A STANDARDS-BASED SECURITY ARCHTECTURE FOR WIRELESS SENSOR NETWORKS

Michael Wurm
Graduate Student Intern
Sun Microsystems Laboratories

Outline

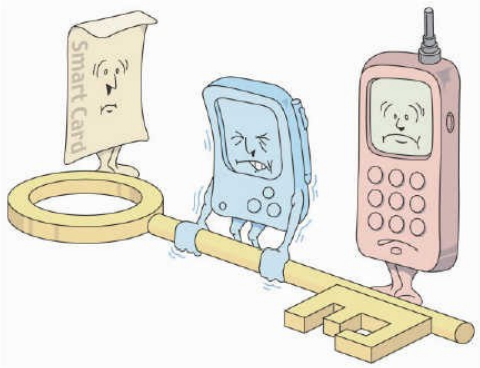
- Elliptic Curve Cryptography and SSL
- Wireless Sensors
- Small Secure Web Server “Sizzle”
- Improving Sizzle's Performance
  - A Small and Fast SSL Stack
  - Implementation of ECC
  - Energy Efficient Communication
- Evaluation of Sizzle

ECC - Elliptic Curve Cryptography

- Highly efficient public-key cryptosystem
- Highest security strength per bit
  - savings in memory, bandwidth, power

Key sizes (in bits), offering equivalent security

RSA	ECC
1,024	160
2,048	224
3,072	256
7,680	384
15,360	521

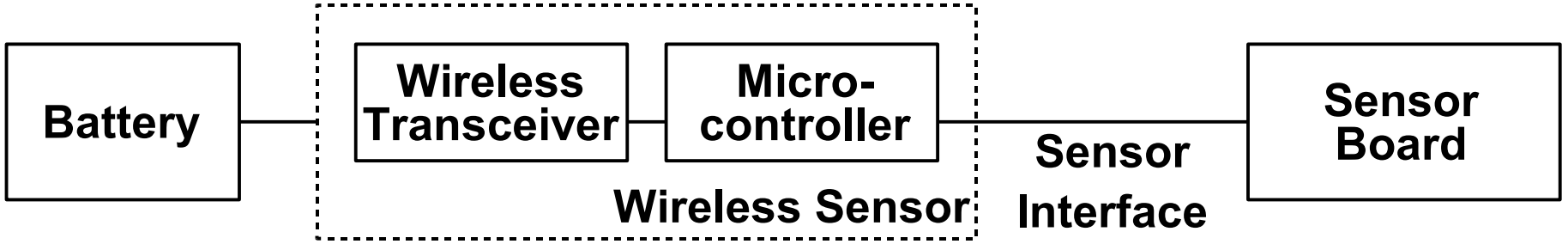


1024-bit keys, like they are used in conventional systems, are too large for small devices

SSL - Secure Sockets Layer

- Handshake
  - Selecting a set of cryptographic algorithms
  - Authentication
  - Establishing a shared secret
- Record Layer
  - Data encryption and decryption
  - Message authentication
- Cryptographic building blocks
  - RSA, ECC; RC4; SHA1, MD5

Wireless Sensors



- Proposed applications:
  - Industrial: temperature, vibration monitoring
  - Health care: patient monitoring
  - Agricultural: monitoring soil chemistry, sunlight
  - Home automation: remote control of appliances
  - Military: battlefield monitoring, intrusion detection
- Security?

Wireless Sensor Platforms

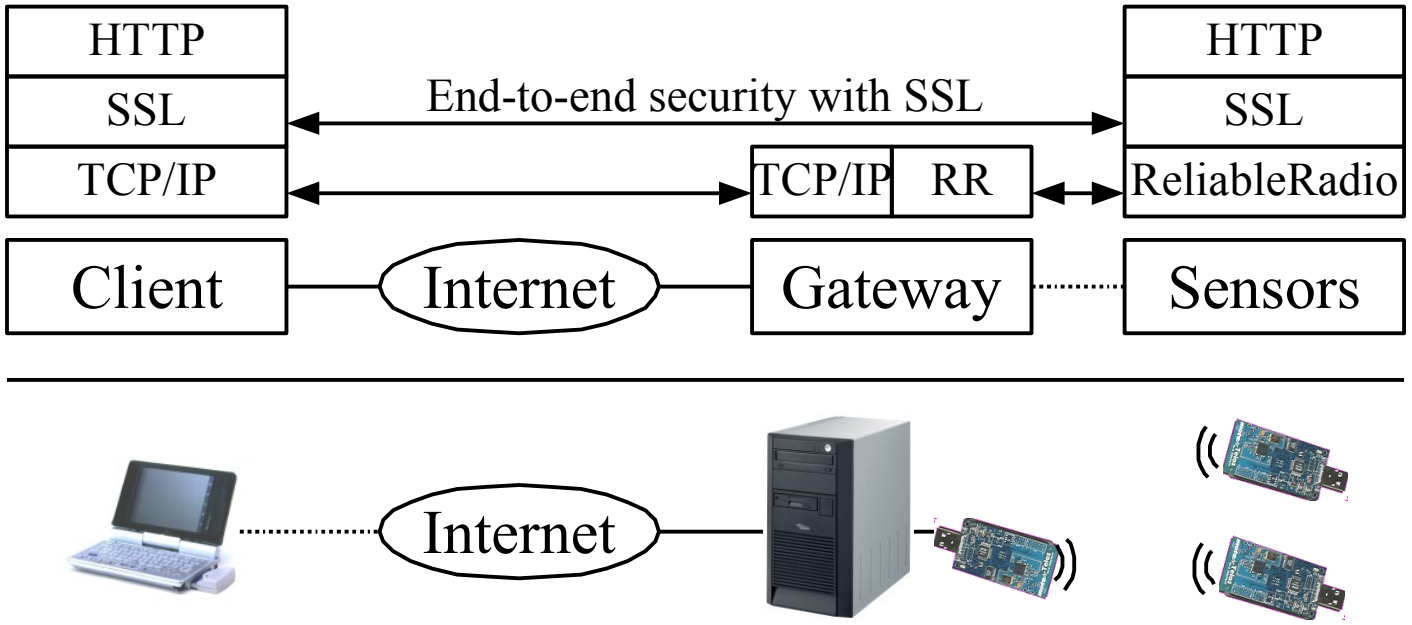
Processor	Atmega	Atmega	TI MSP430	Intel PXA270
Data width	8-bit	8-bit	16-bit	32-bit
Speed	4-8MHz	8MHz	8MHz	500MHz
Radio	CC1000	802.15.4	802.15.4	802.11b (WiFi)
Data rate	20 kbps	250 kbps	250 kbps	2-11 Mbps
Range	10-100m	10-100m	10-100m	100m
Memory				
RAM	4 KB	4 KB	10 KB	128-256 MB
Flash	128 KB	128 KB	48 KB	
Price	~$100*	~$100*	~$100*	~$500
Battery life	Months/Years	Months/Years	Months/Years	Hours

* Target price for large quantities in the next 12-24 months is ~$10

Sizzle

- World's smallest secure web server
  - SSL with RSA and ECDH key exchange
  - implemented in C and assembly
  - 40k FLASH, 3.5k RAM on Telos
- Works with standard web browsers
  - M$IE, etc: can only use RSA
  - Mozilla: can do ECC
- Applications
  - small networks, where sensors must be individually configured / queried / controlled

Sizzle – Architecture



- Gateway
  - Converts TCP/IP to a wireless protocol
  - Provides interface to manage sensors

A Small and Fast SSL Stack

- Implementation of a small set of cryptographic primitives (only three ciphersuites)
- Support for only one elliptic curve allowing efficient implementation
- Client authentication not supported
- Still compliant with the SSL specification
- Performance:
  - Session reuse
  - Persistent HTTP
  - Small HTTP requests

Extent of my Work

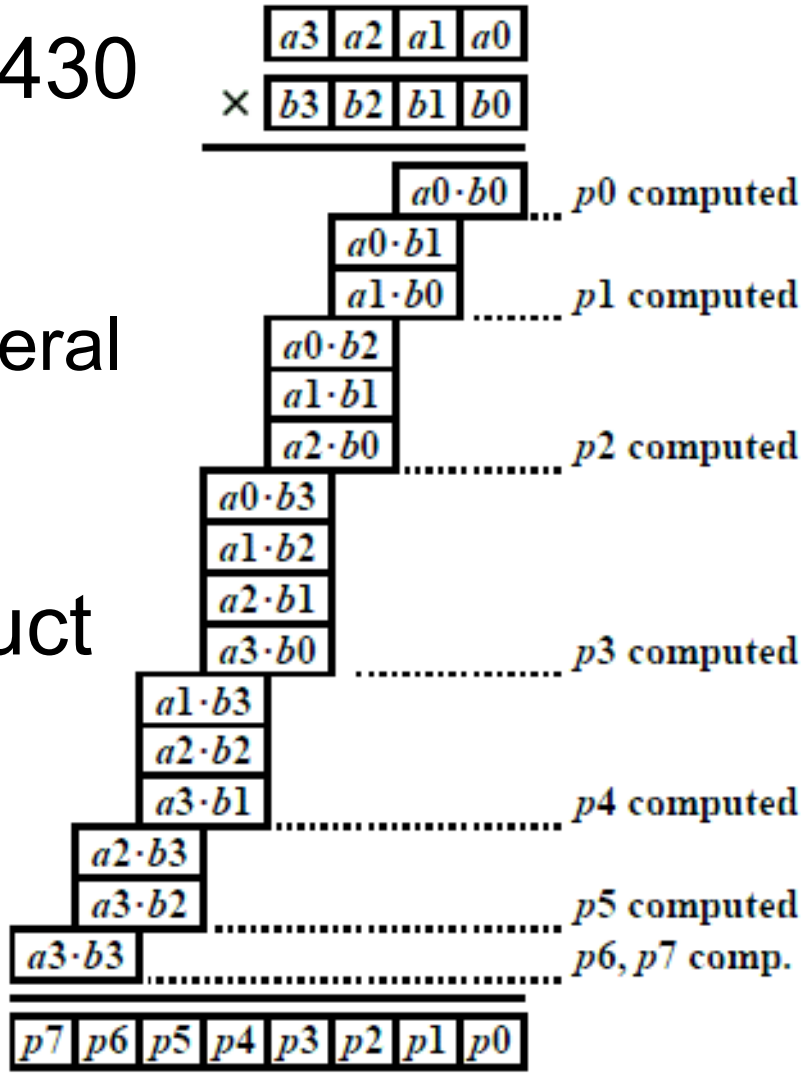
- Porting of Sizzle to the Telos platform
  - implementation of ECC
- Redesign of the communication protocol
  - energy efficiency
- Optimizations
  - speed
  - memory footprint
- Analysis of performance and energy consumption

Implementation of ECC (1/3)

- Only one curve: secp160r1
  - Highly efficient reduction modulo $2^{160}-2^{31}-1$
- Point multiplication
  - on Atmega: 890 ms
  - on MSP430?
    - 16 bits instead of 8 bits
    - expected to be 4 times faster
    - also has multiply-accumulate unit
  - most important part: multiplication of 160-bit integers

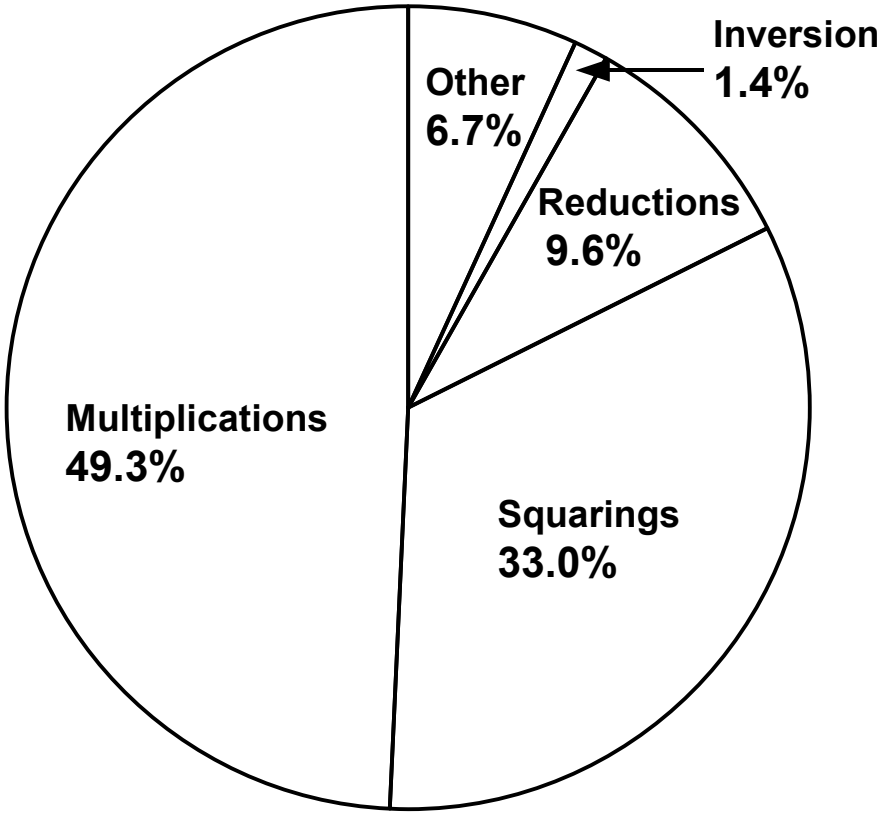
Implementation of ECC (2/3)

- MAC unit of the MSP430
  - 16x16-bit multiplier
  - 32-bit adder
  - integrated as a peripheral
  - slow access
- Long integer multiplication in product scanning form
  - better than operand scanning form when access to MAC is slow



Implementation of ECC (3/3)

- 480 ms on MSP430 (at 8 MHz)
  - versus 890 ms on Atmega



2/3 of that time is spent in instructions moving data between memory and MAC-unit...

Energy Efficient Communication (1/3)

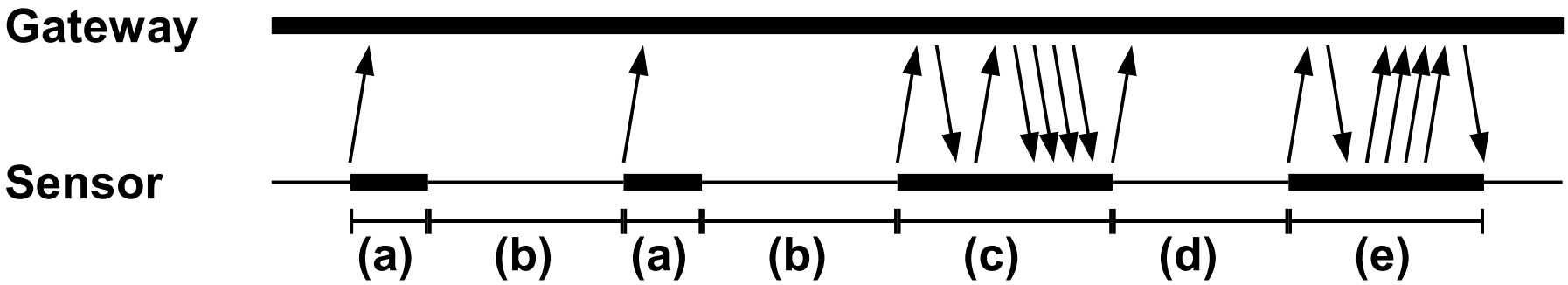
- Problem of idle listening
  - Radio must be in receive mode when expecting incoming packets

Telos Supply Currents at 3V

Operating Mode	Current
Standby	5.1 uA
Microcontroller Idle (Oscillator on)	54.5 uA
Microcontroller Active (8 MHz)	3.9 mA
Radio Transmit	17.7 mA
Radio Receive	20.0 mA

Energy Efficient Communication (2/3)

- Duty-cycle-based approach
  - > sensor polls for incoming connections
  - > trade off: energy conservation vs. latency
  - > gateway can discover devices

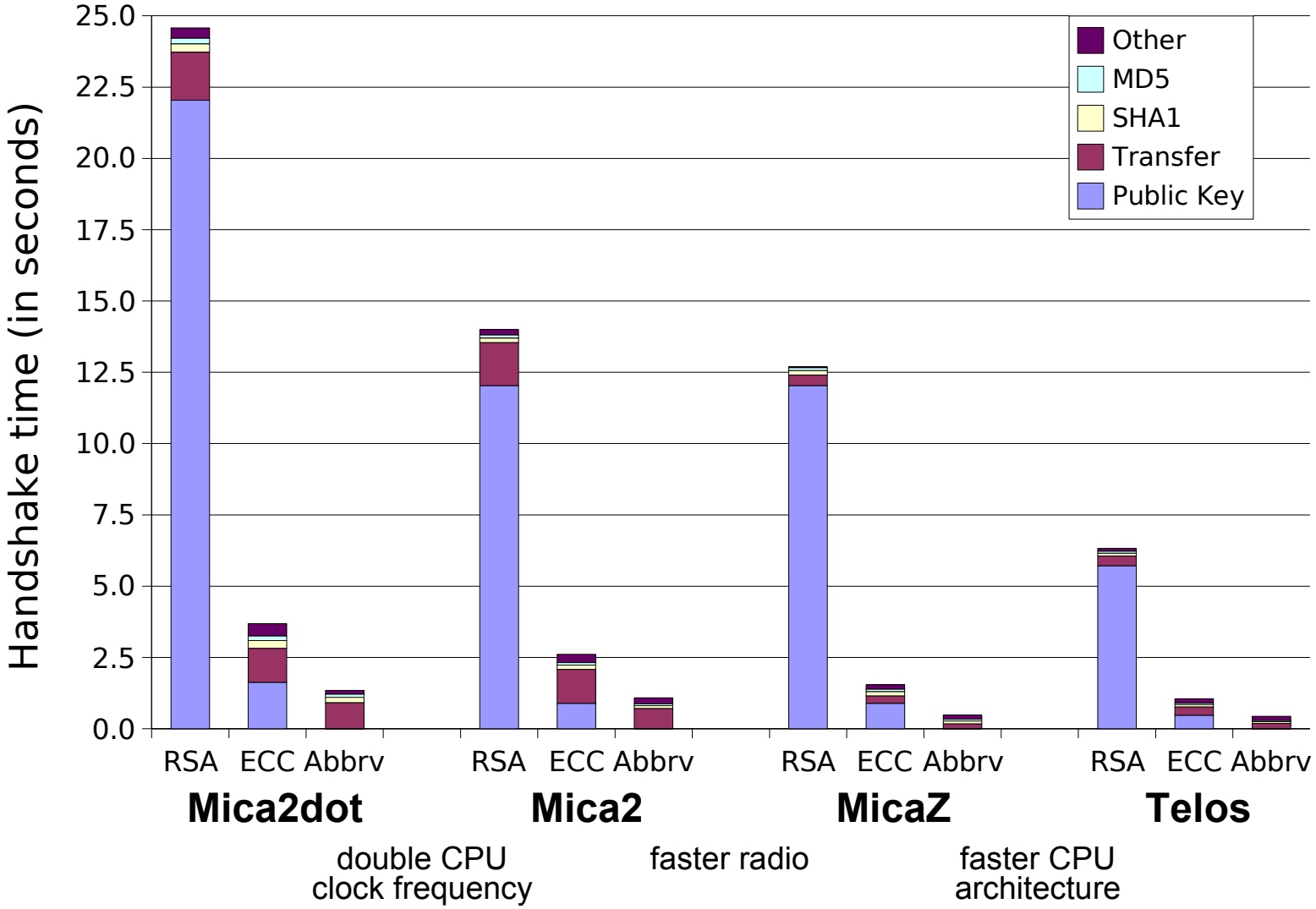


a: device sends polling message
b: no response from the gateway -> device sleeps
c: gateway transmits message right after polling
d: device processes message; radio is turned off
e: device responds

Energy Efficient Communication (3/3)

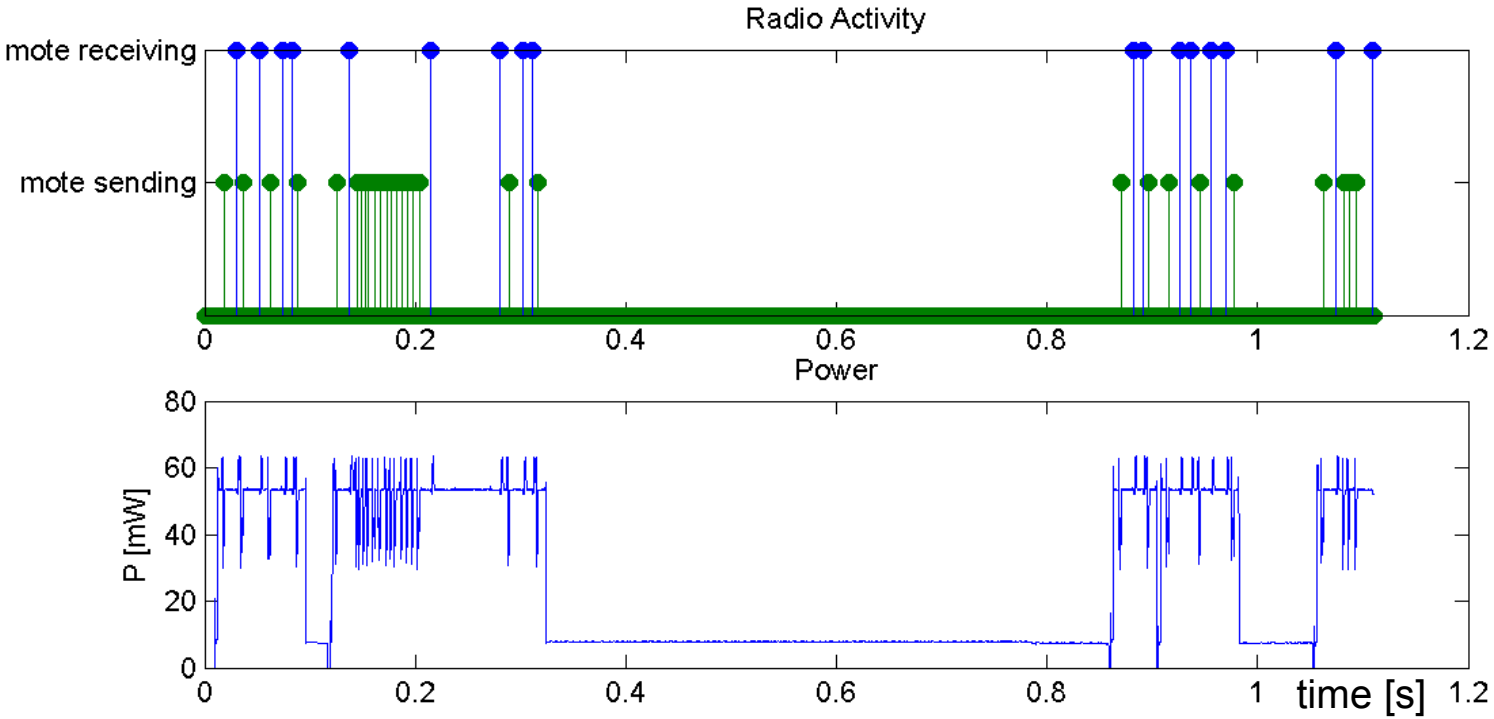
- Example
  - Active period: 25 ms
  - Polling interval: 2.5 seconds
  - 99% of energy saved
- Battery lifetime (2x AA)
  - 3 days when receiver is on
  - 1 year with the setting above

Evaluation – Performance

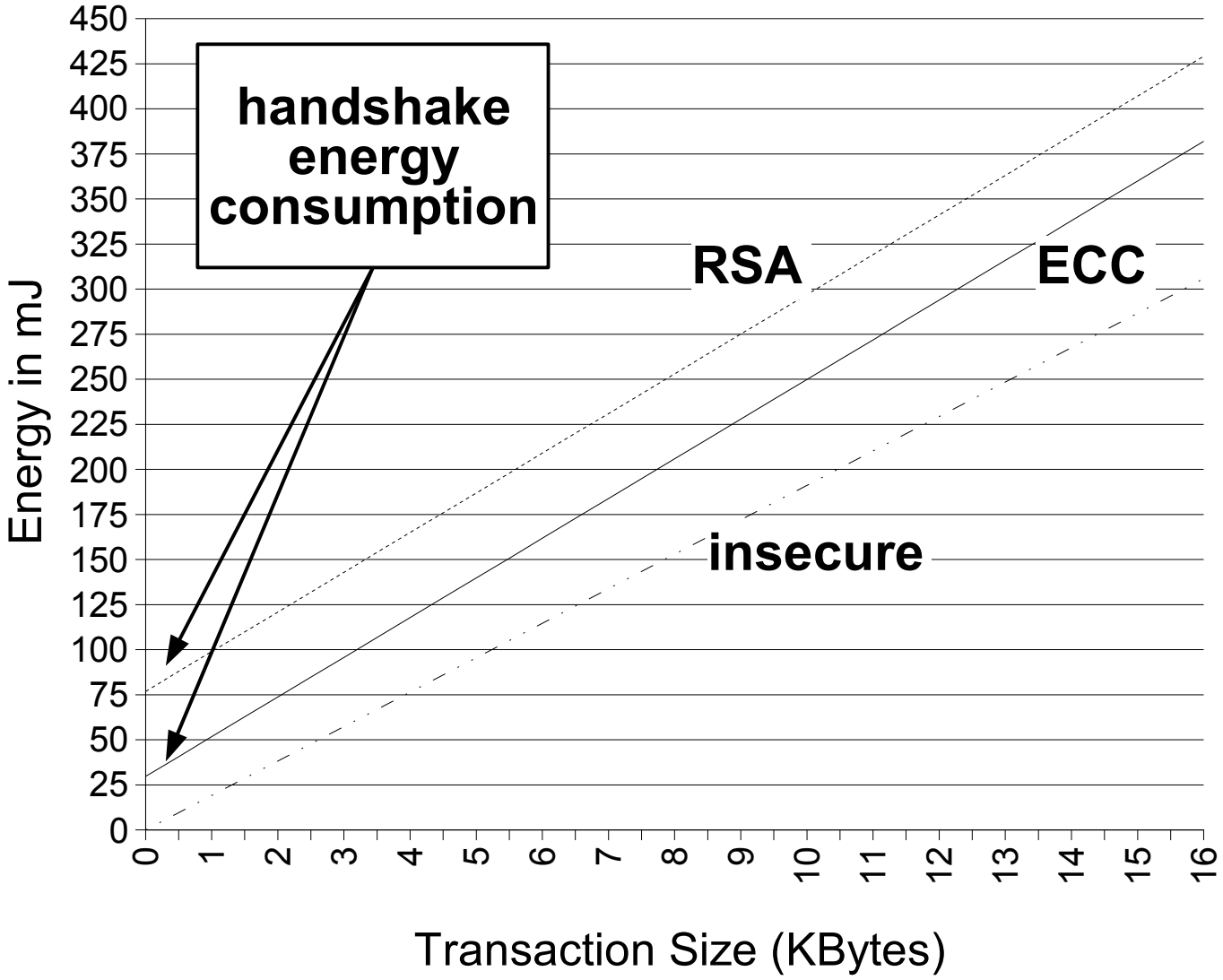


Evaluation – Energy Consumption

	Absolute Cost		Cost relative to RSA		Energy in
	Time [ms]	Energy [mJ]	Time	Energy	Communication
RSA Handshake	6,410	76.9	100.0%	100.0%	41%
ECC Handshake	1,110	29.7	17.3%	38.6%	82%
Session Reuse	422	16.5	6.6%	21.5%	93%



Evaluation – Impact of Security



Results and Future Developments

- Results:
  - Sizzle runs well on Mica2, MicaZ, and Telos platforms
  - Paper, article for a journal, patent, thesis
  - Generated a lot of interest
- What to expect in the future?
  - Influence of the Sizzle project on wireless sensor research
  - Future of ECC / SSL / HTTPS for wireless sensors

Thanks for your interest.

Michael Wurm
mwurm@sime.com