BBS Signature Scheme

From Theory to Implementation

Joël Gabriel Robles Gasser & Miguel Angel Schweizer

Presentation Project 2 BTI3041

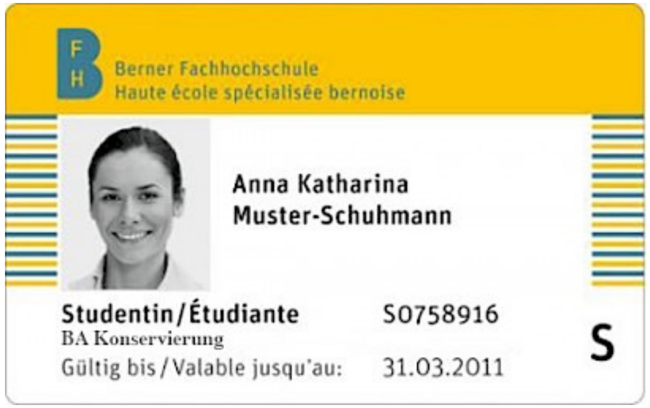
Tasks

- Understanding :
  - Elliptic curves
  - Pairings
  - Why BBS?

- Implement the IETF BBS Signature Scheme Draft Pseudocode into Java code

Example

- BFH Card -> Verified if valid when purchasing something



Berner Fachhochschule
Haute école spécialisée bernoise

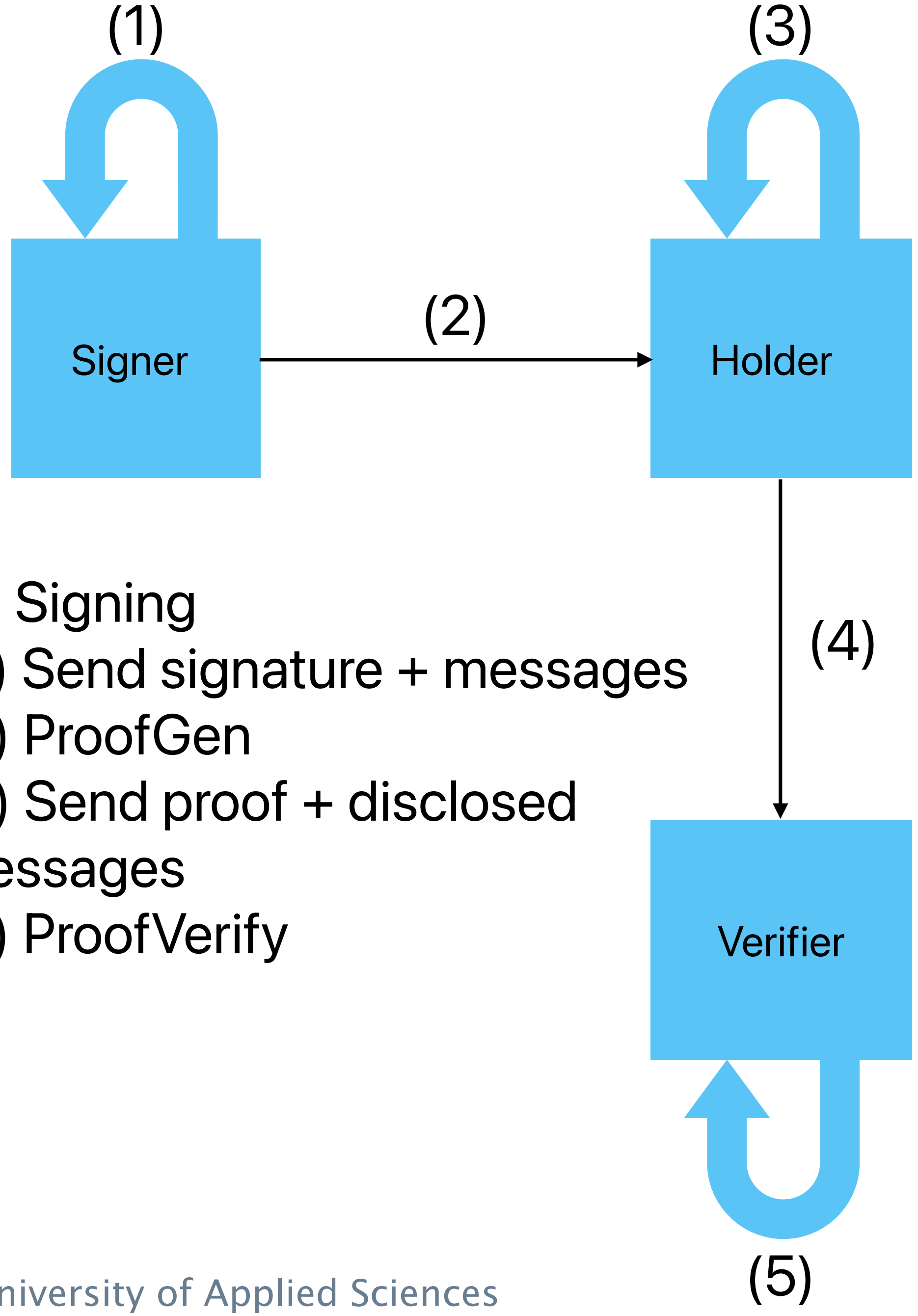
Anna Katharina
Muster-Schuhmann

Studentin/Étudiante S0758916
BA Konservierung
Gültig bis/Valable jusqu'au: 31.03.2011

S

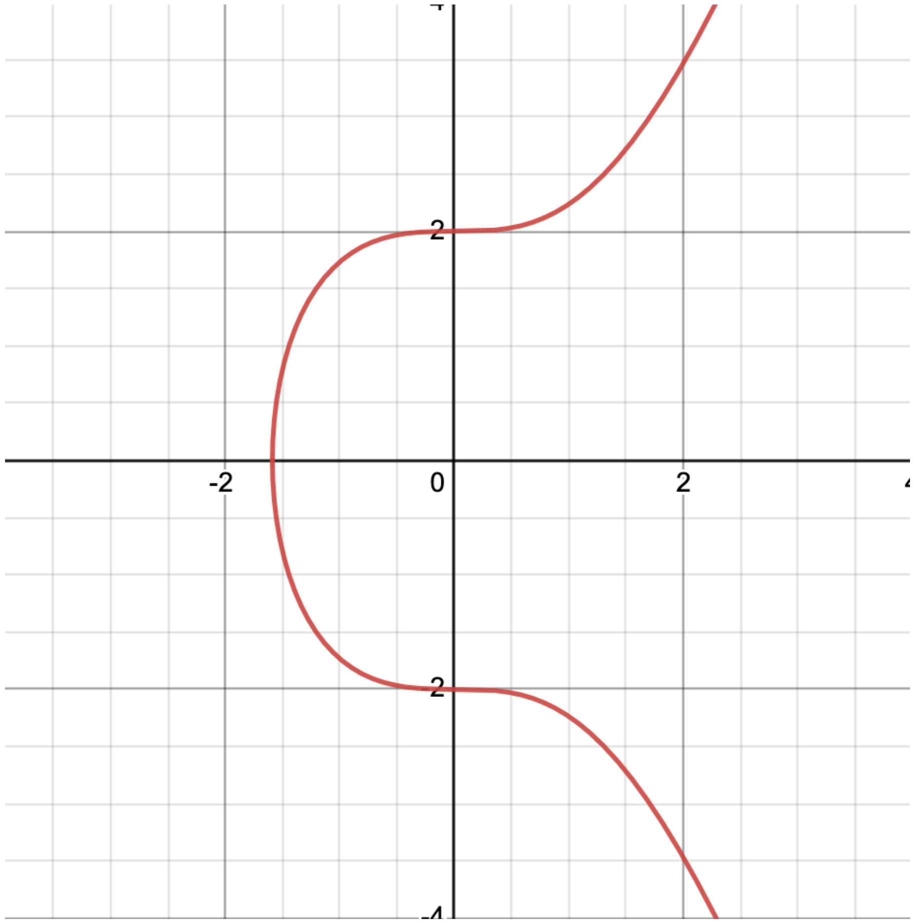
Why is BBS so fancy?

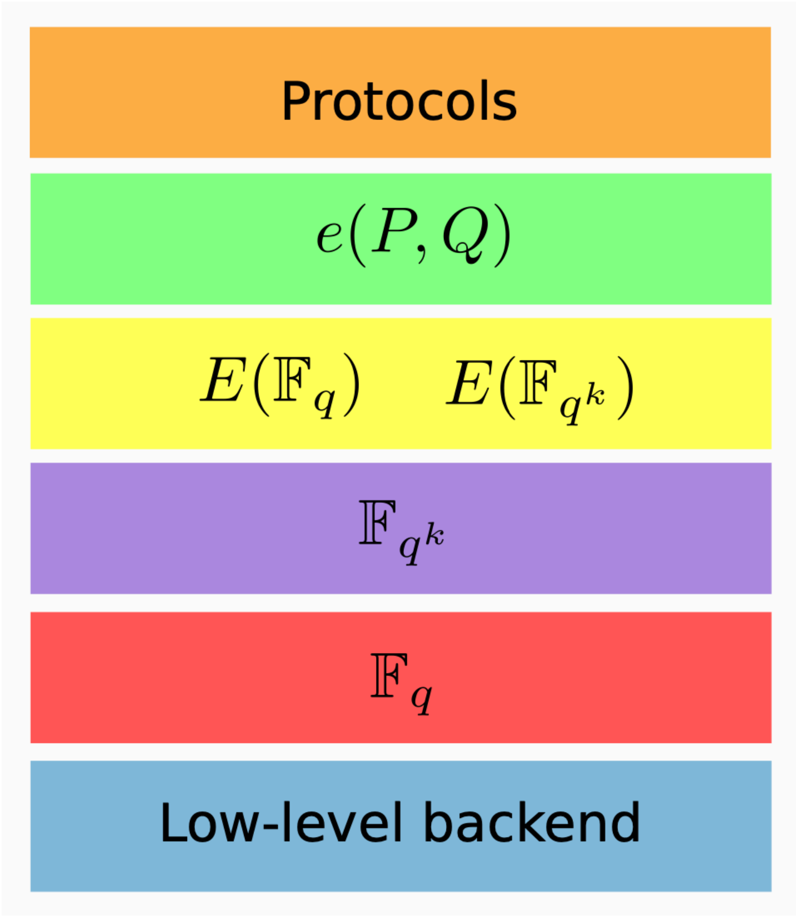
- Verifiable Credentials
- Selective Disclosure
- Proof of Posession
- Unlinkable Proofs



(1) Signing
(2) Send signature + messages
(3) ProofGen
(4) Send proof + disclosed messages
(5) ProofVerify

BLS12-381





Protocols

$e(P, Q)$

$E(\mathbb{F}_q)$ $E(\mathbb{F}_{q^k})$

$\mathbb{F}_{q^k}$

$\mathbb{F}_q$

Low-level backend

Calculations

Slope: $\lambda = \frac{y_q - y_p}{x_q - x_p}$

Point addition: $x_r = \lambda^2 - x_p - x_q$ $\quad$ $y_r = \lambda(x_p - x_q) - y_p$

Point doubling: $\lambda = \frac{3{x_p}^2 + a}{2y_p}$

Calculations in F_q^2:

$$\begin{aligned}
-(a + b\alpha) &= -a + (-b)\alpha \\
(a + b\alpha) + (c + d\alpha) &= (a + c) + (b + d)\alpha \\
(a + b\alpha)(c + d\alpha) &= (ac + rbd) + (ad + bc)\alpha \\
(a + b\alpha)^{-1} &= a(a^2 - rb^2)^{-1} + (-b)(a^2 - rb^2)^{-1}\alpha
\end{aligned}$$

Pairings

- *e(P,Q)*

- Bilinearity

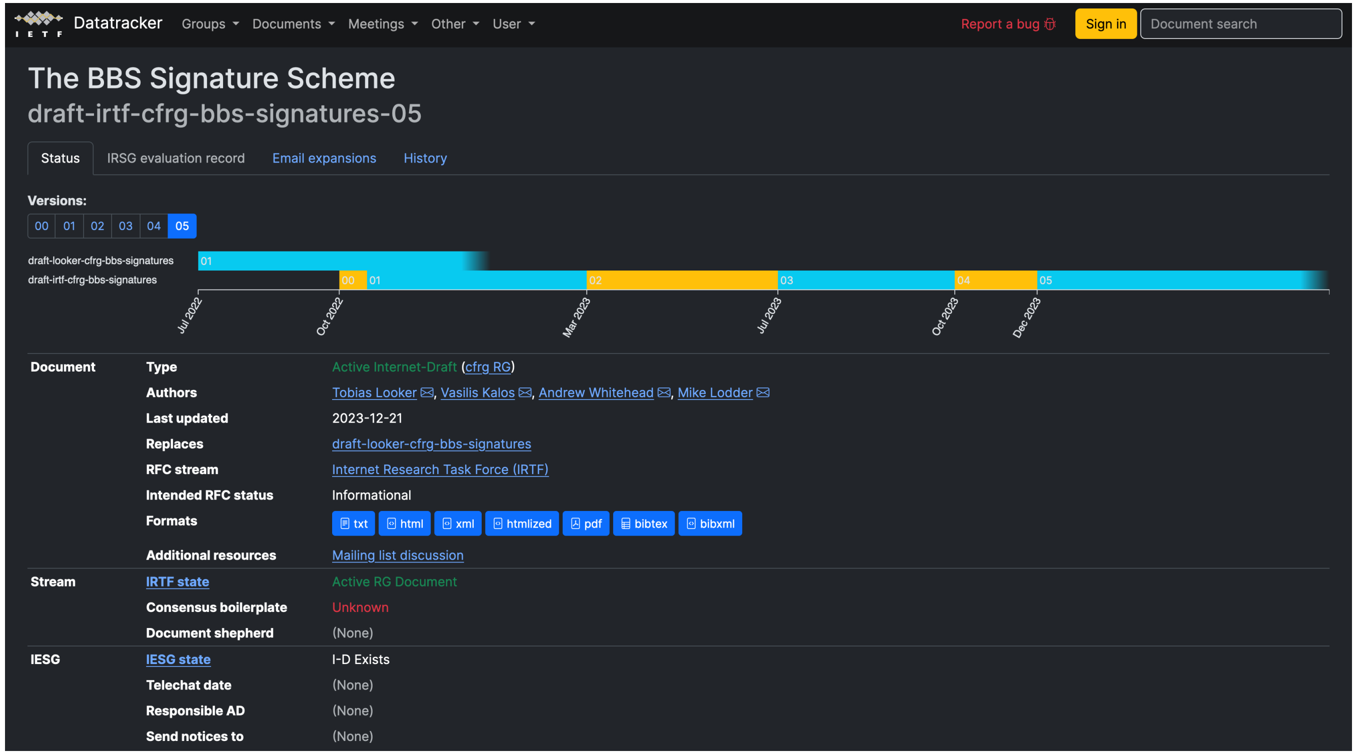
$$B(u+v,w) = B(u,v) + B(v,w)$$

$$B(au,v) = aB(u,v) \text{ \& } B(u,bv) = bB(u,v)$$

- Weil pairing
- Tate pairing
- Ate pairing

The Recipe

- IETF BBS Signature Scheme Draft



The BBS Signature Scheme

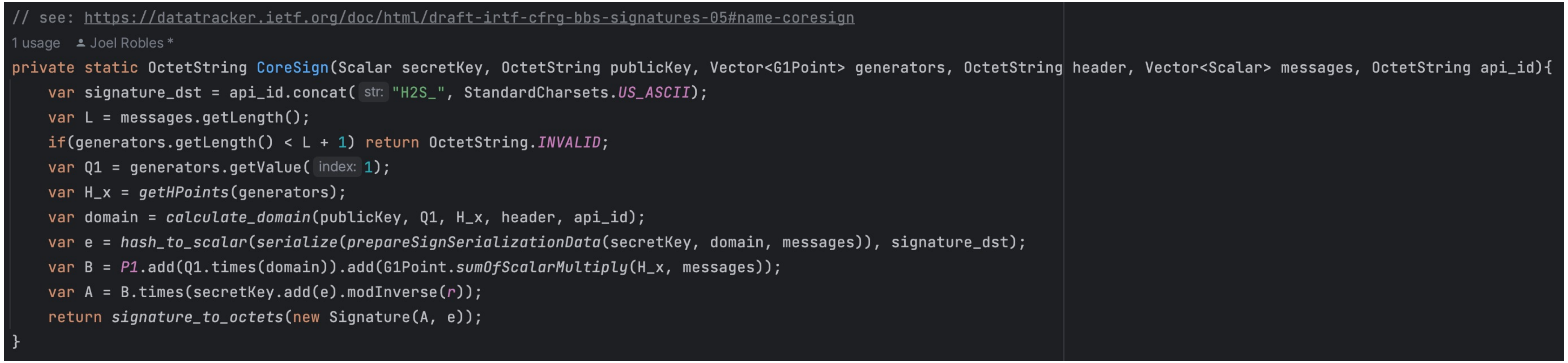
draft-irtf-cfrg-bbs-signatures-05

Versions: 00 01 02 03 04 05

Document	**Type**	Active Internet-Draft (cfrg RG)
	Authors	Tobias Looker, Vasilis Kalos, Andrew Whitehead, Mike Lodder
	Last updated	2023-12-21
	Replaces	draft-looker-cfrg-bbs-signatures
	RFC stream	Internet Research Task Force (IRTF)
	Intended RFC status	Informational
	Formats	txt html xml htmlized pdf bibtex bibxml
	Additional resources	Mailing list discussion
Stream	**IRTF state**	Active RG Document
	Consensus boilerplate	Unknown
	Document shepherd	(None)
IESG	**IESG state**	I-D Exists
	Telechat date	(None)
	Responsible AD	(None)
	Send notices to	(None)

Implementation

- Java
- MCL library
  - Problems with ARM



```
// see: https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-bbs-signatures-05#name-coresign
1 usage   Joel Robles *
private static OctetString CoreSign(Scalar secretKey, OctetString publicKey, Vector<G1Point> generators, OctetString header, Vector<Scalar> messages, OctetString api_id){
    var signature_dst = api_id.concat( str: "H2S_", StandardCharsets.US_ASCII);
    var L = messages.getLength();
    if(generators.getLength() < L + 1) return OctetString.INVALID;
    var Q1 = generators.getValue( index: 1);
    var H_x = getHPoints(generators);
    var domain = calculate_domain(publicKey, Q1, H_x, header, api_id);
    var e = hash_to_scalar(serialize(prepareSignSerializationData(secretKey, domain, messages)), signature_dst);
    var B = P1.add(Q1.times(domain)).add(G1Point.sumOfScalarMultiply(H_x, messages));
    var A = B.times(secretKey.add(e).modInverse(r));
    return signature_to_octets(new Signature(A, e));
}
```

- New library from supervisor Rolf Haenni

Encountered Problems

- Difficult Research
- Advanced Mathematics
- Constant change of the draft
- Some mistakes in the draft
- MCL library
- Test Vectors

Is the goal achieved?

- Implemented Java code:
  - YES!

- Understanding:
  - YES!

- Working Java code for the next step -> Bachelor-Thesis

Questions

Thank you for listening!