"Preview Writeup": In anticipation of a package submission to the NIST Threshold Call

Title: TECLA: Two-party ECDSA from CLAss groups

Subtitle: Two-Party ECDSA from Linearly Homomorphic Encryption over Class Groups

Version: 0.1 (2026-01-09)[1]

Team name: BICYCCLIST

Team members: Cyril Bouvier, Guilhem Castagnos, Dario Catalano, Quentin Combal, Fabien Laguillaumie, Federico Savasta, Ida Tucker

Abstract: This Preview Writeup defines the starting plan for submitting TECLA to the NIST First Call for Multi-Party Threshold Schemes. TECLA is an efficient two-party ECDSA scheme for distributing ECDSA signatures between two parties, where one of them can be compromised and act maliciously. TECLA is built upon Class Group Cryptography and it follows the paradigm of using linearly homomorphic public key encryption (PKE) to distribute shares of the signature and then obtaining the final signature by decryption. The specific PKE building block is the Castagnos-Laguillaumie (CL) encryption scheme, whose main advantage is the reduction of the communication costs compared to other linearly homomorphic PKE. TECLA is accompanied by a benchmarked implementation using the BICYCL public repository, an optimized and specialized library for class group cryptography. The plan is to present the theoretical scheme with its proof of security, the building blocks and the composition of the source code, then to provide the complete package for public use in real-world cases. The package satisfies the requirements of the NIST call, in particular interchangeability and provable security. The scheme has proven simulation-based security against probabilistic polynomial time (PPT) malicious adversaries with static corruptions.

Categories of proposed crypto-systems: Threshold ECDSA Signature (N1.2);

Keywords: Threshold Cryptography; NIST Threshold Call; Threshold ECDSA; Two-Party ECDSA; Class Groups

[1] Preliminary version submitted to NIST-MPTC for review

Preview writeup. This document is provided to NIST for online publication, to foster public awareness and support public discussion within the scope of the NIST First Call for Multi-Party Threshold Schemes [NIST-IR8214C]. This "preview writeup" represents a good-faith plan for a subsequent "package submission". However, until the deadline for package submission, the team may still modify its own composition and the submission plan, including possible changes to the technical scope, and/or the used techniques or achieved results.

Team members: Cyril Bouvier [i1,a1], Guilhem Castagnos [i2,a2], Dario Catalano [i3,a3], Quentin Combal [i4,a1], Fabien Laguillaumie [i5,a1], Federico Savasta [i6,a1], Ida Tucker [i7,a4]

Open Researcher and Contributor Identifiers (ORCID):

i1 (0009-0006-6722-4887); i2 (0009-0004-5857-7889); i3 (0000-0001-9677-944X); i4 (0009-0003-8653-9444); i5 (0000-0001-6464-1139); i6 (0009-0005-4551-1562); i7 (0000-0003-4895-5896)

Affiliations:

[a1] Université de Montpellier, CNRS, LIRMM @ Montpellier, France

[a2] Université de Bordeaux, CNRS, INRIA, IMB @ Talence, France

[a3] Dipartimento di Matematica ed Informatica, University of Catania @ Catania, Italy

[a4] PQShield, United Kingdom

Associateship clarifications:

* Ph.D. student (non-employee). † Associate (visiting researcher). ‡ Work performed while on sabbatical leave.

Main contacts:

- **Team mailing list:** bicycclist@lirmm.fr
- **Primary technical contact person:** Fabien Laguillaumie, fabien.laguillaumie@lirmm.fr
- **Secondary contact person 1:** Guilhem Castagnos, guilhem.castagnos@math.u-bordeaux.fr
- **Secondary contact person 2:** Federico Savasta, federico.savasta@lirmm.fr

Produced by humans. The team hereby confirms that the content in this preview writeup: (i) was produced by the team members, and (ii) was not produced by generative artificial intelligence (AI), with the possible exception of AI-proposed grammar improvements, minor integrated suggestions, or some well-identified and short localized portions of auxiliary content (e.g., some illustration); and (iii) was proofread by the team members.

1. Introduction

Threshold signature schemes allow several parties to split the computation of a signature, in such a way that no single party can compute a new distributed signature by itself. In general, this can be achieved by splitting the signature key using a secret sharing protocol. The secret key is then implicitly defined but unknown to the parties joining the protocol. In threshold signature schemes, n parties jointly generate the keys, but it is sufficient that at least $t+1$ parties, for some threshold t, participate in the signing phase of the protocol to create a new signature, while $\leq t$ of them can not achieve this goal. Threshold signatures solve the "single point of failure" problem, typical in centralized signature schemes. Furthermore, a threshold signature should satisfy interchangeability with regards to signature verification. Use cases of threshold signatures include signatures on documents by a quorum, or the validation of transactions in a blockchain such as Bitcoin.

This preview document presents the main components of TECLA, a two-party ECDSA signature scheme based on the properties of the HPS framework (Hash Proof Systems) ([CS02]) which satisfies interchangeability with regard to plain ECDSA verification. HPS allow to achieve the properties we want to realize and TECLA can be implemented with any construction which satisfies the HPS framework. This proposal is accompanied by a concrete instantiation of an "ECDSA-friendly" HPS using the Castagnos-Laguillaumie encryption scheme (CL) and its optimal implementation using the BICYCL library ([BCIL23]). The implementation results in a very efficient scheme both in terms of computation and communication. The purpose of the package submission is to give an implementation in the context of two-party ECDSA signing, which guarantees provable security against active corruptions, and whose performance is suitable for real-world scenarios. The implemented code is accompanied by benchmarks and performance analysis. The package fits in Category Type N1.2: Signing, Subtype: QV, Families of Specification: ECDSA sign.

2. Specification

2.1. Organization

The specification document will include one cryptosystem, i.e., the TECLA two-party ECDSA protocol. With regard to the directive lines of the NIST call documentation, Sections 5, 6, 7, the specification document will be structured as follows.

The "Main Matter - Preliminaries" section will present the cryptographic tools necessary for understanding TECLA: plain ECDSA, class group cryptography, and common cryptographic primitives (such as the Diffie-Hellman Key Exchange, commitments, zero-knowledge proofs). The "Main Matter - Crypto-System" section will go into details about the main building blocks used in TECLA and discuss potential extensions such as offline/online computation, key refresh, and security against other adversary capabilities.

Specifically, subsections will be dedicated to the following building blocks:

- Castagnos-Laguillaumie (CL) linearly homomorphic public key encryption scheme: we will present a short description of class group cryptography and the motivations behind the technical choices of parameters used for the generation of the class groups of interest, the description of CL, its security, and the hard assumptions for class groups which imply that CL is secure. CL is an encryption scheme of independent interest ([CL15],[CLT18]).
- Honest Verifier Zero Knowlegde Proof with partial extractability (HVZK-PwPE): we will describe the specific primitives and hard assumptions we use to achieve active security in our proposal. A HVZK-PwPE allows to prove knowledge of a message encrypted in a CL ciphertext and the well-formedness of the latter. It does not prove the knowledge of the randomness used, but suffices for our construction. These involved proofs are introduced in [BCL25] and they extend the Zero-Knowledge Proof of Plaintext Knowledge (ZKPoPK) introduced and analyzed in [BDO23], [BCDLMOT24]. HVZKPwPE are of independent interest. These CL-based proofs are Σ-protocols made non interactive via the well-known Fiat-Shamir transform.

2.2. System Model

Protocol overview: TECLA is composed of two main distributed subprotocols: a distributed key generation (DKG) and distributed signing (DSig). In TECLA, there are $n = 2$ parties P_1 and P_2 that participate in both the DKG and DSig phases. The roles of the parties are not symmetric; party P_1 generates the keys of a PKE scheme and encrypts its share of the ECDSA signing key. Then, it sends the encryption to P_2, who performs homomorphic operations on it to compute its part of the signature. P_2 then sends its encrypted partial signature to P_1, and P_1 can decrypt and finish computing the ECDSA signature. Following Table 16 in the NIST documentation [NIST-IR8214C-2pd], the threshold profile in TECLA is $n2$, i.e. $n = 2$ parties and $f = 1$ (only dishonest majority is considered). Parameters to instantiate the PKE require the generation of a class group. TECLA has no trusted setup; the computation of class group parameters can be done via an interactive setup between the two parties. It consists of a unique commit and reveal phase to share the randomness to be used for the generation of parameters. After that, the two parties can locally compute the class group parameters using a deterministic procedure. Regarding ECDSA parameters, we use standard NIST P-curves specific to each NIST security level [FIPS:186-5].

Networking aspects: Regarding networking and communication model, we consider a broadcast and authenticated channel, and the construction can be built upon an unreliable channel, making it suitable for bad network conditions. The communication between parties is asynchronous.

Applicable I/O interfaces: Activation of the signature phase can be done by a simple request from any of the parties. Following the [NIST-IR8214A] documentation, TECLA uses implicit I/O secret-sharing modes (SSO KeyGen, Subsequent (SSI) operation). More in detail, in KeyGen, the two parties run a Diffie-Hellman Key Exchange (DHKE) to obtain the secret shares of the implicit distributed ECDSA signing key. In the signing phase, the parties apply another DHKE to obtain the ECDSA nonce elliptic curve point used for creating a signature. However, even if the nonce

material is secret shared, this is independent of the ECDSA secret key, and the applicable I/O interfaces are (SSO) KeyGen and Subsequent (SSI) Operations.

2.3. Security

Security: Regarding provable security, TECLA is proven secure under the Real/Ideal simulation paradigm against malicious adversaries with static corruptions. It presents a simulation based proof of security for sequential repetitions (no concurrency): the protocol realizes an ECDSA functionality which represents the authentication goals of a two-party ECDSA digital signature and is secure by definition, as for the definition of the Real/Ideal paradigm. On input "key-generation" (KG) from both parties, called once, the functionality computes the ECDSA keys; and on input "sign" and a message m (Sign) from both parties, which can be called many times, it returns a valid ECDSA signature (r, s) on m. We assume security with abort. The security is proven using standard techniques, i.e., by a series of hybrids from the real-world protocol between two parties to the ideal world with the ECDSA functionality and a simulator. Hops between games are proven using statistical arguments and the computational hardness assumption explained in the following dedicated paragraph. The scheme is not subject to trivial attacks against adaptive corruptions. Regarding the satisfied security levels, parameters for the CL encryption scheme and of every building block/component of the protocol are consistent with the standard NIST security levels of 112, 128, 192 and 256 bits.

Adversarial Model: We assume Probabilistic Polynomial Time (PPT) adversaries. The adversary's goal is to distinguish between a real-world (the two-party ECDSA protocol) and an ideal world (with a simulator and a two-party ECDSA functionality introduced in the Security paragraph above). In this cryptosystem proposal, the adversary is malicious and can perform static corruptions, i.e., it decides which party to corrupt at the onset of the protocol, and then tries to reach its goal. In our two-party ECDSA scheme, we consider security with abort, meaning that a corrupted party can learn the output while the honest one does not, without harming the security of the overall protocol.

Hard assumptions: For security to hold, we require that plain centralized ECDSA be existentially unforgeable, along with some additional assumptions. Some of these assumptions are necessary for the CL PKE, which is the main component for distributing parts of the signature in a secure way. The security of our two-party protocol is proven under the Discrete Logarithm Problem over Elliptic Curves (DLP), and the Hard Subgroup Membership problem in the class group (HSM) [CLT18].

3. Open-Source Implementation

Code structure: Our core code will mainly consist of the BICYCL library ([BCIL23], [BIC]), our open-source C++ library for class groups based cryptography, developed and maintained

by team members of the LIRMM. BICYCL implements arithmetic in the ideal class groups of imaginary quadratic fields, along with a set of cryptographic primitives based on class groups. It also implements the operations of our multi-party protocols, including the two-party ECDSA scheme we are submitting, and benchmarks for these operations. BICYCL relies on the GNU Multiple Precision Arithmetic Library (GMP) for arithmetic operations over big integers, as well as the OpenSSL library for arithmetic on elliptic curves. Both libraries are written in C. For simplicity, we plan to include GMP and OpenSSL as external dependencies, installed through the distribution's package manager. Compilation will be performed using GCC and GNU make, included as external dependencies as well. As our core code does not rely on architecture-specific features, we only select the compiler options for general optimization (e.g. `-O3`). Still, we will consider architecture-specific options if we notice significant performance improvements.

Code progress and availability: A first version of our implementation was developed in the BICYCL library in Q4 2024. Since then, fixes were added to improve performance and simplify the interface. Additional work may be needed to make the code clearer and ease the auditing process. The git repository for BICYCL is publicly available and hosted on the LIRMM's GitLab instance [BIC]. To comply with the submission requirements, we will develop additional C++ code to take care of the networking between parties. This code will rely on the BICYCL library to perform protocol operations for each party.

Implementation of the networking model: As TECLA involves only two parties, all communications can be considered broadcasted. In a real-life scenario, parties would need to be able to authenticate each other. However, authentication is outside the scope of our protocol, and our implementation will assume it has been established beforehand. Each party will be run as a separate process, and parties will communicate through the loopback interface available on GNU/Linux systems. The messages will be transmitted as UDP datagrams over IP.

Testing: Testing malicious behaviour is straightforward: only one party at a time can be malicious (both parties being malicious means total corruption). The implementation will check that in each case, the protocol is aborted by the honest party. Sub-optimal networking conditions can be modeled with erroneous or missing transmissions, either chosen arbitrarily or picked at random, that require re-transmission to complete the round.

4. Experimental performance evaluation

Performance: The values from Table 1 are the result of running the TwoPartyECDSA benchmark from BICYCL (commit `e4e5c97`). The values correspond to the mean of several experiments, 10 to 100 depending on the operation, and include the operations of both parties. The machine used for the measurements has the following specifications:

- x64 CPU with 16 cores (from 1.4 to 5 GHz) and 22 threads
- 16 GB of RAM

- SSD with 500 Go of memory
- Ubuntu 24.04 LTS

The resulting timings are quite stable, except for the "Setup" phase (standard deviation of $\sim$1200 ms for security parameter $\kappa = 192$ and $\sim$5000 ms for $\kappa = 256$) due to the generation of the class group discriminant. The operations are performed on the same process, and network latency is not taken into account. Therefore, benchmarking on multiple processes with network communication is expected to give longer timings.

Platform: The baseline platform is an environment similar to the machine we used for Table 1, with more RAM and storage space, so it is expected to give similar timings when running the same benchmarks. Our implementation currently makes use of up to 4 threads per party. In our reference implementation, we may use an additional thread per party to handle the networking aspects. This makes up to 10 threads; therefore, the 16 cores of the baseline platform will be sufficient. For these reasons, we do not expect challenges with using the baseline platform.

Curve	κ	σ	Setup (ms)	Keygen (ms)		Signing (ms)		Keygen (kB)	Signing (kB)
			ST	ST	MT	ST	MT		
P-224	112	40	132	50	30	20	13	1.06	0.66
P-256	128	40	362	83	48	35	23	1.31	0.81
P-384	192	64	3 250	300	160	130	88	2.26	1.37
P-512	256	64	14 400	780	402	340	230	3.43	2.07

Table 1: Two-Party ECDSA benchmark results from BICYCL, where the "ST" columns refer to operations performed on a single thread, and the "MT" columns to operations performed on multiple threads (up to 4 in the current implementation). The "Setup" phase is performed on a single thread. κ refers to the computational security parameter and σ to the statistical security parameter.

5. Licensing, patent claims, and funding

Licenses: Our main library, BICYCL, relies on the GMP and OpenSSL libraries. Both libraries are free software and licensed under OSI approved licenses (LGPL version 3.0 for GMP, Apache license version 2.0 for OpenSSL). BICYCL itself is licensed under GPL version 3.0 or later, a license compatible with both LGPL v3.0 and Apache v2.0. Regarding the additional code needed for the submission, we will release it under GPLv3 as well. Building uses tools from the GNU toolchain (gcc, make, ...) that are licensed under GPL. All the aforementioned licenses are approved by the Open-Source Initiative.

Patent claims: We declare that, to the best of our knowledge, no patents cover the contents of our submission. The copyright holders of our core code are the team members who contributed to the development.

Funding: The research associated with our submission is supported by the French Agence Nationale de la Recherche (ANR) project ANR-21-CE39-0006 SANGRIA, the France 2030 ANR project ANR-22-PECY-003 SecureCompute, the ANR ASTRID program under the national project AMIRAL with reference ANR-21-ASTR-0016, and by ICO, Institut Cybersecurité Occitaine, funded by Région Occitanie, France.

References

[BCDLMOT24] Lennart Braun, Guilhem Castagnos, Ivan Damgård, Fabien Laguillaumie, Kelsey Melissaris, Claudio Orlandi, and Ida Tucker. "An Improved Threshold Homomorphic Cryptosystem Based on Class Groups". In: *SCN 24, Part II*. Ed. by Clemente Galdi and Duong Hieu Phan. Vol. 14974. LNCS. Springer, Cham, September 2024, pp. 24–46. DOI: 10.1007/978-3-031-71073-5_2. Also at ia.cr/2024/717.

[BCIL23] Cyril Bouvier, Guilhem Castagnos, Laurent Imbert, and Fabien Laguillaumie. "I Want to Ride My BICYCL : BICYCL Implements CryptographY in CLass Groups". In: *Journal of Cryptology* 36.3 (July 2023), p. 17. DOI: 10.1007/s00145-023-09459-1. Also at ia.cr/2022/1466.

[BCL25] Agathe Beaugrand, Guilhem Castagnos, and Fabien Laguillaumie. "Efficient Succinct Zero-Knowledge Arguments in the CL Framework". In: *J. Cryptol.* 38.1 (January 2025). DOI: 10.1007/s00145-024-09534-1. URL: https://doi.org/10.1007/s00145-024-09534-1.

[BDO23] Lennart Braun, Ivan Damgård, and Claudio Orlandi. "Secure Multiparty Computation from Threshold Encryption Based on Class Groups". In: *CRYPTO 2023, Part I*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14081. LNCS. Springer, Cham, August 2023, pp. 613–645. DOI: 10.1007/978-3-031-38557-5_20. Also at ia.cr/2022/1437.

[BIC] *BICYCL public git repository*. URL: https://gite.lirmm.fr/crypto/bicycl.

[CL15] Guilhem Castagnos and Fabien Laguillaumie. "Linearly Homomorphic Encryption from DDH". In: *CT-RSA 2015*. Ed. by Kaisa Nyberg. Vol. 9048. LNCS. Springer, Cham, April 2015, pp. 487–505. DOI: 10.1007/978-3-319-16715-2_26. Also at ia.cr/2015/047.

[CLT18] Guilhem Castagnos, Fabien Laguillaumie, and Ida Tucker. "Practical Fully Secure Unrestricted Inner Product Functional Encryption Modulo p". In: *ASIACRYPT 2018, Part II*. Ed. by Thomas Peyrin and Steven Galbraith. Vol. 11273. LNCS. Springer, Cham, December 2018, pp. 733–764. DOI: 10.1007/978-3-030-03329-3_25. Also at ia.cr/2018/791.

[CS02] Ronald Cramer and Victor Shoup. "Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption". In: *EUROCRYPT 2002*. Ed. by Lars R. Knudsen. Vol. 2332. LNCS. Springer, Berlin, Heidelberg, 2002, pp. 45–64. DOI: 10.1007/3-540-46035-7_4. Also at ia.cr/2001/085.

[FIPS:186-5] Lily Chen, National Institute of Standards, Technology, Dustin Moody, Andrew Regenscheid, and Angela Robinson. *Digital Signature Standard (DSS)*. Washington, D.C, 2023. DOI: 10.6028/NIST.FIPS.186-5.

[NIST-IR8214A] Luís T. A. N. Brandão, Michael Davidson, and Apostol Vassilev. *NIST Roadmap Toward Criteria for Threshold Schemes for Cryptographic Primitives*. National Institute of Standards and Technology (NIST) Internal Report (NISTIR) 8214A. 2020. DOI: 10.6028/NIST.IR.8214A. URL: https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8214A.pdf.

[NIST-IR8214C] Luís T. A. N. Brandão and René Peralta. *NIST First Call for Multi-Party Threshold Schemes*. (National Institute of Standards and Technology) NIST Internal Report (NISTIR) 8214C. 2025. DOI: 10.6028/NIST.IR.8214C.

[NIST-IR8214C-2pd] Luís T. A. N. Brandão and René Peralta. *NIST First Call for Multi-Party Threshold Schemes*. (National Institute of Standards and Technology) NIST Internal Report (NISTIR) 8214C 2pd (Second Public Draft). March 2025. DOI: 10.6028/NIST.IR.8214C.2pd.