

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

Title: Distributed ECDSA Signatures

Subtitle: Fireblocks’ CGGMP protocol

Version: 1.0 (2026-01-15)¹

Team name: Fireblocks–3MI Labs

Team members: Michael Adjedj, Tomer Ashur, Amit Singh Bhati, Geoffroy Couteau, Cyprien Delpech de Saint Guilhem, Michael Gutkin, Nikos Makriyannis

Abstract: This preview write-up presents the plan to submit Fireblocks’s *CGGMP* protocol for multi-party ECDSA signing. The submission package will also include contents on secret-sharing, Paillier encryption, Damgård–Fujisaki and ElGamal-in-the-exponent commitments, a vector oblivious linear evaluation construction, as well as several related zero-knowledge proofs.

Proposed crypto-systems: (I) Paillier Homomorphic Encryption (Category S2); (II) Damgård–Fujisaki Commitment Scheme (Category S7); (III) ElGamal-in-the-Exponent Commitment Scheme (Category S7); (IV) Paillier-based VOLE (Category S7); (V) CGGMP: Multi-Party ECDSA Signing (Categories N1.2 and N4).

Keywords: Threshold Cryptography; NIST Threshold Call; ECDSA Key Generation; ECDSA Signature; Identifiable Abort

¹Version submitted to NIST-MPTC for publication

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: Michael Adjedj^{i1,a1}, Tomer Ashur^{i2,a2}, Amit Singh Bhati^{i3,a2}, Geoffroy Couteau^{i4,a3*}, Cyprien Delpech de Saint Guilhem^{i5,a2}, Michael Gutkin^{i6,a1}, Nikos Makriyannis^{i7,a1}

Open Researcher and Contributor Identifiers (ORCID):

i1 (0009-0001-2738-9728); i2 (0000-0001-6091-4857); i3 (0000-0003-0843-4885); i4 (0000-0002-6645-0106); i5 (0000-0002-0147-2566); i6 (0009-0002-5533-7420); i7 (0000-0002-9818-456X)

Affiliations:

^{a1} Fireblocks, New York City, NY, United States of America

^{a2} 3MI Labs, Leuven, Belgium

^{a3} Université Paris Cité, CNRS, IRIS, Paris, France

Associateship clarifications:

* Consultant at Fireblocks Ltd.

Main contacts:

- **Team mailing list:** NIST-MPTC@fireblocks.com
- **Primary technical contact person:** Nikos Makriyannis, nikos@fireblocks.com
- **Secondary contact person 1:** Michael Gutkin, gutkin@fireblocks.com
- **Secondary contact person 2:** Cyprien Delpech de Saint Guilhem, cyprien@3milabs.tech

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

Principal Crypto-Systems. The planned package submission will specify a family of threshold signature crypto-systems for the ECDSA signature scheme based on Fireblock's CGGMP protocol [CGGMP20]. This family fits within categories N1.2 (ECDSA signing) and N4 (subcategory QV-ECC-KeyGen).

This family of protocols is already widely deployed in commercial products to secure financial (crypto-)assets given the prevalence of requirements for ECDSA signatures in these industries and the security advantage (compromise resistance, distributed control) of secret-sharing signing keys that control valuable assets.

Secondary Crypto-Systems. This planned package submission will also include specification for additional crypto-systems that are used within the principal ones. We believe that these additional crypto-systems are sufficiently used as building blocks in the literature of threshold crypto-systems to warrant their inclusion within the Threshold Call. Specifically, this planned package submission will include (i) the Paillier crypto-system (category S2) [Pai99], (ii) the Damgård–Fujisaki commitment scheme (category S7) [DF02], (iii) the ElGamal-in-the-exponent commitment scheme (category S7) [EIG84], and (iv) a protocol for oblivious linear evaluation (OLE) instantiated with gadgets (i)–(iii) (category S7).

Additional Gadgets. To provide security against active corruptions, the principal crypto-systems of this planned package submission make frequent use of further gadgets, mainly zero-knowledge proofs (of knowledge) (ZKP(oK)s). This intended submission will therefore also specify (non-)interactive proof protocols for several languages linked to ECC relations, validity of RSA moduli, validity of parameter tuples, messages within ciphertexts or commitments, etc.

2. Specification

Organization. The planned specification document will, after preliminaries, first specify the following secondary crypto-systems.

1. the (single-party) Paillier homomorphic encryption scheme [Pai99] with Paillier–Blum modulus.
2. The (single-party) Damgård–Fujisaki commitment scheme [DF02].
3. The (single-party) ElGamal-in-the-exponent commitment scheme [EIG84].
4. A two-party VOLE protocol built from schemes 1–3. To achieve active security, this crypto-system will include the specification of non-interactive ZKPoKs for the following relations:
 - (a) *Damgård–Fujisaki parameter validity:* For given Damgård–Fujisaki parameters (N, s, t) , the integer s is an element of the sub-group of \mathbb{Z}_N^* generated by t .

(b) *Paillier modulus well-formedness:*

- i. A public integer N is a Blum modulus for two private odd primes p, q congruent to 3 modulo 4.
- ii. A public integer N is Paillier compatible, i.e., $\gcd(N, \varphi(N)) = 1$.
- iii. A public integer N can be factored into two integers p, q that lie within a certain public range of integers.

(c) *(V)OLE operations:*

- i. **Receiver message.** For a given Paillier ciphertext C encrypted under the receiver's public key, the encrypted plaintext $x = \text{dec}(C)$ lies within a certain public range of integers and is consistent with an El-Gamal commitment.
- ii. **Sender message.** For public Paillier ciphertexts C and D encrypted under the receiver's public key, it holds that $\text{dec}(D) = a \cdot \text{dec}(C) + b$ for range-checked private coefficients a, b that are, respectively, consistent with the dlog of a public EC point and a Paillier ciphertext encrypted under the sender's key.

The document will then specify the CGGMP family of crypto-systems consisting of three phases: key generation, key-refresh and parameter generation, and signing. All of these phases will include a base variant, and a variant with identifiable abort.

- **Key Generation.** A commit-open construction is used to counter rushing adversaries when generating a shared ECDSA private key and its corresponding public key. An interactive ZKPoK of discrete logarithm will also be specified, with a distributed coin-tossing construction to generate the challenge, as it enables identification of cheating adversaries.
- **Key Refresh and Parameter Generation.** Each party generates a Paillier public key and Damgård–Fujisaki commitments and provides ZKPoKs of their validity. Each party samples a nilpotent randomizer and secret-shares it among all the parties. Commit-open and coin-tossing constructions are again used to counter rushing adversaries and biased randomness sampling.
- **Signing.** This final phase can be further divided into two variants: a one-shot variant where the message is known and the signature is computed within a single protocol run, and a pre-sign variant where, first, some random information, independent of the message, is jointly generated by the parties and stored for later use, and secondly, once the message is known, a short protocol is executed to generate the final signature.
 - **Pre-Signing.** Parties sample shares of a random signature nonce and a random multiplicative mask. Two-party VOLE protocols are executed between pairs of parties to compute shares of pre-sign components. In the variant with identifiable abort, additional non-interactive ZKPoKs will be specified for the following relations:
 - * *Consistency of discrete logarithm with ElGamal commitment:* the discrete logarithm of a public group element with respect to a given base is the same as the message within an ElGamal commitment over a (possibly different) given base.

- * *Paillier decryption in the exponent*: Given group elements X, S and two group generators g, h , the private discrete logarithm of X with base g is the plaintext of a given Paillier ciphertext and equal to the discrete logarithm of S with base h .
- **Signing**. Each party retrieves pre-sign information, computes their share of the signature with the message digest, and broadcast their share. Each party then individually verifies each other party's signature share and outputs the combined signature.

System model. The CGGMP family of protocol uses a *verifiable, authenticated, and synchronous broadcast channel*. (The original presentation also uses point-to-point channels for certain messages, but their confidentiality is not required, and the security analysis is conducted as if all messages were sent over the broadcast channel.)

The analysis of the CGGMP protocol is conducted in the *global random oracle model*, and no other setup assumption (trusted or ideal) is used.

The CGGMP-base protocol provides security with abort against t active corruptions; the CGGMP-fancy protocol provides security with *identifiable* abort against t active corruptions. In both cases t can range from 1 to $n - 1$.

Security. *Unforgeability* and *accountability* will be proven as standalone properties of the protocols in a game-based security model. Security as indistinguishability from an *ideal threshold signature functionality* will then be shown to follow from the standalone properties.

The security theorem of the CGGMP protocol assumes: (i) semantic security of the Paillier cryptosystem, (ii) hardness of strong-RSA moduli, (iii) hardness of the DDH problem, and (iv) unforgeability of the (non-threshold) ECDSA.

3. Open-Source Implementation

The core code is written in C and C++17 and is currently divided into two main modules: (i) a “cosigner” module which contains the functions required to run the CGGMP protocol as a signing party, and (ii) a cryptography module with functions implementing the secondary crypto-systems, additional gadgets, and required cryptographic and arithmetic operations. The submission package will include scripts to build the code on the target reference platform and run benchmarks of the various components for varying threshold configurations and parameter choices.

The code does not bundle any dependencies. It requires the following *external* dependencies:

1. libssl from OpenSSL.
2. libuuid (for testing).
3. libsecp256k1 (for testing).

There already is a public repository with an implementation of the CGGMP protocol [Fir25] which will be restructured to match the organisation of the submission. This code currently does not implement the network model.

In addition to testing correctness in the all-honest scenario, the current test suite also tests the behaviour of the code in case of malicious parties or unreliable network.

4. Experimental Performance Evaluation

The current code package’s performance is reported in [Table 1](#). The 112-bit and 128-bit security levels differ in several parameters such as the bit-length of RSA moduli, the number of repetitions of certain ZKPoKs, and various slackness parameters.

Configuration	Phase	112-bit security (s)	128-bit security (s)
3-out-of-3	KeyGen	2.714	13.907
	Sign	0.346	1.063
2-out-of-2	KeyGen	1.380	12.135
	Sign	0.181	0.547

Table 1: Computation time per party of the core computation for the CGGMP protocol over the `secp256k1` elliptic curve, not including computation required for the networking model. Measurements obtained on Intel® Core™ i7-1365U CPU.

Platform. The baseline platform is expected to provide sufficient resources for realistic benchmarking and we do not anticipate particular challenges.

5. Licensing, Patent Claims, and Funding

The *core code* is currently open-sourced under the GPL-3.0 licence [[Fre07](#)]. For the *external dependencies*: `libuuid` is available under the LGPL-2.0 licence [[Fre91](#)], `libssl` is available under the Apache-2.0 licence [[Apa07](#)], and `libsecp256k1` is available under the MIT licence [[MIT](#)].

The team is not aware of any patents with claims covering the content of the planned submission.

References

- [Apa07] Apache Software Foundation. *The Apache License, Version 2.0*. <https://www.apache.org/licenses/LICENSE-2.0>. 2007.
- [CGGMP20] Ran Canetti, Rosario Gennaro, Steven Goldfeder, Nikolaos Makriyannis, and Udi Peled. “UC Non-Interactive, Proactive, Threshold ECDSA with Identifiable Aborts”. In: *ACM CCS 2020*. Ed. by Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna. ACM Press, November 2020, pp. 1769–1787. DOI: [10.1145/3372297.3423367](https://doi.org/10.1145/3372297.3423367). Also at ia.cr/2021/060.
- [DF02] Ivan Damgård and Eiichiro Fujisaki. “A Statistically-Hiding Integer Commitment Scheme Based on Groups with Hidden Order”. In: *ASIACRYPT 2002*. Ed. by Yuliang Zheng. Vol. 2501. LNCS. Springer, Berlin, Heidelberg, December 2002, pp. 125–142. DOI: [10.1007/3-540-36178-2_8](https://doi.org/10.1007/3-540-36178-2_8). Also at ia.cr/2001/064.
- [EIG84] Taher ElGamal. “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”. In: *CRYPTO’84*. Ed. by G. R. Blakley and David Chaum. Vol. 196. LNCS. Springer, Berlin, Heidelberg, August 1984, pp. 10–18. DOI: [10.1007/3-540-39568-7_2](https://doi.org/10.1007/3-540-39568-7_2).
- [Fir25] Fireblocks Ltd. *Fireblocks-MPC*. <https://github.com/fireblocks/mpc-lib/>. 2025.
- [Fre07] Free Software Foundation. *The GNU General Public License version 3.0*. <https://www.gnu.org/licenses/gpl-3.0.en.html>. 2007.
- [Fre91] Free Software Foundation. *The GNU Library General Public License version 2.0*. <https://opensource.org/licenses/lgpl-2-0>. 1991.
- [MIT] MIT. *The MIT License*. <https://opensource.org/licenses/mit>.
- [Pai99] Pascal Paillier. “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes”. In: *EUROCRYPT’99*. Ed. by Jacques Stern. Vol. 1592. LNCS. Springer, Berlin, Heidelberg, May 1999, pp. 223–238. DOI: [10.1007/3-540-48910-X_16](https://doi.org/10.1007/3-540-48910-X_16).
- [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. 2026. DOI: [10.6028/NIST.IR.8214C](https://doi.org/10.6028/NIST.IR.8214C).