

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

Title: Octopus-TapTop and Octopus-Robot

Subtitle: Two-party BBS and Robust Threshold BBS Schemes

Version: Preview Writeup 1.0 (2026-09-20)¹

Team name: Octopus

Team members: Man Ho Au, Robert H. Deng, Liming Gao, Bowen Jiang, Xianhui Lu, Tian Qiu, Guofeng Tang, Ruida Wang, Haiyang Xue, Guomin Yang, Chengru Zhang

Abstract: This document previews the Octopus team’s proposals for the NIST Multi-Party Threshold Cryptography Call, specifying two threshold BBS schemes: a two-party BBS protocol in two passes and a robust t -out-of- n threshold BBS protocol in two rounds.

First, our two-party BBS scheme (denoted as Octopus-TapTop) uses a verifiable random function (VRF) and Castagnos-Laguillaumie (CL) homomorphic encryption to construct a highly efficient two-party signing protocol that requires only two passes. Second, we propose Octopus-Robot, a robust threshold BBS signature scheme supporting a general t -out-of- n access structure. Built upon threshold VRF and threshold CL encryption, it achieves robustness in just two rounds.

Proposed crypto-systems:

- (I) Octopus-TapTop: Two-party BBS (Categories S1)
- (II) Octopus-Robot: t -out-of- n threshold BBS (Categories S1)

Keywords: Threshold Cryptography; NIST Threshold Call; Two-party BBS; t -out-of- n Threshold BBS



Document License: [CC BY 4.0 International](https://creativecommons.org/licenses/by/4.0/)

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: Man Ho Au ^{i1,a1}, Robert H. Deng ^{i2,a3}, Liming Gao ^{i3,a2*}, Bowen Jiang ^{i4,a3*}, Xianhui Lu ^{i5,a2}, Tian Qiu ^{i6,a4}, Guofeng Tang ^{i7,a3}, Ruida Wang ^{i8,a2}, Haiyang Xue ^{i9,a3}, Guomin Yang ^{i10,a3}, Chengru Zhang ^{i11,a5}

Open Researcher and Contributor Identifiers (ORCID):

i1 (0000-0003-2068-9530); i2 (0000-0003-3491-8146); i3 (0009-0004-0211-4148); i4 (0009-0002-8867-866X); i5 (0000-0001-7091-5810); i6 (0009-0006-2429-8771); i7 (0000-0002-0121-5471); i8 (0009-0002-1276-1299); i9 (0000-0002-0173-7894); i10 (0000-0002-4949-7738); i11 (0009-0001-8755-5206)

Affiliations:

^{a1} Department of Computing, The Hong Kong Polytechnic University

^{a2} Institute of Information Engineering, Chinese Academy of Sciences

^{a3} SCIS, Singapore Management University

^{a4} Digital Trust Centre, Nanyang Technological University

^{a5} Ethereum Foundation

Associateship clarifications:

* Ph.D student (non-employee).

Main contacts:

- **Team mailing list:** octopus_mptc@googlegroups.com
- **Primary technical contact person:** Guofeng Tang, tang.guofeng789@gmail.com
- **Secondary contact person 1:** Haiyang Xue, haiyangxue@smu.edu.sg
- **Secondary contact person 2:** Tian Qiu, qtautumn6@gmail.com
- **Secondary contact person 3:** Bowen Jiang, bowen.jiang.2024@smu.edu.sg

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

The scope of this preview document is to present our plan for submitting two-party and general t -out-of- n threshold BBS schemes.

We plan to propose two schemes: 1) Octopus-TapTop, a two-party BBS in two passes, 2) Octopus-Robot, a robust t -out-of- n threshold BBS in two rounds.

BBS is an emerging digital signature standard jointly spearheaded by W3C Verifiable Credentials Group [Gre] and IETF [LKWL], specifically tailored for privacy-preserving digital identity and anonymous credentials. The two submissions focus on threshold BBS in both two-party and multi-party settings, which fall under Category S1: [Special] Signing of the NIST Threshold Call [NIST-IR8214C].

2. Specification

Organization. Our final submission will feature two cryptosystems: the first focuses on a two-party setting, while the second provides a t -out-of- n access structure. We briefly introduce each cryptosystem in the following subsections. Both cryptosystems include a distributed key generation phase executed once, followed by a signing phase that may be run multiple times as required.

2.1. Octopus-TapTop

Octopus-TapTop is a revisited two-party BBS scheme based on our team's paper [WQTNJZXY26]. In the Distributed Key Generation (DKG) phase, the BBS public key is generated, and the corresponding signing key is split into a 2-out-of- n threshold structure via Shamir's secret sharing. Subsequently, the valid BBS signature can only be generated via a two-party signing protocol with two-pass interaction.

2.1.1. System model

The threshold profile is 2-out-of- n , corresponding to a special t -out-of- n threshold BBS with $t = 2$ in the literature. The DKG phase is executed among any $n \geq 2$ parties, after which any 2 of those n parties can cooperatively generate a valid signature.

The parties operate within a synchronous network and are assumed to be connected by authenticated point-to-point channels. We do not assume the availability of a broadcast channel.

The main building blocks of this construction are a verifiable random function (VRF) and additively homomorphic encryption (AHE), instantiated with suitable zero-knowledge proofs. In particular, the VRF can be realized over elliptic curve groups under the DDH assumption.

For the homomorphic encryption layer, options include Castagnos-Laguillaumie [CL15] (CL), Paillier [Pai99], and Joye–Libert [JL13] (JL) schemes. Among these, the plaintext space of CL encryption natively aligns with the BBS group’s modulus, enabling a significantly more efficient construction.

2.1.2. Key Idea

Octopus-TapTop supports a two-party BBS signing protocol. The secret key is additively shared as $x = x_1 + x_2$, and the signature is $\sigma = (A, e)$ for the public key $(X = xG_2, \{H_i\}_{i=1}^\ell)$ on messages $\{m_i\}_{i \in [\ell]}$

$$A = (x + e)^{-1}B, \quad B = G_1 + \sum_{i=1}^{\ell} m_i H_i.$$

Two obstacles stand in the way of a two-pass design. The nonce e must be produced without bias, and the nonlinear inversion $(x + e)^{-1}$ must be carried out without extra interaction.

For the first obstacle, we use a VRF asymmetrically. The observation is that only the second mover can react to the other party’s share and skew the result, so P_1 simply samples β_1 uniformly and proves nothing, while P_2 evaluates the VRF to obtain β_2 together with a zero-knowledge proof of β_2 ’s validity. Both then derive $e = H(\beta_1 \parallel \beta_2)$ with the random oracle H after two-pass communication. This removes the three-pass pattern that commit-then-reveal would require. Note that we require it to be stateful signing, since a repeated sid would let P_1 vary β_1 and bias the output.

For the second obstacle, the inversion $(x + e)^{-1}$ can be inverted to a multiplication $r(x + e)$ with the secret nonce r . The insight is that the mask r is only temporary and canceled in the final division, so it can be chosen unilaterally by P_2 . More specifically, during key generation, P_1 publishes CL ciphertext of the share x_1 under its public key ek_1 , $C_{\text{key}_1} = \text{CL-Enc}(ek_1, x_1)$. In signing, P_2 picks r , sets $C = rB$, and uses additive homomorphism to form

$$D = C_{\text{key}_1}^r \cdot \text{CL-Enc}(ek_1, r(x_2 + e)),$$

a ciphertext of $r(x + e)$, where D is sent together with β_2 and its zero-knowledge proof, so no additional communication is incurred. Party P_1 decrypts to get $d = \text{CL-Dec}(dk_1, D)$ and outputs $A = d^{-1}C$, with x hidden behind the mask.

It extends readily to a 2-out-of- n access structure, where x is shared by Shamir’s secret sharing and is still recovered additively as $x = \lambda_i x_i + \lambda_j x_j$ for the Lagrange coefficients λ_i, λ_j of the two signers.

2.1.3. Security

Octopus-TapTop provides security against a static adversary corrupting up to $f = 1$ party. We prove that Octopus-TapTop securely realizes a two-party $\mathcal{F}_{\text{2BBS}}$ functionality in the random oracle model under the Universal Composability (UC) framework. The completed proof has been provided in our team’s paper [WQTNJZXY26].

2.2. Octopus-Robot

Octopus-Robot is a robust t -out-of- n threshold BBS scheme based on our team's work [TQJXYADL26]. In the DKG phase, the BBS signing key is split into a t -out-of- n threshold structure via Shamir's secret sharing. Any subset of at least t participants can collaboratively generate a valid BBS signature, while $t - 1$ participants cannot. In the case of $n \geq 2t - 1$, our scheme is robust, guaranteeing successful signature generation even in the presence of malicious behavior during signing executions. The signing protocol only requires two rounds of interaction.

2.2.1. System model

The threshold profile is t -out-of- n . We use authenticated and synchronized broadcast channels to consistently detect malicious participants.

Octopus-Robot relies on threshold Verifiable Random Function (VRF) [DEFLQR25] and threshold Castagnos-Laguillaumie (CL) encryption [BDO23] schemes along with their associated zero-knowledge proofs. Specifically, the threshold VRF enables any subset of t or more participants to deterministically derive the VRF output for a given input. We use a DDH-based VRF, in which a zero-knowledge proof of a simple DDH relation is required to verify the validity of evaluations. At the same time, we require the threshold CL scheme to support multiplication in BBS signatures. The threshold structure allows any t or more parties to decrypt a ciphertext, which is the cornerstone of robustness in a two-round signing protocol.

2.2.2. Key Idea

The BBS signature is written as

$$A = \frac{\gamma(G_1 + \sum_{k \in [\ell]} m_k H_k)}{\gamma(x + e)},$$

so the protocol must generate e , then produce shares of $\gamma(x + e)$ and of $\gamma(G_1 + \sum_{k \in [\ell]} m_k H_k)$ and open them. Each of these two tasks costs two rounds in prior work, and the contribution is to compress each into one.

The first round rests on a threshold VRF. An ordinary VRF is not enough under robustness, because the protocol does not abort on a bad share but discards it and aggregates the rest, so an adversary can still bias the outcome by choosing whether to supply a valid output. With a t -out-of- n threshold VRF, the key is shared, any t valid partial evaluations combine into one fixed group element Z , and the nonce follows as $e = H(Z)$ with the random oracle H , after one round of communication.

The second insight is that the ciphertext of $\gamma(x + e)$ need not wait for the nonces. Separating γx from γe lets the first round carry both $\text{CL.Enc}(\gamma_i)$ and $\text{CL.Enc}(\gamma_i x)$, the latter obtained homomorphically from C_x , so once e is known, the ciphertext of $\gamma(x + e)$ is assembled by local computation and the second round needs only a threshold decryption. The numerator is a group

element, so a threshold ElGamal scheme is used instead: $\gamma_i B$ is encrypted in the first round and combined locally; the second round then performs threshold decryption. To bind the same γ_i across the two encryption schemes, we define a consistency relation and give the corresponding Σ -protocol spanning the class group and the elliptic curve group.

2.2.3. Security

Octopus-Robot is proven to securely realize the $\mathcal{F}_{\text{BBS}}$ functionality in the random oracle model against a malicious adversary statically corrupting up to $t - 1$ parties. Specifically, it achieves “best-of-both-worlds”: security with identifiable abort under a dishonest majority and robustness under an honest majority. Because it guarantees that if there are insufficient honest parties, the signing protocol aborts while identifying the malicious parties; conversely, if the honest majority condition holds ($n \geq 2t - 1$), the protocol is guaranteed to successfully output a valid signature. The completed proof has been provided in our team’s paper [TQJXYADL26].

3. Open-Source Implementation

1. **Code structure:** Our reference implementation will provide complete Rust implementations of Octopus-TapTop and Octopus-Robot in a single, layered codebase. A shared primitive layer will implement curve and pairing operations, class-group arithmetic, encryption, secret sharing, and zero-knowledge proofs. For BLS12-381 arithmetic, we will use `blstrs` with `blst` as its backend. CL encryption and related class-group primitives will be implemented in Rust using `rug` and FFI bindings to GMP. Separate protocol state machines will implement distributed key generation and signing for both schemes. A networking layer and command-line tools will support multiprocess execution, testing, known-answer tests, and benchmarking on the NIST baseline platform.
2. **Code progress and availability:** Both source papers are accompanied by public C++ proof-of-concept implementations of Octopus-TapTop² and Octopus-Robot.³ We will use these implementations as references while reimplementing both schemes in Rust to meet the requirements of the NIST Threshold Call. Each Rust implementation will be audited against its corresponding protocol specification. At submission, the repository will be made public at a fixed commit with instructions for building, running, testing, and benchmarking the code.
3. **Implementation of the networking model:** Octopus-TapTop uses authenticated point-to-point channel and Octopus-Robot uses broadcast channel. Each party will run as a separate process and communicate with its peers over TCP. Octopus-Robot’s synchronized broadcast will be implemented over these authenticated connections. Configuration files will specify party identities, public keys, network addresses, threshold parameters, and session information.

²Octopus-TapTop: https://github.com/Xiaofei-Wu-20/Two-Party_Two-Passes_BBS_plus

³Octopus-Robot: <https://github.com/Jiangjiang-jiang/Multiparty-BBS-Plus-Implementation>.

4. **Testing:** Testing will cover primitives, protocol, and multiprocess executions of DKG and signing. We will introduce malicious deviations and network disruptions at different protocol stages, and verify that honest parties either complete correctly or terminate without accepting invalid results.

4. Licensing and Patent Claims

1. **Licensing:** The Rust code developed for this submission will be released under MIT OR Apache-2.0. All Bundled Dependencies and External Dependencies will use licenses as required by the NIST Threshold Call. The package will record licenses using SPDX expressions, preserve required notices, identify the license applicable to each component, and verify that the compiled reference implementation can be redistributed under an allowed license.
2. **Patent claims:** To our knowledge, no patent has or could have claims covering the contents of our submission.

References

- [BDO23] Lennart Braun, Ivan Damgård, and Claudio Orlandi. “Secure Multiparty Computation from Threshold Encryption Based on Class Groups”. In: *CRYPTO*. Vol. 14081. Springer, 2023, pp. 613–645. DOI: [10.1007/978-3-031-38557-5_20](https://doi.org/10.1007/978-3-031-38557-5_20). Also at ia.cr/2022/1437.
- [CL15] Guilhem Castagnos and Fabien Laguillaumie. “Linearly homomorphic encryption from DDH”. In: *Cryptographers’ Track at the RSA Conference*. Springer, 2015, pp. 487–505. DOI: [10.1007/978-3-319-16715-2_26](https://doi.org/10.1007/978-3-319-16715-2_26). Also at ia.cr/2015/047.
- [DEFLQR25] Poulami Das, Andreas Erwig, Sebastian Faust, Philipp-Florens Lehwald, Julian Loss, Ziyang Qu, and Siavash Riahi. “BIP32-Compatible Threshold Wallets”. In: *AsiaCCS*. ACM, 2025, pp. 856–872. DOI: [10.1145/3708821.3710830](https://doi.org/10.1145/3708821.3710830). Also at ia.cr/2023/312.
- [Gre] Manu Sporny Greg Bernstein. *BBS Data Integrity Cryptosuite*. <https://github.com/w3c/vc-di-bbs>.
- [JL13] Marc Joye and Benoît Libert. “Efficient cryptosystems from 2 k-th power residue symbols”. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2013, pp. 76–92. DOI: [10.1007/978-3-642-38348-9_5](https://doi.org/10.1007/978-3-642-38348-9_5).
- [LKWL] Tobias Looker, Vasilis Kalos, Andrew Whitehead, and Mike Lodder. *The BBS Signature Scheme*. <https://datatracker.ietf.org/doc/draft-irtf-cfrg-bbs-signatures/09/>.
- [Pai99] Pascal Paillier. “Public-key Cryptosystems Based on Composite Degree Residuosity Classes”. In: *Advances in Cryptology – EUROCRYPT ’99*. Springer, 1999, pp. 223–238. DOI: [10.1007/3-540-48910-X_16](https://doi.org/10.1007/3-540-48910-X_16).
- [TQJXYADL26] Guofeng Tang, Tian Qiu, Bowen Jiang, Haiyang Xue, Guomin Yang, Man Ho Au, Robert H. Deng, and Kwok-Yan Lam. “Robot: Robust Threshold BBS+ in Two Rounds”. In: *SP*. IEEE, 2026, pp. 1332–1349. Also at ia.cr/2026/651.
- [WQTNJZXY26] Xiaofei Wu, Tian Qiu, Guofeng Tang, Yuqing Niu, Bowen Jiang, Jun Zhou, Haiyang Xue, and Guomin Yang. “Two-Party BBS+ Signature in Two Passes”. In: *ESORICS*. 2026. Also at ia.cr/2026/573.
- [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).