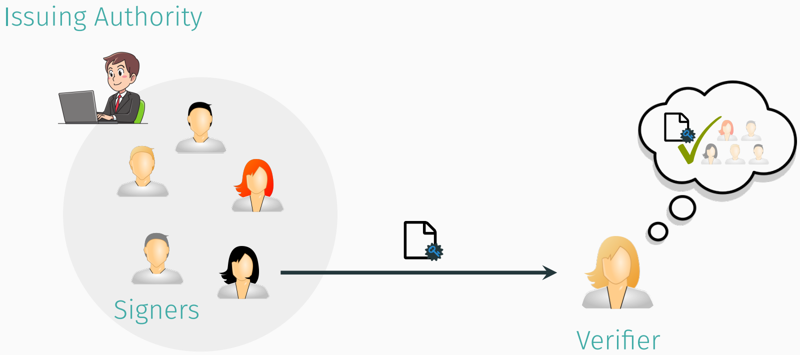
Highly-Efficient Fully-Anonymous Dynamic Group Signatures

David Derler[‡], Daniel Slamanig[§]

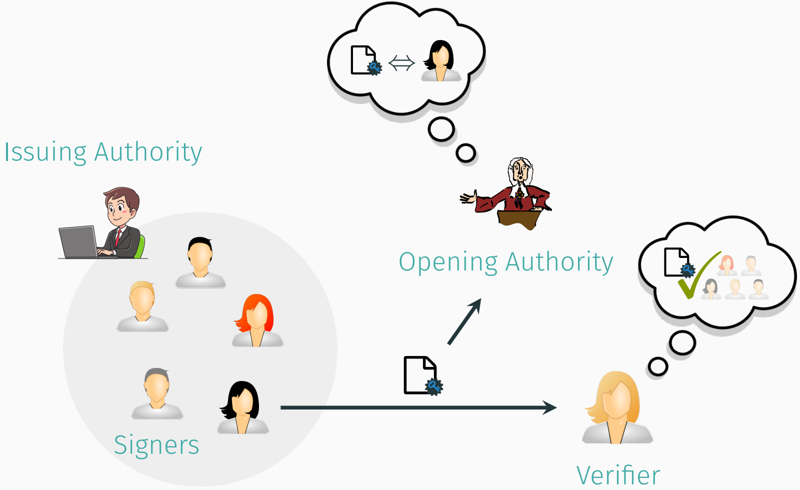
June 7, 2018—AsiaCCS 2018, Incheon, Korea

‡ TU Graz § AUSTRIAN INSTITUTE OF TECHNOLOGY

Group Signatures



Group Signatures



Group Signatures - Security

Anonymity

- Signers stay anonymous
- Full anonymity: Even when signer keys leak! [BSZ05]

Traceability

- Opening authority can trace valid signatures to signers

Non-frameability

- Nobody can produce signatures for honest signers

Opening soundness [SSEHO12]

- Only signer can claim ownership of honest signatures

Why (Revocable) Privacy in Authentication?

- Revealing unique user ID allows tracking!
- Proof of group membership often sufficient
  - floating car data, toll systems, parking, ticketing, etc.
- Re-identification (opening) required
  - E.g., court order

Our Goals

High Efficiency

- Signers typically computationally constrained
  - E.g., smart NFC tickets for public transportation

Full Anonymity

- Anonymity of signers even if keys get public and arbitrary signatures get opened

Dynamic Groups

- Dynamic enrollment of users instead of static setup

Overview

- Existing paradigms & our construction
- Comparison to existing schemes
- Benchmarks
- Conclusions

Group Signature Paradigms

Sign-Encrypt-Prove (SEP)

- GS is an encrypted membership certificate (=signature) + signature of knowledge

Sign-Randomize-Prove (SRP)

- GS is a randomizable signature (unlinkable) + proof of knowledge of membership certificate

So far we have SEP schemes with full anonymity, but SRP schemes only provide weaker anonymity

We propose the first SPR scheme with full anonymity

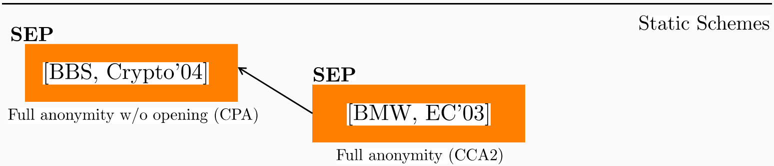
Static Schemes

SEP

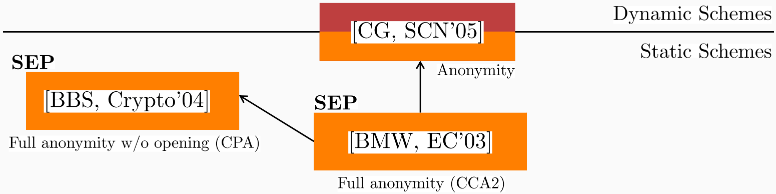
[BMW, EC'03]

Full anonymity (CCA2)

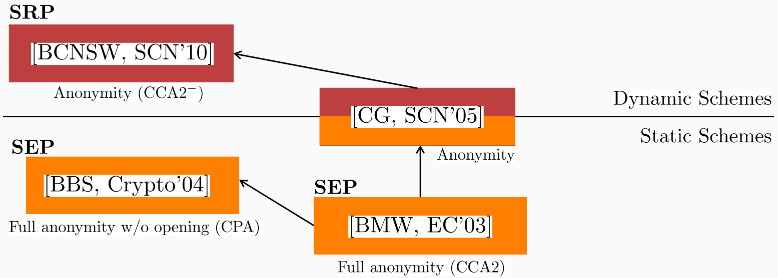
Group Signature Models



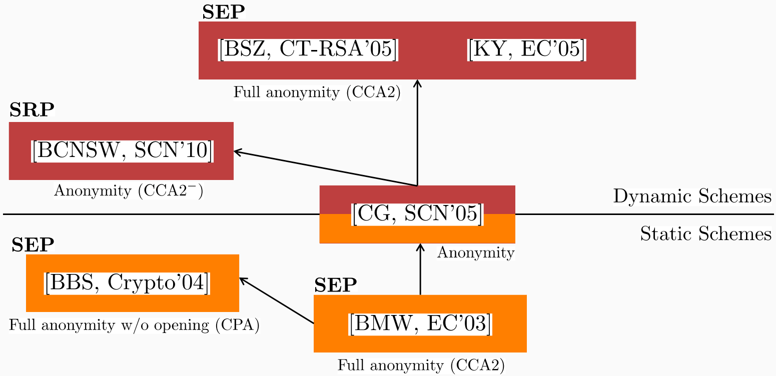
Group Signature Models



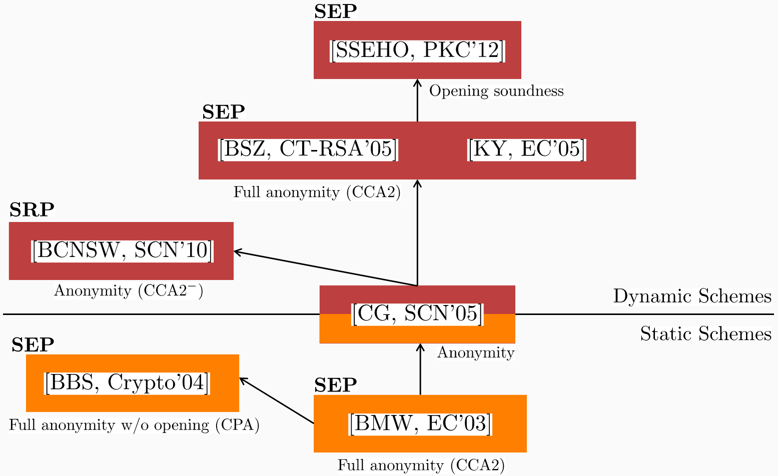
Group Signature Models



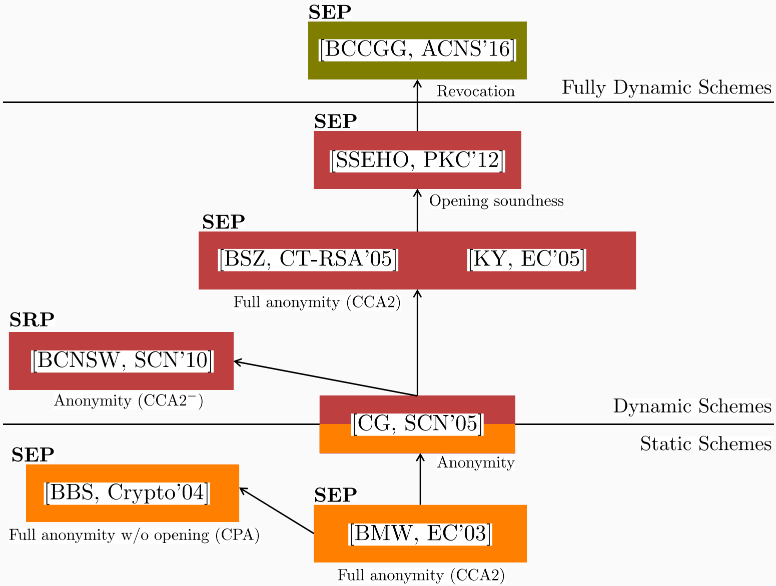
Group Signature Models



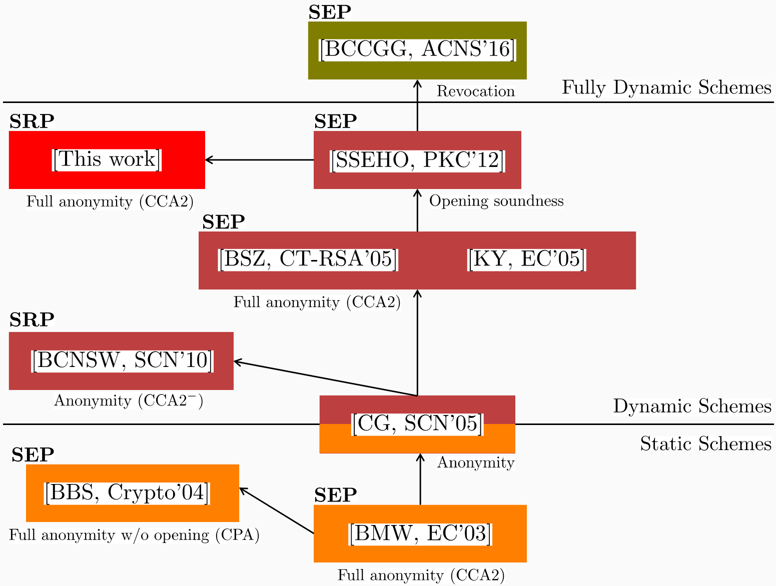
Group Signature Models



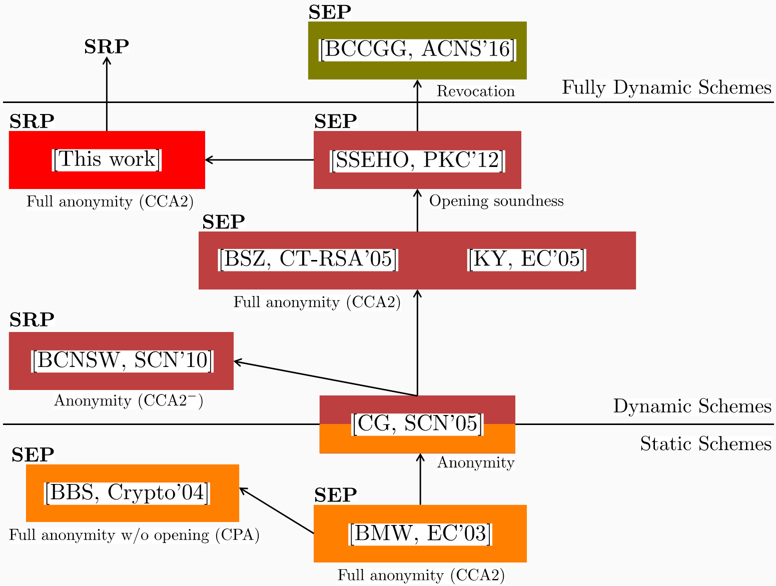
Group Signature Models



Group Signature Models



Group Signature Models



Construction - Setting

Asymmetric bilinear map (pairing)

- $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = p$
- $e(g^a, \hat{g}^b) = e(g, \hat{g})^{ab}$ (bilinearity)
- $e(g, \hat{g}) \neq 1_{\mathbb{G}_T}$ (non-degeneracy)
- $e(\cdot, \cdot)$ efficiently computable (efficiency)

Construction - Setting

Asymmetric bilinear map (pairing)

- $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = p$
- $e(g^a, \hat{g}^b) = e(g, \hat{g})^{ab}$ (bilinearity)
- $e(g, \hat{g}) \neq 1_{\mathbb{G}_T}$ (non-degeneracy)
- $e(\cdot, \cdot)$ efficiently computable (efficiency)

SXDH setting

- DDH assumed to hold in $\mathbb{G}_1$ and $\mathbb{G}_2$

$$(g^a, g^b, g^{ab}) \approx (g^a, g^b, g^r)$$

and

$$(\hat{g}^a, \hat{g}^b, \hat{g}^{ab}) \approx (\hat{g}^a, \hat{g}^b, \hat{g}^r)$$

Structure Preserving Signatures [AFG+10]

Signature scheme

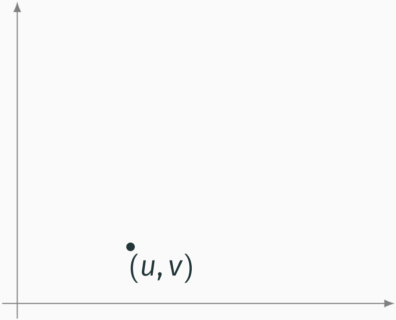
- Sign group element vectors
- Signatures and public keys consist only of group elements
- Verification uses solely
  - pairing-product equations

$$\prod_i \prod_j e(A_i, \hat{B}_j)^{a_{ij}} = Z$$

  - group membership tests

Our Construction - Building Blocks

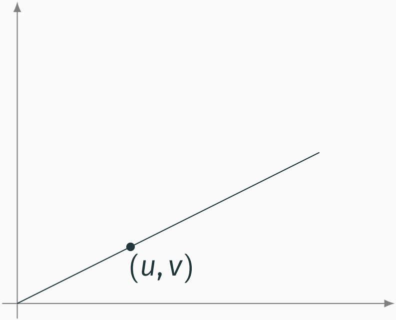
Structure-preserving signatures on EQ classes (SPS-EQ) [HS14,FHS18]



- Vector of group elements

Our Construction - Building Blocks

Structure-preserving signatures on EQ classes (SPS-EQ) [HS14,FHS18]

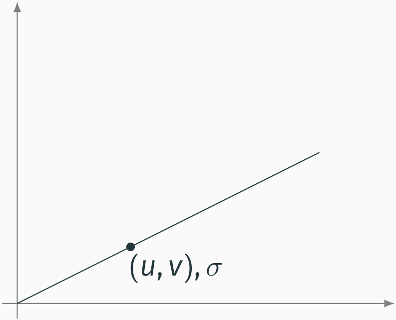


- Vector of group elements
- EQ classes

  $\sim_{\mathcal{R}}$ mutual ratios of DLOGs

Our Construction - Building Blocks

Structure-preserving signatures on EQ classes (SPS-EQ) [HS14,FHS18]

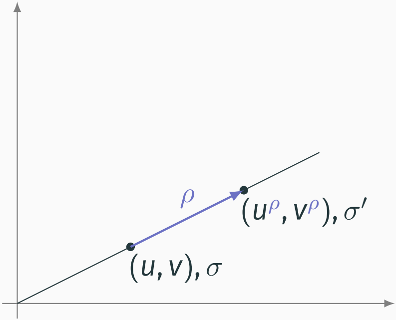


- Vector of group elements
- EQ classes

  $\sim_{\mathcal{R}}$ mutual ratios of DLOGs
- Sign representative

Our Construction - Building Blocks

Structure-preserving signatures on EQ classes (SPS-EQ) [HS14,FHS18]

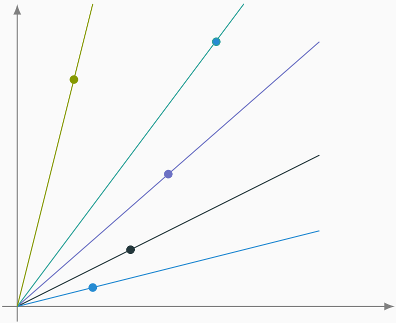


- Vector of group elements
- EQ classes

  $\sim_{\mathcal{R}}$ mutual ratios of DLOGs
- Sign representative
- Switch representative (publicly)

Our Construction - Building Blocks

Structure-preserving signatures on EQ classes (SPS-EQ) [HS14,FHS18]



- Vector of group elements
- EQ classes

  $\sim_{\mathcal{R}}$ mutual ratios of DLOGs
- Sign representative
- Switch representative (publicly)

Perfect adaption [FHS15]

- Adapted signatures indistinguishable from fresh ones

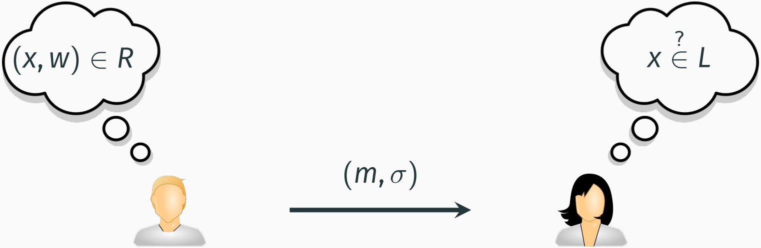
Class-Hiding msg. space [FHS15]

- No advantage in distinguishing classes using signatures

Our Construction - Building Blocks

Signatures of knowledge (SoK) [CL06, BCC+15]

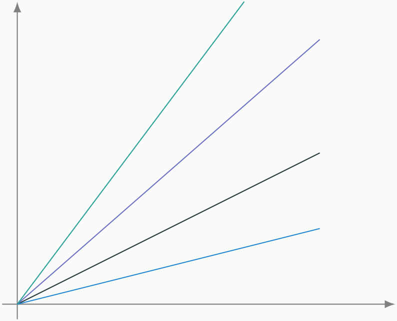
- NP-language L w.r.t. relation R
- $x \in L \iff \exists w : (x, w) \in R$



$\sigma \leftarrow \mathsf{Sign}(x, w, m)$ $\checkmark / \times \leftarrow \mathsf{Verify}(x, m, \sigma)$

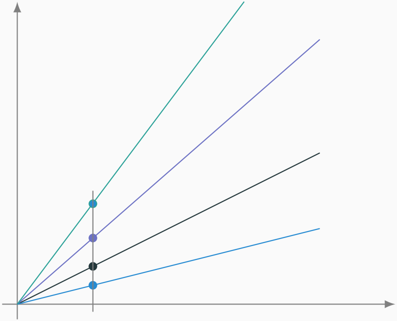
Guarantees: signer knows w, yet signature does not “leak” w

Our Construction



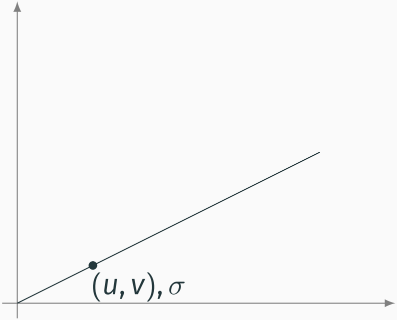
- User signing keys
  - 1 EQ-class per user

Our Construction



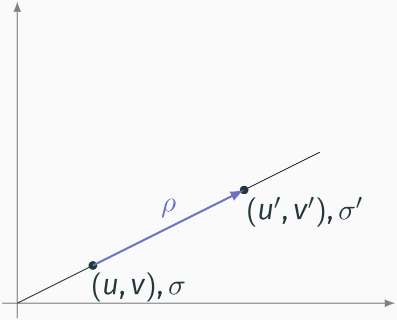
- User signing keys
  - 1 EQ-class per user
  - Sign representative (1 component fixed)

Our Construction



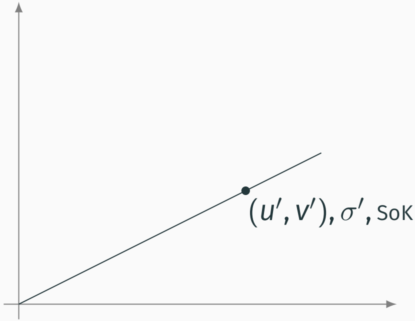
- User signing keys
  - 1 EQ-class per user
  - Sign representative (1 component fixed)
- Group signature on m

Our Construction



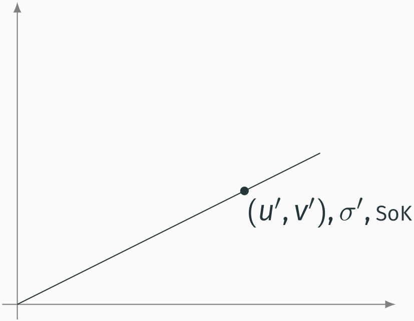
- User signing keys
  - 1 EQ-class per user
  - Sign representative (1 component fixed)
- Group signature on m
  - Randomize signing key

Our Construction



- User signing keys
  - 1 EQ-class per user
  - Sign representative (1 component fixed)
- Group signature on m
  - Randomize signing key
  - \+ SoK w.r.t. ρ s.t. $u^{\rho} = u'$

Our Construction



- User signing keys
  - 1 EQ-class per user
  - Sign representative (1 component fixed)
- Group signature on m
  - Randomize signing key
  + SoK w.r.t. ρ s.t. $u^{\rho} = u'$

Security (very roughly)

- **Anonymity:** Perfect adaption, DDH on msg. space + SoK
- **Traceability:** Unforgeability of SPS-EQ
- **Non-frameability:** co-CDHI [1]+ SoK

[1] Diffie-Hellman Inversion assumption in Type-3 groups

Comparison: Performance

Scheme	Anon.	Signature Size	Signature Cost	Verification Cost
[BCN+10]	CCA^-	1273bit	351ms	1105ms
[PS16]	CCA^-	1018bit	318ms	777ms
[BBS04]	CPA	2289bit	1545ms	2092ms
[BBS04] (prec.)	CPA	2289bit	1053ms	1600ms
Our work	CPA	**2037bit**	**266ms**	**886ms**
Our work	CCA2	3309bit	**771ms**	**1290ms**
Our work (switch)	CCA2	3563bit	**703ms**	**1154ms**
[DP06]	CCA2	2290bit	1380ms	2059ms
[DP06] (prec.)	CCA2	2290bit	1020ms	1353ms
[LMPY16]	CCA2	2547bit	1688ms	2299ms

Comparison: Performance

Scheme	Anon.	Signature Size	Signature Cost	Verification Cost
[BCN+10]	CCA^-	1273bit	351ms	1105ms
[PS16]	CCA^-	1018bit	318ms	777ms
[BBS04]	CPA	2289bit	1545ms	2092ms
[BBS04] (prec.)	CPA	2289bit	1053ms	1600ms
Our work	CPA	**2037bit**	**266ms**	**886ms**
Our work	CCA2	3309bit	**771ms**	**1290ms**
Our work (switch)	CCA2	3563bit	**703ms**	**1154ms**
[DP06]	CCA2	2290bit	1380ms	2059ms
[DP06] (prec.)	CCA2	2290bit	1020ms	1353ms
[LMPY16]	CCA2	2547bit	1688ms	2299ms

- Uses performance values from [UW14]
  (Group operations/pairings on ARM-Cortex-M0+ using 254-bit BN curves)
- Based on counting expensive operations
- **Our** Sign **only requires** $\mathbb{G}_1$ **operations!**

Comparison: Assumptions

Scheme	Anon.	Assumptions
[BCN+10]	CCA^-	Interactive
[PS16]	CCA^-	GGM
[BBS04]	CPA	q-Type (non-static) & DCR
[BBS04] (prec.)	CPA	q-Type (non-static) & DCR
Our work	CPA	GGM
Our work	CCA2	GGM
Our work (switch)	CCA2	GGM
[DP06]	CCA2	q-Type (non-static) & DCR
[DP06] (prec.)	CCA2	q-Type (non-static) & DCR
[LMPY16]	CCA2	standard

Much easier impl. than other CPA/CCA2 candidates

- Combines simplicity of CCA^- schemes w. CPA/CCA2 security

Benchmarking Results

Comparison

- Our CCA2-fully anonymous scheme
- vs. the scheme in [DP06]

Setting

- Intel Core i7-4790, 16 GB RAM
- Ubuntu 17.04
- JMH benchmarking framework
- IAIK BN pairing implementation

Benchmarking Results

Security level

- According to recent estimations [BD17]
- 100 bit $\rightarrow$ 256 bit BN curves
- 128 bit $\rightarrow$ 462 bit BN curves

Increased throughput upon signing

100 bit: **2** $\times$ faster

128 bit: **2.5** $\times$ faster

Observations

- Our advantage increases with increasing security level
- In contrast to others, no $\mathbb{G}_T$ operations

Conclusions

Efficiency

- ✓ **Fastest** known group signature scheme
- ✓ **Fastest** singing and verification among CPA/CCA2
- ✓ **Shortest** signatures among CPA
- Slightly more progressive assumptions

Much easier impl. than other CPA/CCA2 candidates

- Combines simplicity of CCA^{-} schemes w. CPA/CCA2 security

Favorable properties

- No $\mathbb{G}_T$ operations for signing
- Even more favorable with increasing security level

Thank you! Questions?

@drl3c7er