More Efficient Digital Signatures with Tight Multi-User Security

Denis Diemert Kai Gellert Tibor Jager Lin Lyu

IT Security and Cryptography Group
University of Wuppertal

PKC 2021

BERGISCHE
UNIVERSITÄT
WUPPERTAL

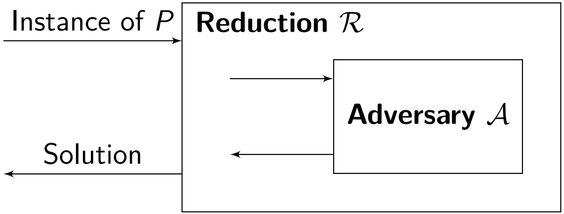
This work

- Tightly-secure signatures in the multi-user setting with adaptive corruption
- First generic construction based on lossy identification schemes and OR-Proofs
  - We build upon the work of Abe et al. (AC'02) and Fischlin et al. (EC'20)
- Strong unforgeability: first tightly multi-user-secure signature with adaptive corruption
- Short signatures: Instantiated with DDH signature consists only of $3\mathbb{Z}_q$ elements
- Perfect candidate to instantiate tightly-secure authenticated key exchange (AKE)

Tightly Multi-User-Secure Signatures

Cryptographic Reductions

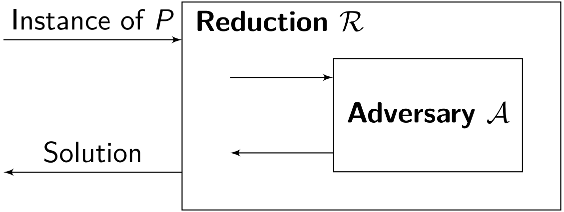
- Hardness of problem $P \implies$ security of scheme Π
- Proof: Adversary $\mathcal{A}$ breaking scheme $\Pi \implies$ algorithm $\mathcal{R}$ solving problem P



- $\mathcal{A}$ with success $\epsilon \quad \rightsquigarrow \quad \mathcal{R}$ with success $\epsilon/\ell \qquad (\ell$: security loss$)$

Larger security loss $\ell \Rightarrow$ weaker security garantuees $\Rightarrow$ harder instance of $P \Rightarrow$ inefficient deployment

Tight Cryptographic Reductions



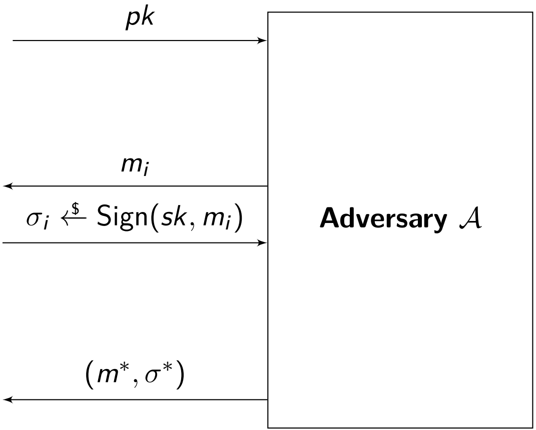
Definition (Tight Reduction)

We say a reduction $\mathcal{R}$ is tight if $\quad \text{time}_{\mathcal{R}} \approx \text{time}_{\mathcal{A}} \quad$ and $\quad \epsilon_{\mathcal{R}} \geq \epsilon_{\mathcal{A}}/\ell \quad$ (ℓ small).

- That is, security loss ℓ is a small constant
- Optimal choice of parameters $\Rightarrow$ optimal balance between security and efficiency

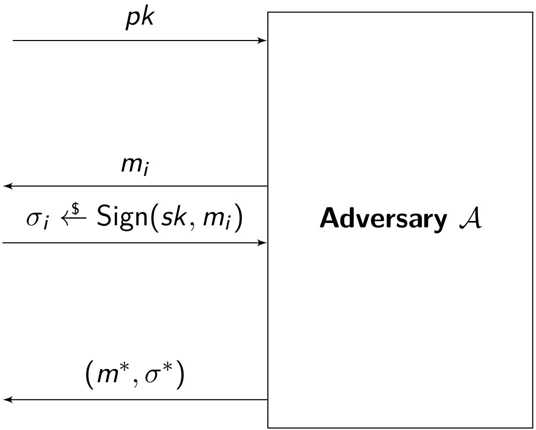
EUF-CMA

"Single-User Security"



EUF-CMA

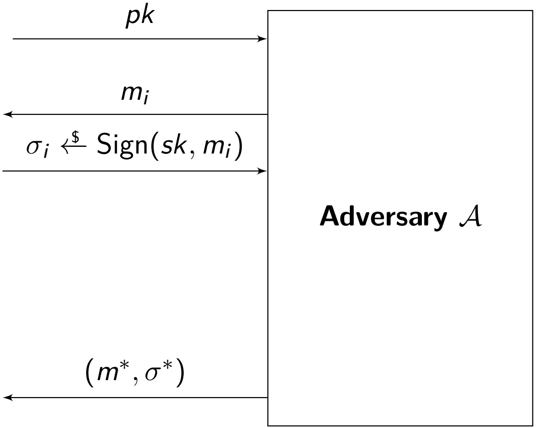
"Single-User Security"



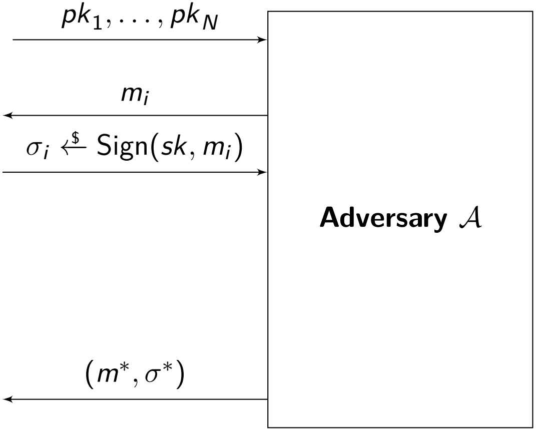
Adversary $\mathcal{A}$ wins if

1. (m^*, σ^*) is valid, and
2. $\mathcal{A}$ did not query a signature for m^*.

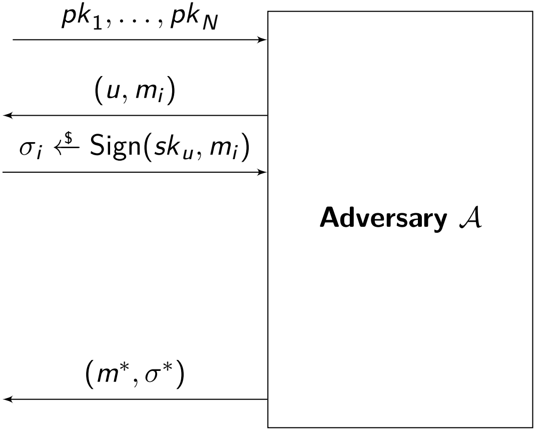
A Multi-User Variant – MU-EUF-CMA$^{\text{corr}}$



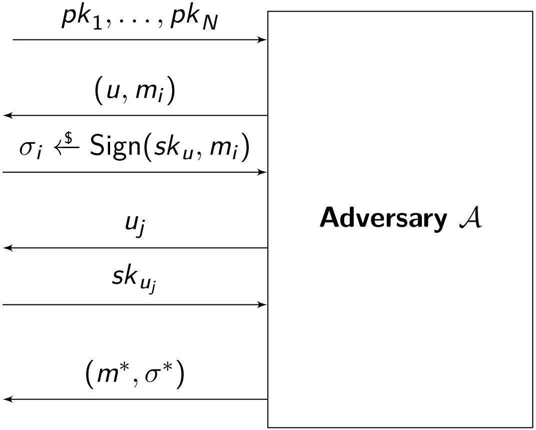
A Multi-User Variant – MU-EUF-CMA$^{\text{corr}}$



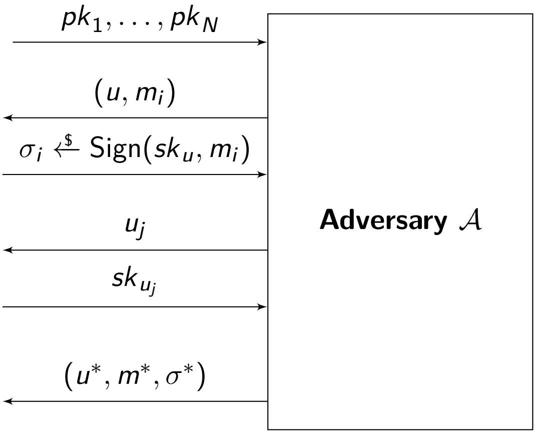
A Multi-User Variant – MU-EUF-CMA$^{\text{corr}}$



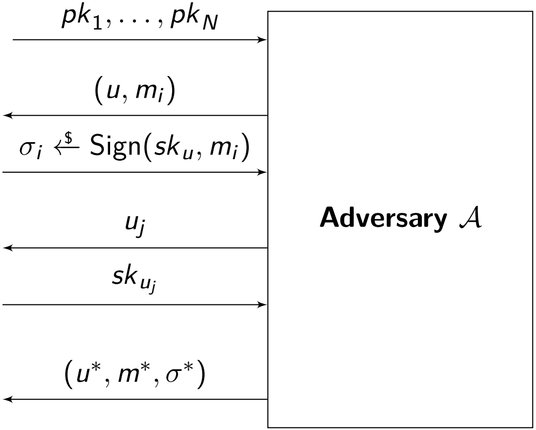
A Multi-User Variant – MU-EUF-CMA$^{\mathrm{corr}}$



A Multi-User Variant – MU-EUF-CMA$^{\text{corr}}$



A Multi-User Variant – MU-EUF-CMA$^{\mathrm{corr}}$



Adversary $\mathcal{A}$ wins if

1. (m^*, σ^*) is valid under pk_{u^*},
2. $\mathcal{A}$ did not query a signature for m^* under sk_{u^*}, and
3. $\mathcal{A}$ did not query for sk_{u^*}.

EUF-CMA $\implies$ MU-EUF-CMA$^{\text{corr}}$

- Reduction is a straightforward guessing argument:
  - Guess user $\hat{u}$ for which the adversary outputs a forgery

EUF-CMA $\implies$ MU-EUF-CMA$^{\mathsf{corr}}$

- Reduction is a straightforward guessing argument:
  - Guess user $\hat{u}$ for which the adversary outputs a forgery
- "Problem" with this reduction: it is only successful if guess $\hat{u}$ is correct, i.e.

$$\epsilon_R \geq \frac{1}{N} \cdot \epsilon_A$$

$\implies$ Reduction is not tight! Loss ℓ is linear in #users N

Difficulty of Constructing Tightly-Secure MU-EUF-CMA$^{\mathrm{corr}}$ Signatures

A (seemingly) Paradox to Solve

- To avoid guessing, the reduction needs to satisfy
  1. Knowing all secret keys of *all* users (to answer corruption queries), AND
  2. Being able to extract a solution to the underlying assumption from a forgery while knowing the secret key of the corresponding instance

Difficulty of Constructing Tightly-Secure MU-EUF-CMA$^{\mathrm{corr}}$ Signatures

A (seemingly) Paradox to Solve

- To avoid guessing, the reduction needs to satisfy
  1. Knowing all secret keys of *all* users (to answer corruption queries), AND
  2. Being able to extract a solution to the underlying assumption from a forgery while knowing the secret key of the corresponding instance

Impossibility of a Tight Reduction

- Bader et al. (EC'16): Impossibility of tightly-MU-EUF-CMA$^{\mathrm{corr}}$-secure signatures under non-interactive assumptions
  - Result only holds for signatures schemes satifying certain properties

Construction

Lossy Identification Schemes (LID) – Abdalla et al. (EC'12)

Syntax like a "standard" identification protocol:

$$(pk, sk) \xleftarrow{\$} \mathsf{LID.Gen}$$

Prover: sk		**Verifier:** pk
$(\mathsf{cmt}, \mathsf{st}) \xleftarrow{\$} \mathsf{LID.Prove}_1(sk)$	$\xrightarrow{\quad\mathsf{cmt}\quad}$	
	$\xleftarrow{\quad\mathsf{ch}\quad}$	$\mathsf{ch} \xleftarrow{\$} \mathsf{CSet}$
$\mathsf{resp} \leftarrow \mathsf{LID.Prove}_2(sk, \mathsf{cmt}, \mathsf{ch}, \mathsf{st})$	$\xrightarrow{\quad\mathsf{resp}\quad}$	**return** $\mathsf{LID.Vrfy}(pk, \mathsf{cmt}, \mathsf{ch}, \mathsf{resp})$

Properties of LID

Lossiness

- "Lossy" key generation algorithm: $pk \xleftarrow{\$} \mathsf{LID.LossyGen}$
- Impossible to find a valid transcript if ID scheme is in lossy mode
- Normal pk is indistinguishable from lossy pk

Additional properties: completeness, simulatability and uniqueness

Properties of LID

Lossiness

- "Lossy" key generation algorithm: $pk \xleftarrow{\$} \mathsf{LID.LossyGen}$
- Impossible to find a valid transcript if ID scheme is in lossy mode
- Normal pk is indistinguishable from lossy pk

Additional properties: completeness, simulatability and uniqueness

Commitment Recoverability (Kiltz et al. (C'16)) – Intuition

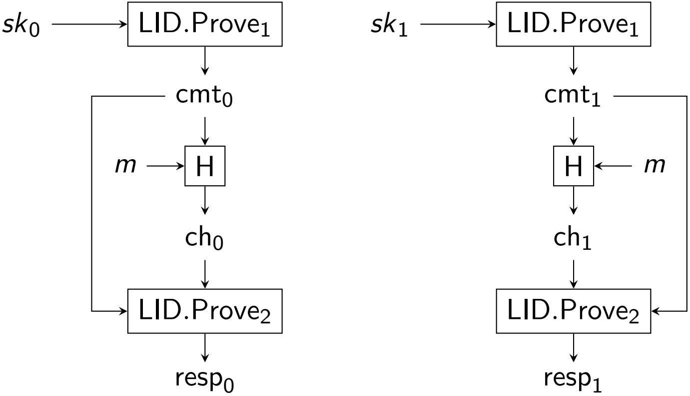
Algorithm LID.Sim that on input $(pk, \mathsf{ch}, \mathsf{resp})$ outputs cmt s.t. $\mathsf{LID.Vrfy}(pk, \mathsf{cmt}, \mathsf{ch}, \mathsf{resp}) = 1$

Intuition of the Construction

How to solve the paradox to achieve tight multi-user security?

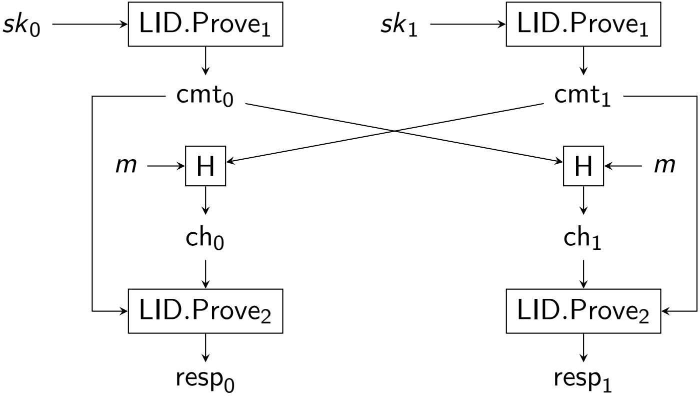
- Basic idea: Use a “double signature” (Bader et al. (TCC’15))
- Signature consists indistinguishable “real” and “fake” component
- Foundation:
  - Signature based on LID by Abdalla et al. (EC’12) (Fiat-Shamir transform)

Construction



Signature: $\sigma = (\mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{resp}_0, \mathsf{resp}_1)$

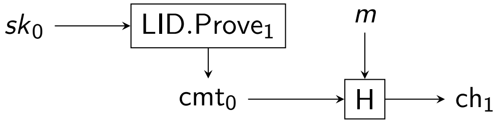
Construction – "Sequential" OR-Proofs by Abe et al. (AC'02)



Signature: $\sigma = (\mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{resp}_0, \mathsf{resp}_1)$

Construction – "Sequential" OR-Proofs by Abe et al. (AC'02)

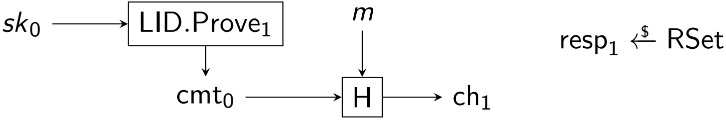
Assumption: $pk = (pk_0, pk_1)$, sk_0



Output: $\sigma = (\mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{resp}_0, \mathsf{resp}_1)$

Construction – “Sequential” OR-Proofs by Abe et al. (AC’02)

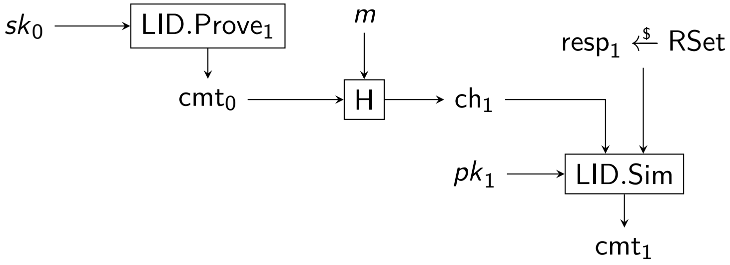
Assumption: $pk = (pk_0, pk_1)$, sk_0



Output: $\sigma = (\mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{resp}_0, \mathsf{resp}_1)$

Construction – “Sequential” OR-Proofs by Abe et al. (AC’02)

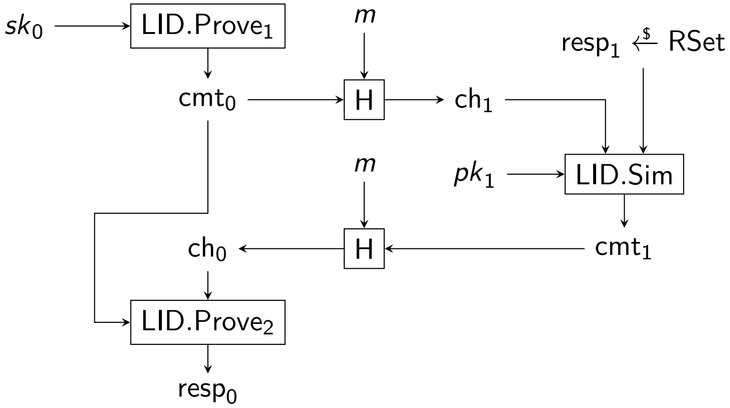
Assumption: $pk = (pk_0, pk_1)$, sk_0



Output: $\sigma = (\mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{resp}_0, \mathsf{resp}_1)$

Construction – “Sequential” OR-Proofs by Abe et al. (AC’02)

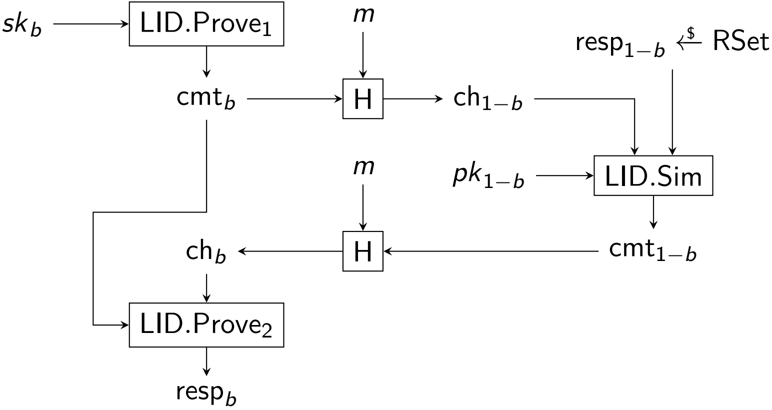
Assumption: $pk = (pk_0, pk_1)$, sk_0



Output: $\sigma = (\mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{resp}_0, \mathsf{resp}_1)$

Construction – "Sequential" OR-Proofs by Abe et al. (AC'02)

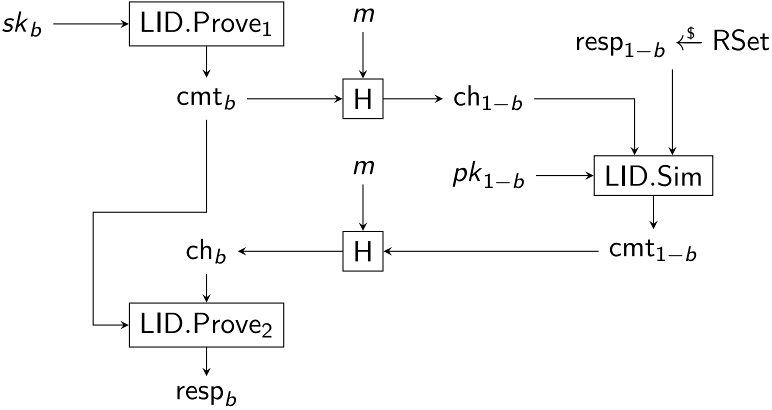
Input: $pk = (pk_0, pk_1)$, $sk = (b, sk_b)$, m



Output: $\sigma = (\mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{resp}_0, \mathsf{resp}_1)$

Construction – Our Refined Variant

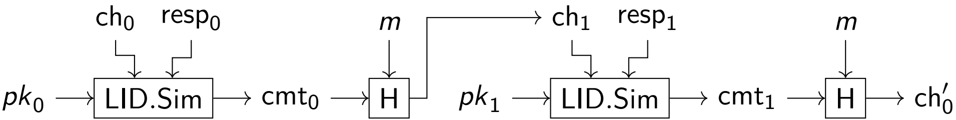
Input: $pk = (pk_0, pk_1)$, $sk = (b, sk_b)$, m



Output: $\sigma = (\mathsf{ch}_0, \mathsf{resp}_0, \mathsf{resp}_1)$

Construction – Verification

Input: $pk = (pk_0, pk_1)$, $\sigma = (\mathsf{ch}_0, \mathsf{resp}_0, \mathsf{resp}_1)$



Output: $1 \iff \mathsf{ch}_0' = \mathsf{ch}_0$

Security

- Fischlin et al. (EC'20): Tight "single-user" security in the NPROM
- Our result: Tight multi-user security (MU-sEUF-CMA$^{\mathsf{corr}}$) in the NPROM
  - "real" and "fake" component of the signature are indistinguishable for any user
  - Adversary outputs with probability $1/2$ a forgery for the "fake" component
  - This enables to construct a tight reduction to the lossiness of the LID scheme

Comparision with Existing Tightly Multi-User-Secure Signatures

Existing tightly MU-EUF-CMA$^{\text{corr}}$-secure signatures

Bader et al. (BHJKL) (TCC'15):

- First tightly MU-EUF-CMA$^{\text{corr}}$-secure signatures
- Standard model, pairing-based
- Large signatures $\implies$ impractical
- "almost-tight" variant with shorter signatures

Gjøsteen and Jager (GJ) (C'18):

- Based on ("parallel") OR-Proofs (Cramer et al. (C'94))
- Requires a programmable random oracle
- Efficient signatures size

Comparison to Existing MU-EUF-CMA$^{\text{corr}}$-secure signatures

Scheme	$\lvert\sigma\rvert$	$\lvert pk\rvert$	Loss	Assumption	Setting	sEUF
BHJKL 1	$\mathcal{O}(\lambda)\lvert\mathbb{G}\rvert$	$\mathcal{O}(1)\lvert\mathbb{G}\rvert$	$\mathcal{O}(1)$	DLIN	Pairings	–
BHJKL 2[1]	$3\lvert\mathbb{G}\rvert$	$\mathcal{O}(\lambda)\lvert\mathbb{G}\rvert$	$\mathcal{O}(\lambda)$	SXDH	Pairings	–
GJ	$2\lvert\mathbb{G}\rvert + 2\lambda + 4\lvert q\rvert$	$2\lvert\mathbb{G}\rvert$	$\mathcal{O}(1)$	DDH	PRO	–
Ours	$3\lvert q\rvert$	$4\lvert\mathbb{G}\rvert$	$\mathcal{O}(1)$	Lossy ID	NPRO	✓

λ: Security parameter

$\lvert\mathbb{G}\rvert$: Size of the an element of group $\mathbb{G}$

$\lvert q\rvert$: Size of the binary representation of q, order of $\mathbb{G}$

[1] Flaw in the proof. Personal communication with one of the authors.

Impact on Tightly-Secure AKE Protocols

Impact on Tightly-Secure AKE Protocols

- Tight MU-EUF-CMA$^{\text{corr}}$-secure signature are the main building block tightly-secure AKE
- Tight security particularly interesting for AKE, due to the large scale use (e.g., TLS)

Protocol	**With GJ Sigs.** **Bytes**	**With our scheme**[2] **Bytes**
GJ (C'18)	544	288
TLS 1.3 (JoC'2?, ACNS'21)	640	384
SIGMA-I (ACNS'21)	640	384
LLGW (AC'20)	544	288
JKRS (EC'21)	416	288

[2] For more details, consider Table 2 in our paper.

Summary

- We construct the first strong and (currently) most efficient MU-EUF-CMA$^{\mathsf{corr}}$-secure signature scheme
- Our construction is perfectly suitable for instantiating tightly-secure AKE:
  - Strong unforgeability $\Rightarrow$ strong authentication (matching conversations)
  - Short signatures $\Rightarrow$ efficient key exchange

`https://eprint.iacr.org/2021/235`