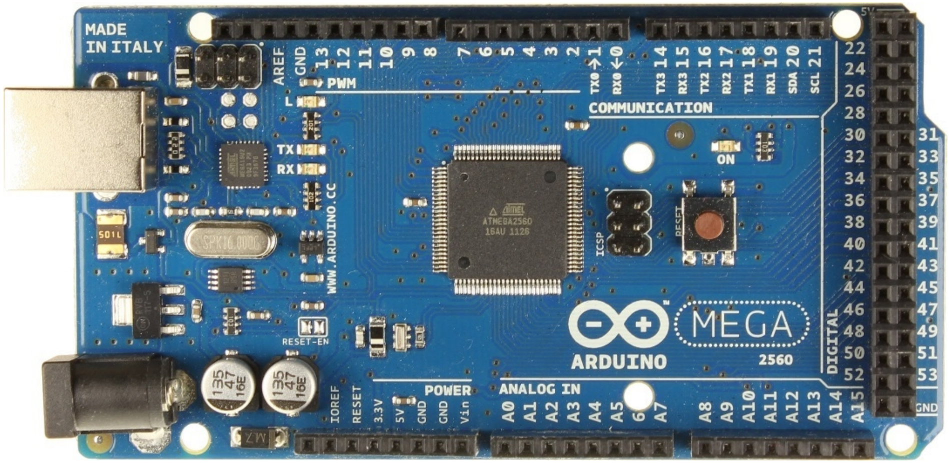
Practical Experiences with crypto on 8-bit

draft-ietf-lwig-crypto-sensors-01

Mohit Sethi, Jari Arkko, Ari Keränen, Heidi-Maria Back

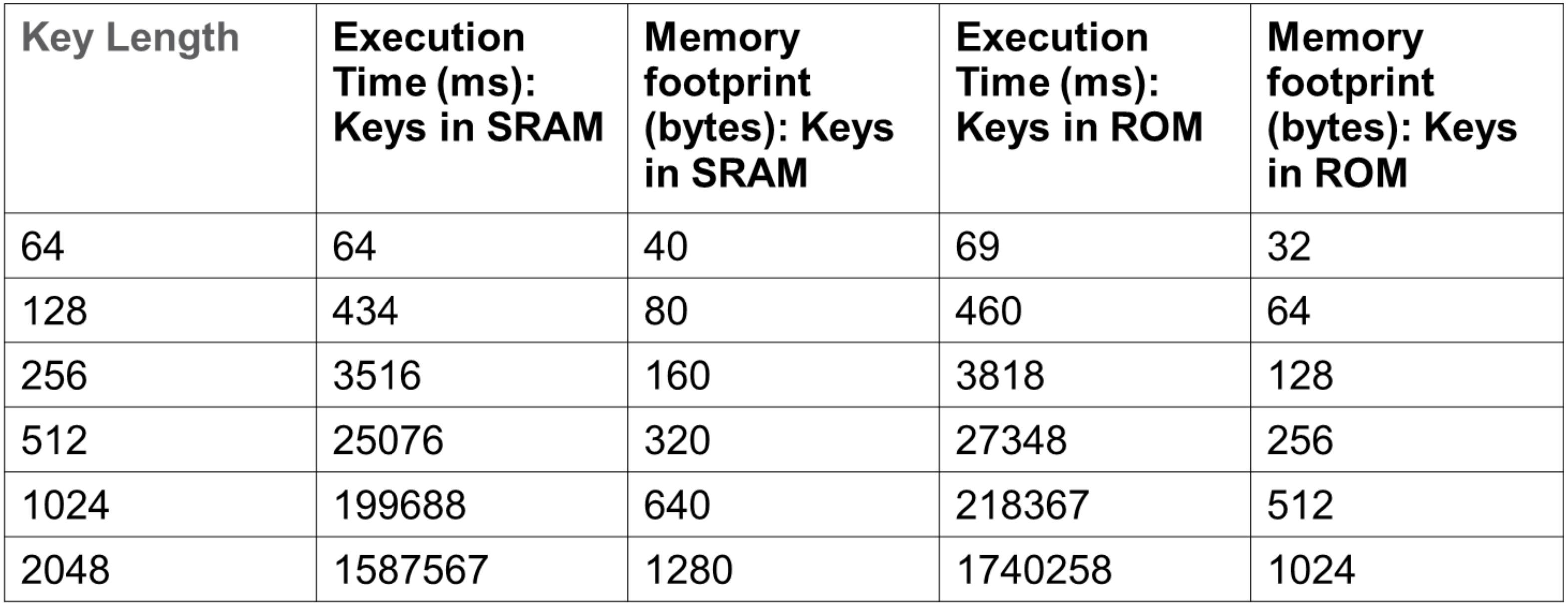
Public Key Experiences

- Can we do Public key crypto on (really) small 8-bit devices 2-5 kB of RAM (Class 0/1)
  - What is available off-the shelf to a developer?
  - How hard it is to run these? (How much time and hacking does it need)
  - How is the performance?



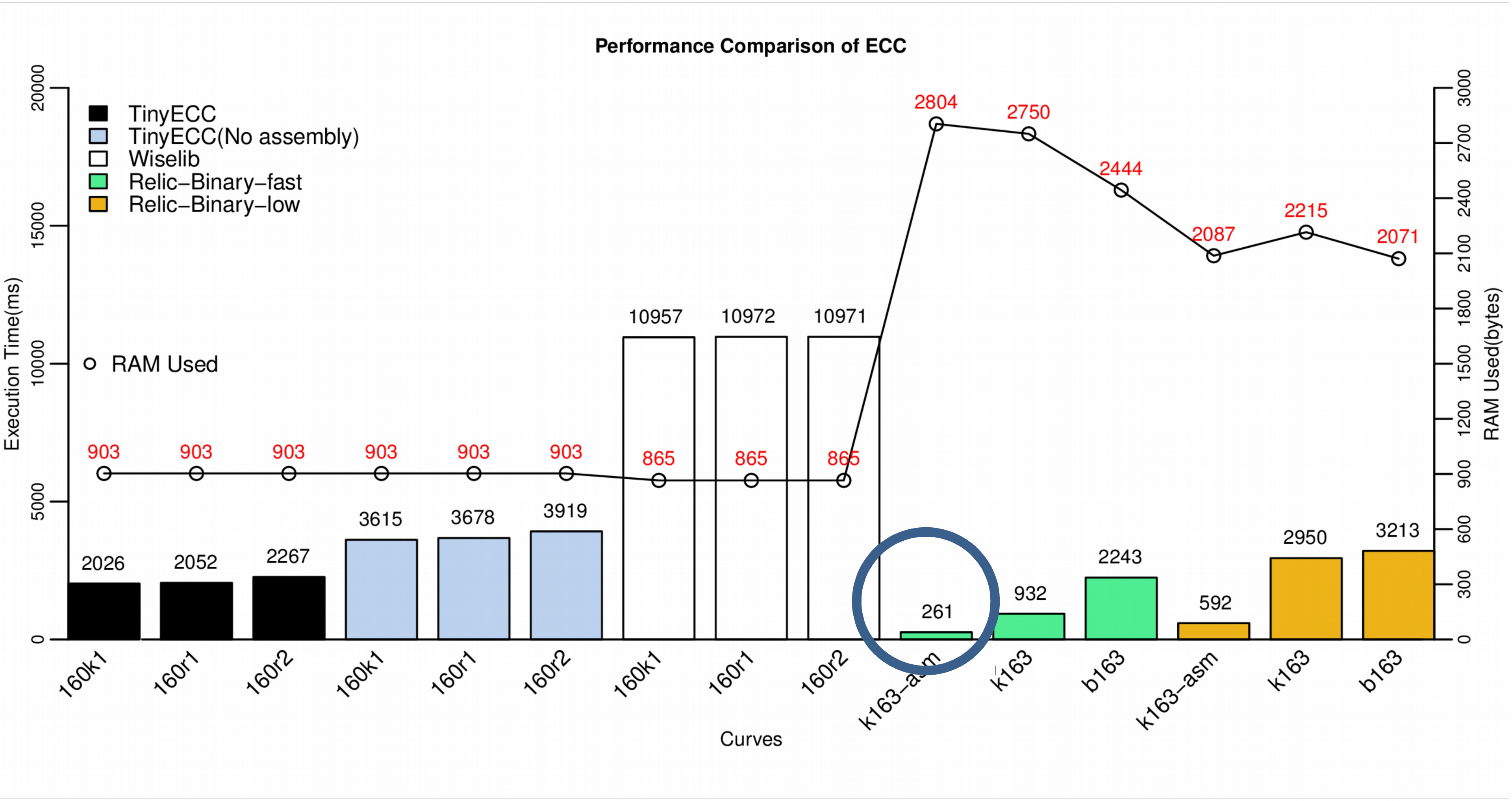
PK Experiences - RSA

- http://www.emsign.nl/
- AVRCryptoLib



Key Length	Execution Time (ms): Keys in SRAM	Memory footprint (bytes): Keys in SRAM	Execution Time (ms): Keys in ROM	Memory footprint (bytes): Keys in ROM
64	64	40	69	32
128	434	80	460	64
256	3516	160	3818	128
512	25076	320	27348	256
1024	199688	640	218367	512
2048	1587567	1280	1740258	1024

ECDSA libraries

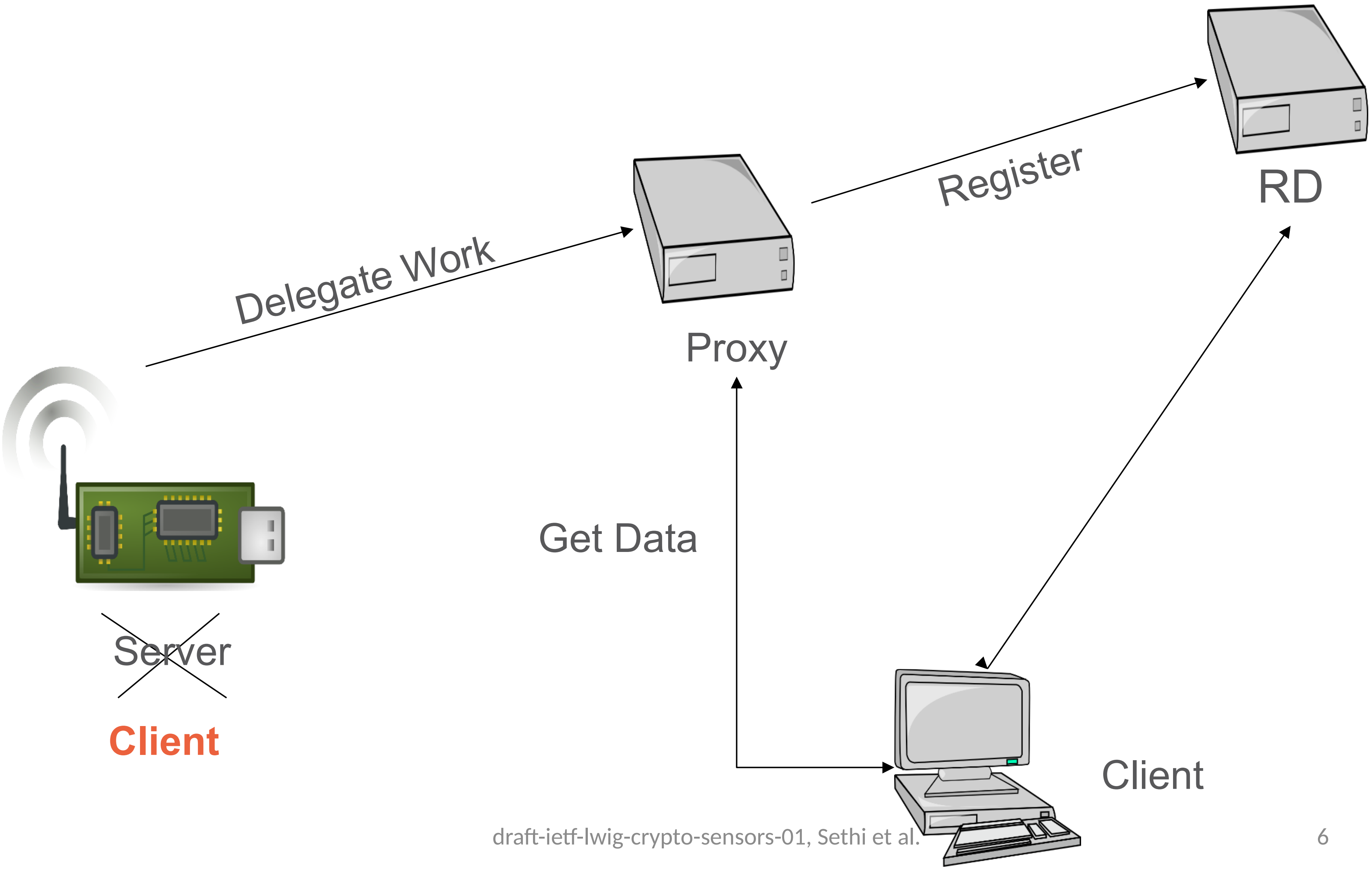


EdDSA libraries

- Edwards-curve Digital Signature Algorithm (EdDSA)
- https://tools.ietf.org/html/draft-irtf-cfrg-eddsa-05
- NaCl and µNaCl high-speed software library
- Public domain
- Signing* = 23,216,241 clock cycles ~ 1,4 sec
- Verification* = 32,634,713 clocks cycles ~ 2 sec

* NaCl on 8-Bit AVR Microcontrollers, Michael Hutter, Peter Schwabe, http://link.springer.com/chapter/10.1007/978-3-642-38553-7_9

Example application



Example application

Flash memory consumption (for the entire prototype (including Relic crypto + CoAP + Arduino UDP etc. libraries)	**51 kB**
SRAM consumption (for the entire prototype including client + key generation + signing the hash of message + COAP + UDP)	**4678 bytes**
Execution time for creating the key pair + sending registration message + time spent waiting for acknowledgement	**2030 ms**
time for signing the hash of message+ sending update	**987 ms**
Signature overhead	**42 bytes**

What we learnt

- Chosen prototype platform was unnecessarily restrictive in the amount of code space
  - we chose this platform on purpose to demonstrate something that is as small and difficult as possible
- Power requirements necessary to send or receive messages are far bigger than those needed to execute cryptographic operations
- No good reason to choose platforms that do not provide sufficient computing power to run the necessary operations

Discussion

- Feasibility
- Message freshness
- Symmetric vs Asymmetric
- Link vs Network vs Transport vs Application

Changes

- Working group adoption call during last IETF
- Interest in the group and confirmed on the mailing list
- Thanks for feedback: Akbhar, Rahul, Daniel, John, Abhijan, Renzo, Raghavendra
- Remove reference to DTLS group keys
- Fix editorial suggestions
- Update reference to Pub/Sub broker
- Smaller key lengths are for reference only
- Ready for WGLC
- More reviews are welcome