Securing WSNs and the IoT: Performance Analysis of Identity-based Signatures

Tobias Markmann

tobias.markmann@haw-hamburg.de

23.04.2014

Outline

1. Introduction
2. Background
3. Identity-based Signature Schemes
4. Evaluation
5. Results
6. Discussion

1. Introduction

- Constrained devices communicating in a network
- Identification of devices/things
- Varying communication media

Secure identification and communication between devices

Identification in Networks

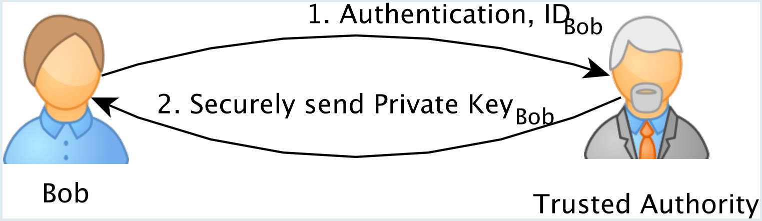
- Identification by address:
  - EMail address: alice@wonderland.lit
  - Internet: 2a02:2028:ad:d411:be05:43ff:fe18:2bf
- Authenticaiton of identiy
  - Unique private data only the true identity knows
  - Authenticate communication using secret keys

2. Cryptography Background

- Asymmetric Signatures
  - Public key/private key signatures
  - Widespread use: World Wide Web, Passports, ...
  - Easy and flexible trust concepts
- Identity-based Signatures
  - Form of asymmetric signature
  - Arbitrary choice of public key
  - Trust via central commonly trusted authority

ID-based Cryptography Workflow

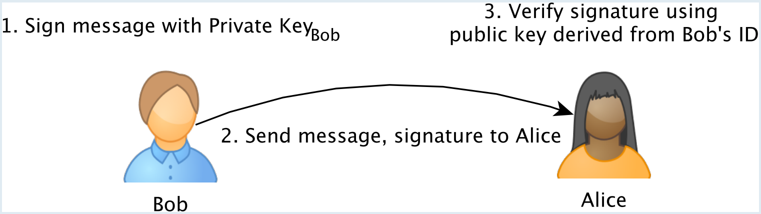
1. Setup $\rightarrow$ system parameters (SP) and master secret key (msk)
2. KeyExtraction$(SP, msk, ID) \rightarrow$ secret key for ID (s_{ID})



3. Authentication and Verification

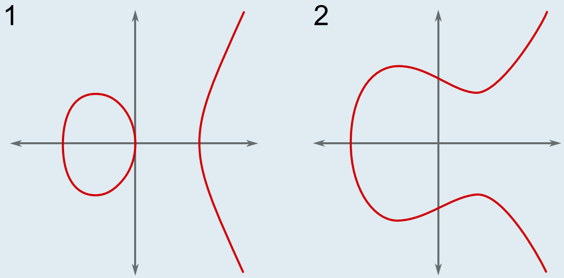
   Sign$(SP, s_{ID}, m) \rightarrow (\sigma)$

   Verify$(SP, ID, m, \sigma) \rightarrow 1/0$



2. Mathematical Background

- RSA
- Elliptic Curves
- Pairings



$y^2 = x^3 - x$ $\qquad$ $y^2 = x^3 - x + 1$

2.1. RSA

RSA Cryptosystem

- 2 large primes p, q at random
- $N = p \cdot q$
- $1 < e < \psi(N)$ and $gcd(e, \psi(N)) = 1$
- $d = e^{-1} \bmod N$
- Sign: $s = H(m)^d \bmod N$
- Verify: $h = s^e \bmod N$, $h \stackrel{?}{=} H(m)$

Complexity

- Signature verification and generation equally expensive
- Practice: pick small e, e.g. 65537
- Result: Faster verification than generation

2.2. Elliptic Curves

- Motivation
- Basics
- Group Law

Motivation for Elliptic Curves

- Discrete logarithm problem in finite fields ($\mathbb{F}_p$)
  - Let $p = 128(2^{800} + 25) + 1$, 807-bit prime
  - Problem: find $\lambda \in \mathbb{Z}$, such that $2 \equiv 3^\lambda \bmod p$
  - For modern security, p needs to be greater than **3000** bits
- DLOG in $\mathbb{F}_p$:
  subexponential complexity $\longrightarrow$ security requires big p
- DLOG in elliptic curves:
  only exponential complexity algorithm known $\longrightarrow$ smaller numbers

Basics of Elliptic Curve Crypto

- Elliptic curve formula of form:

$$E_{A,B} : Y^2 = X^3 + AX + B$$

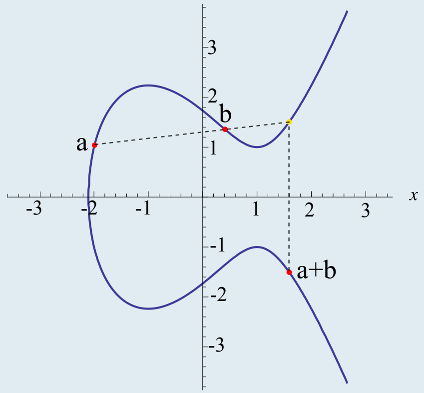
- Curve defined over $\mathbb{F}_p$, $\mathbb{F}_{2^m}$ or $\mathbb{F}_{p^m}$

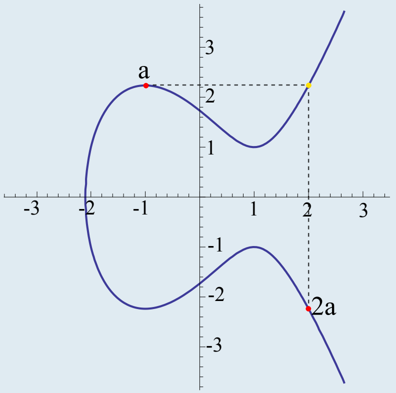
- Example: "Curve25519"

  – $E : Y^2 = X^3 + 486662X^2 + X$,
  – over $\mathbb{F}_p$, $p = 2^{255} - 19$

Groups over Elliptic Curves

- $E(K) = \{(x, y) \in K^2$: x,y satisfy the elliptic curve equation$\} \bigcup \{\mathcal{O}_E\}$
- Point addition
- Point doubling





- Scalar multiplication: $nP = \underbrace{(x, y) + (x, y) + ... + (x, y)}_{n \text{ times}}$
- Point P as generator of group $G(E(K))$ with a large prime order

2.3. Pairing-based Cryptography

Definition (symmetric):

- G, G_t two abelian groups
- $e : G \times G \longrightarrow G_t$
- $P, Q \in G, a, b \in \mathbb{Z}$
- Properties:
  1. Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$
  2. Non-degenerate: $e(P, Q) \neq 1$
  3. Efficiently computable: Miller's algorithm

Groups:

- Example: $G \subseteq E(\mathbb{F}_p)$ and $G_t \subseteq \mathbb{F}_{p^\alpha}^*$
- $\alpha = 2, 6, \ldots$

PBC Example: BLS Signature

Key Generation:

- Random $sk \in \mathbb{Z}_q$ as secret key
- Public key is $pk = g^{sk}$, g is generator of group G

Signature Generation:

- Sign(sk, m) $\rightarrow H(m)^{sk}$

Signature Verification:

- Verify(pk, m, σ) $\rightarrow$ valid if $e(g, \sigma) = e(pk, H(m))$
- $e(g, \sigma) = e(g, H(m)^{sk}) = e(g^{sk}, H(m)) = e(pk, H(m))$

3.1 SH-IBS

- Original proposal by Adi Shamir in 1984
- Based on the RSA cryptosystem

SH-IBS: Description

Setup:

- Like RSA: master private key (MPK) and master secret key (MSK)
- Define two hash functions:
  1. $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_n$
  2. $H_2 : \mathbb{Z}_n \times \{0,1\}^* \rightarrow \mathbb{Z}_n$

Key Extraction:

- Identity ID, ID's secret key s_{ID}
- $s_{ID} = H_1\left(ID\right)^d \bmod n$

Signature Generation:

- Random $r \in \mathbb{Z}_n$
- $t = r^e \bmod n$
- $s = s_{ID} \cdot r^{H_2(t,m)} \bmod n$
- $\sigma_m = (s, t)$

Signature Verification:

- Holds if the signature is valid:
- $s^e \stackrel{?}{=} H_1(ID) \cdot t^{H_2(t,m)} \bmod n$

SH-IBS: Complexity

Storage Complexity:

- Signature size: $\mathbb{Z}_N \times \mathbb{Z}_N$

Computational Complexity:

- Generation: 2 modular exponentiation in $\mathbb{Z}_N \equiv \mathcal{O}(\log e + \log \frac{N}{2})$
- Verification: 2 modular exponentiation in $\mathbb{Z}_N \equiv \mathcal{O}(\log e + \log \frac{N}{2})$
- e being the master public key

3.2 vBNN-IBS

- Proposed by Cao, Kou, Dang and Zhao in 2008
- As part of "IMBAS: Identity-based multi-user broadcast authentication in wireless sensor networks"
- Security based on elliptic curve discrete logarithm problem

vBNN-IBS: Description

Setup:

- Elliptic-curve setup according to security parameter
- Random master secret key $x \in \mathbb{Z}_p$
- Master public key: $P_0 = xP$
- Define two hash functions:
  1. $H_1 : \{0,1\}^* \times \mathbb{G} \to \mathbb{Z}_p$
  2. $H_2 : \{0,1\}^* \times \{0,1\}^* \times \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_p$

Key Extraction:

- Random $r \in \mathbb{Z}_p$, $R = rP$
- $s = r + H_1(ID, R) \cdot x$
- $s_{ID} = (R, s)$

vBNN-IBS: Description (cont.)

Signature Generation:

- Random $y \in \mathbb{Z}_p$, $Y = yP$
- $h = H_2(ID, m, R, Y)$
- $z = y + hs$
- $\sigma = (R, h, z)$

Signature Verification:

- $c = H_1(ID, R)$
- $T = zP - h(R + cP_0)$
- Holds if signature is valid:
- $h \stackrel{?}{=} H_2(ID, m, R, T)$

vBNN-IBS: Complexity

Storage Complexity:

- Signature size: $G(E(\mathbb{F}_q)) \times \mathbb{Z}_p \times \mathbb{Z}_p$

Computational Complexity:

- Generation: 1 exponentiation in $G(E(\mathbb{F}_p))$
- Verification: 3 exponentiations in $G(E(\mathbb{F}_p))$

3.3 TSO-IBS

- Proposed by Tso, Gu, Okamoto and Okamoto in 2007
- Utilizes bilinear pairings over elliptic curves
- Provides ID-based signatures with message recovery
  - **For fixed size messages**
  - For variable size messages
- Message recovery:
  - Signature includes message
  - Recoverable by any receiver
  - Reduce overall size of authenticated message

TSO-IBS: Description

Setup:

- ECC setup
- G_1 and G_2 of order q, $|q| = l_1 + l_2$
- Random $s \in \mathbb{Z}_q^*$ (MSK)
- $P_{Pub} = sP$ (MPK)
- $\mu = \hat{e}(P, P)$

Key Extraction:

- $s_{ID} = (H(ID) + s)^{-1}P$

- 4 hash functions:
  1. $H : \{0,1\}^* \longrightarrow \mathbb{Z}_p^*$
  2. $H_1 : \{0,1\}^* \longrightarrow \{0,1\}^{l_1+l_2}$
  3. $F_1 : \{0,1\}^{l_1} \longrightarrow \{0,1\}^{l_2}$
  4. $F_2 : \{0,1\}^{l_2} \longrightarrow \{0,1\}^{l_1}$

TSO-IBS: Description (cont.)

Signature Generation:

- $m \in \{0,1\}^{l_1}$ and compute random $r_1 \in \mathbb{Z}_q^*$
- $\alpha = H_1(ID, \mu^{r_1}) \in \{0,1\}^{l_1+l_2}$
- $\beta = F_1(m) \| (F_2(F_1(m)) \bigoplus m)$ and $r_2 = [\alpha \bigoplus \beta]$
- $U = (r_1 + r_2)s_{ID}$, final signature $\sigma = (r_2, U)$

Signature Verification:

- $P_{ID} = H(ID)P + P_{Pub}$
- $\tilde{\alpha} = H_1(ID, \hat{e}(U, P_{ID}) \cdot \mu^{-r_2})$
- $\tilde{\beta} = r_2 \bigoplus \tilde{\alpha}$ and $\tilde{m} = |\tilde{\beta}|_{l_1} \bigoplus F_2({}_{l_2}|\tilde{\beta}|)$
- Valid if ${}_{l_2}|\tilde{\beta}| = F_1(\tilde{m})$

TSO-IBS: Complexity

Storage Complexity:

- Authenticated message size: $|q| + |G_1|$
- Signature size: $|q| + |G_1| - l_1$, for messages of size l_1
- Implemented with $|G_1|$ = 193 bytes and l_1 = 32 bytes

Computational Complexity:

- Generation: 1 exponentiation in G_2, 1 EC multiplication in G_1
- Verification: 1 pairing, 1 exponentiation in G_2, 1 EC multiplication in G_1

3.4 Comparative Overview

Scheme	Signing	Verification	Size
SH-IBS	2 mod. exp. in $\mathbb{Z}_N$	2 mod exp. in $\mathbb{Z}_N$	$\mathbb{Z}_N \times \mathbb{Z}_N$
vBNN-IBS	1 $\cdot$ in $G(E(\mathbb{F}_p))$	3 $\cdot$ in $G(E(\mathbb{F}_p))$	$G(E(\mathbb{F}_q)) \times \mathbb{Z}_p \times \mathbb{Z}_p$
TSO-IBS	1 ^ in G_2, 1 EC $\cdot$ in G_1	1 $\hat{e}()$, 1 ^ in G_2, 1 EC $\cdot$ in G_1	$\|q\| + \|G_1\| - l_1$

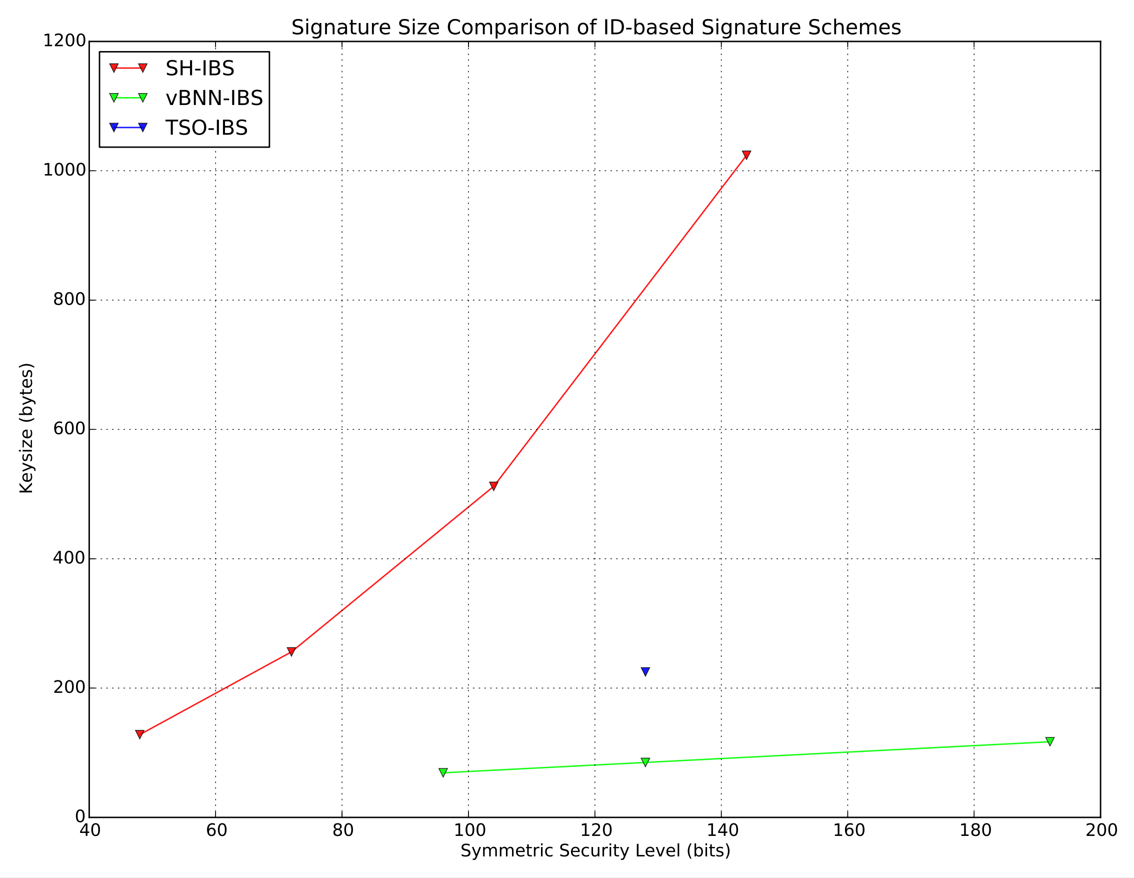
4. Evaluation

- All IBS schemes implemented in C/C++
- Using Relic Toolkit
  - Open source (LGPL)
  - C library, some assembler
  - Protocols, big numbers, elliptic curve, pairings
  - Supported architectures: AVR, MSP, ARM, X86, X86_64
- C++ wrapper
  - Safety: memory management and bounds checking
  - Convenience: operator overloading (+, *, ^, %, ==, =)

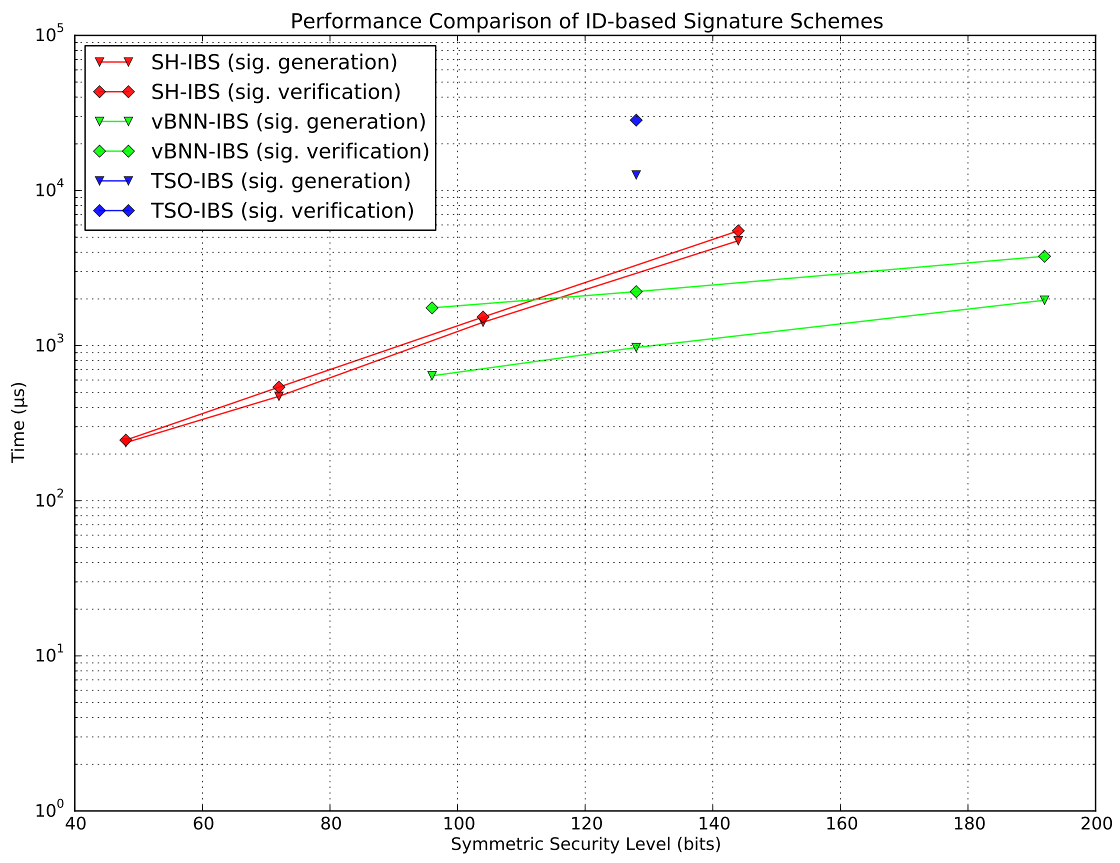
Benchmark

- Benchmark size of signature
- Benchmark timings for
  - Signature generation
  - Signature verification
- For SH-IBS N of size 512, 1024, 2048 and 4096 bits
- For vBNN-IBS curves over $\mathbb{F}_p$ with size of p 192, 256 and 384 bits
- For TSO-IBS a super-singular curve over $\mathbb{F}_p$ with size of p 1536 bits (SLOW)
- Security levels converted to symmetric level according ECRYPT II

Benchmark: Signature Size



Benchmark: Timings



Discussion

- vBNN-IBS shows a speed advantage at good security levels
- VBNN-IBS has smaller signatures overall
- TSO-IBS shows bad performance, due to SS-P1536 curve
- SH-IBS performance shines at lower security levels (like ECDSA vs. RSA)

Outlook

- Evaluation on constrained hardware
  - e.g. Rasberry Pi or sensor nodes
- Signature schemes based on asymmetric pairings
  - Higher efficiency
- Investigating use of Edwards curves
  - Requires dedicated implementation for improved security/performance

Further Reading / Watching

- Upcoming Project 1 Report
- 3rd BIU Winter School on Cryptography 2013

  https://www.youtube.com/playlist?list=PLXF_IJaFk-9C4p3b2tK7H9a9axOm3EtjA

  http://crypto.biu.ac.il/winterschool2013/
- Math ⋂ Programming

  http://jeremykun.com/category/cryptography/
- Relic Toolkit

  https://code.google.com/p/relic-toolkit/

Thanks!

Questions?

Image Sources

- http://upload.wikimedia.org/wikipedia/commons/2/23/Bugaboo_forest_fire.jpg
- http://i1.ytimg.com/vi/L8TkhHgkBsg/maxresdefault.jpg
- http://www.blogcdn.com/www.engadget.com/media/2013/01/pebble2f0a6577.jpg
- http://en.wikipedia.org/wiki/File:ECClines-3.svg
- https://www.imperialviolet.org/2010/12/04/ecc.html