Comparison of innovative asymmetric signature schemes for WSNs

ACM WiSec, Alexandria, VA, USA

Benedikt Driessen, 31.03.2008

escrypt
Embedded Security

Why use asymmetric signature schemes in WSNs?

WSNs from the security perspective:

- Cheap & low-power (i.e. „unprotected“) hardware
- Exposed to:
  - eavesdropping
  - data modification
  - physical attacks

Applications for WSNs:

- Monitoring (health/environment), industrial control, etc.
- Critical applications require message integrity & authenticity

Asymmetric signature schemes...

- 👍 ..can provide adequate security mechanisms
- 👍 ..can avoid certain problems of symmetric schemes
- 👎 ..are perceived as being too slow

Typical WSN Hardware: **MICAz**

MICAz:

- 2 AA batteries (1000mAh)
- ATMega128L

ATMega128L, 8-bit uP:

- Clock: 7,37MHz
- ROM: 128kB
- RAM: 4kB

Implied constraints:

(1) Energy
(2) Computational power
(3) RAM

Potential, asymmetric **signature schemes**

Standard schemes:

- RSA (1978)
- ECDSA (1985)

Innovative (i.e. “young”) schemes:

- NTRUSign (2001)
- XTR-DSA (2000)
- MQ-based schemes (e.g. SFLASH v1-v3, **broken 2007**)

Focus of this work:

- Compare XTR-DSA, NTRUSign to a high-speed implementation of ECDSA
- Security level equivalent to an 80-bit symmetric key
- Optimized for speed
- Implementation in NesC (a C dialekt) and ASM

ECDSA

Characteristics:

- 👍 Compact signatures
- 👍 Quite well researched
- 👍 Fast
- 👎 Complex arithmetic

Major optimizations:

- Field arithmetic in ASM (based on work by Uhsadel)
- Fixed-point methods for..
  - ..single scalar multiplication: kP
  - ..simultaneous scalar multiplication: kP+lQ

XTR-DSA

Characteristics:

- 👍 Compact signatures
- 👍 Fast (supposedly as fast as ECDSA)
- 👎 Complex arithmetic
- 👎 Not as well researched as ECDSA

Major optimizations:

- Field arithmetic in ASM
- Montgomery Multiplication
- Stam's exponentiation techniques

NTRUSign

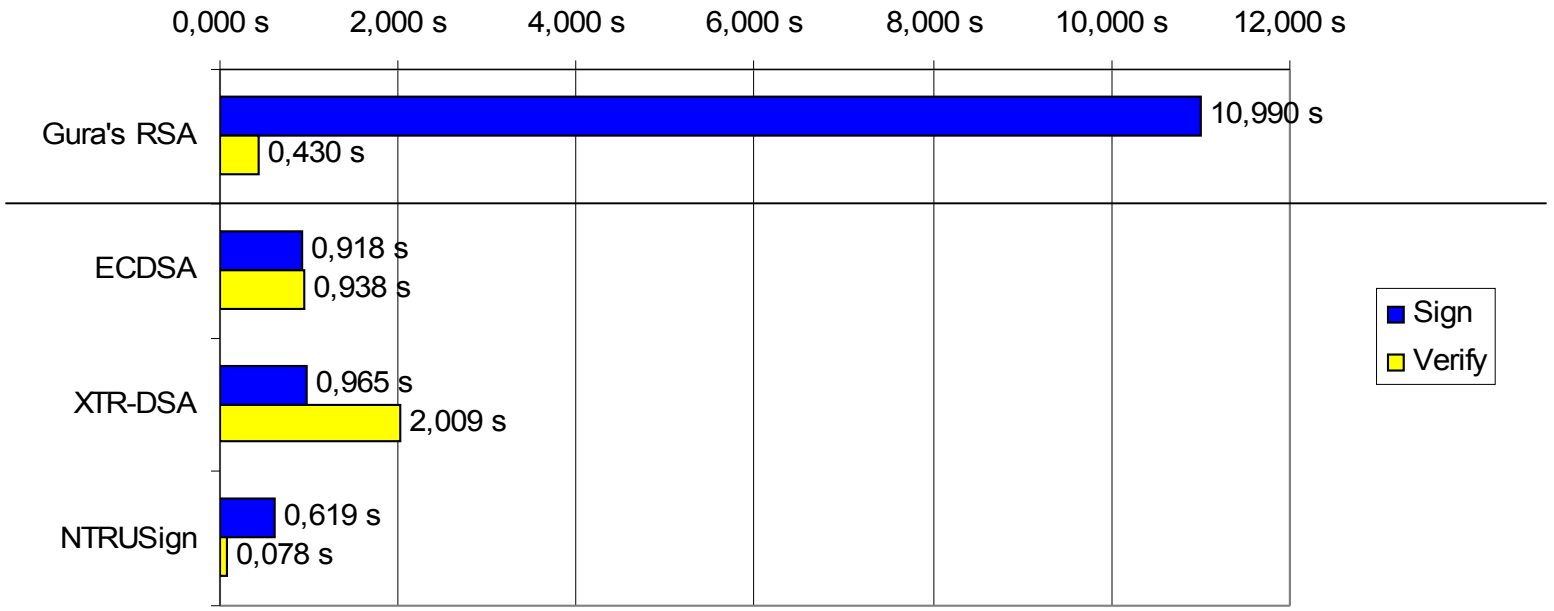
Characteristics:

- 👍 **Really** fast
- 👍 Simple arithmetic (convolution of polynomials)
- 👎 Recent attacks on NTRU
- 👎 Large signatures

Major optimizations:

- Choice of parameter-set with low N
- Karatsuba-like approaches for..
  - ..single convolution: f*g
  - ..simultaneous convolution: f1*g+f2*g

Results



NTRUSign wins

- Pure NesC implementation
- No precomputation time (ECDSA: 46s, XTR-DSA: 4s)

Thank you!

Questions?