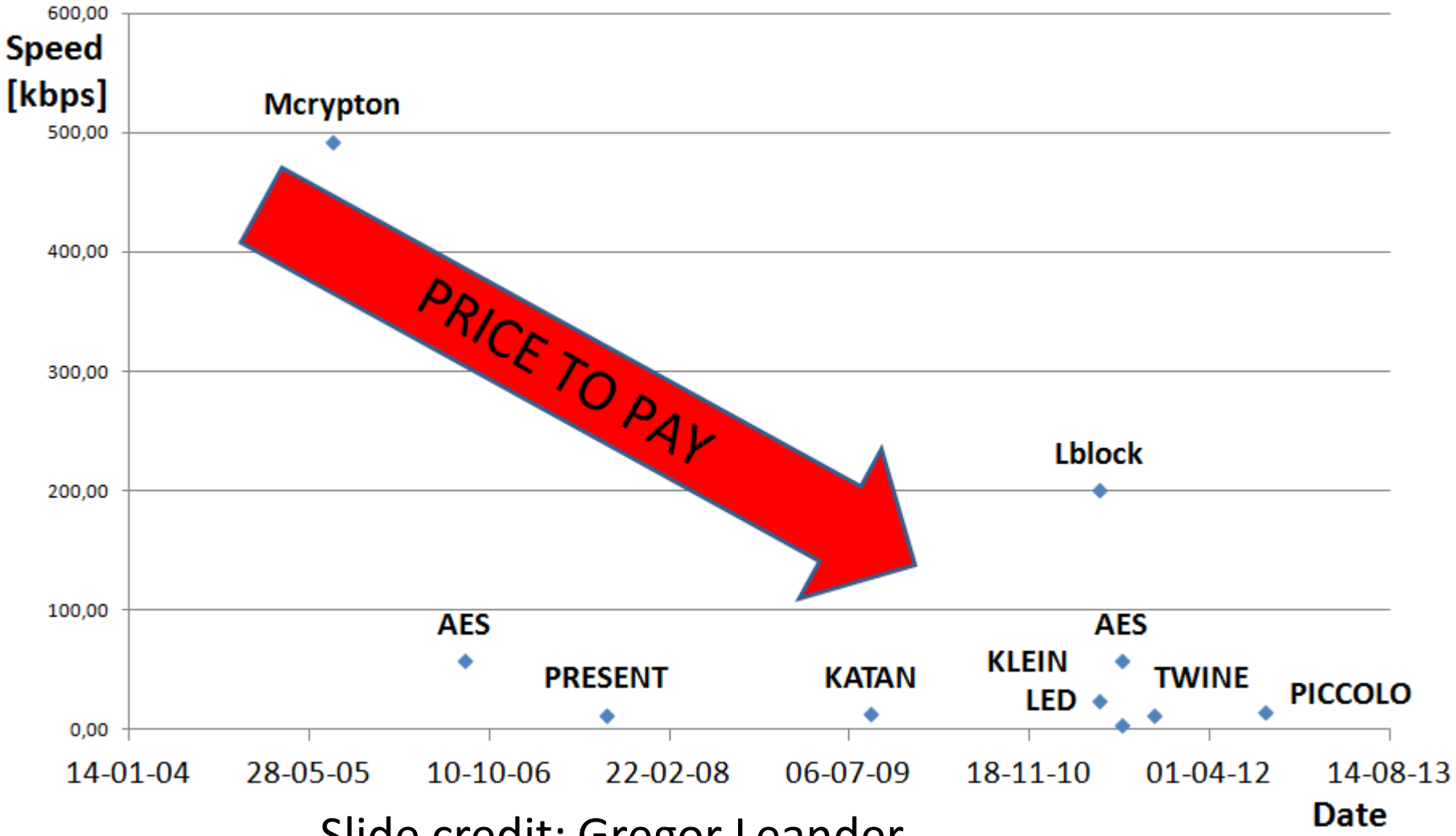
- Two orders of magnitude per dollar per decade increase in computation
- Three orders of magnitude per dollar per decade increase in storage
- Four orders of magnitude per dollar per decade increase in bandwidth

Progress in academic research on lightweight crypto?



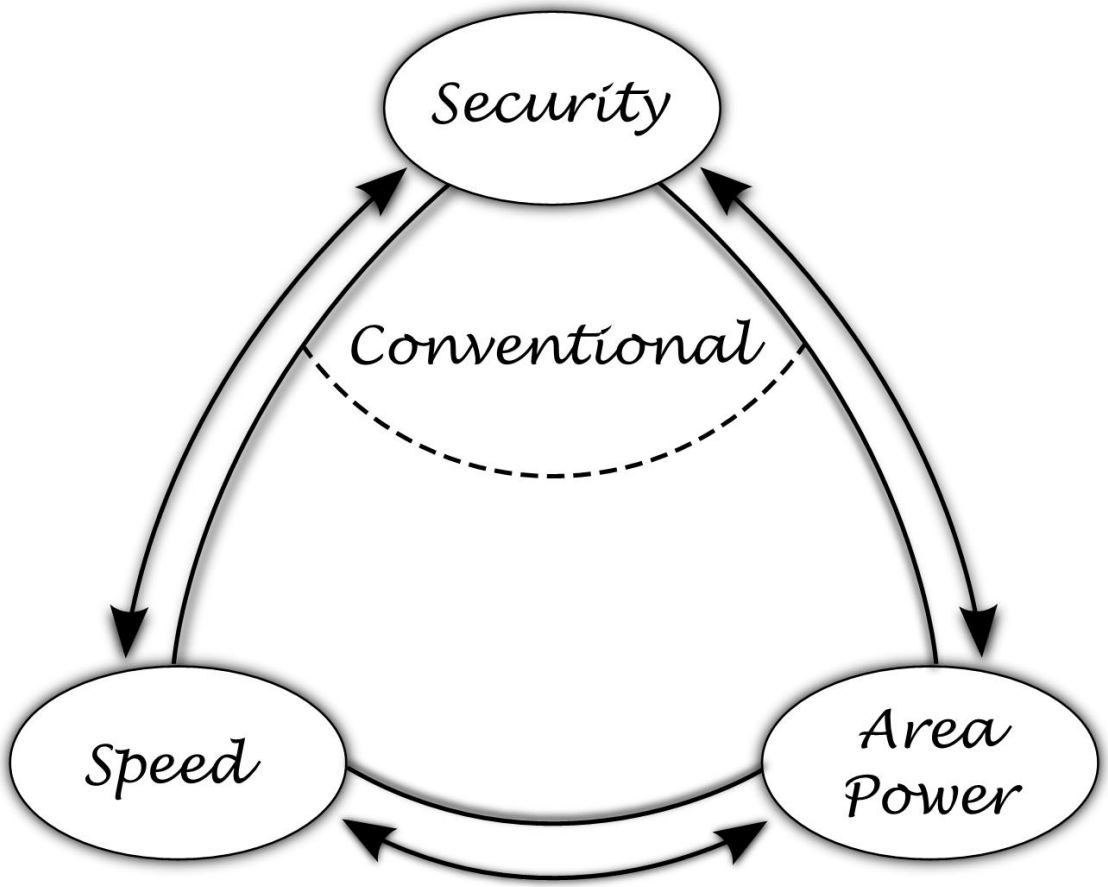
Slide credit: Gregor Leander

Progress in academic research on lightweight crypto?

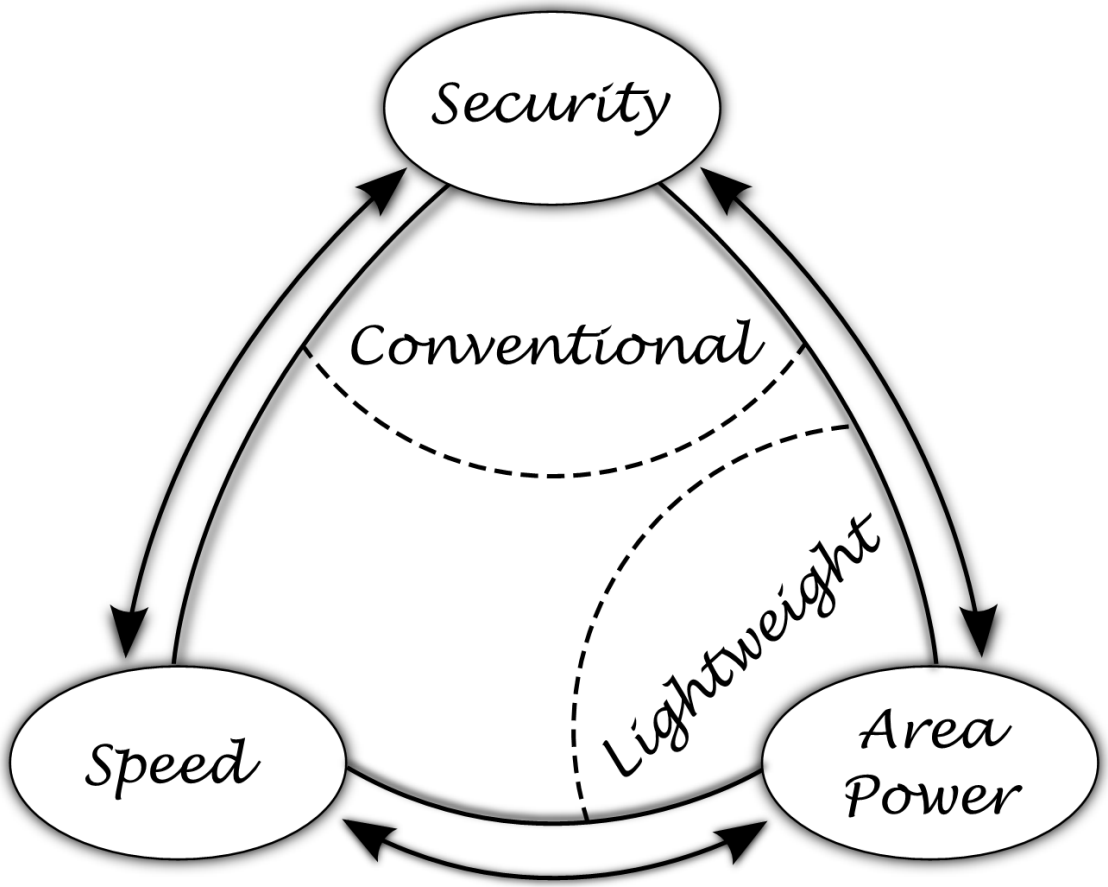


Slide credit: Gregor Leander

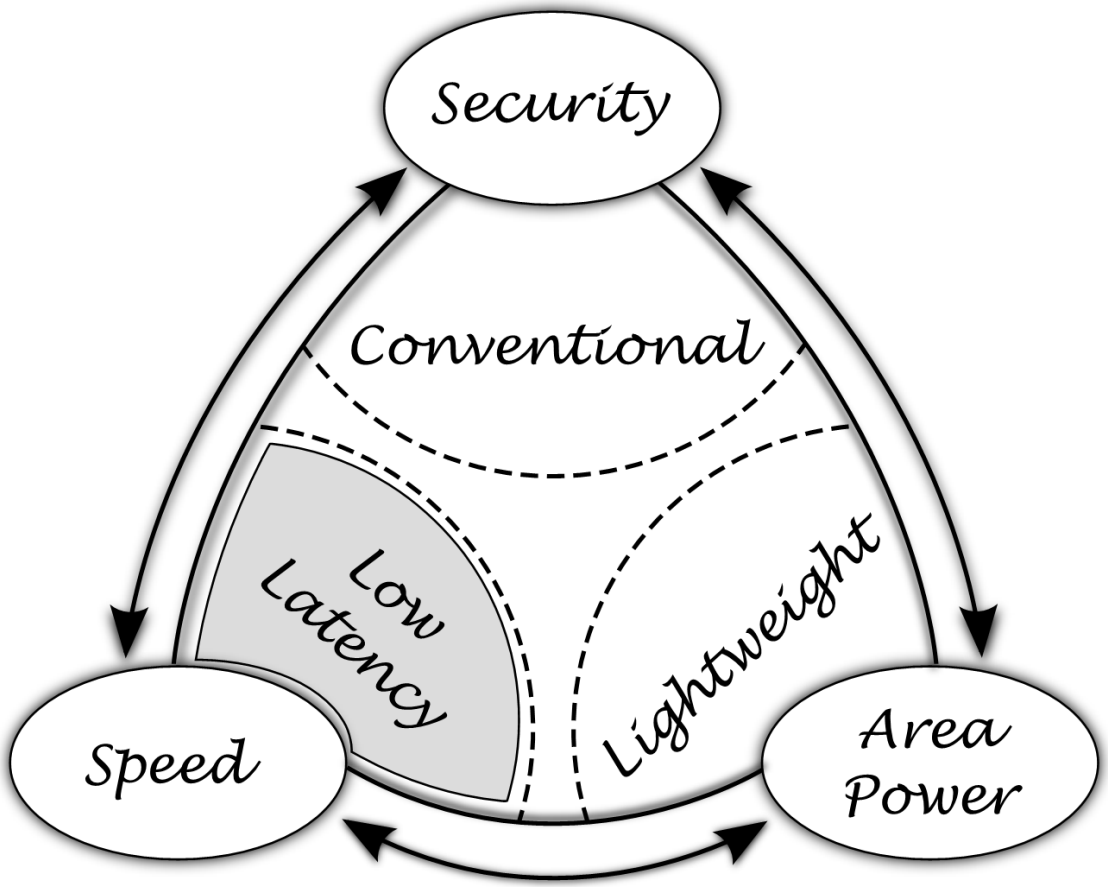
Trade-offs in Cryptography



Trade-offs in Cryptography



Trade-offs in Cryptography



Low-latency designs

Latency = #clock cycles * critical path length

- Low-latency implies high-throughput
- But high-throughput does not imply low-latency, because of
  - heavy use of pipelining
  - parallelization
- Has good potential to also be “low-energy”

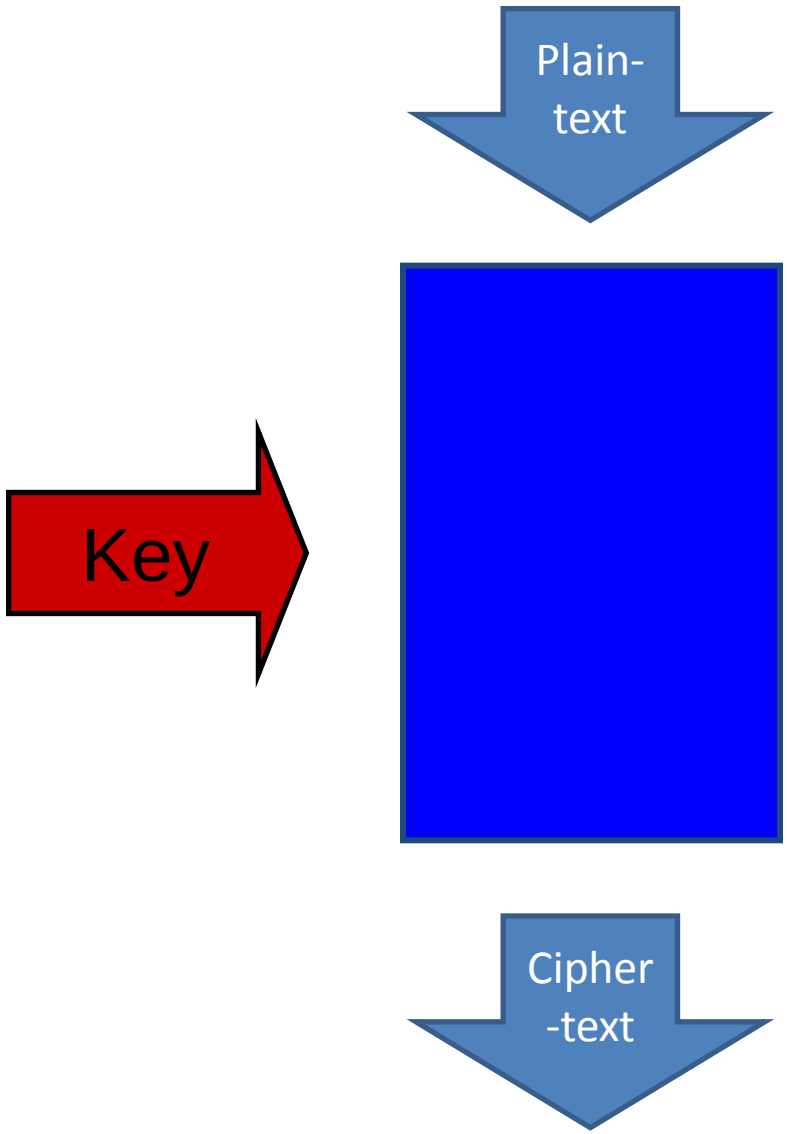
Implications on high-level structure

- No Feistel structure
- No modular additions
- SBOX-based SPN seems good choice

Encryption/Decryption

- All components involutive?
  - Big constraint on choices
- Feistel?
  - Key schedule overhead
- Other options?

What is a block cipher?

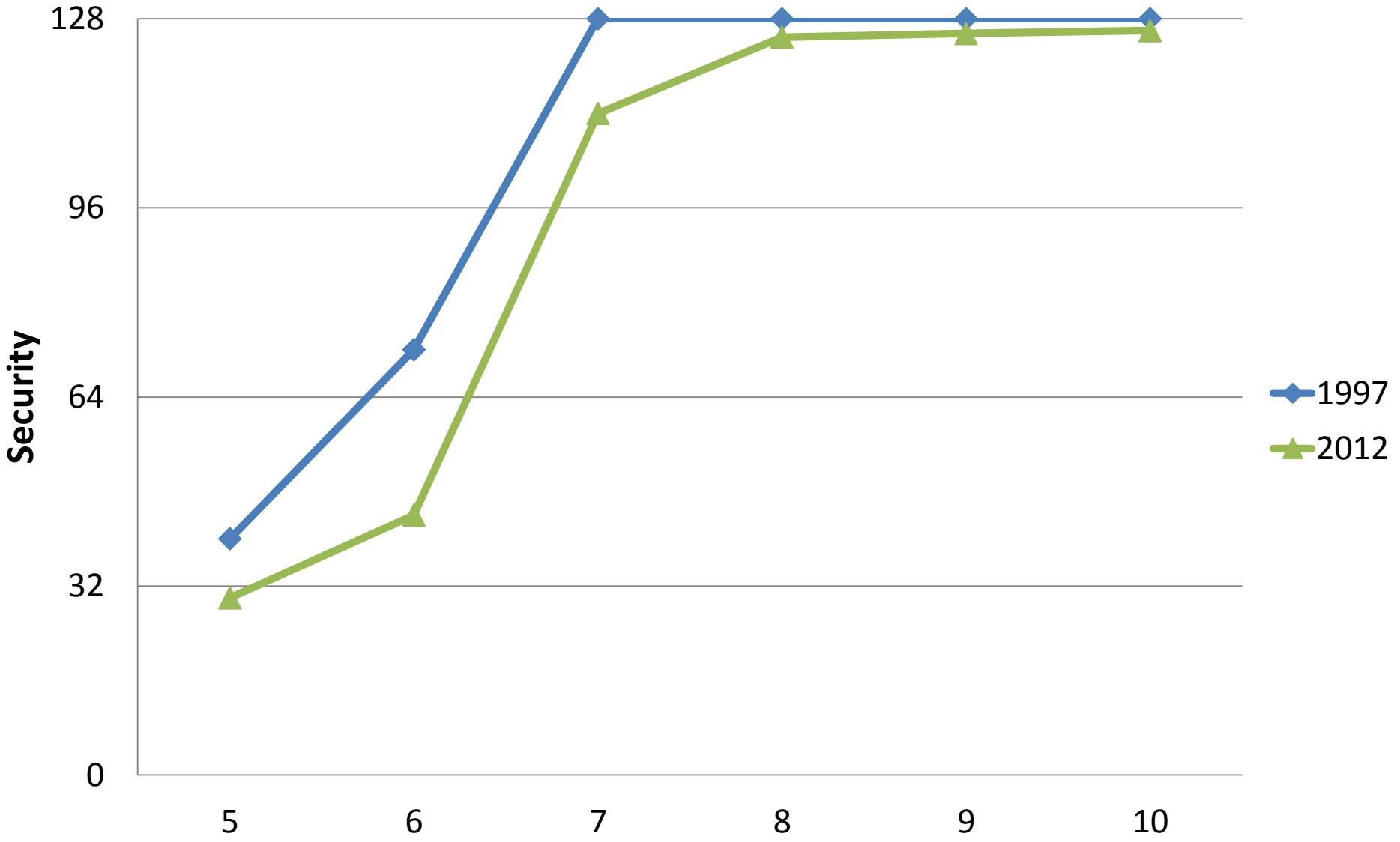


"Ideal" if

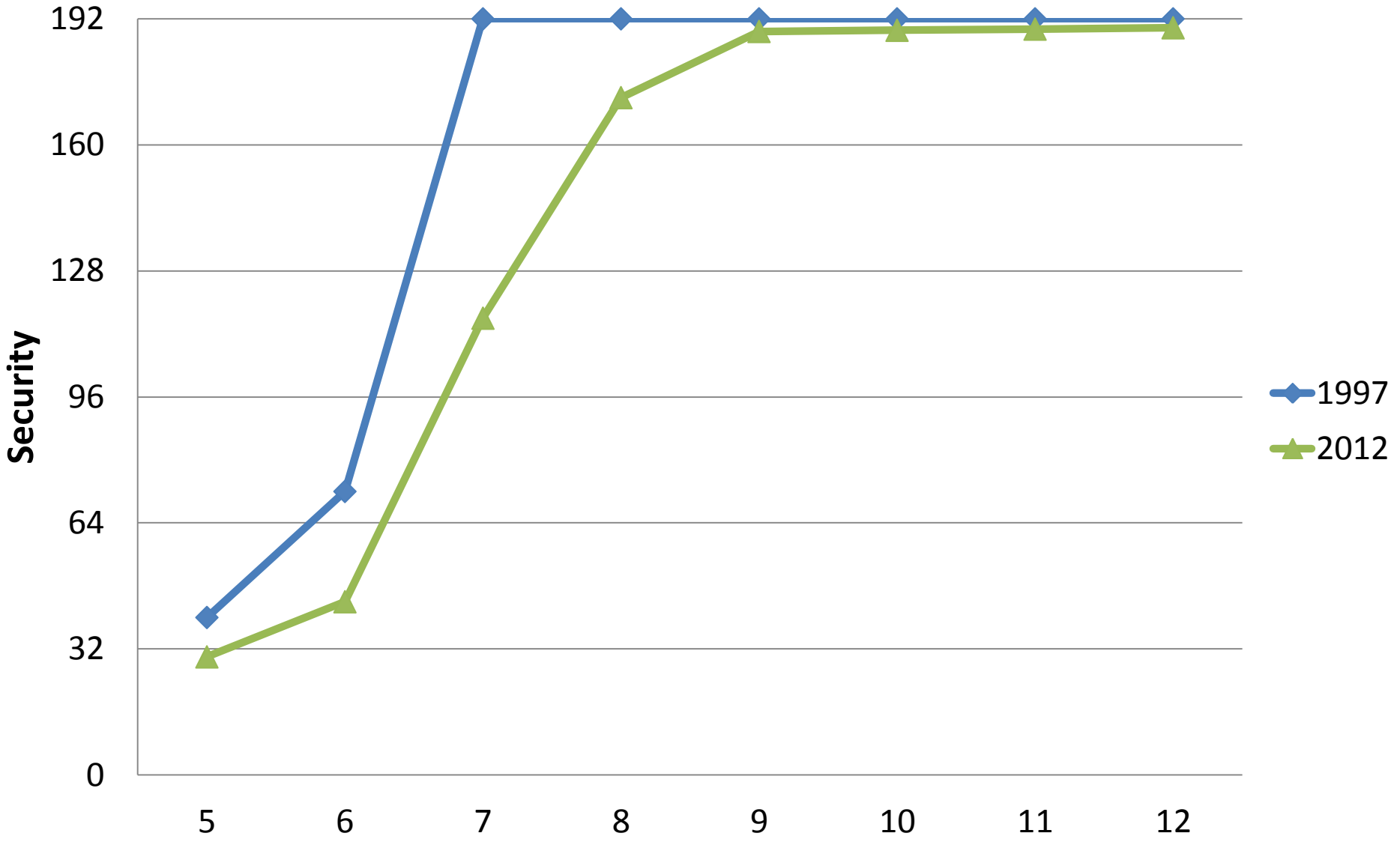
1) Knowledge of a set of plaintext/ciphertext pairs does not allow to deduce new plaintext/ciphertext pairs

2) Finding a key requires testing all keys

Evolution of AES-128 security



Evolution of AES-192 security



Resembling an ideal cipher?

- For a “lightweight” cipher, this is maybe too much to ask for?

- Related-key attacks may not be relevant
- High data-complexity attacks are not too important
  - How to formulate this in a security claim?

PRINCE - A Low-latency Block Cipher for Pervasive Computing Applications

by

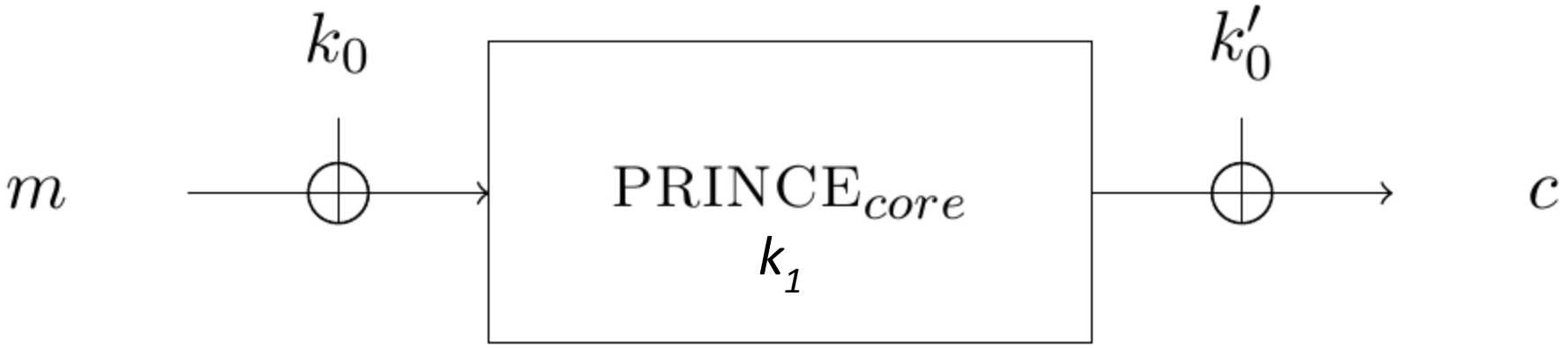
Julia Borghoff and Anne Canteaut and Lars R. Knudsen and Gregor Leander and Christan Rechberger and Soeren S. Thomsen (DTU)

Elif Bilge Kavun and Tolga Yalcin and Tim Güneysu and Christof Paar (RUB)

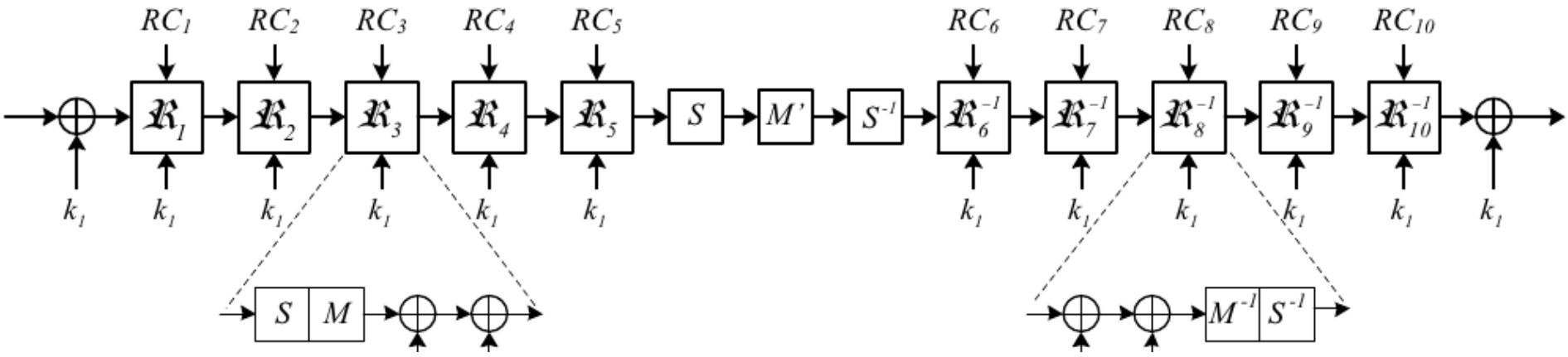
Miroslav Knezevic and Ventzi Nikov and Peter Rombouts (NXP)

PRINCE: Overview

- Claim is **126-n bit security** for an adversary with access to 2^n input/output pairs
- FX construction (similar to DES-X)



PRINCEcore



S-layer
M-layer
RC_i-add
k_1-add

$\Re_i$

k_1-add
RC_i-add
M^{-1}-layer
S^{-1}-layer

$\Re_i^{-1}$

PRINCEcore details

- S-layer: 4-bit sbox
- M-layer: only M' is an involution, M is SR o MR'
- ki-add: master key is simply added as round key
- RCi-add: constants have high HW but
  have special structure

PRINCEcore details

- S-layer: 4-bit sbox
- M-layer: only M' is an involution, M is SR o MR'
- ki-add: master key is simply added as round key
- RCi-add: constants have high HW but have special structure

RC_0	0000000000000000
RC_1	13198a2e03707344
RC_2	a4093822299f31d0
RC_3	082efa98ec4e6c89
RC_4	452821e638d01377
RC_5	be5466cf34e90c6c
RC_6	7ef84f78fd955cb1
RC_7	85840851f1ac43aa
RC_8	c882d32f25323c54
RC_9	64a51195e0e3610d
RC_{10}	d3b5a399ca0c2399
RC_{11}	c0ac29b7c97c50dd

RC_0	0000000000000000
RC_1	13198a2e03707344
RC_2	a4093822299f31d0
RC_3	082efa98ec4e6c89
RC_4	452821e638d01377
RC_5	be5466cf34e90c6c
RC_6	7ef84f78fd955cb1
RC_7	85840851f1ac43aa
RC_8	c882d32f25323c54
RC_9	64a51195e0e3610d
RC_{10}	d3b5a399ca0c2399
RC_{11}	c0ac29b7c97c50dd

PRINCEcore details

- S-layer: 4-bit sbox
- M-layer: only M' is an involution, M is SR o MR'
- ki-add: master key is simply added as round key
- RCi-add: constants have high HW but have special structure

RC_0	0000000000000000
RC_1	13198a2e03707344
RC_2	a4093822299f31d0
RC_3	082efa98ec4e6c89
RC_4	452821e638d01377
RC_5	be5466cf34e90c6c
RC_6	7ef84f78fd955cb1
RC_7	85840851f1ac43aa
RC_8	c882d32f25323c54
RC_9	64a51195e0e3610d
RC_{10}	d3b5a399ca0c2399
RC_{11}	c0ac29b7c97c50dd

$RC_i + RC_{11-i}$ = c0ac29b7c97c50dd !!!

$PRINCE_{core}$ key-schedule

Master key: k

Round keys: ki

First half of the rounds: ki = k

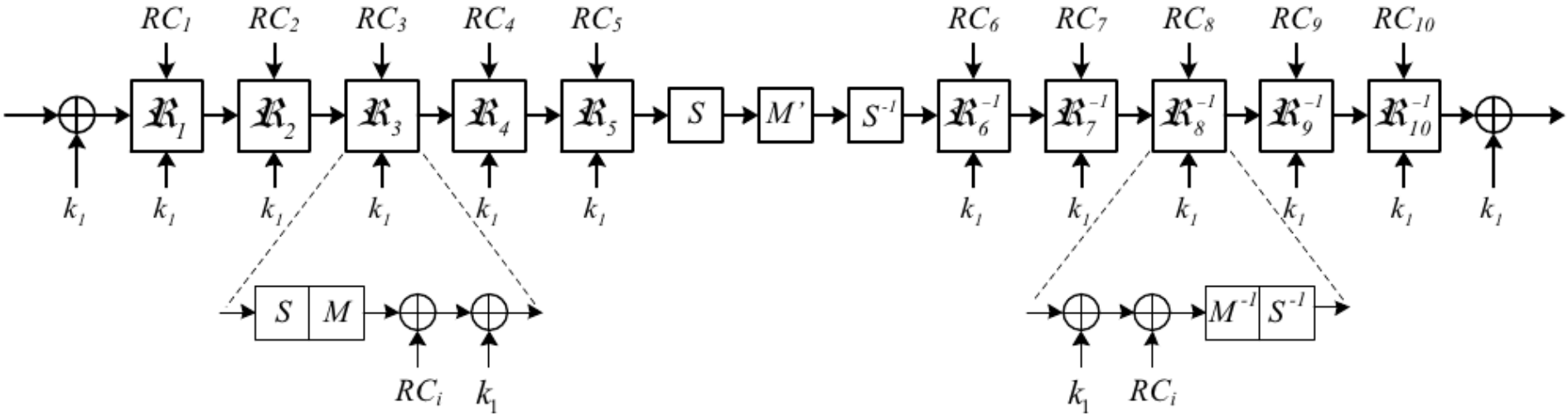
Second half of the rounds: ki = k+Alpha

Alpha-reflection property

Since M' is involution,

$$\text{PRINCEcore}_k(x) = \text{PRINCEcore}^{-1}_{k+\text{Alpha}}(x)$$

Allows for very simple implementation of decryption



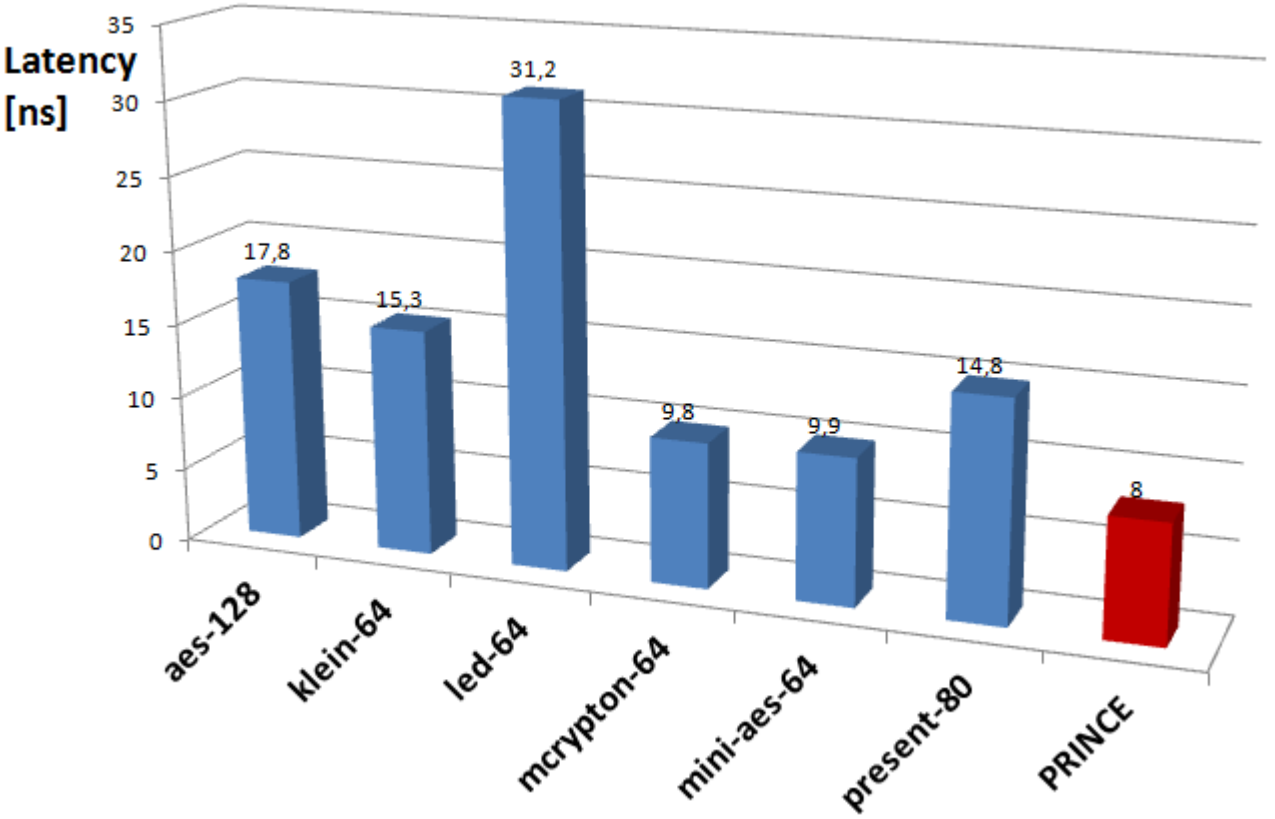
Related-key attacks

- May not be relevant in many settings
- However: Using very strong related-key properties, it is possible to speed-up key recovery in the single-key model. E.g. attack on eStream candidate Moustique
- For Prince this leads to a loss of 1-bit of security in a straight-forward way

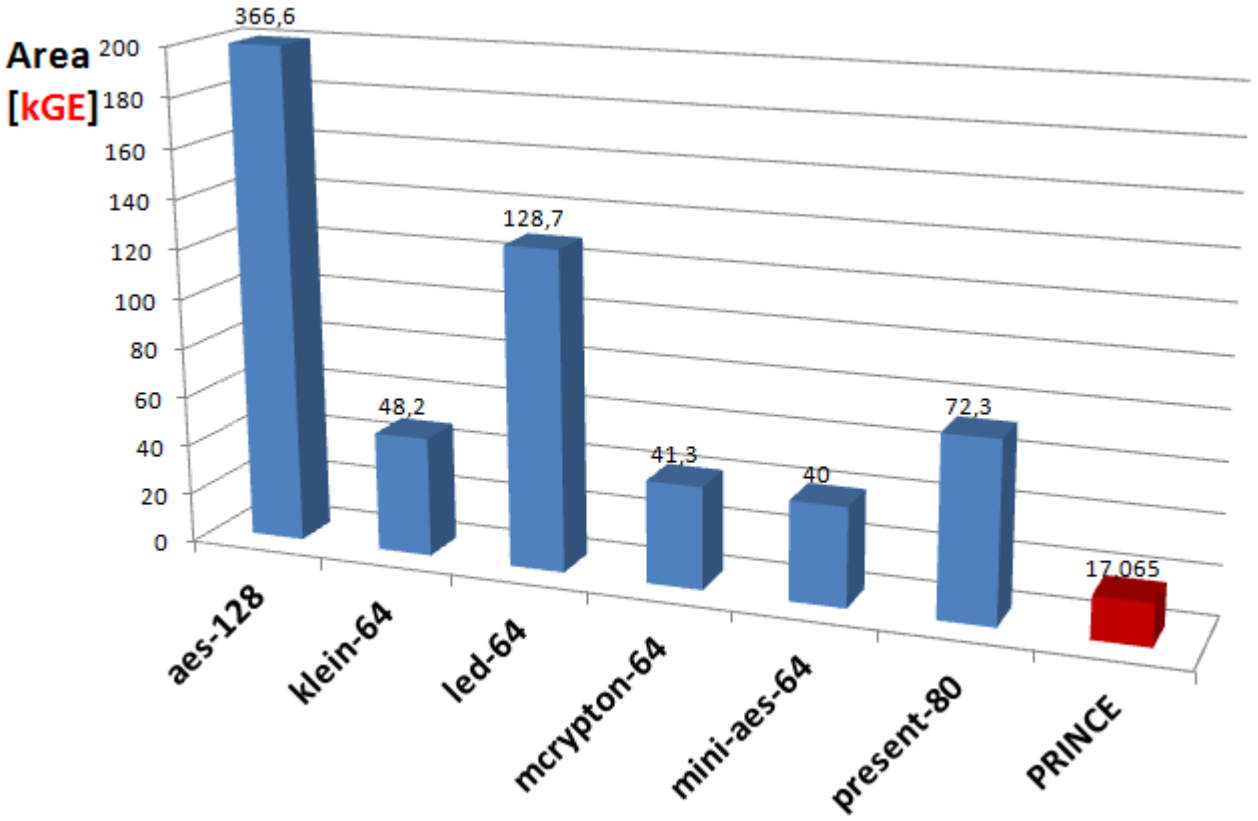
Advantages of PRINCE

- Decryption for free (=encryption with related key)
- Alpha-reflection method better than choosing all components to be involutions: More choice for Sboxes
- Less multiplexers needed
- Generic reductionist proof possible
- Small number of relatively simple rounds → low latency
- Bounds against various classical attacks (wide-trail strategy) applicable, but still lightweight building blocks

Latency comparison



Area comparison



Symmetric crypto research → real world (1/2)

- Consolidating lots of research on s-boxes, linear layer construction, SPN designs…
- Meets very tough constraints from industry

Symmetric crypto research → real world (2/2)

- We convinced NXP management to allow us to publish the design ideas + security analysis (AC 2012)
  - Lots of “free” external cryptanalysis already after 1 year, increases confidence. Even more now, 3 years later.

- Both sides are happy:
  - Industry gets problems solved, plan for global deployment in a few years.
  - Researchers get interesting problems to work on
  - Inspires both theory and practice

Selected cryptanalysis

- Reflection Cryptanalysis of Prince-like ciphers, FSE 2013 and JoC
- Security Analysis of Prince, FSE 2013
- Sieve-in-the-middle: Improve MITM Attacks, Crypto 2013
- Improved MITM Attacks on AES-192 and Prince
- On the Security of the core of PRINCE against Biclique and Differential attacks
- Multiple-differential attacks on Round-Reduced Prince, FSE 2014
- Multi-user collisions: Applications to Discrete Logs, Even-Mansour and Prince, Asiacrypt 2014
- Cryptanalytic Time-Memory-Data Tradeoffs for FX-Constructions with Applications to PRINCE and PRIDE
- Various side-channel and fault attack papers

Early cryptanalysis

- All those focus to achieve as many rounds as possible, even at the cost of getting very close to the $D*T<2^{126}$ bound.

- How to change the incentives?

Input from Industry

- Care about cryptanalysis
- Care about practical attacks
- Was usually not very concrete

The 15.000 EUR
PRINCE cryptanalysis competition
makes it more concrete

The PRINCE Challenge

Setting 1: Given 2^{20} chosen plaintexts/ciphertexts

- How fast can you break 4 rounds?
- How fast can you break 6 rounds?
- How fast can you break 8 rounds?
- How fast can you break 10 rounds?
- How fast can you break 12 rounds?

The PRINCE Challenge

Setting 2: Given 2^{30} known plaintexts

- How fast can you break 4 rounds?
- How fast can you break 6 rounds?
- How fast can you break 8 rounds?
- How fast can you break 10 rounds?
- How fast can you break 12 rounds?

Timeline

Start in March 2014

Round 1 (August 2014)

Round 2 (April 2015)

Round 3 (April 2016)

Winners of round 1

Patrick Derbez

SnT, University of Luxembourg

Léo Perrin

SnT, University of Luxembourg

Paweł Morawiecki

Polish Academy of Sciences, Computer Science Institute, and Kielce University of Commerce, Poland

Winners of round 2

PATRICK DERBEZ

Best results in the 8-round CP category

RALUCA POSTEUCĂ and GABRIEL NEGARĂ

Best results in the 6-round CP category

The PRINCE Challenge

Setting 1: Given 2^{20} chosen plaintexts/ciphertexts

- How fast can you break 4 rounds?
  - Winner: Pawel, 2^{7} CP, time 2^{11}
- How fast can you break 6 rounds?
  - Winners: Patrick, 2^{16}CP, time $2^{33.7}$ and Léo 2^{15}CP, time 90min
  - **New Winner: Raluca and Gabriel , $2^{14.6}$CP, time 2^{37}**
- How fast can you break 8 rounds?
  - Winner: Patrick: 2^{16}CP, time 2^{50}-2^{67}
  - **New Winner: Patrick: 2^{16}CP, time $2^{66.4}$**
- How fast can you break 10 rounds?
- How fast can you break 12 rounds?

The PRINCE Challenge

Setting 2: Given 2^{30} known plaintexts

- How fast can you break 4 rounds?
  - Patrick: 2^5 KP, time 2^{43}
- How fast can you break 6 rounds?
  - Patrick: 2^6 KP, time 2^{101}
- How fast can you break 8 rounds?
- How fast can you break 10 rounds?
- How fast can you break 12 rounds?

Prizes

- Best result for …
  - 4-round challenges: Chocolate/Beer
  - 6-round challenges: Chocolate/Beer
  - 8-round challenges: Chocolate/Beer
  - 10-round challenges: Chocolate/Beer
  - 12-round challenges: more Chocolate/Beer

- First attack with less than 2^{64} time, 2^{45} bytes memory on…
  - **8-rounds: 1.000 Euros**
  - **10-round: 4.000 Euros**
  - **12-round: 10.000 Euros**

Round 3

submit convincing technical report to

prince-challenge@compute.dtu.dk

- Deadline: End of April 2016, before Eurocrypt

- Committee:
  - Gregor Leander (RUB)
  - Ventzi Nikov (NXP)
  - Christian Rechberger (DTU)
  - Vincent Rijmen (KUL)

Conclusions

- "Lightweight" cipher should **not** (only) mean
  - Lightweight security
  - Low gate-count
  - Low latency

  …but simply an evolution of the state of the art almost 2 decades after the design on Rijndael/AES
- Ciphers are only core building blocks
- Time for industry to benefit from recent developments in academia?
- Prince seems to be a competitive lightweight cipher
  - Even for completely unrelated homomorphic computations (see e.g. eprint 2014/233)

Last-Minute Addendum

Table seen earlier this morning during the SIMON/SPECK presentation

algorithm	area (GE)	latency (ns)	clock (MHz)
PRINCE	9522	22.9	43.7
SIMON 64/128	9516	22.88	437.1
	5072	31.90	344.9
SPECK 64/128	6377	52.36	191.0

This means that PRINCE needs about 10x less power and 10xless energy
PRINCE implemented in the same architecture as Simon/Speck would need much less area.