Efficient Pairings and ECC for Embedded Systems

Thomas Unterluggauer, Erich Wenger
thomas.unterluggauer@iaik.tugraz.at
erich.wenger@iaik.tugraz.at

CHES, Busan, Korea
September 23-26, 2014

Overview

- Pairing- and ECC-based cryptography
- High HW/SW requirements vs. embedded scenario
- Three ARM Cortex-M0+-based designs:
  - Pure software
  - MAC instruction-set extension
  - Drop-in hardware accelerator
- 162 ms @ 48 MHz for 254-bit Barreto-Naehrig curve
- 10.1 kGE for drop-in and 49 kGE in total

Outline

1 Bilinear Pairings

2 HW/SW Platform

3 Drop-in Module

4 Results and Related Work

5 Conclusion

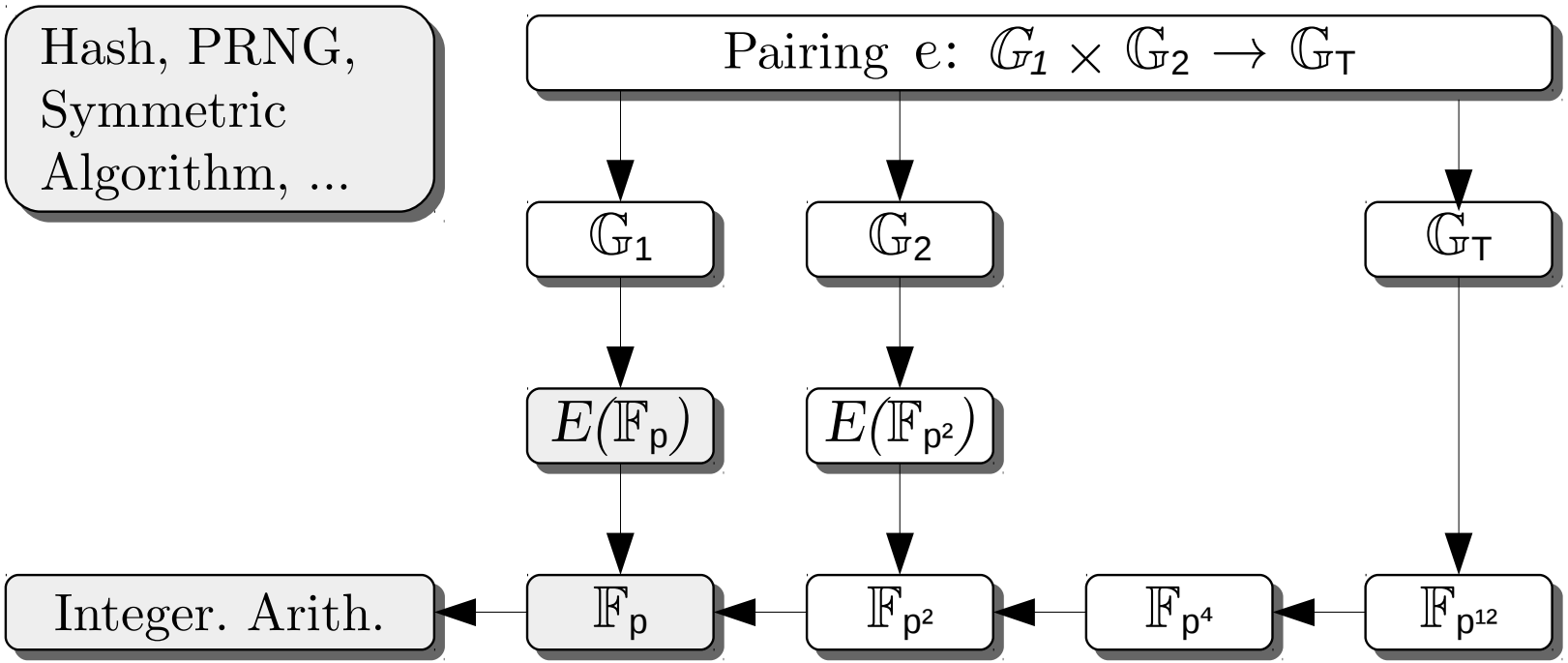
Bilinear Pairings

- $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ are cyclic groups of order n.
- Bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$.
  - Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$.
  - Non-degeneracy: $e(P, Q) \neq 1$.
  - Computability: $e(P, Q)$ can be computed efficiently.

Bilinear Pairings on BN curves

- Elliptic curves by Barreto and Naehrig (2006):
  - $y^2 = x^3 + b$
- Optimal Ate pairing:
  - $a(Q, P) : \mathbb{G}_2 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T : E'(\mathbb{F}_{p^2}) \times E(\mathbb{F}_p) \rightarrow \mathbb{F}_{p^{12}}$.
  - $a(Q, P) = f_{\lambda,Q}(P)^{(p^{12}-1)/n}$
  - Miller (2005):
    - $f_{i+j,P} = f_{i,P} f_{j,P} \frac{\ell_{[i]P,[j]P}}{\nu_{[i+j]P}}$.
  - λ has low Hamming weight
- BN254 (Gouvêa et al., [GOL12])
- BN158 (Nogami et al., [NAS+08])

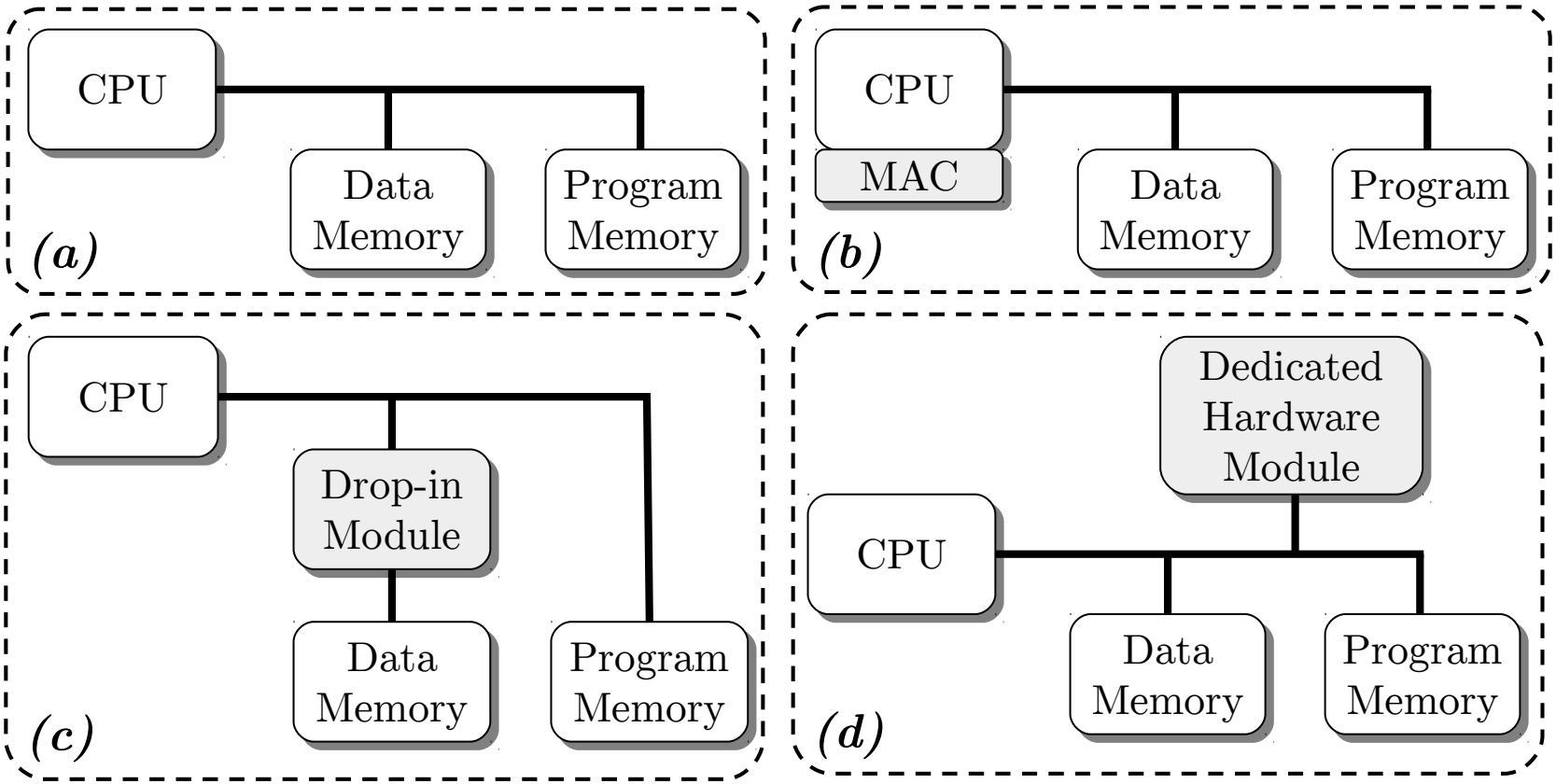
Pairing Arithmetic



Design Requirements

- Pairing-based cryptography for embedded environments
- Usable within interactive protocols
- Minimal modifications to microprocessor platform
- Small overall hardware requirements
- Side-channel countermeasures

Hardware Architectures



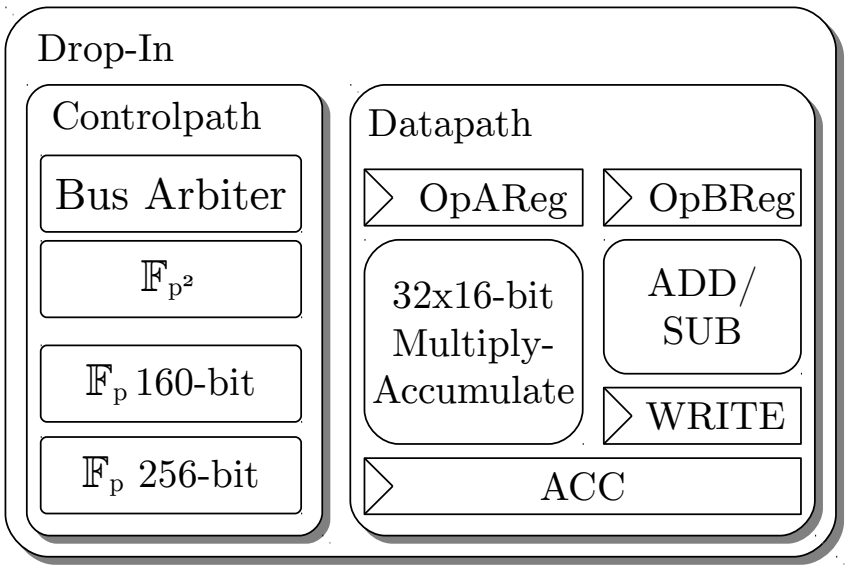
Platform Framework

- 32-bit ARM Cortex-M0+ (self-built)
- Optimized, state-of-the-art pairing framework in C
  - Side-channel countermeasures
  - Performance
  - Low memory consumption
    - Memory-optimized algorithms
    - Stack allocation and memory reutilization
    - Compiler optimizations

Finite-Field Arithmetic

- Constant-time, assembler-optimized prime-field arithmetic
- Crucial prime-field multiplication
  - Montgomery multiplication (separated product scanning)
  - Three registers form accumulator
  - BN254: sparse prime
- $32 \times 32 \rightarrow 32$ bit multiplier
  - Aligning four $16 \times 16 \rightarrow 32$ bit multiplications
  - 80% of runtime are $\mathbb{F}_p$ multiplications
  - $32 \times 32 \rightarrow 64$ bit multiply-accumulate
  - `TST` instruction reinterpreted (if bit in control register is set)

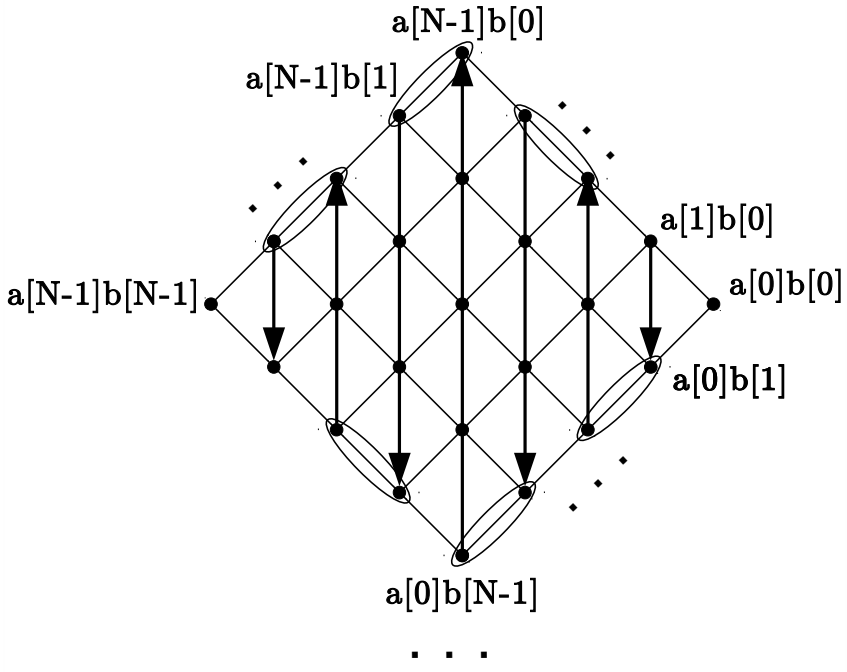
Drop-in Module



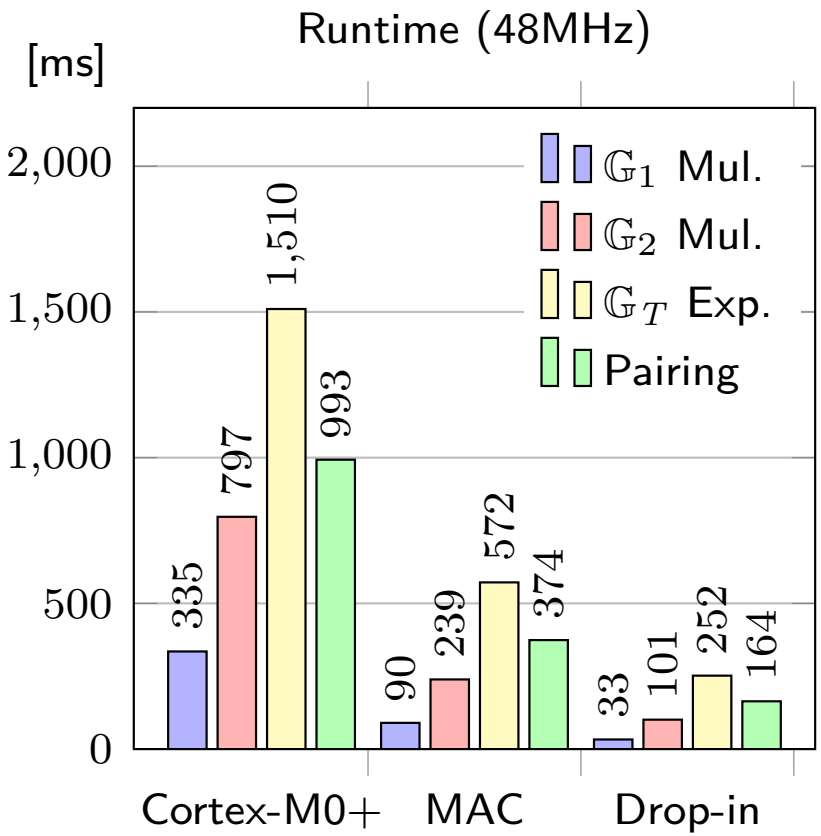
- Unrolled state machines
- FIPS Montgomery mult.
- $W \times W/2$-bit multiplier
  - $2N^2 + N$ W-bit integer multiplications
  - $4N^2$ load operations
- 5-stage pipeline
- Lightweight bus arbiter

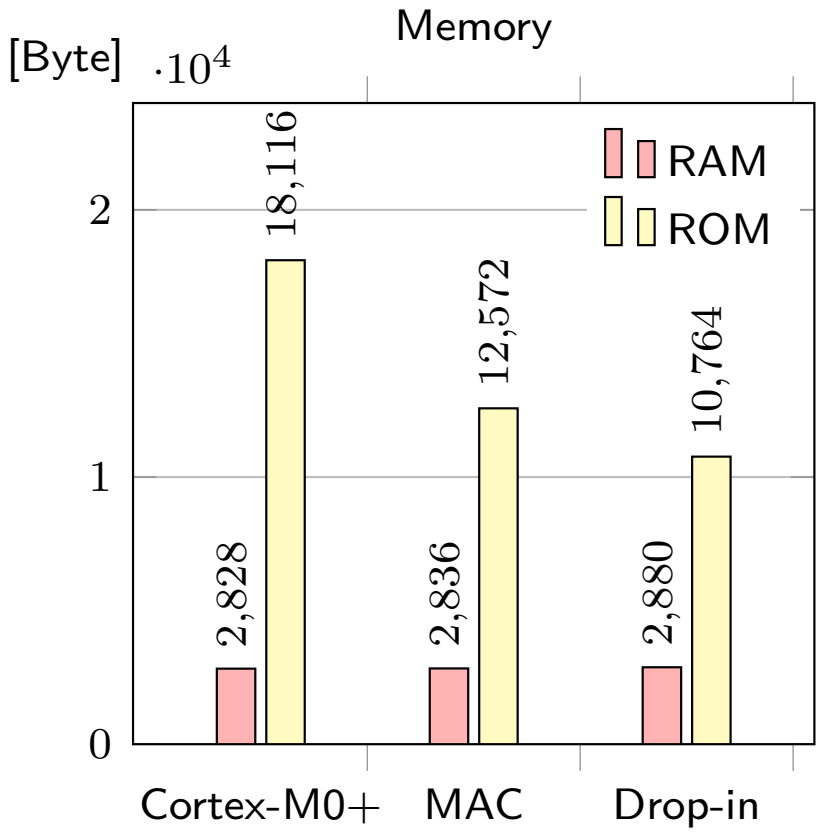
Drop-in Module

- Fully utilized bus
  - Zig-zag product scanning
- Memory-mapped registers
- Invocation of the drop-in:
  - Copy constants to RAM
  - Start+wait or wait+start
    - Parallel control flow



Results (BN254)

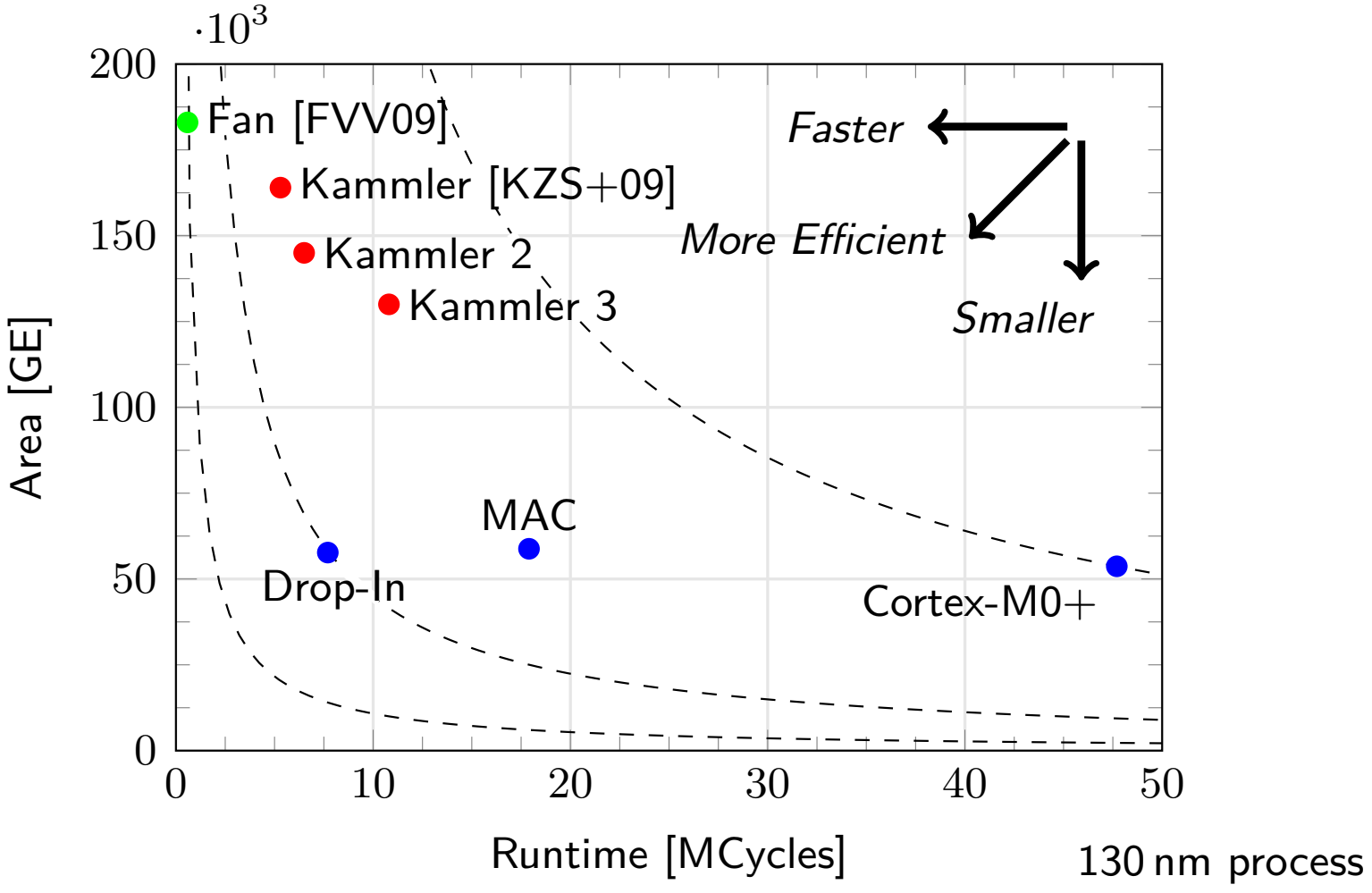




Related Work (Pairing in SW)

	Gouvêa [GOL12]	**Devegili** [DSD07]	**Gouvêa** [GOL12]	**Ours**
Platform	MSP430	SmartMIPS	MSP430X MPY32	Cortex-M0+
Multiplier	16 bit	32 bit	32 bit	16 bit
RAM (KB)	6.5	<16.0	6.5	2.8
ROM (KB)	36.0	-	34.4	18.1
Runtime (MCycles)	79.4	90.5	47.7	47.6
Frequ. (MHz)	8	26	25	48
Runtime (sec)	9.9	2.5	1.9	0.99

Related Work (Pairing in HW)



Conclusion

- Pairing- and ECC-based cryptography in embedded systems
  - Small, practical, and applicable interactively
- HW/SW codesign of three ARM Cortex-M0+-based platforms:
  - Less than 50 kGE
  - Runtimes ranging from 162 ms to 1 sec
- Most promising: drop-in hardware accelerator
  - 11 kGE of dedicated hardware

Efficient Pairings and ECC for Embedded Systems

Thomas Unterluggauer, Erich Wenger
thomas.unterluggauer@iaik.tugraz.at
erich.wenger@iaik.tugraz.at

CHES, Busan, Korea
September 23-26, 2014