# Ultra-Lightweight Cryptography

F.-X. Standaert
*UCL Crypto Group*

European brokerage event, Cryptography
Paris, September 2016

# Outline

- Introduction
- Symmetric cryptography
  - Hardware implementations
  - Software implementations
  - Technology scaling
- Conclusion

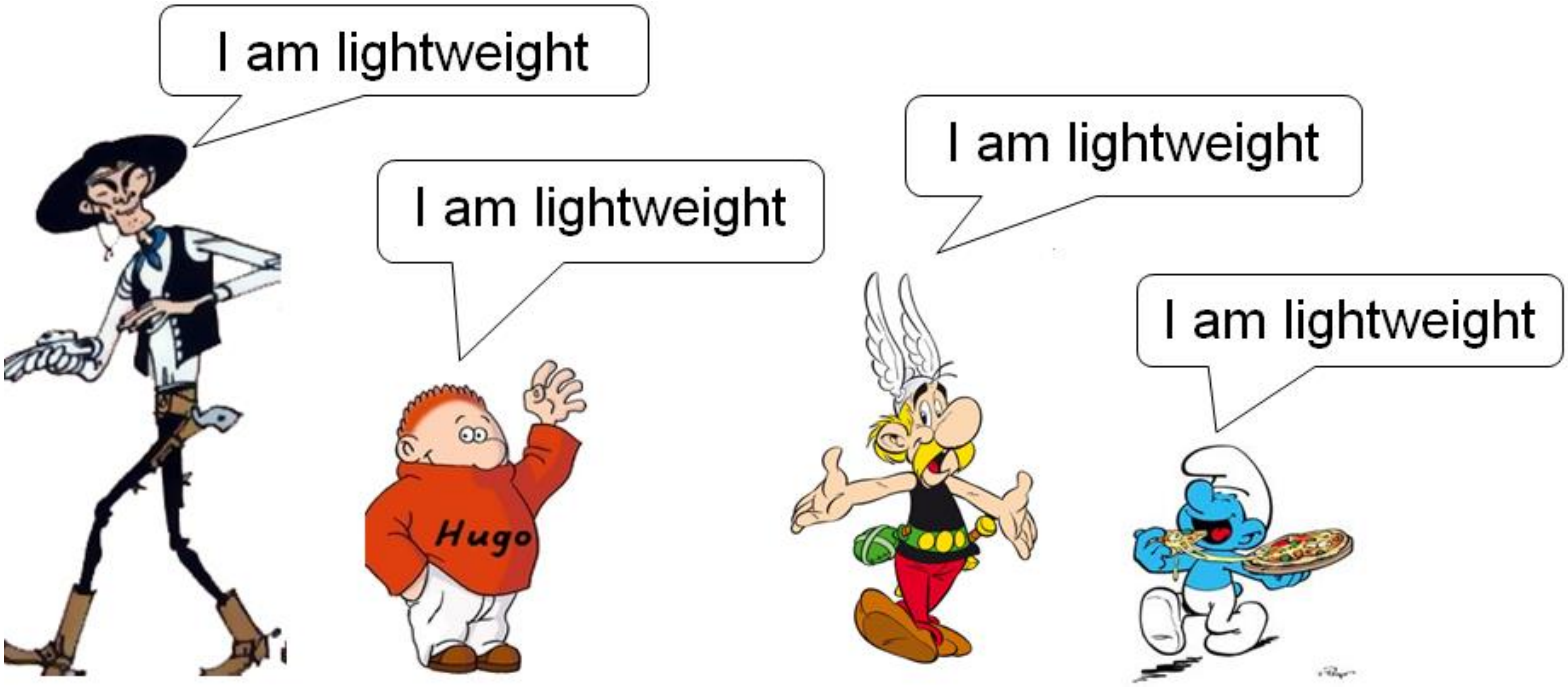
# Outline

- **Introduction**
- Symmetric cryptography
  - Hardware implementations
  - Software implementations
  - Technology scaling
- Conclusion

# Outline



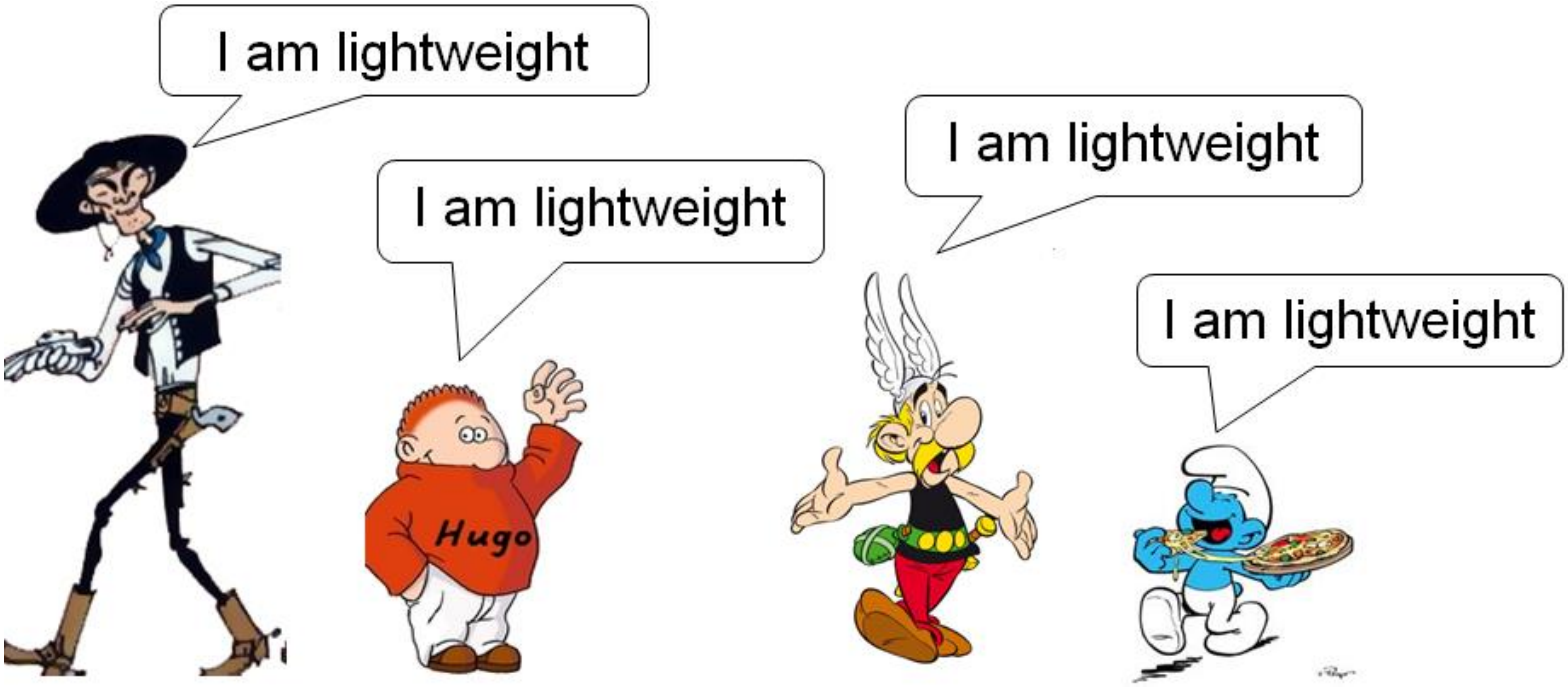
- Evaluation criteria are usually relative…



# Outline



- Evaluation criteria are usually relative…




… and reflect algorithmic & implementation choices

# Outline

- Introduction
- **Symmetric cryptography**
  - **Hardware implementations**
  - Software implementations
  - Technology scaling
- Conclusion

# Performance metrics



| application constraints | physical units | relative to | pre-layout units | HW design goals | algorithmic design goals | relevance w.r.t. algorithms |
|---|---|---|---|---|---|---|
| AREA | um² | time or energy constraints | #gates | share resources | reduce components cost & versatility | weakly discriminant *** |
| INST. POWER (dynamic) | W (J/sec) | time or energy constraints | switching activity | reduce datapath | reduce components cost & versatility | somewhat arbitrary ** |
| THROUGHPUT | bit/sec | area or power constraints | #cycles (& block size) | unroll, parallelize & pipeline | minimize the total combinatorial cost | very arbitrary * |
| ENERGY | J/enc, J/bit | area or power constraints | #cycles X POWER | unroll | minimize the total combinatorial cost | somewhat discriminant *** |

# Performance metrics



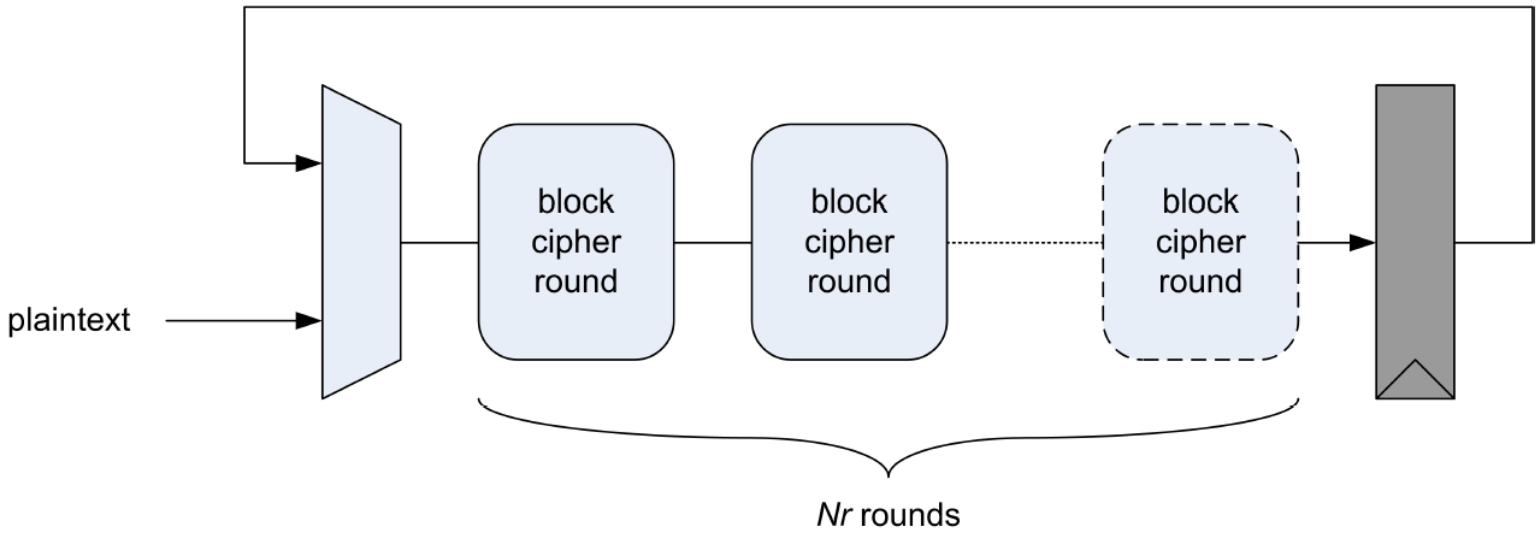
| application constraints | physical units | relative to | pre-layout units | HW design goals | algorithmic design goals | relevance w.r.t. algorithms |
|---|---|---|---|---|---|---|
| AREA | um² | time or energy constraints | #gates | share resources | reduce components cost & versatility | weakly discriminant *** |
| INST. POWER (dynamic) | W (J/sec) | time or energy constraints | switching activity | reduce datapath | reduce components cost & versatility | somewhat arbitrary ** |
| THROUGHPUT | bit/sec | area or power constraints | #cycles (& block size) | unroll, parallelize & pipeline | minimize the total combinatorial cost | very arbitrary * |
| ENERGY | J/enc, J/bit | area or power constraints | #cycles X POWER | unroll | minimize the total combinatorial cost | somewhat discriminant *** |

=> Can be more or less reflective of algorithms

# Case study



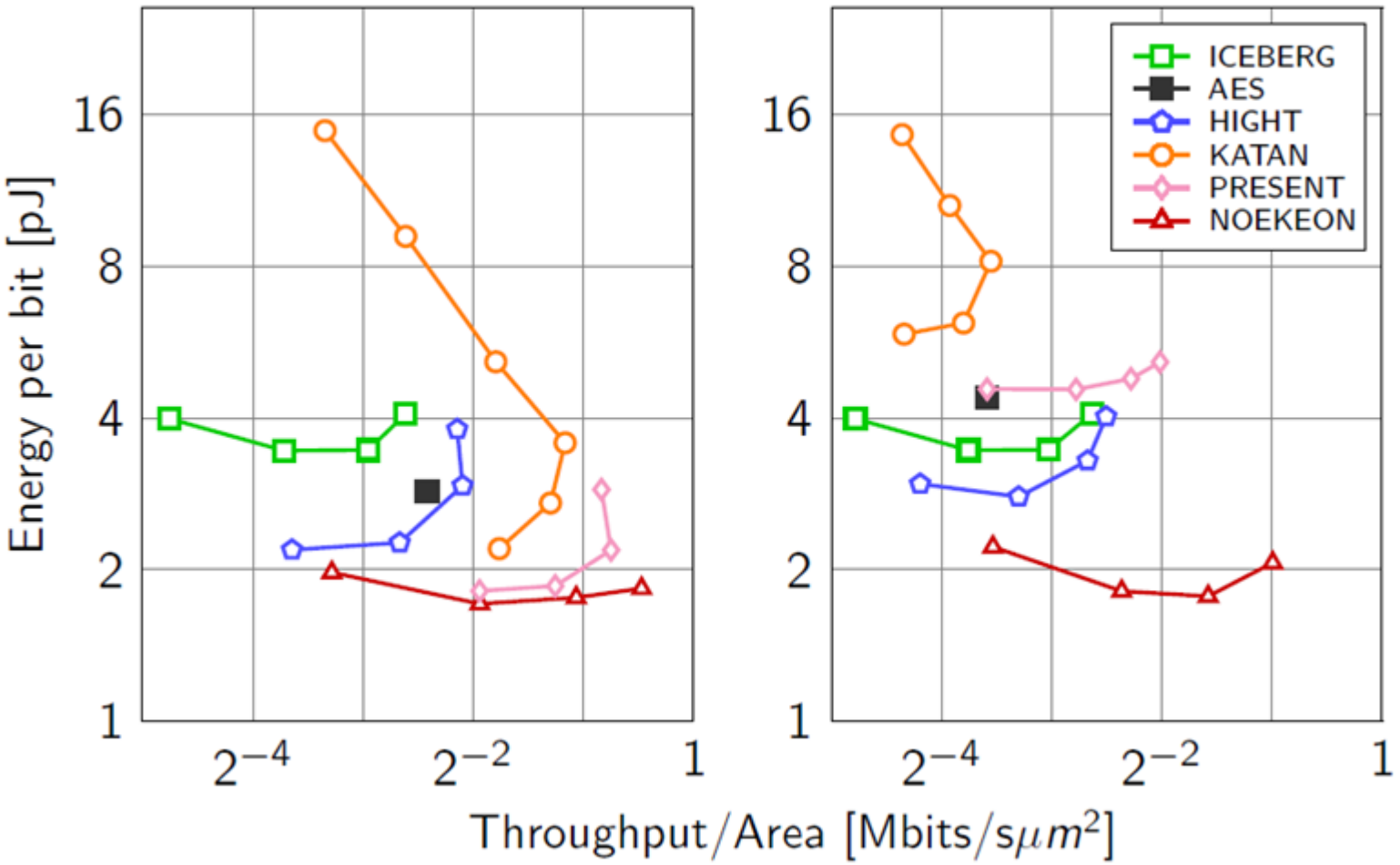
- Flexible block cipher architecture




- 3 core options: enc., dec., enc./dec.
- 65-nanometer CMOS technology

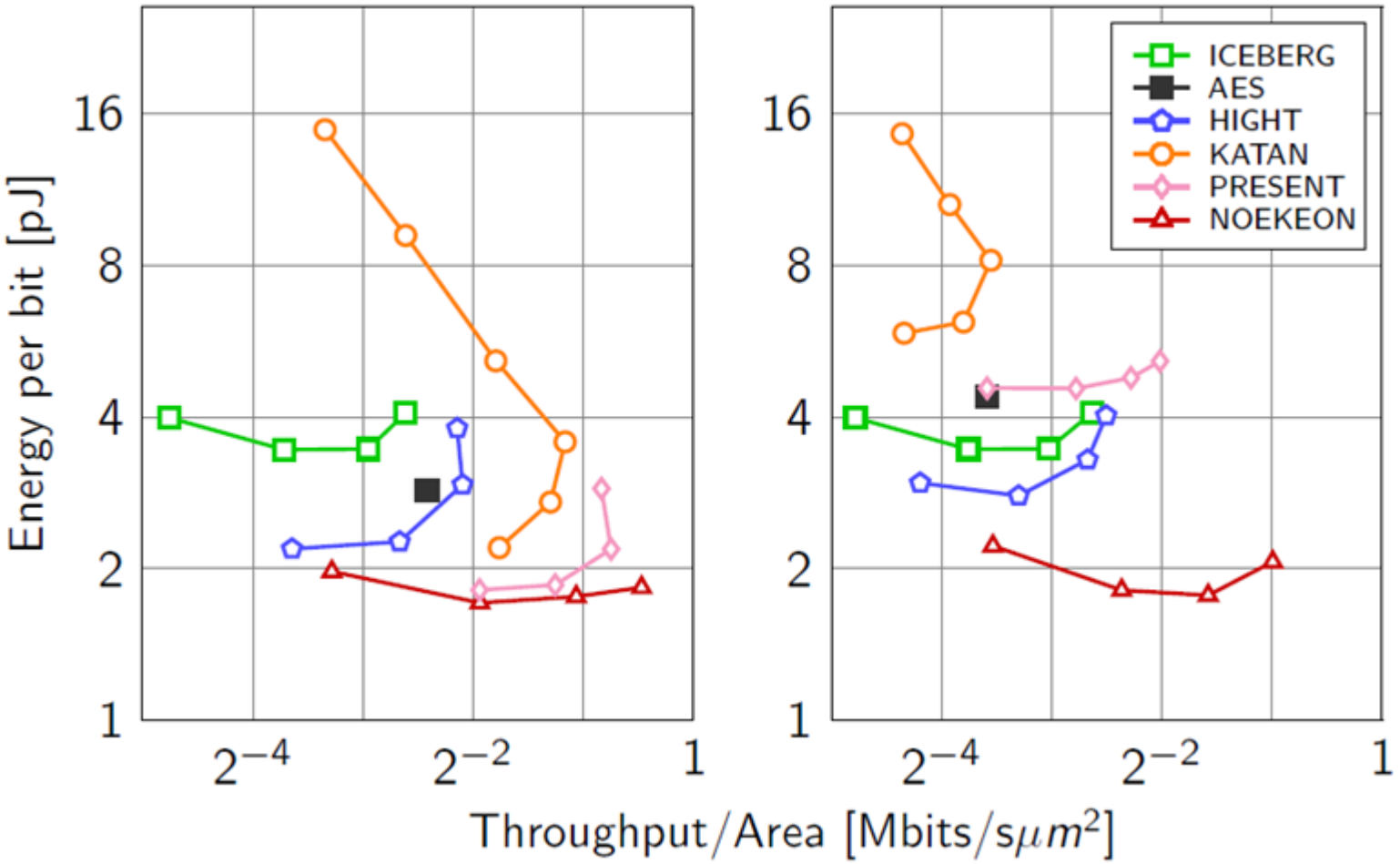
# Combined metric (Enc., Enc./Dec.)





# Combined metric (Enc., Enc./Dec.)

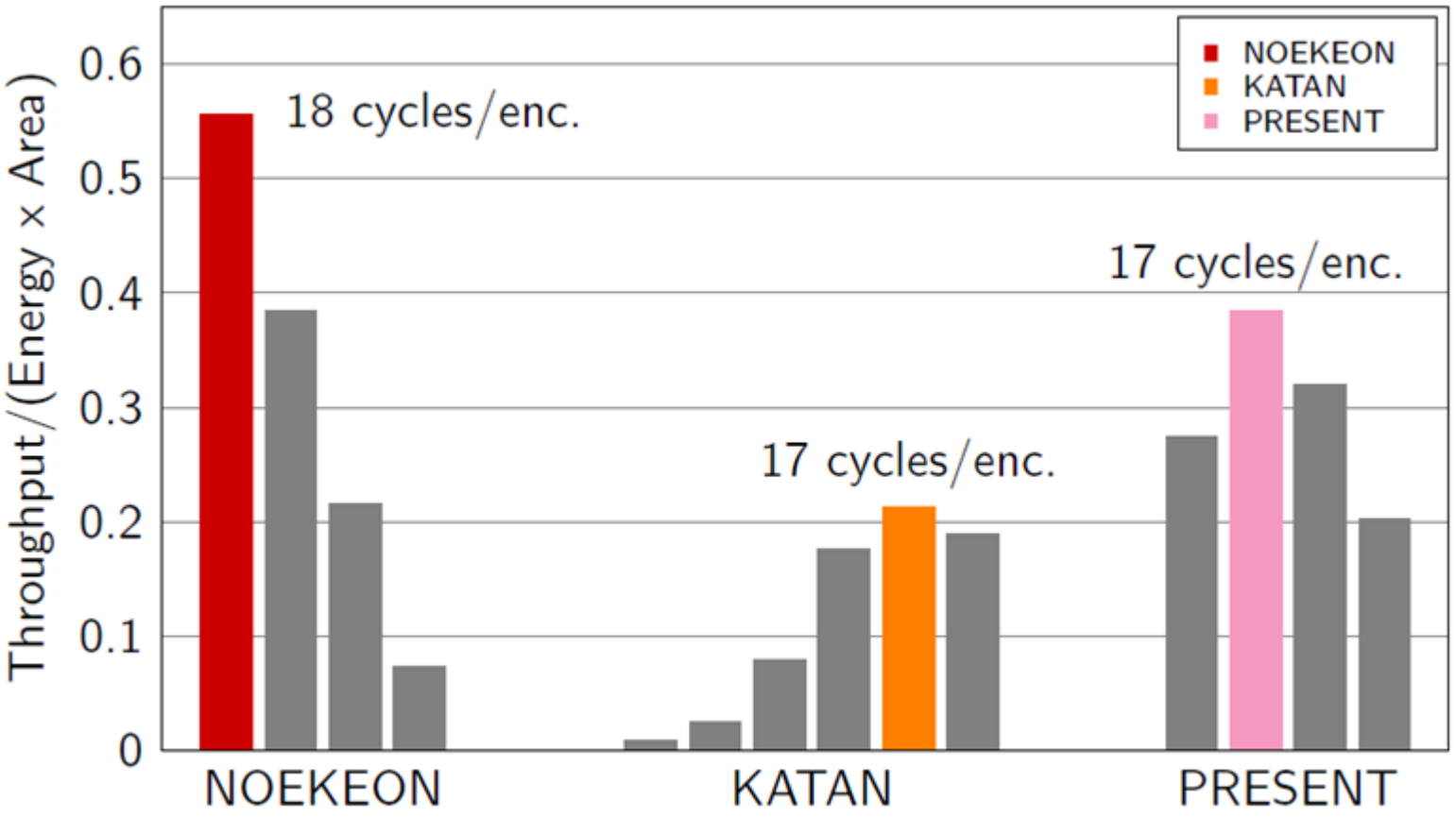





- Mostly reflects different key schedulings

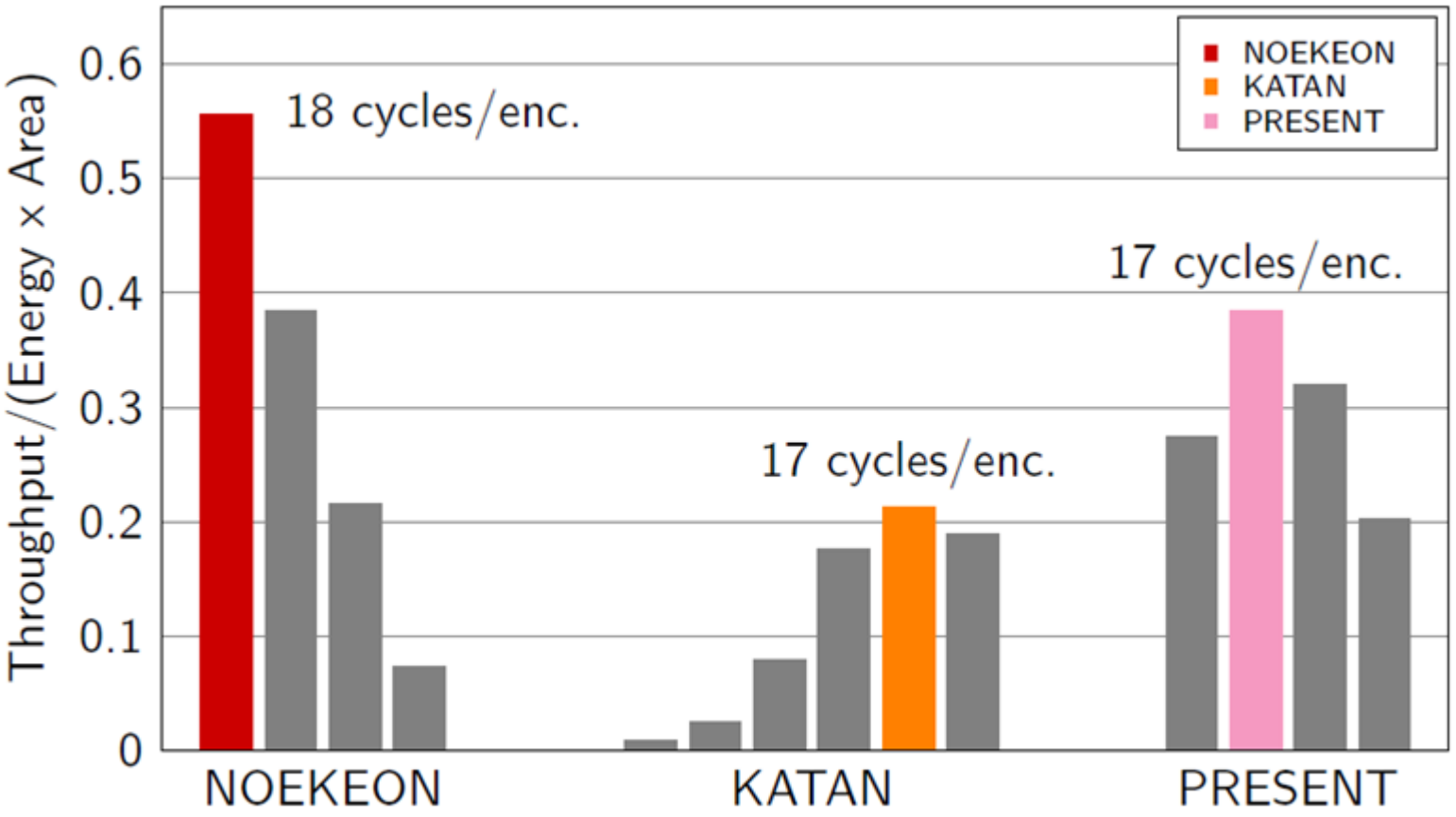
# Number of rounds per cycle (Enc.)





# Number of rounds per cycle (Enc.)





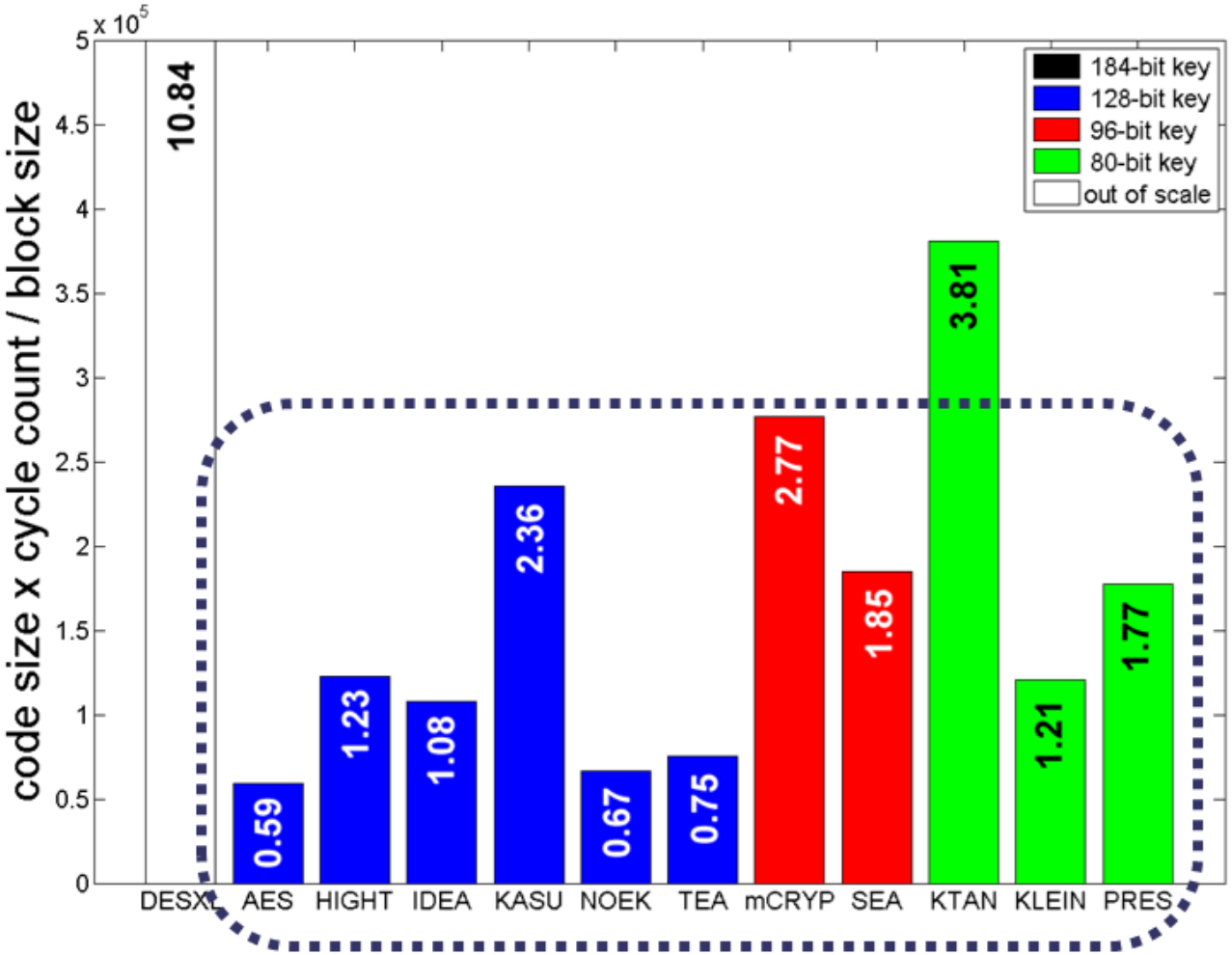

- Suggests “complexity limit” has been reached

# Outline

- Introduction
- **Symmetric cryptography**
  - Hardware implementations
  - **Software implementations**
  - Technology scaling
- Conclusion

# Combined metric (Atmel AVR, Enc.)






- Time vs. code size tradeoff (because HW is fixed)

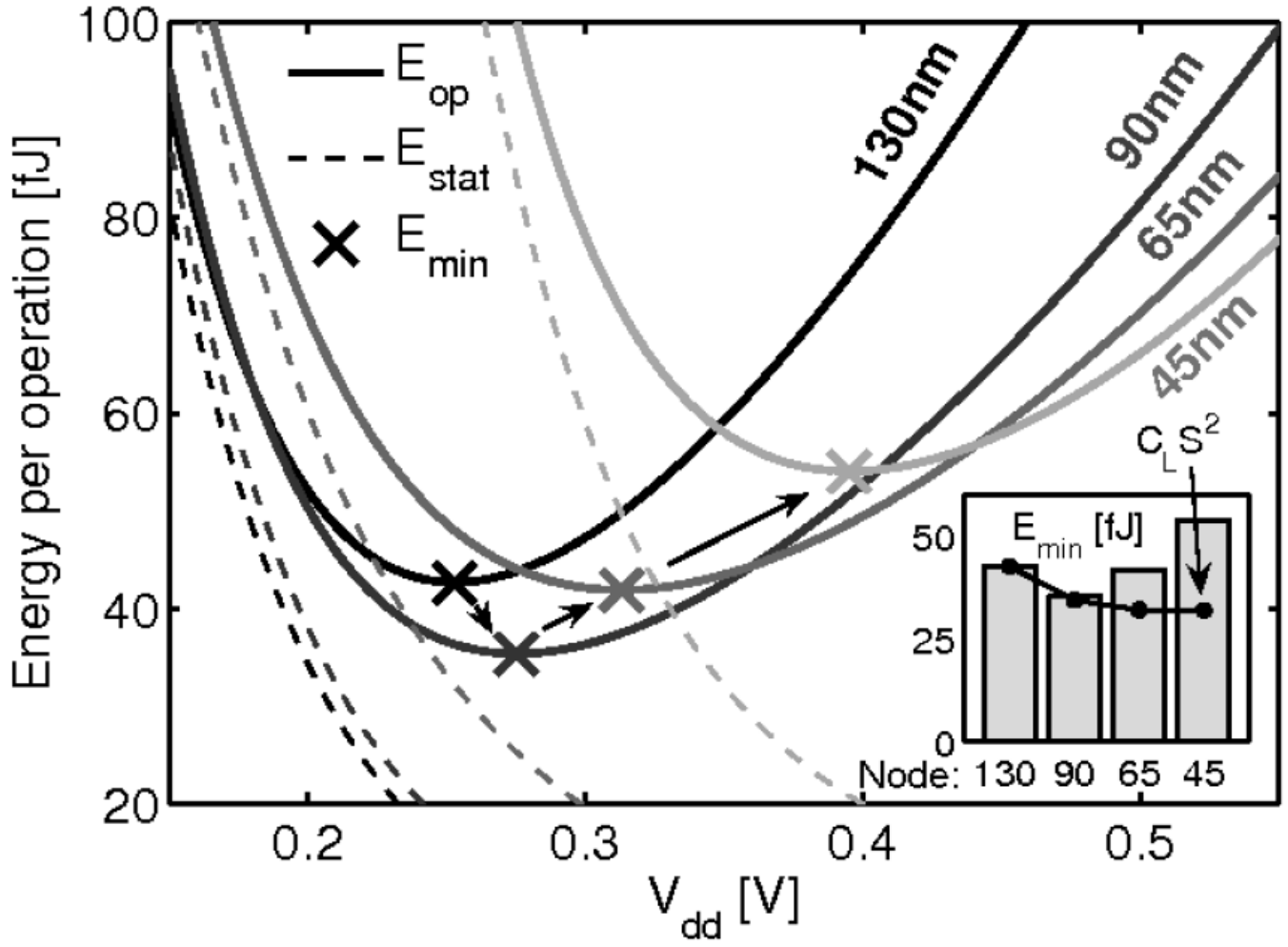
# Outline

- Introduction
- **Symmetric cryptography**
  - Hardware implementations
  - Software implementations
  - **Technology scaling**
- Conclusion

# Technology scaling

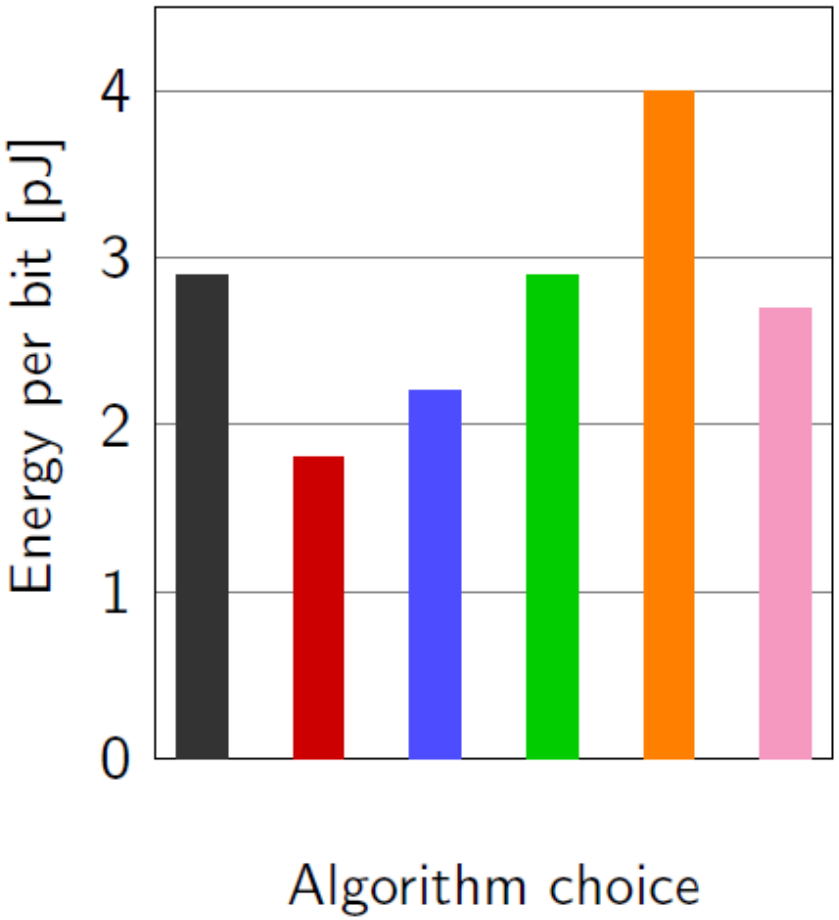


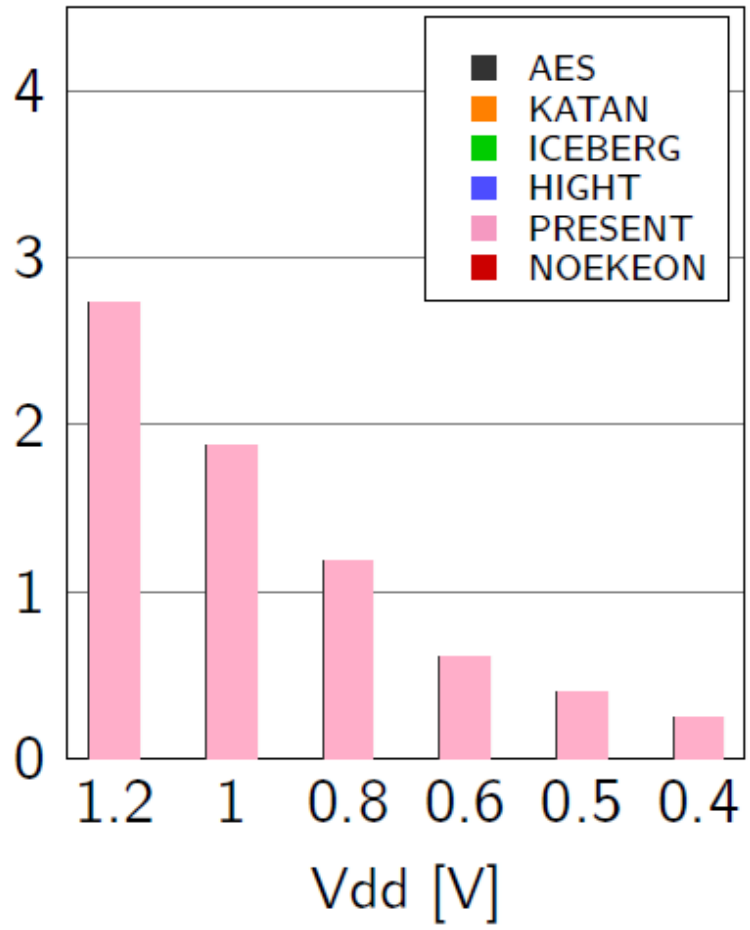
- Between and within technologies (*f* and Vdd)



# 65-nanometer case study

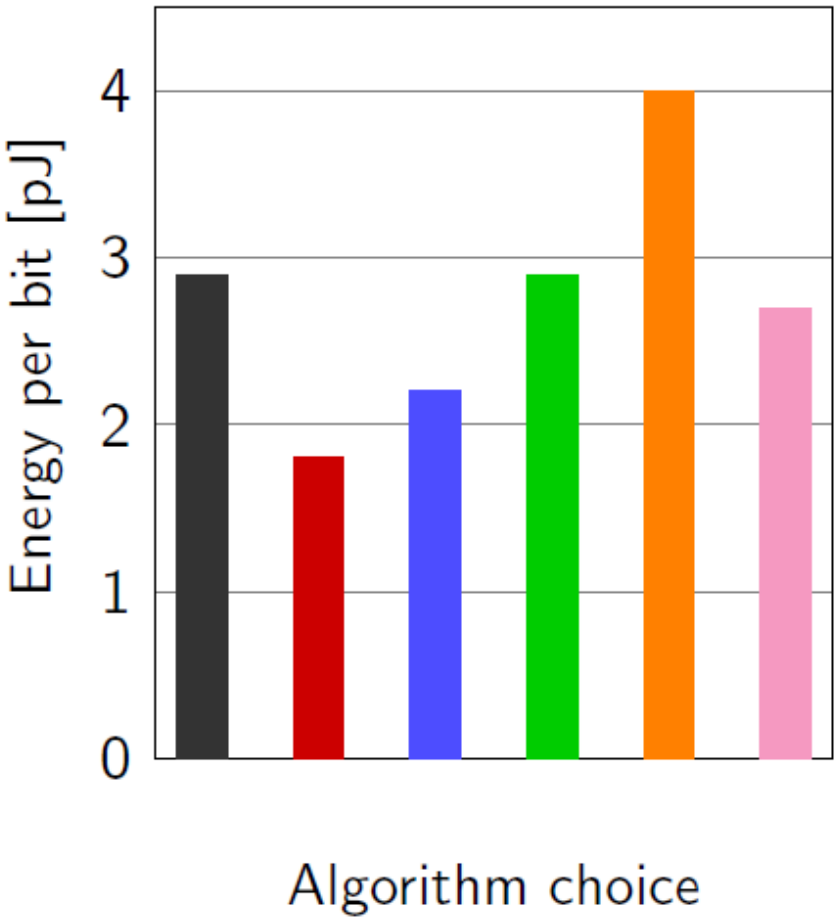


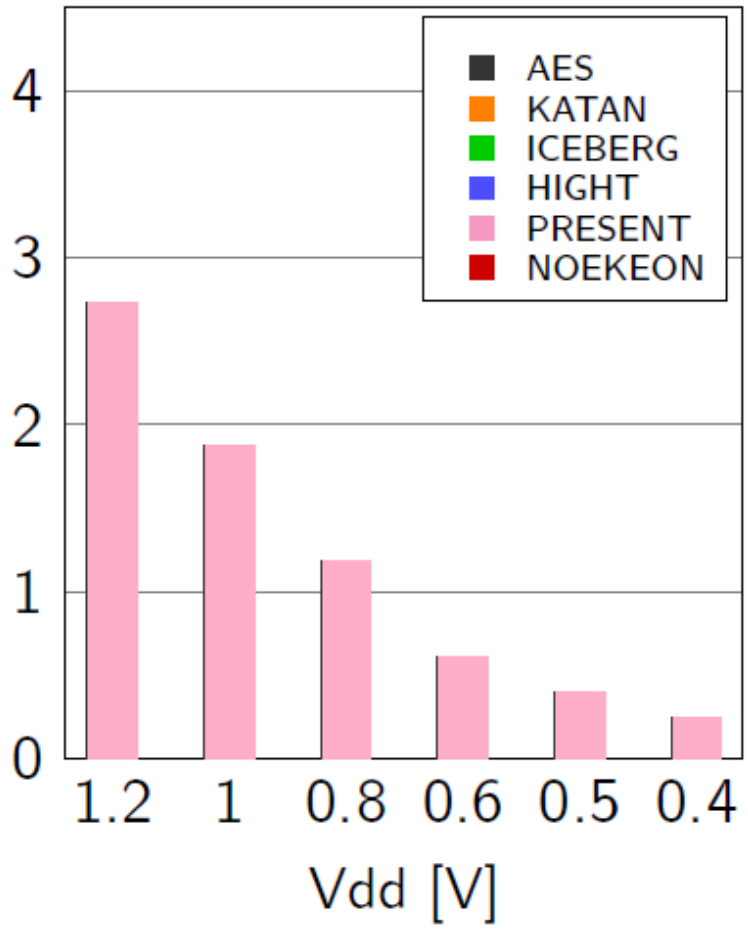


# 65-nanometer case study






- Questions the relevance of algorithmic changes

# Outline

- Introduction
- Symmetric cryptography
  - Hardware implementations
  - Software implementations
  - Technology scaling
- **Conclusion**

# Lessons learned



- Gate count limit probably reached for rounds
- Ciphers differ more by other aspects, e.g.
  - Key scheduling
  - Enc./Dec. combinations

# Lessons learned



- Gate count limit probably reached for rounds
- Ciphers differ more by other aspects, e.g.
  - Key scheduling
  - Enc./Dec. combinations

- Simple & regular designs help (a lot)
  - Compact implementations more “revealing”
- Technology scaling (mostly) helps

# Lessons learned



- Gate count limit probably reached for rounds
- Ciphers differ more by other aspects, e.g.
  - Key scheduling
  - Enc./Dec. combinations

- Simple & regular designs help (a lot)
  - Compact implementations more “revealing”
- Technology scaling (mostly) helps

- AES is already quite lightweight
- NOEKEON is ultra lightweight

# Research challenges



- How to design a key scheduling algorithm?
- How to efficiently combine Enc. and Dec.?

# Research challenges



- How to design a key scheduling algorithm?
- How to efficiently combine Enc. and Dec.?

- ***New metrics for new applications?***

# Research challenges



- How to design a key scheduling algorithm?
- How to efficiently combine Enc. and Dec.?

- ***New metrics for new applications?***
  - *Low energy for IoT (e.g., Midori, NOEKEON)*

# Research challenges



- How to design a key scheduling algorithm?
- How to efficiently combine Enc. and Dec.?

- ***New metrics for new applications?***
  - *Low energy for IoT (e.g., Midori, NOEKEON)*
  - *Low latency for bus encryption (e.g., Prince)*

# Research challenges



- How to design a key scheduling algorithm?
- How to efficiently combine Enc. and Dec.?

- ***New metrics for new applications?***
  - *Low energy for IoT (e.g., Midori, NOEKEON)*
  - *Low latency for bus encryption (e.g., Prince)*
  - *Side-channel resistant ciphers (e.g., LS-designs)*
    - *& fault attacks, tamper resistance, …*

# Research challenges



- How to design a key scheduling algorithm?
- How to efficiently combine Enc. and Dec.?

- ***New metrics for new applications?***
  - *Low energy for IoT (e.g., Midori, NOEKEON)*
  - *Low latency for bus encryption (e.g., Prince)*
  - *Side-channel resistant ciphers (e.g., LS-designs)*
    - *& fault attacks, tamper resistance, …*
  - *Ciphers for MPC, FHE (privacy applications)*

# Research challenges



- How to design a key scheduling algorithm?
- How to efficiently combine Enc. and Dec.?

- ***New metrics for new applications?***
  - *Low energy for IoT (e.g., Midori, NOEKEON)*
  - *Low latency for bus encryption (e.g., Prince)*
  - *Side-channel resistant ciphers (e.g., LS-designs)*
    - *& fault attacks, tamper resistance, …*
  - *Ciphers for MPC, FHE (privacy applications)*
  - *From ciphers to modes (e.g., authenticated enc.)*

# Research challenges



- ***New metrics for new applications?***
  - *Post quantum ciphers (e.g., LPN, LWE, LWR)*

# Research challenges



- ***New metrics for new applications?***
  - *Post quantum ciphers (e.g., LPN, LWE, LWR)*
  - *Communication complexity!*
    - Sending 1 bit over a wireless channel is one order of magnitude more energy consuming than computing 1 cycle

# Research challenges



- ***New metrics for new applications?***
  - *Post quantum ciphers (e.g., LPN, LWE, LWR)*
  - *Communication complexity!*
    - Sending 1 bit over a wireless channel is one order of magnitude more energy consuming than computing 1 cycle
  - Trojan-resilience, surveillance
    - Reverse firewalls, distributed computing, …

# Research challenges



- ***New metrics for new applications?***
  - *Post quantum ciphers (e.g., LPN, LWE, LWR)*
  - *Communication complexity!*
    - Sending 1 bit over a wireless channel is one order of magnitude more energy consuming than computing 1 cycle
  - Trojan-resilience, surveillance
    - Reverse firewalls, distributed computing, …

- Implementation (at large) matters
  - Specific challenges => need of specific solutions
  - But always part of something “bigger”
  - More open source (software & hardware) needed

# THANKS

http://perso.uclouvain.be/fstandae/


http://uclouvain.be/crypto/