Cryptography for the Internet of Things

Diego F. Aranha, dfaranha@unb.br

Departament of Computer Science
University of Braslia

Introduction

Definition

The Internet of Things (IoT) is a scenario in which objects, animals or people are provided with unique identifiers and the ability to automatically transfer data over a network without requiring human-to-human or human-to-computer interaction.

Many applications:

- Smart cities (lighting, waste management, environment, traffic)
- Incident response (access control, detection of fire or radiation)
- Retail (supply chain control, logistics)
- Home automation (intrusion detection, smart spaces).

Important: Devices need to be small and pervasive, thus resource-constrained and limited tamper-resistance.

Typical platform

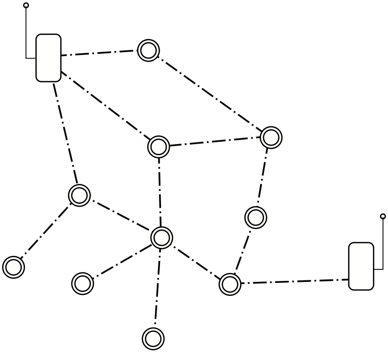
MICAz Mote:

- ATMega128 processor, 7.3828 MHz of clock frequency
- 4KB of RAM memory, 128KB of ROM memory
- Simple two-stage pipeline
- Limited instruction set and battery life
- High cost of memory instructions and data transmission

The problem

Challenge

Since the nodes must be cheap and even disposable, *protecting the communication* between resource-constrained nodes is **hard.**

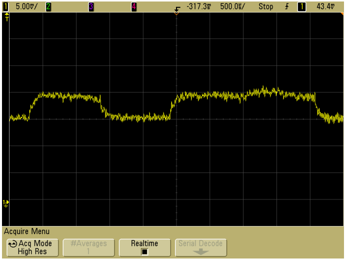


Important: Not only application data, but routing and other metadata!

Side-channel attacks

Computation leaks information correlated with data:

- Execution time and cache timing
- Power consumption and accoustic emanation
- Fault injection.



Bigger challenge

What if data is secret? Algorithms and implementations need to be made **regular**, adding non-trivial performance penalty!

Solutions

1. Lightweight symmetric cryptography
   - Block and stream ciphers (PRESENT, Spongent)
   - Hash functions and MACs (Quark, Marvin)

2. Asymmetric cryptography
   - Number-theoretic cryptography (ECC, PBC)
   - Post-quantum cryptography (Hash functions, lattices and codes)

3. Physical and computational assumptions
   - PUF-based cryptography (SRAM, oscillators)

Previous and ongoing work

1. Key distribution protocols
   - Speed records for ECC-based key agreement [1]
   - Pairing-based non-interactive key agreement (TinyPBC) [2]
   - Currently: Network coding and secure routing.
2. Digital signatures
   - Efficient implementation of ECC-based signatures [3]
   - Signature length, computational cost and energy consumption
   - Currently: New curves and implementation techniques.
3. PUF-based protocols
   - Banking applications
   - Currently: Mutual and multi-factor transaction authentication.

Previous and ongoing work

RELIC cryptographic library:

- GPL license
- Reproducibility of published results
- 78K LoC (*OpenSSH* has 50K)
- 13 released versions
- 2 developers
- 2500 downloads by 3000 unique visitors from 97 countries.

`http://code.google.com/p/relic-toolkit/`

References

D. F. Aranha, L. B. Oliveira, J. López, and R. Dahab.
Efficient implementation of elliptic curve cryptography in wireless sensors.
Advances in Mathematics of Communications, 4(2):169–187, 2010.

L. B. Oliveira, D. F. Aranha, C. P. L. Gouvêa, M. Scott, D. F. Câmara, J. López, and R. Dahab.
TinyPBC: Pairings for Authenticated Identity-Based Non-Interactive Key Distribution in Sensor Networks.
Computer Communications, 4(2):169–187, 2011.

L. B. Oliveira, A. Kansal, C. P. L. Gouvêa, D. F. Aranha, J. López, B. Priyantha, M. Goraczko, and F. Zhao.
Secure-TWS: Authenticating Node to Multi-user Communication in Shared Sensor Networks.
The Computer Journal, 55(4):384–396, 2012.