

TECLA and THE CLASH: Two-Party and Threshold ECDSA signatures from Class Groups Cryptography

BiCyCCLiST

Cyril Bouvier¹, Guilhem Castagnos², Dario Catalano³, Quentin Combal¹, Fabien Laguillaumie¹, **Federico Savasta**¹, Ida Tucker⁴

January 27, 2026*

¹Université de Montpellier, CNRS, LIRMM @ Montpellier, France

²Université de Bordeaux, INRIA, CNRS, IMB UMR 5251, F-33405 @ Talence, France

³Università di Catania @ Catania, Italy



⁴PQShield, United Kingdom



Università di Catania

*Presented at MPTS 2026: NIST Workshop on Multi-Party Threshold Schemes 2026

Table of contents

1. Threshold ECDSA
2. The CL Encryption Scheme
3. TECLA: Two-Party ECDSA from CLAss Group Cryptography
4. THE CLASH: THreshold ECDSA from CLASs Group Linearly Homomorphic Encryption

Threshold ECDSA

Threshold Signatures

(f, n) – threshold signature scheme: n parties, a threshold $f < n$



- 👍 The signing key is divided between more servers
- 👍 The signing key is hidden to each party
- 👍 No single point of failure

Highlights on Threshold ECDSA schemes

Several variants of Threshold ECDSA:

- **Linearly Homomorphic PKE:**
 - Paillier-based:
 - Two-party: [Lin17]
 - Threshold: [GGN16],[GG18],[CGG⁺20]
 - **CL-based:**
 - Two-party: [CCL⁺19]*
 - Threshold: [CCL⁺20]*,[CCL⁺23], [JTX25]
- **Oblivious Transfer:**
 - Two-party: [DKLs18], [Kon25]
 - Threshold: [DKLs19], [DKLs24]
- ... and many more
 - from PCGs: [ANO⁺22]
 - from Generic MPC: [DOK⁺20]
 - from exponent-VRF: [DNP25]
 - from class group Non-Interactive Multiplication: [LLZD25]

* Castagnos, Catalano, Laguillaumie, **Savasta**, Tucker @ CRYPTO '19, PKC '20

Towards LHE from class groups: Paillier based ECDSA

Paillier PKE*:

- Linearly Homomorphic
- 👍 Quite fast computation
- 👍 Well established

Paillier and ECDSA ([Lin17],[GGN16],[GG18],[CGG⁺20]):

- 👎 Inconsistency between ECDSA modulo q and Paillier modulo N
 - Expensive techniques needed (e.g. **range proofs**) for correctness
 - Leakage management necessary \Rightarrow more convoluted security proofs

*Paillier @ EUROCRYPT '99

The CL Encryption Scheme

Group with an easy discrete logarithm (DL) subgroup*

- $G = \langle g \rangle$ cyclic group of order $q \cdot s$ such that $\gcd(q, s) = 1$, q large prime, s **unknown** integer
- $F = \langle f \rangle \leq G$, $|F| = q$
- $G^q = \langle g_q \rangle = \{x^q, x \in G\} \leq G$, $|G^q| = s$, s.t. $G = F \times G^q$
- DL is easy in F (DL: given f and $h = f^x$, find $x \in \mathbb{Z}/q\mathbb{Z}$)

Hard Subgroup Membership problem HSM**:

- Hard to distinguish q -th powers in G : $\{x \stackrel{\$}{\leftarrow} G\} \approx_c \{x \stackrel{\$}{\leftarrow} G^q\}$

Concrete Instantiations: We have them!

- **Class Groups** (of imaginary quadratic fields)

* Castagnos and Laguillaumie @ CT-RSA 2015

** Castagnos, Laguillaumie, Tucker @ ASIACRYPT 2018

The CL encryption scheme

Setup($1^\lambda, q$)

- Run $\text{Gen}(1^\lambda, q)$ //Deterministic pp generation
- Return the output $\text{pp} = (\tilde{s}, g, f, g_q, \hat{G}, G, G_q, F)$
//Order upperbound, group generators and group representations

KeyGen(pp)

- Pick $\text{sk} \leftarrow \mathcal{D}_q$ and
 $\text{pk} := g_q^{\text{sk}}$
- Return (pk, sk)

Enc(pk, m)

- Pick $r \xleftarrow{\$} [S]$
- Return $c = (g_q^r, \text{pk}^r f^m)$

Dec(sk, c = (c₁, c₂))

- Compute $M = c_2 / c_1^{\text{sk}}$
- Return $\text{DL.Solve}(M)$

EvalScal(pk, c = (c₁, c₂), α)

- Compute $c'_1 = c_1^\alpha$ and
 $c'_2 = c_2^\alpha$
- Sample $r \xleftarrow{\$} [S]$
- Return $(c'_1 \cdot g_q^r, c'_2 \cdot \text{pk}^r)$

EvalAdd(pk, c = (c₁, c₂), c' = (c'₁, c'₂))

- Compute $c''_1 = c_1 \cdot c'_1$ and
 $c''_2 = c_2 \cdot c'_2$
- Sample $r \xleftarrow{\$} [S]$
- Return $(c''_1 \cdot g_q^r, c''_2 \cdot \text{pk}^r)$

CL strong points

- CL is **ind-cpa** secure under the hardness of the **HSM** assumption
- Solving HSM reduces to **class number computation**
- Best known algorithms (index calculus method):

$$L_{|\Delta_K|}[1/2, c + o(1)] \quad \text{complexity}^*$$

👍 CL ciphertexts are **shorter** than Paillier ciphertexts

👍 Encryption and decryption are **faster** with CL

Security level	Encryption		Decryption		Ct size	
	Paillier	CL	Paillier	CL	Paillier	CL
112-bit sec.	6.2 ms	6.1 ms	6.2 ms	7.1 ms	4.1 kbit	3.7 kbit
128-bit sec.	14.4 ms	10.8 ms	14.4 ms	13.2 ms	6.1 kbit	4.7 kbit
192-bit sec.	146.5 ms	41.9 ms	148.7 ms	57.4 ms	15.4 kbit	8.7 kbit
256-bit sec.	864.4 ms	120.0 ms	876.1 ms	149.5 ms	30.7 kbit	14.0 kbit

Table 1: Timings and sizes comparison for Paillier and CL

*Best algorithm for **Factoring** is $L_N[1/3, c' + o(1)]$

TECLA: Two-Party ECDSA from CLASS Group Cryptography

Before TECLA - Lindell @ CRYPTO '17

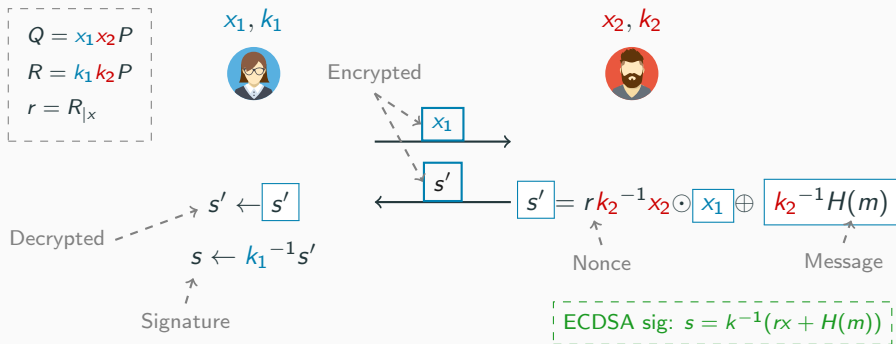


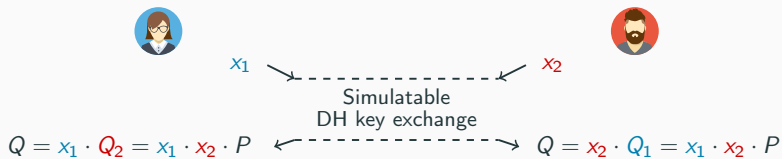
Figure 1: [Lin17] signing idea

Lindell @ CRYPTO '17:

- Encryption scheme: Paillier
- Signature from Lin. Hom. Ops

- Main idea: safe to reveal x_1

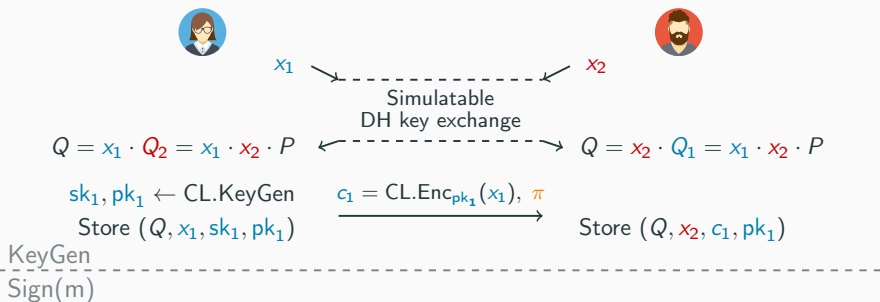
CCLST19 protocol



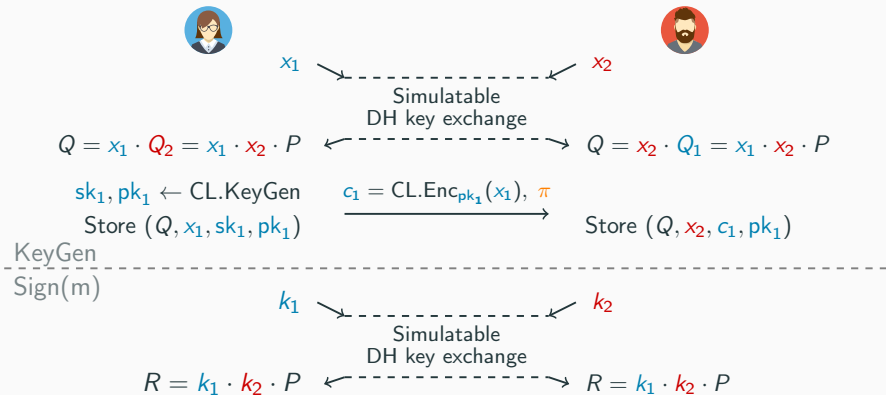
KeyGen

Sign(m)

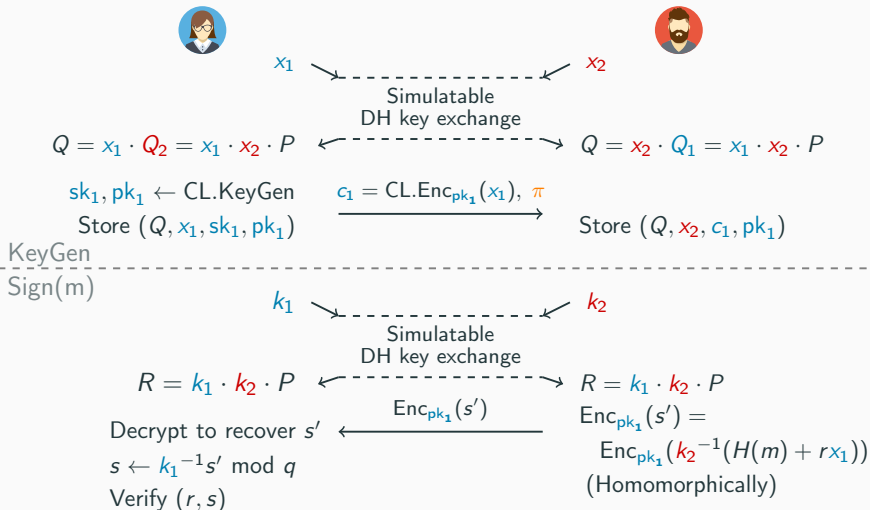
CCLST19 protocol



CCLST19 protocol



CCLST19 protocol



Zero Knowledge for CL ciphertext

ZKs for CL statements

Problem CL encryption is **not surjective** (differently from Paillier)

Goal Proof that $c_x \leftarrow \text{Enc}_{pk}(x)$ is a valid encryption and variants

Solution **Ad hoc** and **expensive** proofs \rightarrow fast proofs/arguments with additional assumptions on the class group

TECLA

Hint Only proof of plaintext knowledge is enough \Rightarrow no full witness knowledge to be proven

Output Fast proofs, no repetitions \Rightarrow **One shot!**

Tool ZKP with **Partial Extractability***

*Beaugrand, Castagnos and Laguillaumie @ Journal of Cryptology 2025
Braun, Damgård and Orlandi @ CRYPTO 2023

Summary of improvements in TECLA

Castagnos et al. @ Crypto '19

- It follows Lindell's idea, but in the setting of **class groups**
- 👍 Only one ZKP for CL for key generation from one party
- 👎 ... but with binary challenge \Rightarrow **several repetitions**
- 👍 **No range proofs**: CL msg space = ECDSA msg space
- 👍 **Security proof**: simulation based security

TECLA

- 👍 Only one ZKP for CL in the key generation from one party ...
but only one repetition!
- 👍 Implemented using the BICYCL library \Rightarrow improved timings!

Security statement - TECLA

Assuming that:

- DL assumption holds in the elliptic curve group \mathbb{G}^*
- RO_C assumption holds for $CL.Setup(1^\lambda)$ **
- HSM assumption holds in G
- CL is δ_S -smooth •
- DE assumption holds in \mathbb{G} and G ••
- the commitment scheme is non-malleable and equivocal

TECLA two-party ECDSA protocol has simulation based security in the presence of a malicious static adversary (under the ideal/real definition)

* DL: Discrete Logarithm

** RO_C : C-Rough Order - Braun et al. @CRYPTO '23

• δ_S -smoothness - Cramer, Shoup @ EUROCRYPT '02

•• DE: Double Encoding - Castagnos et al. @ CRYPTO '19

A library for implementing class group cryptography

BICYCL



- Free and Open Source (GPLv3)
- Based on GMP and OpenSSL
- Arithmetic of class groups of quadratic imaginary number fields
- Cryptographic primitives for class groups cryptography
- Implementation of CL, TECLA and THE CLASH
- Benchmarks

BICYCL library: <https://gite.lirmm.fr/crypto/bicycl>^{*} 

BICYCL original paper: [BCIL23]^{**}

^{*} Bouvier, Castagnos, Combal, Imbert, Laguillaumie

^{**} Bouvier, Castagnos, Imbert, Laguillaumie @ Journal of Cryptology 2023

Benchmarks of TECLA

Curve	Setup	KeyGen		Signing	
	ST	ST	MT	ST	MT
112-bit sec.	132 ms	50 ms	30 ms	20 ms	13 ms
128-bit sec.	362 ms	83 ms	48 ms	35 ms	23 ms
192-bit sec.	3 250 ms	300 ms	160 ms	130 ms	88 ms
256-bit sec.	14 400 ms	780 ms	402 ms	340 ms	230 ms

Table 2: Two-Party ECDSA benchmark results from BICYCL - Timings. ST = single thread, MT = multi thread (up to 4 in the current implementation)

	κ	σ	KeyGen	Signing
P-224	112	40	1.06 kB	0.66 kB
P-256	128	40	1.31 kB	0.81 kB
P-384	192	64	2.26 kB	1.37 kB
P-521	256	64	3.43 kB	2.07 kB

Table 3: Two-Party ECDSA benchmark results from BICYCL - Bandwidth consumption. ST = single thread, MT = multi thread

THE CLASH: **T**Hreshold **E**CDSA
from **C**LASs Group Linearly
Homomorphic Encryption

Before THE CLASH

Starting point Castagnos et al. @ PKC'20 Threshold ECDSA signature scheme:

- KeyGen and Signing in the style of Gennaro et al. 2018
 - Based on Multiplicative to Additive share conversion (MtA) (using Paillier)
 - $a \cdot b = c \Rightarrow \alpha + \beta = c$
- Class Groups variant \rightarrow CL-based MtA
- 👎 Heavy interactive setup for Class Group parameters generation

THE CLASH

- 👍 Simplification of Setup of the scheme from different ZK techniques
- 👍 Optimized implementation with the BICYCL library

THE CLASH: Interactive Setup and Key Generation

Parameters: n parties, any threshold $f < n$

Interactive Setup:

- Commit-and-reveal + reconstruction phases for randomness generation for the class group
- Run the CL.Setup deterministic algo \rightarrow pp_{CL} parameters

Interactive Key Generation:

- (f, n) -Feldman VSS \rightarrow private shares u_i of the secret key x
- Generation of the public verification key $Q = x \cdot P$

THE CLASH: Interactive Signing

Interactive Signing: ($m > f$ players)

- Shares u_i s into additive shares x_1, \dots, x_m s.t. $Q = \sum_i^m x_i \cdot P$ (via [Lagrange int.](#)) \rightarrow subset of signers
- Compute jointly the nonce $R = k^{-1} \cdot P$ and the signature s or abort
 - Encryption c_{k_i} of shares k_i of k and [ZKPoPK*](#) for c_{k_i}
 - CL-based MtA share conversion + Masking
 - Other ZKPoKs: Schnorr proofs and variants \rightarrow [light](#) proofs
 - Signature construction

*ZKP of Plaintext Knowledge - Braun, Damgård and Orlandi @ CRYPTO '23

Security statement - THE CLASH

Assuming:

- standard ECDSA is an **existentially unforgeable** signature scheme
- **DDH** assumption holds in the elliptic curve group \mathbb{G}^*
- **RO_C** assumption holds for $\text{CL.Setup}(1^\lambda)$
- **HSM** assumption holds in G
- CL is **δ_ζ -smooth**
- the commitment scheme is **non-malleable and equivocal**

THE CLASH (f, n) -threshold ECDSA protocol is an **existentially unforgeable** threshold signature scheme against **malicious static** corruptions of **any threshold $f < n$** of parties.

* DDH: Decisional Diffie-Hellman

Benchmarks of THE CLASH

Threshold Profile*	Parties	Communication (Bytes)		Timings (s)	
		KeyGen	Sign	KeyGen	Sign
nSfD	3	1 412	7 202	0.015	0.219
nSfD	5	2 716	13 216	0.016	0.410
nSfD	7	4 276	19 230	0.018	0.593
nMfD	10	7 096	28 251	0.023	0.881
nLfD	100	359 536	298 881	1.24	9.12

Table 4: Communication cost and timings for party in THE CLASH for a threshold $f = n - 1$ and $\kappa = 128$ bits of security

* nSfD: $4 \leq n \leq 8$,

nMfD: $9 \leq n \leq 64$,

nLfD = $65 \leq n \leq 1024$.

(fD: dishonest majority)

We plan to submit a package with TECLA and THE CLASH and their implementation.

Current Status and Todos

- Implementation of TECLA ✓
- Implementation of THE CLASH ✓
- Security proof of TECLA ✗
- Security proof of THE CLASH ✓
- More optimization for Multi-Threading (TECLA, THE CLASH)
✗

✓ ready

✗ not ready yet, WIP

Thank you
for your attention



D. Abram, A. Nof, C. Orlandi, P. Scholl, and O. Shlomovits.

Low-bandwidth threshold ECDSA via pseudorandom correlation generators.

In *2022 IEEE Symposium on Security and Privacy*, pages 2554–2572. IEEE Computer Society Press, May 2022.



C. Bouvier, G. Castagnos, L. Imbert, and F. Laguillaumie.

I want to ride my BICYCL : BICYCL implements CryptographY in CLass groups.


Journal of Cryptology, 36(3):17, July 2023.



G. Castagnos, D. Catalano, F. Laguillaumie, F. Savasta, and I. Tucker.


Two-party ECDSA from hash proof systems and efficient instantiations.

In *CRYPTO 2019, Part III, LNCS 11694*, pages 191–221. Springer, Cham, August 2019.

 G. Castagnos, D. Catalano, F. Laguillaumie, F. Savasta, and I. Tucker.


Bandwidth-efficient threshold EC-DSA.

In *PKC 2020, Part II, LNCS 12111*, pages 266–296. Springer, Cham, May 2020.

 G. Castagnos, D. Catalano, F. Laguillaumie, F. Savasta, and I. Tucker.

**Bandwidth-efficient threshold ec-dsa revisited:
Online/offline extensions, identifiable aborts proactive and adaptive security.**

Theoretical Computer Science, 939:78–104, 2023.

 R. Canetti, R. Gennaro, S. Goldfeder, N. Makriyannis, and U. Peled.

UC non-interactive, proactive, threshold ECDSA with identifiable aborts.

In *ACM CCS 2020*, pages 1769–1787. ACM Press, November 2020.



R. Cramer and V. Shoup.

Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption.

In *EUROCRYPT 2002, LNCS 2332*, pages 45–64. Springer, Berlin, Heidelberg, April / May 2002.



J. Doerner, Y. Kondi, E. Lee, and a. shelat.

Secure two-party threshold ECDSA from ECDSA assumptions.

In *2018 IEEE Symposium on Security and Privacy*, pages 980–997. IEEE Computer Society Press, May 2018.



J. Doerner, Y. Kondi, E. Lee, and a. shelat.

Threshold ECDSA from ECDSA assumptions: The multiparty case.

In *2019 IEEE Symposium on Security and Privacy*, pages 1051–1066. IEEE Computer Society Press, May 2019.



J. Doerner, Y. Kondi, E. Lee, and a. shelat.

Threshold ECDSA in three rounds.

In *2024 IEEE Symposium on Security and Privacy*, pages 3053–3071. IEEE Computer Society Press, May 2024.



H. Dahari-Garbian, A. Nof, and L. Parker.

Trout: Two-round threshold ECDSA from class groups.

In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security, CCS 2025, Taipei, Taiwan, October 13-17, 2025*, pages 380–393. ACM, 2025.



A. P. K. Dalskov, C. Orlandi, M. Keller, K. Shrishak, and H. Shulman.

Securing DNSSEC keys via threshold ECDSA from generic MPC.

In *ESORICS 2020, Part II, LNCS 12309*, pages 654–673.
Springer, Cham, September 2020.



R. Gennaro and S. Goldfeder.

Fast multiparty threshold ECDSA with fast trustless setup.

In *ACM CCS 2018*, pages 1179–1194. ACM Press, October 2018.



R. Gennaro, S. Goldfeder, and A. Narayanan.

Threshold-optimal DSA/ECDSA signatures and an application to bitcoin wallet security.

In *ACNS 2016, LNCS 9696*, pages 156–174. Springer, Cham, June 2016.



B. Jiang, G. Tang, and H. Xue.

Three-round (robust) threshold ECDSA from threshold CL encryption.

In *ACISP 25, Part I, LNCS 15658*, pages 224–244. Springer, Singapore, July 2025.



Y. Kondi.

Two-party ECDSA signing at constant communication overhead.

Cryptology ePrint Archive, Paper 2025/1813, 2025.



Y. Lindell.

Fast secure two-party ECDSA signing.

In *CRYPTO 2017, Part II, LNCS 10402*, pages 613–644. Springer, Cham, August 2017.



Y. Lyu, Z. Li, H.-S. Zhou, and X. Deng.

Threshold ecdsa in two rounds.

In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security, CCS '25*, page

2937–2950, New York, NY, USA, 2025. Association for Computing Machinery.



C. Studio.

Flat profile avatar collection.

Avatar icons by Ceria Studio in CC Attribution License via SVG Repo.