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

Title: Coordinated MPC

Subtitle: Message delivery protocol designed for MPC systems with a central Coordinator

Version: Preview Writeup 0.1 (2026-03-03)[1]

Team name: Dfns

Team members: Denis Varlakov, Antoine Urban, Nikita Sorokovikov

Abstract: MPC protocols are, by nature, interactive, often composed of several communication rounds, and require delivery layer to fulfill security properties at stake of security of the entire system. Implementing own delivery layer is error-prone, especially when other requirements are taken into account: such as operational simplicity, zero-downtime, and in-depth security. To fill the gap between the MPC protocol implementation and its real-world deployment, we submit a message delivery protocol in which a central party, Coordinator, relays messages, but is cryptographically restricted to Deny of Service attacks only. Protocol is designed and optimized specifically for MPC systems, it uses NIST-approved primitives relying either on standard or Post-Quantum assumptions, and provides MPC-specific features such as 1-round consensus on list of participants, globally unique execution identifier derivation, provable attribution of decryption, and a UC security proof, allowing it to be composed with other UC protocols.

Proposed crypto-systems: (I) Message delivery protocol (Categories S7).

Keywords: Message Delivery; Communication Layer; Threshold Cryptography; NIST Threshold Call

[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: Denis Varlakov [i1,a1], Antoine Urban [i2,a1], Nikita Sorokovikov [i3,a1]

Open Researcher and Contributor Identifiers (ORCID):

i1 (0009-0000-8497-3536); i2 (0009-0000-3670-0146);

Affiliations:

[a1] Research team, Dfns @ France

Main contacts:

- **Team mailing list:** nist-submission@dfns.co
- **Primary technical contact person:** Denis Varlakov, denis@dfns.co
- **Secondary contact person:** Antoine Urban, antoine@dfns.co

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 Protocols (MPC in general) are interactive by nature, and their security ultimately depends on the one of message delivery layer: failure to properly implement the later compromises integrity of the whole system.

Usually, an implied choice of delivery layer for MPC protocol is a mesh of peer-to-peer channels. It aligns well with spirit of distributed computations, and simple to explain: just say that all Parties establish pairwise TLS connections. However, in this submission we show that a communications model, which uses a central entity, called the Coordinator, to relay messages between protocol participants, has **smaller attack surface**, **lower operational cost** compared to the peer-to-peer mesh, and has **better performance** for consensus operations.

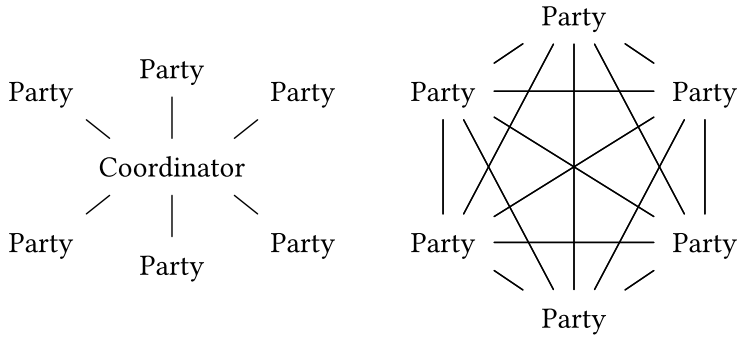

Figure 1: Open connections in a system with the Coordinator vs. in a peer-to-peer mesh

Attack surface. In peer-to-peer mesh, Parties have to have a network port exposed to the Internet, which widens their attack surface: it gives more opportunities for a network scanning and zero-day exploits. Depending on a threat model, it may be an unacceptable risk to leave an exposed port on a machine that possesses the secret key material.

Operational cost. In peer-to-peer mesh, each Party should be able to discover the network address of each other participant. Network address may change over time, and everyone on the network should be notified as briefly as possible to avoid downtime. Implementing and maintaining such discovery mechanism increases the operational cost. Besides, it's often required to set up a firewall so no network connections are possible unless explicitly allowed, which again increases the cost.

In our model, it's Parties who connect to the Coordinator, so they only need to know its address, and they do not need to have any port exposed to the Internet.

Coordinated system is more efficient as well in regard to resource consumption: it requires only $O(n)$ channels to be open, in contrast to peer-to-peer mesh that maintains $O(n^2)$ open connections in total.

Consensus. As we will demonstrate in our submission, having a central Party with a complete view on the system helps streamlining the consensus operations. For instance, we are able to

generate a globally unique protocol execution nonce at cost of 1 communication rounds, whereas in fully distributed systems it would take several rounds involving commit-reveal protocol.

Output reconstruction. Moreover, many Threshold Protocols explicitly rely on having a central Party that performs the last step of the protocol [KG20] [BLPP26] [CGGMP21] [BDLR25] [KU25] which is an optimization that the Coordinator could easily take advantage of.

Due to the advantages of the star topology, it is often used in real-world deployments of MPC systems. However, implementing a message delivery protocol that operates through the Coordinator requires developing a nontrivial crypto-system which has to preserve all security properties needed by the MPC protocol and, at the same time, has to minimize operational cost and to be efficient.

In this submission, we propose for the standardization a message delivery protocol (Category S7) that uses the Coordinator as a medium for the communications. It is designed specifically for running MPC protocols that involve 2 or more rounds and require authenticated and encrypted peer-to-peer channels, provides MPC-focused features, and targets production deployment with zero downtime.

1.1. Applicability

The proposed message delivery protocol targets wide range of MPC protocols, notably the ones that have more than 2 participants, involve at least 2 communication rounds, require authenticated and encrypted peer-to-peer channels, and/or make use of one of the MPC-focused features we provide.

Among write-ups submitted to NIST in the 1st phase of previews, we have identified:

- at least 5 protocols that match the criterias above: Amber [BPLP26], THE CLASH [BCCCLST26], Mithril [CDPENP26], PiVer [ABCJM+26], Quorus [BCEPT26], and Symphony [MPST26].
- 3 other submissions explicitly rely on the Coordinator: FROST [CGeGK+26], Haystack [KLL26], and Tanuki [BEKKK+26].

Note: many submissions do not explicitly state their round complexity and/or whether they assume authenticated/encrypted communication channels, so we weren't able to verify their applicability.

Other MPC protocols may still benefit from the delivery layer. For instance, if the protocol does not need encrypted peer-to-peer channels, using the proposed delivery layer introduces an overhead of handshakes, encryption and decryption, but it can be justified by the additional security properties our framework guarantees. E.g. the CGGMP threshold signing protocol submitted in [AABCDGM26] does not require message encryption, but it needs an execution nonce and message authentication that we provide.

2. Specification

2.1. Assumptions

The messaging protocol comes with two ciphersuites relying either on standard or on Post-Quantum assumptions.

2.2. Network

The message delivery protocol is abstracted from the underlying network stack managing a connection between the Party and the Coordinator. They may communicate over TCP, Websocket, QUIC or something else. In order to facilitate simultaneous execution of independent MPC protocols, The Party and the Coordinator may have several network connections/streams open in parallel. However, within one MPC protocol execution, the Party and the Coordinator shall use the same connection/stream that ensures delivery and preserves order of the sent messages.

Assumption on the network that we make is only required for correctness. If it doesn't hold, it may cause a Deny-of-Service, but it does not affect security.

We do not assume any other properties from the network stack. Notably, we do not require authentication or confidentiality as our protocol provides these properties. It's recommended, however, to use encrypted transport channels to hide metadata transmitted in the plaintext, such as multicast messages and Parties public keys.

It's recommended that the Party initiates the network connection to the Coordinator, so the Party does not expose any network ports to the Internet.

2.3. Security

Security proof. We prove security of the message delivery protocol in UC model, in which the adversary controls the Coordinator and subset of Parties, and at least one Party is assumed to be honest. We prove that adversary is restricted to Deny-of-Service attack: the Coordinator may choose to stop relaying messages or cause the channel to be closed by relaying a malformed message.

Notably, in a channel between two honest Parties, the active adversary cannot change content of a message, reorder messages, inject a message from another protocol execution, send message twice, learn the content of the peer-to-peer message, and so on.

Sessions. The Coordinator orchestrates the handshake protocol between each pair of instances of the Parties as they become online to establish peer-to-peer cryptographic sessions. The session is then used across different protocol executions, but its lifetime is limited by an upper bound on amount of messages that can be encrypted or until the Party's instance restarts. Lifetime is kept

reasonably short to enforce the Forward Secrecy. Since sessions are initiated by the Coordinator, Signers allocate fixed-size memory for the sessions to prevent malicious Coordinator from exploding memory of the Signer. If a new session cannot be fit into allocated memory, an older session is deleted to free up space.

Authenticated, Coordinator-Relayed Multicast channel. We provide an authenticated but unencrypted multicast channel. A message sent to this channel will be relayed to all other Parties, however, there's no cryptographic guarantee that all honest Parties receive the same message: if the Coordinator and a Party are malicious, they may collude to send different messages to different Parties.

Globally Unique Execution Nonce. For each protocol execution, we derive an execution nonce which is globally unique as long as there's at least one honest Party. The messaging protocol uses the nonce to bind messages in the multicast channel, but we also expose the nonce, as some MPC protocols require it to protect from ZK-proofs replay attacks, e.g. [CGGMP21] [KU25].

Consensus on common data. Most (if not all) MPC protocols assume that the honest Parties have the same view on the list of participants. In our protocol, we cryptographically ensure consensus on common data, such as the list of participants and the execution nonce.

Verifiable Attribution of Decryption to Sender. We provide a mechanism that allows the receiver of peer-to-peer message to attribute the decryption to the sender which can be verified by a third party. This may be used to prove that the sender aborts the protocol execution by sending a malformed ciphertext that fails to decrypt with AEAD error, by sending a junk plaintext that doesn't correspond to valid protocol message, by sending invalid ZK-proof, and so on. In order to enable this mechanism, Parties carry out a slightly more expensive version of the handshake, the resulting session has lower bound on amount of messages it can encrypt.

In our scheme, a sender S transmits a signed ciphertext c with signature σ to a receiver R. Formally, we achieve the following security properties:

- **Completeness:** As long as c is correctly signed, R can always generate a valid attribution π that successfully convinces an honest third-party verifier V of the exact decryption outcome. This holds true whether the outcome is a valid plaintext m or an AEAD decryption failure.
- **Soundness:** No probabilistic polynomial-time malicious receiver can produce a valid attribution π^* that convinces an honest third-party verifier of a different decryption outcome.
- **Bounded Leakage:** The public verification of proof π reveals a decryption of a message sent within encrypted session, but this does not compromise the forward or backward security of the session.

We construct the *Verifiable Attribution* gadget using Merkle Trees and standard digital signatures. The Soundness and Non-Repudiation properties reduce directly to the Collision Resistance of the underlying hash function and the EUF-CMA of the sender's static signing key.

3. Open-Source Implementation

The implementation is currently in development, we will open-source it once it's done under dual MIT/Apache-2.0 license. It will contain a library with an interface that can be plugged-in into any network transport channel, such as TCP, Websocket, QUIC and others.

4. Experimental Performance Evaluation

Once sessions are established, performance of the message delivery protocol is bound primarily by network latency: each message sent to Coordinator and the relayed to the recipient. Each sent message is encrypted and signed: typically it's negligible compared to network delay, but if Post-Quantum ciphersuite is used, the signing takes more time and the signature increases size of transmitted message.

We will measure and compare i) handshakes performance in different ciphersuites (classic, Post-Quantum, with/without Verifiable Attribution of Decryption), ii) performance of some TSS protocols communicating via the message delivery protocol using different ciphersuites over real network, iii) the bandwidth reduction achieved by the Authenticated Multicast gadget versus an $O(n^2)$ peer-to-peer TLS mesh.

5. Licensing, Patent Claims, and Funding

The submission code will be released under dual MIT/Apache-2.0 license, all external dependencies have compatible licenses.

To best of our knowledge, the message delivery protocol is not subject to any existing patents.

References

[AABCDGM26] Michael Adjedj, Tomer Ashur, Amit Singh Bhati, Geoffroy Couteau, Cyprien Delpech de Saint Guilhem, Michael Gutkin, and Nikos Makriyannis. *Distributed ECDSA Signatures: Fireblocks' CGGMP protocol*. Preview Writeup submitted to the NIST First Call for Multi-Party Threshold Schemes. Version 1.0. Available at: https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/Fireblocks-c-PW01.pdf. January 2026.

[ABCJM+26] Shahla Atapoor, Karim Baghery, Daniele Cozzo, Robin Jadoul, Hossein Moghaddas, Georgio Nicolas, Robi Pedersen, Mahdi Rahimi, Jannik Spiessens, and Barry Van Leeuwen. *PiVer:* Π *Verifiable Secret Sharing Framework*. Preview Writeup submitted to the NIST First Call for Multi-Party Threshold Schemes. Version 0.1. Available at: https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/PiVer-PW01.pdf. January 2026.

[BCCCLST26] Cyril Bouvier, Guilhem Castagnos, Dario Catalano, Quentin Combal, Fabien Laguillaumie, Federico Savasta, and Ida Tucker. *THE CLASH: THreshold ECDSA from CLASs groups linearly Homomorphic encryption*. Preview Writeup submitted to the NIST First Call for Multi-Party Threshold Schemes. Version 0.1. Available at: https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/BICYCCLIST-b-PW01.pdf. January 2026.

[BCEPT26] Alexander Bienstock, Leo de Castro, Daniel Escudero, Antigoni Polychroniadou, and Akira Takahashi. *Quorus: Scalable Threshold ML-DSA from MPC*. Preview Writeup submitted to the NIST First Call for Multi-Party Threshold Schemes. Version 0.1. Available at: https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/Quorus-PW01.pdf. January 2026.

[BDLR25] Renas Bacho, Sourav Das, Julian Loss, and Ling Ren. *Adaptively Secure Three-Round Threshold Schnorr Signatures from DDH*. Cryptology ePrint Archive, Paper 2025/1009. 2025. URL: https://eprint.iacr.org/2025/1009.

[BEKKK+26] Cecilia Boschini, Thomas Espitau, Aaron Kaiser, Shuichi Katsumata, Darya Kaviani, Russell W. F. Lai, Giulio Malavolta, Thomas Prest, Peter Schwabe, Akira Takahashi, Kaoru Takemure, and Mehdi Tibouchi. *Tanuki: Two-round Threshold Signatures from Lattices*. Preview Writeup submitted to the NIST First Call for Multi-Party Threshold Schemes. Version 0.1. Available at: https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/Tanuki-PW01.pdf. January 2026.

[BLPP26] Katharina Boudgoust, Oleksandra Lapiha, Rafaël del Pino, and Thomas Prest. *IND-CCA Lattice Threshold KEM under 30 KiB*. Cryptology ePrint Archive, Paper 2026/021. 2026. URL: https://eprint.iacr.org/2026/021.

[BPLP26] Katharina Boudgoust, Rafael del Pino, Oleksandra Lapiha, and Thomas Prest. *Amber: a Family of Efficient Lattice-Based IND-CCA Threshold KEMs*. Preview Writeup submitted to the NIST First Call for Multi-Party Threshold Schemes. Version 0.1. Available at: https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/Amber-PW01.pdf. January 2026.

[CDPENP26] Sofía Celi, Gustavo Delerue, Rafael del Pino, Thomas Espitau, Guilhem Niot, and Thomas Prest. *Mithril: Efficient Threshold ML-DSA from Secret Sharing with Short Shares*. Preview Writeup submitted to the NIST First Call for Multi-Party Threshold Schemes. Version 1.0. Available at: https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/Mithril-PW01.pdf. January 2026.

[CGeGK+26] Elizabeth Crites, Conrado Gouvea, Jack Grigg employment, Ian Goldberg, Jonathan Katz, Chelsea Komlo, Mary Maller, Simon Rastikian, Stefano Tessaro, Nikita Sorokovikov, Denis Varlakov, and Chenzhi Zhu. *FROST: Flexible Round-Optimized Schnorr Threshold Signatures: A Threshold Scheme Interchangeable with EdDSA*. Preview Writeup submitted to the NIST First Call for Multi-Party Threshold Schemes. Version 0.1. Available at: https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/FROST-PW01.pdf. January 2026.

[CGGMP21] Ran Canetti, Rosario Gennaro, Steven Goldfeder, Nikolaos Makriyannis, and Udi Peled. *UC Non-Interactive, Proactive, Threshold ECDSA with Identifiable Aborts*. Cryptology ePrint Archive, Paper 2021/060. 2021. DOI: 10.1145/3372297.3423367. URL: https://eprint.iacr.org/2021/060.

[KG20] Chelsea Komlo and Ian Goldberg. *FROST: Flexible Round-Optimized Schnorr Threshold Signatures*. Cryptology ePrint Archive, Paper 2020/852. 2020. URL: https://eprint.iacr.org/2020/852.

[KLL26] John Kelsey, Stefan Lucks, and Nathalie Lang. *Haystack: Threshold and Distributed Stateful Hash-Based Signatures*. Preview Writeup submitted to the NIST First Call for Multi-Party Threshold Schemes. Version 0.1. Available at: https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/Haystack-PW01.pdf. January 2026.

[KU25] Jonathan Katz and Antoine Urban. "Honest-Majority Threshold ECDSA with Batch Generation of Key-Independent Presignatures". In: *IACR Communications in Cryptology* 2.1 (April 8, 2025). DOI: 10.62056/a0l5wol7.

[MPST26] Hiraku Morita, Erik Pohle, Peter Scholl, and Daniel Tschudi. *Symphony: Threshold Evaluation of Symmetric Primitives: A protocol family for threshold AES, SHA2, SHA3 and G-/C-/H-/KMAC evaluation in the three-party, honest majority setting*. Preview Writeup submitted to the NIST First Call for Multi-Party Threshold Schemes. Version 0.1. Available at: https://csrc.nist.gov/csrc/media/Projects/threshold-cryptography/documents/TCall-1/Symphony-PW01.pdf. January 2026.

[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.