Compilation of Public Comments on NISTIR 8214C ipd

NIST Multi-Party Threshold Cryptography Project[1]

Updated: April 25, 2023

The present document recalls the context of the call for public comments about the NISTIR 8214C ipd — NIST First Call for Multi-Party Threshold Schemes (Initial Public Draft) — published on January 25th, 2023, and compiles twelve corresponding public comments received by email via nistir-8214c-comments (at) nist (dot) gov. A future document will publish a "diff" between the final and the ipd versions of NISTIR 8214C, along with comments on the changes between the two versions.

Contents

[1] Webpage: https://csrc.nist.gov/projects/threshold-cryptography; email address: nistir-8214C-comments@nist.gov.

1 Context for the Public Comments

1.1 Announcing the draft and the period of public comments

The NISTIR 8214C ipd — NIST First Call for Multi-Party threshold Schemes (Initial Public Draft) — with digital object identifier (doi) 10.6028/NIST.IR.8214B.ipd, is publicly available via the NIST Computer Security Resource Center (CSRC), at: https://csrc.nist.gov/publications/detail/nistir/8214b/draft. The draft call had an associated recommended period for public comments, until April 10th, 2013, as publicly announced in various forms, including:

1. "NISTIR 8214C ipd" contacts page (see copy in Section 1.3)
2. CSRC publication page: https://csrc.nist.gov/publications/detail/nistir/8214c/draft
3. MPTC-forum: https://groups.google.com/a/list.nist.gov/g/MPTC-forum
4. IACR news item: https://iacr.org/news/item/20034
5. Twitter (@NISTCyber): https://twitter.com/NISTcyber/status/1618294983228030977 and https://twitter.com/NISTcyber/status/1644334373892923392
6. Public presentations: https://csrc.nist.gov/projects/threshold-cryptography/presentations

1.2 Feedback on previous related documents

Prior NIST publications/events related to threshold schemes have also received public comments with topics that may be relatable to NISTIR 8214C ipd. Some useful links:

1. NISTIR 8214B ipd (2022-Aug-12) — comments: https://csrc.nist.gov/csrc/media/Publications/nistir/8214b/draft/documents/NISTIR-8214B-ipd-public-feedback.pdf
2. Call2021a for Feedback (2021-Jul-02) — comments: https://csrc.nist.gov/csrc/media/projects/threshold-cryptography/documents/MPTC-Call2021a-Feedback-compilation.pdf
3. Workshop MPTS'20 (2020-Nov-04–06): https://csrc.nist.gov/events/2020/mpts2020
4. NISTIR 8214A (2020-Nov-08 / 2021-Jul-07) — comments: https://csrc.nist.gov/CSRC/media/Publications/nistir/8214a/final/documents/nistir-8214a-diff-comments-received.pdf
5. Workshop NTCW'19 (2019-Mar-11–12): https://csrc.nist.gov/Events/2019/NTCW19
6. NISTIR 8214 (2018-Jul-26 / 2019-Mar-01) — diff with comments: https://csrc.nist.gov/CSRC/media/Publications/nistir/8214/final/documents/nistir-8214-diff-comments-received.pdf

1.3 The contacts page of NISTIR 8214C ipd

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

NIST Technical Series Policies

Copyright, Fair Use, and Licensing Statements
NIST Technical Series Publication Identifier Syntax

Publication History

This version is the initial public draft (ipd).

How to cite this NIST Technical Series Publication

Luís T. A. N. Brandão, René Peralta (2023). NIST First Call for Multi-Party Threshold Schemes (Initial Public Draft). (National Institute of Standards and Technology, Gaithersburg, MD) NIST IR 8214C ipd.
https://doi.org/10.6028/NIST.IR.8214C.ipd

NIST Author ORCID identifiers

Luís T. A. N. Brandão: 0000-0002-4501-089X
René Peralta: 0000-0002-2318-7563

Contact Information

nistir-8214C-comments@nist.gov

Public Comment Period

January 25, 2023 – April 10, 2023

Submit Comments

Only via email: nistir-8214C-comments@nist.gov

All comments are subject to release under the Freedom of Information Act (FOIA).

1.4 The optional template made available for comments

The following word document was made available to facilitate comments.

Public comments about the NIST IR 8214C ipd
"NIST First Call for Multi-Party Threshold Schemes" (initial public draft)

FirstA LastA[1] · FirstB LastB[2] · FirstC LastC[3]

Month day, 2023

[[REMOVE THIS PORTION: This is a suggested but not mandatory template. Once filled, preferably with no more than six pages, export to a file in portable document format (PDF) and send by email to nistir-8214c-comments@nist.gov with the subject "Public comments on NIST IR 8214C ipd (Call for threshold)", before the public comment closes (April 11, 2023).]]

1. Generic scope of submissions and organization into two categories (Section 3)
<Comments go here>

2. Requirements and recommendations for submissions (Section 4)
<Comments go here>

3. Technical requirements (Section 5)
<Comments go here>

4. Primitives and threshold modes in Cat1 (Section 6)
<Comments go here>

5. Subcategories in Cat2 (Section 7)
<Comments go here>

6. Details of subcategory ___ (Appendix A)
<Comments go here> (duplicate this section as applicable)

7. Submission checklists (Appendix B)
<Comments go here>

8. Diverse editorial feedback
<Comments go here>

9. Other comments
<Comments go here>

[1] Fill in with affiliations and possible disclaimers.
[2] Fill in with affiliations and possible disclaimers.
[3] Fill in with affiliations and possible disclaimers.

Page 1 of 1

2 Informal Lists of Topics

The following topics, informally gathered based on the received comment sets, are intended as a mere auxiliary index of the content in the actual comments (detailed in Section 3).

#	Main topics (informal)
#1	Scope; quantum resistance.
#2	Innovation; models.
#3	Threshold motivation and alternatives; some expired patents.
#4	Mandatory checks; KAT values; implementation complexity.
#5	Fully homomorphic encryption (FHE).
#6	Threshold & oblivious pseudo-random functions (PRF); keygen; robustness; asynchronicity.
#7	Shamir Secret-sharing (safe evaluation points)
#8	Scope; keygen; adaptive security; key-refresh; bounds; broadcast; thresholds; party's state.
#9	Attribute-based encryption (ABE): ciphertext-policy, key-policy, multi-authority.
#10	All-or-nothing transform (AONT) and homomophic encryption.
#11	Implementation dependencies, KAT values in randomized multi-party runs.
#12	Robustness.

2.1 From A. Thompson (see Comment Set #1)

1. Emphasis of quantum resistance; examples of PQC selected schemes.
2. Clarify the types of adversaries.

2.2 From S. Ranellucci (see Comment Set #2)

1. Innovation: notes on state of the art, system setup, system model, applicability of protocols.
2. Continuous progress in the area.
3. MPC protocols vs. traditional primitives.
4. Broad scope vs narrow focus.
5. Protocol evolution.
6. Conservative assumptions, vulnerabilities, scrutiny, attacks.
7. Validation of key-shares.
8. Making assumptions explicit.

2.3 From F. Sudia (see Comment Set #3)

1. Theshold motivation, risk assessments.
2. Confidentiality, integrity, availability, authenticity.

3. Auditability of various properties.
4. PKI decentralization.
5. ulti-signatures, secret-sharing of secret-shares.
6. Malicious design, testing, proof of use of key-shares.
7. References to 4 expired patents.

2.4 From J. Miller, and J. van de Pol (see Comment Set #4)

1. Mandatory checks with corresponding KAT values.
2. Implementation complexity.
3. Auditability of implementation and API.
4. Veto power and liveness property in threshold profiles.

2.5 From A. Badawi, A. Alexandru, N. Genise, D. Micciancio, Y. Polyakov, Saraswathy R.V., and V. Vaikuntanathan (see Comment Set #5)

1. FHE based on LWE / Ring-LWE.
2. Motivation for FHE, use-cases, standardization efforts.
3. FHE as a gadget for MPC.
4. FHE in passive model.
5. ZKP for actively secure (threshold) FHE.
6. Building blocks with increasing levels of complexity for active security.
7. Various concrete FHE schemes in three categories.
8. Three classes of use-cases: keygen/decryption; homomorphic additions; general computations.
9. Threshold friendliness across the three categories of FHE schemes.
10. Open-source implementations (existing libraries).
11. 26 bibliographic references.

2.6 From F. Benhamouda, S. Halevi, H. Krawczyk, and T. Rabin (see Comment Set #6)

1. Cat2 also relevant because of applications.
2. Threshold PRF/OPRF for key-management and decentralized identities.
3. Threshold PRFs, and threshold oblicious PRFs.
4. Threshold randomness beacon for keygen or as a gadget.
5. Clarification of gadgets vs. subprotocols (e.g., agreement, and broadcast encryption).

6. Protocol robustness (e.g., guaranteed output delivery) and liveness.
7. Asynchronous channels.
8. DKG is also usable for ephemeral secrets.

2.7 From H. Maji (see Comment Set #7)

1. Security of secret-sharing when subject to leakage from the secret shares.
2. For example, some bits of binary representation of each share may leak.
3. Distinguisher attack when shares are evaluated at consecurity integers.
4. Safer evaluation places in Shamir secret sharing.
5. Security when selecting random evaluation places.
6. Derandomization of evaluation places, Mersenne prime modulus.
7. Needed caution in selecting evaluation places.
8. Five bibliographic references

2.8 From J. Katz, C. Komlo, X. Meng, and N. Smart (see Comment Set #8)

1. Broadness of call, divising attention across subcategories.
2. Threshold schemes for signatures are more mature than for other key agreement.
3. Keygen state of operations needs to be explict in submissions.
4. Adaptive security in long-term vs. short-term.
5. Possible different schemes aiming static and (optional) adaptive security.
6. Encouragement of partial analysis against adaptive attacks.
7. Key-refresh sub-protocols as standalone submissions.
8. Concrete security bounds (and note when proof bounds are not tight but only asymptotic).
9. Communication: broadcast vs. point-to-point communication model.
10. Threshold profiles: clarify the participation threshold (k) possibly different from $f + 1$.
11. Recommend modeling the state of parties (inc. erasures and session identifiers for concurrency).

2.9 From G. Alpáar, L. Botros, A. de la Piedra, and M. Venema (see Comment Set #9)

1. Considering ABE vs. MPC fields, clarify terminology, models and functionalities of ABE.
2. Ciphertext-policy (CP)-ABE and Key-policy (KP)-ABE, collusion resistance.
3. Multi-authority (MA)-ABE (multiple domains, separation of attributes).
4. In CP-ABE, access structure can eb determined durine encryption.

5. Three types of MA-ABE: thresholdized, distributed, and decentralized.
6. Distributed and decentralized: authorities (typically) manage unique sets of attributes.
7. Decentralized (more difficult) & distributed: autonomy of user; keys from multiple authorities.
8. Thresholdized: each authorityhas a key (secret-share?) for each attribute.
9. Examples gadgets for ABE: access trees; LSSS matrices.
10. Methodologies exist to evaluate performance of pairing-based ABE (reference material?).
11. Clarify which design goal (e.g., keygen, encryption, ...) is being optimized.
12. 13 bibliographic references.

2.10 From G. Seghaier, and J. Doget (see Comment Set #10)

1. All-or-nothing transform (AONT).
2. Example of deployed application, with patented parts, available for licensing.
3. Distributed multi-cloud environment; threshold interfaces different from those in IR 8214A.
4. Multi-cloud storage (redundancy & availability); proxy orchestrator (entry point for client).
5. Client encrypts; the proxy homomorphically secret-shares data across multiple clouds.
6. Future submissions: open-source vs. copyright vs. copyleft+commercial-license.
7. Client transforms the input/output data using homomorphic encryption scheme.
8. Consideration of AONT in Cat1 or Cat2, possibly as a gadget.

2.11 From a. shelat, J. Doerner, E. Lee, and Y. Kondi (see Comment Set #11)

1. Whether to submit all dependencies: compilers, interpreters, libraries, languages.
2. Stable vs. nightly features of programming languages.
3. How to consider KAT values for multi-party, multi-round protocols.
4. Upcoming submissions.

2.12 From T. Ruffing (see Comment Set #12)

1. Robustness (protocol termination in spite of attacker) is security property against the attacker.
2. Importance depends on application, e.g., one-time keygen vs. threshold signing or decryption.
3. Make explicit the robustness guarantees/breakdowns.
4. Robustness may rely on additional adversarial assumptions.
5. Considerations on the terminology for robustness.
6. Broadcast has several flavours; relevant to know which one is considered in a proposal.

3 Received Public Comments

Comment Set #1: From A. Thompson

Public comments about the NIST IR 8214C ipd
"NIST First Call for Multi-Party Threshold Schemes" (initial public draft)

Alyssa Thompson[1]

April 6, 2023

Generic scope of submissions and organization into two categories (Section 3)
An increased emphasis on quantum resistant techniques may be beneficial. In Table 2, add Crystals-Dilithium as an example for C2.1 and Crystal-Kyber as an example for C2.3.

Technical requirements (Section 5)
In section 5.2.3, it could be made clearer that protection against covert and passive adversaries is assumed. In addition, in the definition of an active adversary, can the adversary change the number of corrupt parties during computation? If so, please update the language to be more complete.

Subcategories in Cat2 (Section 7)
For C2.8, consider changing "gadget" to "framework" or "technique" since the term "gadget" doesn't appear widely in the multi-party literature. It seems like a catch-all type here and it's not clear if that is the intention. Technique or Framework would better represent GC and related mechanisms to create threshold systems.

Other comments
Section A.1.4, line 1312, change "bot" to "not."

[1] NSA

Comment Set #2: From S. Ranellucci

From: Samuel Ranellucci
Sent: Monday, April 10, 2023 12:37
To: nistir-8214C-comments; Arash Afshar
Subject: comments about “NIST First Call for Multi-Party Threshold Schemes”

Dear Luís Brandão, Rene Peralta,
The comments about “NIST First Call for Multi-Party Threshold Schemes” can be found below.
Thank you and NIST for your effort.
Best Regards,
Samuel Ranellucci
Coinbase

Introduction

This document provides feedback on the NIST draft titled “NIST First Call for Multi-Party Threshold Schemes”. This document will contain multiple sections, each describing independent feedback.

Mandates can Harm Innovation

I have many years of experience developing applied MPC protocols. Here are some of my observations from working in this field.

1. Very often, when I need to implement some MPC functionality, I have to significantly improve the state of the art.
2. Both the system setup and the security model heavily influence the choice of protocol, the customization and the optimizations that I can use. A protocol that is acceptable in one context can be completely useless in another context.
3. Significantly more efficient protocols keep appearing in the literature and the eprint archive (https://eprint.iacr.org/)
4. There is generally no “best protocol” for a given functionality.

These issues don’t exist for more traditional cryptographic primitives such as encryption schemes and collision-resistant hash functions.

Comment Set #2: From S. Ranellucci

In many cases, companies and organizations look at the NIST standards to create requirements. FIPS certifications often require NIST-standardized primitives. In many cases, this is a good thing and raises the bar in terms of security and auditability of software. Unfortunately, by virtue of the issues I mentioned above, in the particular case of MPC protocols, this might not be the case. Since organizations might be mandated to use a protocol that is not necessarily the best for their unique situation.

Overly Broad Scope

This draft has a scope that is too extensive. By my calculation, In excess of 40 different standards could be provided where each standard either realizes a different functionality or provides security in a different security model. At this point, a narrower focus would have a better result. For example by focusing on a subset of such protocol for now and extending it to others in the future.

Protocol Evolution

When traditional cryptographic primitives (encryption, digital signatures and collision-resistant hash functions) have been standardized it was the case that subsequent improvements within 5-10 years were non-disruptive. I do not believe this will be the case for MPC. As a result, it is plausible that standardized MPC schemes might be subsumed within a relatively short amount of time.

Conservative Assumptions and Constructions with Strong Validation

In the last few years, many protocols in the literature have been shown to have vulnerabilities. The paper https://eprint.iacr.org/2020/945.pdf proposed a novel attack for breaking blind signatures by using clever mathematical techniques. It is therefore essential that protocols should use conservative assumptions and constructions that have stood the test of time. In particular, the generic group model, and the one-more discrete logarithm assumption are suspect. In addition, we note that intense scrutiny must be given to any proposal since many well-known protocols have been revealed to have significant vulnerabilities. We list below some papers detailing significant attacks or issues in well-known constructions.

Primitive	Attacks, Issues

Comment Set #2: From S. Ranellucci

Oblivious Transfer Extension	eprint.iacr.org/2022/192 eprint.iacr.org/2019/074 eprint.iacr.org/2019/706
Threshold ECDSA	eprint.iacr.org/2021/1621 https://blog.verichains.io/p/vsa-2022-120-multichain-key-extraction
Base Oblivious Transfer	eprint.iacr.org/2017/370.pdf eprint.iacr.org/2016/624.pdf

---

Key-Share Validation

When using threshold cryptography, the secret key is often shared between parties. During threshold signing or threshold encryption, it is necessary to ensure that each party provides the correct share of the secret key. Therefore every standard should ensure that the adversary cannot tamper with the shares of the key. For example, consider the case where a non-interactive version of AES encryption is susceptible to related-key attacks, but its thresholdized version must not be susceptible to such attacks.

Assumptions

There should be a section stating the assumptions needed so that the protocol is secure. Each assumption should be written as a warning label. For example,

Security Requirements

- One-time trusted setup.
- All honest participants must participate in every signing operation.
- Security relies on the one-more discrete log assumption in the generic group model.
- Secure as long as three of the four participants are honest.

Comment Set #3: From F. Sudia

Public Comments About the NIST IR 8214C ipd
"NIST First Call for Multi-Party Threshold Schemes" (initial public draft)

Frank W. Sudia[1]

April 10, 2023

To: nistir-8214c-comments@nist.gov

Re: Public Comments on NIST IR 8214C ipd (Call for Threshold)

2. Requirements and Recommendations for Submissions (Section 4)

As we embark on threshold standardization, it is well to recall the reasons threshold systems were initially developed, which included the requirement that the implemented system be able to pass Internal, External (i.e., Big X), and Federal Reserve or DoD Audits, remind ourselves to perform adequate risk assessments, and to manage the level of reliance to be placed on a given system or key, based on those risk assessments. No algorithm, no matter how clever, can function outside of some physical implementation, including associated message protocols, while being subject to audit, control methodologies, operational risk, and reliance management.

Recall the key info-sec principles of confidentiality, integrity, availability, and authenticity (CIAA), each of which can be defeated in various ways. Any responsible development team seeking to get their app into production will need to comply with a tall stack of app-development and security standards, and convince their Auditors to let them go live. How each mathematical quantity will be generated, stored, managed, used, verified, backed up, restored, where, when and by whom, in order to achieve CIAA objectives, must be explained and documented.

In addition to verifying the physical implementation, the Auditors will also ask you to prove that the system has the Abstract properties you say it has, and is immune from tampering by malicious adversaries. Is this really a 3 out of 5 system, and can we be assured there are no secret quorums or secret shares, let alone that one party somehow has the entire key, etc?

On the policy / marketing level, teams should recall how unpopular PKI has been, with experts decrying its risks, privacy groups deploring any central ID system, and government agencies who wish to "see through" everything, for whom every day there's no functioning PKI is another good day. To minimize such policy headwinds, non-centralized uses, such as *the critical need to more reliably protect Bitcoin wallets*, should be prioritized.

3. Technical Requirements (Section 5)

Turning to technical matters, in many cases the going-in Availability issue of how can we plausibly backup an important key in case of system failure, might just as well be addressed by a non-threshold (e.g., 3 out of 3) scheme using *High School Algebra*, as proposed in the pioneer US Patent 5,825,880 (expired). The eventual standard should include such non-threshold split key systems, as a limiting case, which may be simpler to implement and audit, yet could suffice for many users.

Another feature larger users may find attractive would be a well-defined option in the protocol to generate sub-splits, wherein each key share can be "split" (or co-generated) again, to one or more successive tiers, as a further security and backup/recovery risk reduction feature.

[1]Independent researcher. No conflicts of interest.

Comment Set #3: From F. Sudia

All such protocols should be future-proofed by including variable algorithm IDs and key lengths.

As remarked in this Request For Comments, "On the other extreme, a proposed protocol must not allow the major safety properties of interest to be trivially broken in case of adaptive corruptions, as in the classical example of a protocol that delegates all capabilities to a small quorum that is difficult to guess in advance, but whose overall corruption (by an adaptive adversary) would be disastrous." Page 22, lines 912-915. Not to mention that such a system violates Audit principles, defrauds anyone who participates in or relies on it, creates financial, legal, and reputational risks, and may constitute a felony of tampering with an official system, warranting jail terms.

The enumeration of Adversaries must be expanded to include not only corrupt protocol participants or malicious intermediaries, but also corrupt sponsor personnel. Any persons or groups designing, setting up, or effectuating key generation operations may be in a position to mis-design or mis-configure the process, either through their own math skills, exploits from criminal groups, or with the assistance of prior clients or employers (some with preeminent math skills) who they are still secretly working for. The notion that “you” will somehow be able to verify all this yourself is most likely false, since in reality some developer or consultant will be the one doing it.

Hence in order to convince your many Audit teams that your system has the properties you say it has, and that adversaries (including any double-agents among your staff or consultants) have not altered them, it will be preferable if its critical properties can be verified, e.g., via a test suite of protocols, viewable by all participants, and logged onto some blockchain, which demonstrate them.

For example, you could start by challenging all participants to create all partial signatures, to assure that no signatures with less that the stated quorum of signers are valid, including proof (possibly zero knowledge) that each partial signature was indeed made by the key shares that each participant supposedly has. Such proofs could be run again each time the system is modified to add, delete, or replace a key share, to assure that the claimed properties (or any new set of properties) still hold, and (perhaps for a nominal fee) upon demand of any participant.

As a worst-case scenario, proof that the conditions held *during a given signing operation* might be block-chained and appended to the signature, as a further proof of its genuineness.

Many variations of such proofs might arise, but I will not develop any of them here. (Nor am I planning to research, develop, or patent any further ideas related to this topic.)

Many thanks to NIST for initiating this effort, and best of luck to the proposing teams.

9. Other Comments

Some Related Patents (all expired):

1. Sudia et al, US Patent 5,825,880, Multi-Step Digital Signature Method and System, 10-20-1998 (the *High School Algebra* version)
2. Brickell et al, US Patent 5,867,578, Adaptive Multi-Step Digital Signature System and Method of Operation Thereof, 2-2-1999 (the advanced math version)
3. Asay et al, US Patent 5,903,882, Reliance server for electronic transaction system, 5-11-99 (part 2 of the spec, written by me, may be more readable)
4. Sudia et al, US Patent 6,209,091, Multi-Step Digital Signature Method and System, 3-27-01 (further claims re delegation)

Comment Set #4: From J. Miller, and J. van de Pol

Public comments about the NIST IR 8214C ipd

"NIST First Call for Multi-Party Threshold Schemes" (initial public draft)

Jim Miller[1] · Joop van de Pol[2]

April 10, 2023

1. Generic scope of submissions and organization into two categories (Section 3)
No comments.

2. Requirements and recommendations for submissions (Section 4)

- Regarding item S8 in Section 4.4.2, we recommend requiring that each protocol specification clearly states all mandatory checks that are needed for security (including trivial things like zero mod group order).
- Regarding item S10 in Section 4.2.2, we recommend also mentioning a consideration for 'implementation complexity'. It is difficult to encapsulate this in a specific metric, but some threshold schemes are more complicated to implement than others, which has security implications for creating and auditing implementations. As an example,
- Regarding item M2 in Section 4.3, we recommend mentioning 'auditability' as a requirement for reference implementations, both for the implementation itself and its API. I.e., it should be straightforward to audit the reference implementation, and it should be straightforward to audit the correct use of the API by implementations using the scheme.
- Regarding item X4 in Section 4.4, we recommend requiring that there are "known-answer test" values (KAT) for each of the aforementioned mandatory checks (cf. the comment on item S8), such that verification of these KAT values shows that the mandatory checks are implemented. Preferably, KAT values are provided for all combinations of positive and negative cases of these checks.

3. Technical requirements (Section 5)
Regarding item T5 in Section 5.5, we recommend mentioning the concept of 'veto power' or other liveness properties in relation to the chosen threshold profiles.

4. Primitives and threshold modes in Cat1 (Section 6)
No comments.

5. Subcategories in Cat2 (Section 7)
No comments.

6. Details of subcategories (Appendix A)
No comments.

7. Submission checklists (Appendix B)
No comments.

8. Diverse editorial feedback
No comments.

9. Other comments
No comments.

[1] Trail of Bits
[2] Trail of Bits

Comment Set #5: From A. Badawi, A. Alexandru, N. Genise, D. Micciancio, Y. Polyakov, Saraswathy R.V., and V. Vaikuntanathan

Comments on NIST First Call for Multi-Party Threshold Schemes

Ahmad Al Badawi[1], Andreea Alexandru[1], Nicholas Genise[1], Daniele Micciancio[1,3], Yuriy Polyakov[1], Saraswathy R.V.[1], and Vinod Vaikuntanathan[1,2]

[1]Duality Technologies
[2]MIT
[3]UCSD

April 10, 2023

Our comments are for Fully Homomorphic Encryption (FHE) schemes based on LWE and Ring/Module LWE over power-of-two cyclotomic rings, since that is what is most commonly implemented in open-source libraries. Our comments could apply to other FHE schemes with different hardness assumptions as well (e.g., NTRU).

Comment 1: Motivation for Standardizing (Threshold) Fully Homomorphic Encryption

We strongly believe in an organized community effort to standardize both FHE and its threshold variants. The state-of-the-art schemes for (threshold) FHE are quantum-resistant and can be applied to many significant use cases, such as AES enciphering, (federated) machine learning and statistics, secure database queries, Beaver triple generation in secure multi-party computation (MPC), and many more. There are two existing prominent standardization efforts for FHE: `homomorphicencryption.org` (since 2017) and `https://www.iso.org/standard/83139.html` (ISO from 2021).

FHE as a gadget. (Threshold) FHE is already an important cryptographic gadget in itself. For example, semi-maliciously[1] secure FHE [AJL+12] is used as a gadget in actively

[1]Semi-malicious security is where the adversary can choose its randomness malicously otherwise following the protocol. RLWE-based threshold FHE schemes achieve some form of semi-malicious security "for free" in the passive security model [AJL+12].

Comment Set #5: From A. Badawi, A. Alexandru, N. Genise, D. Micciancio, Y. Polyakov, Saraswathy R.V., and V. Vaikuntanathan

secure MPC protocols [KPR18, CHI+21]. Standardizing FHE schemes in this simpler security model is more straightforward than doing it directly in the active security model. Conversely, actively secure (threshold) FHE will likely use other important gadgets, such as zero-knowledge proofs.

A path towards standardization for actively secure FHE. Standardizing actively secure (threshold) FHE schemes is a significant effort since (i) FHE algorithms in themselves are intricate, and (ii) state-of-the-art open-source FHE libraries use the passive security (also called semi-honest) model. A potential simpler path to standardization is to standardize building blocks, like the following, presented in order of complexity:

Class 1: Threshold key generation and decryption.

Class 2: HE schemes with linearly-homomorphic operations/additive homomorphic encryption (AHE).

Class 3: FHE schemes, with relinearization/key switching (usually required by non-linear homomorphic operations, e.g., multiplications or bootstrapping).

Note that the first class is roughly a subset of the second class, and the second class is roughly a subset of the third class. The last class encompasses FHE schemes, which all require circular security. The first two classes are less complex, and there are many use cases of interest to the community utilizing only the capabilities in the first two classes, as described in Comment 2. Moreover, (R)LWE-based schemes are key-homomorphic, so the insights learned from standardizing the first class will likely be valuable in the second and the third classes.

A potential course of action for standardization is to design a standard template for transforming a passively secure or a semi-maliciously secure (threshold) FHE scheme into an actively secure scheme via zero-knowledge proofs.

FHE Schemes. The most common FHE schemes can be separated into three categories: (i) Brakerski-Gentry-Vaikuntanathan [BGV12] (BGV) and Brakerski [Bra12]/Fan-Vercauteren [FV12] (BFV), (ii) Ducas-Micciancio [DM15] (DM, also called FHEW)/Chillotti-Gama-Georgieva-Izabachène [CGGI16] (CGGI, also called TFHE), and (iii) Cheon-Kim-Kim-Song [CKKS17] (CKKS, also called HEAAN). BGV and BFV are schemes supporting SIMD encrypted computations for arithmetic circuits modulo a prime power, DM and CGGI are schemes supporting binary or small-precision arithmetic (larger-precision plaintext spaces require either very large parameters or many small-precision bootstrapping operations), and CKKS supports SIMD fixed-point-like arithmetic circuits. The main method for threshold decryption in all three categories, against a passive adversary, is noise flooding.

Comment Set #5: From A. Badawi, A. Alexandru, N. Genise, D. Micciancio, Y. Polyakov, Saraswathy R.V., and V. Vaikuntanathan

Comment 2: Use Cases

The AES enciphering use case mentioned in the NIST call falls in the third class described above, requiring the full capabilities of FHE. However, there are already important use cases in the first two classes. These include all the use cases in Category 1 from the NIST call, some forms of federated learning and inference, and generating the offline setup for MPC. For completeness, we mention a few existing works in each class, focusing on active security.

Class 1. Here, the use cases are the same as for Category 1 in the NIST call. This first class of algorithms—threshold key generation and decryption—would build upon the NIST PQC standardization effort. For instance, threshold-izing Kyber falls into this class. Therefore, the insights gained in standardizing this first class in the active security model for adaptive, mobile adversaries will apply to the latter two classes as well. Gladius [CCMS21], which threshold-ized Saber, falls under this class, but it remains to be seen how its techniques—FO transform, LWR, etc.—apply to threshold FHE (efficiently).

Class 2. The use cases for this class cover computations with plaintext-ciphertext additions/multiplications and ciphertext additions, such as secure aggregation and voting, some forms of federated learning and inference, polynomial statistics, and Beaver triple generation for MPC setup. For instance, Gazelle [JVC18] and Cheetah [HLHD22] use BFV encryption for evaluating the linear layers of a neural network (and garbled circuits, oblivious transfer and secret-sharing for the non-linear layers), in a semi-honest two-party setting. Further, Badawi et al. [BJL+21] use BFV to compute CNN inference with plaintext-ciphertext multiplications, and their solution can be threshold-ized. These works can be adapted to be circuit-private via noise flooding. In the active security model, Diogenes [CHI+21] uses RLWE-ciphertext additions with zero-knowledge proofs to achieve distributed RSA modulus generation against an active dishonest majority. Aranha et al. [ABGS22] use actively secure threshold BGV (both threshold key generation and threshold decryption) and zero-knowledge proofs to construct an efficient form of secure voting. In the context of providing an actively secure setup for secure multi-party computation, Overdrive [KPR18] employs BGV for addition and plaintext-ciphertext multiplication to generate Beaver triples, using noise flooding for circuit privacy and zero-knowledge proofs for active security. Their solution is for two parties, but can be easily threshold-ized, e.g., by following the BGV distributed key generation and decryption protocols from [DKL+13, RST+22].

Class 3. Many impactful use cases are in this class: AES enciphering [GHS12] (using BGV over non-power-of-two cyclotomic rings), neural network training and inference [SPT+21], decision tree training [LMP22], private set intersection (PSI) [CMdG+21][2], private information retrieval (PIR) [ACLS18][3] [GH19], and many more. The referenced works here are

[2] `https://github.com/microsoft/APSI`
[3] `https://github.com/microsoft/SealPIR`

Comment Set #5: From A. Badawi, A. Alexandru, N. Genise, D. Micciancio, Y. Polyakov, Saraswathy R.V., and V. Vaikuntanathan

mostly in the passively-secure model, but the motivation towards making them actively secure is clear. Brakerski et al. [BHP17] show actively secure distributed setup generation, encryption and decryption protocols for an LWE-based homomorphic encryption scheme, without practical implementation. We do not know of any practical works in this direction. It is worth noting that in this class, even under a common passive security model (which currently is still an active area of research), standardization has clear technical challenges.

Comment 3: FHE Schemes' Threshold Friendliness

All three categories of FHE schemes: (1) BGV/BFV, (2) DM/CGGI, and (3) CKKS can be threshold-ized up to linear operations (class 2 above) in the actively secure model with similar techniques and impacts on performance. On the other hand, finding efficient ways to perform threshold decryption in the active, adaptive, mobile security model for the third class of algorithms (non-linear operations) is a technical challenge worthy of a serious community effort. We now describe the current state of the art in each category's passively-secure (or semi-malicious) version and the respective performance degradation from thresholdization.

BGV/BFV. BGV and BFV are the two schemes most suitable for threshold key generation and decryption, as-is, without affecting performance. The reason is that in BFV/BGV, threshold decryption is done efficiently by noise flooding with uniformly random noise in the RNS representation. This adds two extra RNS limbs (50-63 bit moduli in the ciphertext modulus) for 100+ bits of statistical security, while three limbs can easily achieve 128-bits of statistical security with many adversarial queries (well over 2^{50} assuming statistical security degrades with the square root of the number of queries).

DM/CGGI and CKKS. Conversely, DM/CGGI and CKKS suffer from a larger performance degradation from noise flooding. This is because the former category must increase the parameters in a homomorphic accumulator, and the latter must increase its scaling factor to over 64 bits, requiring 128-bit arithmetic, even in the RNS setting. Therefore, threshold decryption alone introduces technical challenges simply in the passive security model for these two categories of schemes.

Comment 4: Open-Source Implementations

Here we summarize existing implementations available in open-source libraries.

Active Libraries There are three main active FHE libraries, and two active MPC libraries with some FHE algorithms:

Comment Set #5: From A. Badawi, A. Alexandru, N. Genise, D. Micciancio, Y. Polyakov, Saraswathy R.V., and V. Vaikuntanathan

1. OpenFHE[4] [BBB+22] builds off of previous libraries, such as PALISADE[5], HEAAN[6], and HElib[7]. It implements the N out of N key generation/threshold decryption and $t > N/2$ out of N Shamir secret sharing threshold key generation/decryption described in [AJL+12] for BGV/BFV and CKKS in the passive security model. OpenFHE also has an implementation of the non-threshold version of the BGV and BFV schemes, the DM/CGGI schemes and the CKKS scheme in the recent IND-CPA$^{\mathrm{D}}$ security model [LMSS22]. The latter's cryptanalysis is very similar to the cryptanalysis of threshold RLWE decryption.

2. Lattigo[8] implements threshold key generation/decryption for BFV/BGV and CKKS for Shamir sharing with general t out of N sharing together with N out of N sharing in the passive security model [MBH22], as well as the corresponding non-threshold versions.

3. TFHE-rs[9] is a (non-threshold) implementation of FHE over the torus (TFHE).

4. MP-SPDZ[10] is an MPC library that has actively secure Beaver triple generation from BGV and an implementation of actively secure distributed key generation for BGV. The active security is specific to the application of Beaver triples and does not include a generic actively-secure threshold BGV decryption.

5. SecretFlow[11] is an MPC/FHE framework with AHE implementations of BFV and CKKS, and an implementation of CGGI.

Inactive Libraries The following libraries are not nearly as active, or supported, as the libraries above, or are completely inactive. SEAL[12] has FHE implementations of BFV, BGV, and CKKS schemes without threshold variants. PALISADE implements BFV, BGV, and CKKS in both threshold and non-threshold settings and DM/CGGI in a non-threshold setting. HElib implements BGV and CKKS, and has support for non-power-of-two cyclotomic rings, in addition to power-of-two cyclotomics. NFLlib[13] has an RNS implementation of BFV.

[4] https://github.com/openfheorg/openfhe-development
[5] https://palisade-crypto.org/
[6] https://github.com/snucrypto/HEAAN
[7] https://github.com/homenc/HElib
[8] https://github.com/tuneinsight/lattigo
[9] https://github.com/zama-ai/tfhe-rs
[10] https://github.com/data61/MP-SPDZ
[11] https://github.com/secretflow/secretflow
[12] https://github.com/microsoft/SEAL
[13] https://github.com/quarkslab/NFLlib

Comment Set #5: From A. Badawi, A. Alexandru, N. Genise, D. Micciancio, Y. Polyakov, Saraswathy R.V., and V. Vaikuntanathan

References

[ABGS22] Diego F. Aranha, Carsten Baum, Kristian Gjøsteen, and Tjerand Silde. Verifiable mix-nets and distributed decryption for voting from lattice-based assumptions. *IACR Cryptol. ePrint Arch.*, page 422, 2022.

[ACLS18] Sebastian Angel, Hao Chen, Kim Laine, and Srinath T. V. Setty. PIR with compressed queries and amortized query processing. In *IEEE Symposium on Security and Privacy*, pages 962–979. IEEE Computer Society, 2018.

[AJL+12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *EUROCRYPT*, volume 7237 of *LNCS*, pages 483–501. Springer, 2012.

[BBB+22] Ahmad Al Badawi, Jack Bates, Flávio Bergamaschi, David Bruce Cousins, Saroja Erabelli, Nicholas Genise, Shai Halevi, Hamish Hunt, Andrey Kim, Yongwoo Lee, Zeyu Liu, Daniele Micciancio, Ian Quah, Yuriy Polyakov, R. V. Saraswathy, Kurt Rohloff, Jonathan Saylor, Dmitriy Suponitsky, Matthew Triplett, Vinod Vaikuntanathan, and Vincent Zucca. Openfhe: Open-source fully homomorphic encryption library. In *WAHC@CCS*, pages 53–63. ACM, 2022.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325. ACM, 2012.

[BHP17] Zvika Brakerski, Shai Halevi, and Antigoni Polychroniadou. Four round secure computation without setup. In *Theory of Cryptography - 15th Intl. Conference, TCC 2017, Proceedings, Part I*, volume 10677 of *LNCS*, pages 645–677. Springer, 2017.

[BJL+21] Ahmad Al Badawi, Chao Jin, Jie Lin, Chan Fook Mun, Sim Jun Jie, Benjamin Hong Meng Tan, Xiao Nan, Khin Mi Mi Aung, and Vijay Ramaseshan Chandrasekhar. Towards the alexnet moment for homomorphic encryption: Hcnn, the first homomorphic CNN on encrypted data with gpus. *IEEE Trans. Emerg. Top. Comput.*, 9(3):1330–1343, 2021.

[Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer, 2012.

[CCMS21] Kelong Cong, Daniele Cozzo, Varun Maram, and Nigel P. Smart. Gladius: LWR based efficient hybrid public key encryption with distributed decryption. In *ASIACRYPT (4)*, volume 13093 of *LNCS*, pages 125–155. Springer, 2021.

[CGGI16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *ASIACRYPT (1)*, volume 10031 of *Lecture Notes in Computer Science*, pages 3–33, 2016.

[CHI+21] Megan Chen, Carmit Hazay, Yuval Ishai, Yuriy Kashnikov, Daniele Micciancio, Tarik Riviere, Abhi Shelat, Muthu Venkitasubramaniam, and Ruihan Wang. Diogenes: Lightweight scalable RSA modulus generation with a dishonest majority. In *IEEE Symposium on Security and Privacy*, pages 590–607. IEEE, 2021.

[CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic encryption for arithmetic of approximate numbers. In *ASIACRYPT (1)*, volume 10624 of *Lecture Notes in Computer Science*, pages 409–437. Springer, 2017.

[CMdG+21] Kelong Cong, Radames Cruz Moreno, Mariana Botelho da Gama, Wei Dai, Ilia Iliashenko, Kim Laine, and Michael Rosenberg. Labeled PSI from homomorphic encryption with reduced computation and communication. In *CCS*, pages 1135–1150. ACM, 2021.

Comment Set #5: From A. Badawi, A. Alexandru, N. Genise, D. Micciancio, Y. Polyakov, Saraswathy R.V., and V. Vaikuntanathan

[DKL+13] Ivan Damgård, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, and Nigel P. Smart. Practical covertly secure MPC for dishonest majority - or: Breaking the SPDZ limits. In *Computer Security - ESORICS 2013 - 18th European Symposium on Research in Computer Security. Proceedings*, volume 8134 of *LNCS*, pages 1–18. Springer, 2013.

[DM15] Léo Ducas and Daniele Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. In *EUROCRYPT (1)*, volume 9056 of *Lecture Notes in Computer Science*, pages 617–640. Springer, 2015.

[FV12] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptol. ePrint Arch.*, page 144, 2012.

[GH19] Craig Gentry and Shai Halevi. Compressible FHE with applications to PIR. In *TCC (2)*, volume 11892 of *Lecture Notes in Computer Science*, pages 438–464. Springer, 2019.

[GHS12] Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 850–867. Springer, 2012.

[HLHD22] Zhicong Huang, Wen-jie Lu, Cheng Hong, and Jiansheng Ding. Cheetah: Lean and fast secure two-party deep neural network inference. In *USENIX Security Symposium*, pages 809–826. USENIX Association, 2022.

[JVC18] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha P. Chandrakasan. GAZELLE: A low latency framework for secure neural network inference. In *USENIX Security Symposium*, pages 1651–1669. USENIX Association, 2018.

[KPR18] Marcel Keller, Valerio Pastro, and Dragos Rotaru. Overdrive: Making SPDZ great again. In *EUROCRYPT (3)*, volume 10822 of *LNCS*, pages 158–189. Springer, 2018.

[LMP22] Zeyu Liu, Daniele Micciancio, and Yuriy Polyakov. Large-precision homomorphic sign evaluation using FHEW/TFHE bootstrapping. In *ASIACRYPT (2)*, volume 13792 of *Lecture Notes in Computer Science*, pages 130–160. Springer, 2022.

[LMSS22] Baiyu Li, Daniele Micciancio, Mark Schultz, and Jessica Sorrell. Securing approximate homomorphic encryption using differential privacy. In *CRYPTO (1)*, volume 13507 of *LNCS*, pages 560–589. Springer, 2022.

[MBH22] Christian Mouchet, Elliott Bertrand, and Jean-Pierre Hubaux. An efficient threshold access-structure for rlwe-based multiparty homomorphic encryption. *IACR Cryptol. ePrint Arch.*, page 780, 2022.

[RST+22] Dragos Rotaru, Nigel P. Smart, Titouan Tanguy, Frederik Vercauteren, and Tim Wood. Actively secure setup for SPDZ. *J. Cryptol.*, 35(1):5, 2022.

[SPT+21] Sinem Sav, Apostolos Pyrgelis, Juan Ramón Troncoso-Pastoriza, David Froelicher, Jean-Philippe Bossuat, Joao Sa Sousa, and Jean-Pierre Hubaux. POSEIDON: privacy-preserving federated neural network learning. In *NDSS*. The Internet Society, 2021.

Comment Set #6: From F. Benhamouda, S. Halevi, H. Krawczyk, and T. Rabin

Public comments about the NIST IR 8214C ipd

"NIST First Call for Multi-Party Threshold Schemes" (initial public draft)

Fabrice Benhamouda[1] · Shai Halevi[1] · Hugo Krawczyk[1] · Tal Rabin[2]

April 10, 2023

Generic scope of submissions and organization into two categories (Section 3)
Another possible justification for a CAT2 submission could be its current use in interesting applications. (I.e., in addition to threshold-friendliness or advanced properties.) For example, the use of threshold PRF/OPRF in key management systems, decentralized identities and wallets can be used as a motivation for submitting a threshold-PRF protocol.

An important set of protocols to call out explicitly are threshold PRFs. These are used in many of the sub-categories that are listed in the call, including symmetric-key encryption/decryption, key derivation, key management, key distribution centers, randomness beacons, and more. Threshold Oblivious PRFs should be considered too as PRFs whose inputs are hidden from the Threshold PRF servers (as such they are applicable in the above cases as well as additional uses such as password protocols and privacy-preserving ticketing systems). These should be mentioned as examples of interesting CAT2 primitives.

Another threshold primitive that should be mentioned is a threshold implementation of a randomness beacon. Either in the keygen sub-category C2.5 or as a gadget C2.8.

The scope of the "gadgets" sub-category can be made clearer. For example, many protocols have sub-protocols that could stand on their own; would you consider those sub-protocols as "gadgets"? Some examples are Agreement on a Common Subset (e.g., agreeing on the set of qualifying dealers) or multi-recipient encryption (a-la-[BBS PKC'03]) that's used for a dealer to broadcast encrypted shares to many shareholders. If such sub-protocols are in-scope, then you should add some examples like that (both in section 3 and in 7.2.3).

3. Technical requirements (Section 5)
An important model distinction which is not mentioned in the draft call is whether the protocols are "robust", i.e., ensure guaranteed output delivery. For example, guarantee that a threshold signature system always generates requested signatures. This should be (at least) on par in terms of significance as the various adversary models that are called out in 5.2.3 (Active/Adaptive/Proactive). I.e., say that "the specification must consider the robustness and liveness guarantees of the protocol, e.g., whether they provide Guaranteed Output Delivery".

Similarly, the treatment of asynchronous channels must be elevated to at least as important consideration as the adversary model. It is not just "the pitfalls of deployment in environments with weaker guarantees (e.g., with asynchronous and unreliable channels". Submissions must discuss the guarantees (if any) that they provide in an asynchronous model. Note that the asynchronous setting is the most realistic in scenarios with large number of parties.

Details of subcategory CS.5 (Keygen) (Appendix A)
As you point out in A.5, DKG protocols are sometimes used to generate ephemeral randomness (e.g., in the case of ECDSA and Schnorr signatures). In this light, consider weakening the text in

[1] Algorand Foundation, USA. {fabrice,shai,hugo}@algorand.foundation
[2] University of Pennsylvania, PA, USA. talr@seas.upenn.edu

Comment Set #6: From F. Benhamouda, S. Halevi, H. Krawczyk, and T. Rabin

A.5. "*the focus on DKG is only on the generation of the private/secret keys and (when applicable) the public parameters that depend on them*". Maybe just add "long-term or ephemeral private/secret keys".

Comment Set #7: From H. Maji

Public comments about the NIST IR 8214C ipd "NIST First Call for Multi-Party Threshold Schemes" (initial public draft)

Hemanta K. Maji

1 Improving the Security of Shamir's Secret-Sharing Scheme

Additive and Shamir's secret-sharing schemes are the foundations of nearly all cryptography and privacy technologies. Their security is analyzed in a model where an adversary obtains a few secret shares, and the remaining ones remain hidden. However, side-channel attacks can leak partial information from all the secret shares. Security against such attacks is not ensured by typical security analysis of these secret-sharing schemes. The objective is to make these secret-sharing schemes secure against leakage attacks.

Notation. To illustrate such vulnerabilities, consider secret-sharing among n parties where the reconstruction threshold is k. For example, the threshold for the additive secret-sharing scheme is $k = n$. Suppose the secret-sharing schemes are over the prime field F_p, where the field's order is p, an odd prime. The secret is s, and the secret shares are $s_1, s_2, \ldots, s_n$, respectively, for the parties.

Attack model. Suppose the secret shares are stored in their binary representation. The length of the secret and secret shares represents the security parameter λ. For example, we know that $2^{\lambda-1} < p < 2^{\lambda}$ Consider the model where the adversary can leak m bits from each secret share. This is a very standard model for leakage [ISW03] and is also widely studied in *masking techniques*. Our objective is to ensure that the joint distribution of the leakage is (essentially) independent of the secret.

Representative Attack on the Additive Secret-sharing Scheme. Consider a very elementary attack on the additive secret-sharing scheme. The adversary obtains the *least significant bit* of each secret share. Define $\mathrm{LSB}(x) = 0$ if $x \in \{0, 2, \ldots, p-1\}$; and $\mathrm{LSB}(x) = 1$, otherwise. The adversary sees the joint distribution $(\mathrm{LSB}(s_1), \ldots, \mathrm{LSB}(s_n))$. We aim to ensure that the joint leakage distribution is independent of the secret; otherwise, this leakage poses a security threat.

Consider $n = 2$ and the secret $s = 0$. Note that the secret shares are as follows

$$(s_1, s_2) \in \Big\{(0,0), (1, p-1), (2, p-2), \ldots, (p-1, 1)\Big\}.$$

Then, the leakage is

$$(\mathrm{LSB}(s_1), \mathrm{LSB}(s_2)) \in \Big\{(0,0), (1,0), (0,1), \ldots, (0,1)\Big\},$$

respectively. Note that $\mathrm{LSB}(s_1) \oplus \mathrm{LSB}(s_2) = 1$ with probability $1 - 1/p$.

Comment Set #7: From H. Maji

Likewise, for secret $s = 1$, the secret shares are

$$(s_1, s_2) \in \left\{(0,1), (1,0), (2, p-1), \ldots, (p-1, 2)\right\}.$$

In this case, $\mathrm{LSB}(s_1) \oplus \mathrm{LSB}(s_2) = 1$ with probability $2/p$. Therefore, the *parity-of-the-parities* distinguisher can distinguish these two secrets with probability $1 - 3/p$, which is negligibly close to probability 1. This distinguisher was proposed in [MNP+21].

This attack extends to arbitrary n, and the distinguishing advantage of the parity-of-the-parities distinguisher is (roughly) $(2/\pi)^k$ [MNPW22, MNP+22], which is also optimal. Therefore, for any constant k, the additive secret-sharing scheme is constant-insecure. *Thus, this elementary attack completely breaks the additive secret-sharing scheme.*

Extension of the Attack to Shamir's Secret-sharing Scheme. Shamir's secret-sharing scheme inherits this vulnerability if one is not careful in choosing the modulus and the evaluation places. For a secret $s \in F_p$ and distinct evaluation places $X_1, \ldots, X_n \in F_p^*$, Shamir's secret-sharing scheme prepares the secret shares as follows. Pick a random polynomial $P(Z) \in F_p[Z]$ such that $\deg P(Z) < k$ and $P(Z = 0) = s$. The secret shares are $s_1 = P(X_1), \ldots, s_n = P(X_n)$.

Consider $p = 1 \mod k$. Consider the solutions of the equation $\zeta^k = 1$ in F_p^*. Denote these solutions by $\left\{1, \omega, \omega^2, \ldots, \omega^{k-1}\right\}$. Note that these solutions are a multiplicative subgroup of F_p^*. Suppose we have evaluation places $\{X_{i_1}, \ldots, X_{i_k}\} = \left\{\rho, \rho\omega, \ldots, \rho\omega^{k-1}\right\}$, for some $\rho \in F^*$. Observe that

$$s_{i_1} + s_{i_2} + \cdots s_{i_k} = ks.$$

Essentially, the secret-sharing scheme reduces to the additive secret-sharing scheme. The parity-of-that-parties attack will have $(2/\pi)^k$ advantage in distinguishing two secrets.

What is known? A Monte-Carlo Result. There are some results known for security. We know that, for every prime p, if one picks each evaluation place of Shamir's secret-sharing scheme uniformly at random, they are (exponentially) secure [MNP+21]. However, no algorithm is known to distinguish secure evaluation places from insecure ones. This problem is the standard "searching hay in a haystack" problem, which is very common in mathematics and computer science. There are indications that it is a highly challenging problem to solve.

Consequently, one needs to derandomize the result of [MNP+21]. For example, NIST cannot recommend choosing evaluation places at random and hoping the particular choice is secure. Note that even for $n = k = 2$, this derandomization problem is challenging.

Derandomization Results. We have derandomized this problem for $n = k = 2$ and $(n = 3, k = 2)$. We present an efficient algorithm that takes as input $X_1, \ldots, X_n$) and outputs whether this choice of evaluation places is secure against physical bit leakage.

We recommend using Mersenne prime modulus, i.e., the prime $p = 2^\lambda - 1$. In the sequel, we consider $n = k = 2$ to present the key technical innovations. Suppose an adversary leaks the i-th and the j-th least significant bits from secret share s_1 and s_2, respectively. In a Mersenne prime modulus, the leakage is equivalent to the LSB attack on Shamir's secret-sharing scheme with evaluation places $(2^i X_1, 2^j X_2)$. Consequently, an algorithm to determine security against the LSB attack suffices. For example, if the following set of evaluation places

$$\left\{(2^i X_1, 2^j X_2) \colon i, j \in \{0, 1, 2, \ldots, \lambda - 1\}\right\}.$$

Comment Set #7: From H. Maji

If all these evaluation places are secure against the LSB attack, then the scheme with evaluation places (X_1, X_2) is secure against all physical bit leakage attacks.

Now, consider determining the security of the evaluation places (X_1, X_2). This problem reduces to determining the orthogonality of the following two functions.

$$\begin{aligned} &\text{(Function 1)} \qquad Y = \mathrm{sgn}_p(X_1 \cdot X) \\ &\text{(Function 2)} \qquad Y = \mathrm{sgn}_p(X_2 \cdot X), \end{aligned}$$

where

$$\mathrm{sgn}_p(x) \;:=\; \begin{cases} +1, & x \mod p \in \{1, 2, \ldots (p-1)/2\} \\ -1, & \text{otherwise.} \end{cases}$$

We call these functions "signs of lines." The following sum determines the orthogonality

$$\sum \;:=\; \sum_{x \in F} \mathrm{sgn}_p(X_1 \cdot x) \cdot \mathrm{sgn}_p(X_2 \cdot x).$$

This sum is proportional to the following integral involving "square wave functions."

$$I \;:=\; \int_0^1 \mathrm{sgn}(\sin(2\pi \cdot X_1 t)) \cdot \mathrm{sgn}(\sin(2\pi \cdot X_2 t))\; \mathrm{d}t.$$

Here $\mathrm{sgn}(\cdot)$ is the real-values sign function. In an unpublished work [MNPY23], we present the algorithm that identifies secure evaluation places. Evaluation places deemed "not secure" are, in fact, insecure. This work also identifies new attacks on these secret-sharing schemes.

Conclusions. Physical bit leakage is an elementary leakage model – simpler the attack model, the greater the security threat. Securing against these side-channel attacks is essential. Such attacks completely break the additive secret-sharing scheme. One needs to be extremely careful in choosing the modulus and the evaluation places to be secure against this attack class. Although randomly chosen evaluation places are secure with high probability, no algorithm is known to identify the secure ones. More research on this problem must be directed from the cryptographic research community.

Comment Set #7: From H. Maji

References

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Heidelberg, Germany. doi:10.1007/978-3-540-45146-4_27. 1

[MNP+21] Hemanta K. Maji, Hai H. Nguyen, Anat Paskin-Cherniavsky, Tom Suad, and Mingyuan Wang. Leakage-resilience of the shamir secret-sharing scheme against physical-bit leakages. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part II*, volume 12697 of *Lecture Notes in Computer Science*, pages 344–374, Zagreb, Croatia, October 17–21, 2021. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-77886-6_12. 2

[MNP+22] Hemanta K. Maji, Hai H. Nguyen, Anat Paskin-Cherniavsky, Tom Suad, Mingyuan Wang, Xiuyu Ye, and Albert Yu. Tight estimate of the local leakage resilience of the additive secret-sharing scheme & its consequences. In Dana Dachman-Soled, editor, *3rd Conference on Information-Theoretic Cryptography, ITC 2022, July 5-7, 2022, Cambridge, MA, USA*, volume 230 of *LIPIcs*, pages 16:1–16:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITC.2022.16. 2

[MNPW22] Hemanta K. Maji, Hai H. Nguyen, Anat Paskin-Cherniavsky, and Mingyuan Wang. Improved bound on the local leakage-resilience of shamir's secret sharing. In *IEEE International Symposium on Information Theory, ISIT 2022, Espoo, Finland, June 26 - July 1, 2022*, pages 2678–2683. IEEE, 2022. doi:10.1109/ISIT50566.2022.9834695. 2

[MNPY23] Hemanta K. Maji, Hai H. Nguyen, Anat Paskin-Cherniavsky, and Xiuyu Ye. Security of shamir's secret-sharing against physical bit leakage: Secure evaluation places, 2023. https://www.cs.purdue.edu/homes/hmaji/papers/MNPY23.pdf. 3

Comment Set #8: From J. Katz, C. Komlo, X. Meng, and N. Smart

Public Comments about the NIST IR 8214C ipd

"NIST First Call for Multi-Party Threshold Schemes" (initial public draft)

Jonathan Katz* Chelsea Komlo*† Xianrui Meng* Nigel Smart‡

April 10, 2023

1 Generic Scope of Submissions

The scope (i.e., number of subcategories) of the call is very broad. This overly broad scope runs the risk of dividing the attention of the research community both during the submission phase and the subsequent analysis phase. NIST may miss out on some submissions simply because submitters don't have the time to contribute to submissions in multiple categories.

Some subcategories are also much less mature than others. (For example, there has been much more work on threshold signatures than on threshold key agreement.) It may therefore be premature to ask for submissions in certain subcategories at this time.

2 Technical Requirements

2.1 KeyGen

For subcategories C1.1–C1.4 and C2.1–C2.4, submissions should explicitly note the required initial state (e.g., key shares) of the parties, as well as how that initial state is assumed to be established for purposes of the security analysis. This would give submissions the flexibility to assume a trusted dealer (that could, in turn, be realized by an appropriate KeyGen protocol submitted in subcategories C1.5/C2.5), and/or to include their own specialized KeyGen protocol as part of the submission. This will help to facilitate analysis and security proofs, and also allow for better integration with submissions to subcategories C1.5/C2.5.

2.2 Adaptive/Proactive Security

While adaptive security on long timescales makes sense—and the example in lines 912–915 of a "secure" protocol that is trivially broken by an adaptive adversary is well-motivated—the formal model of adaptive security that allows an adversary to instantaneously corrupt parties may be too strong for some practical applications, and achieving that notion of adaptive security can require modifications that severely impact efficiency (as noted in lines 910–912). Thus, while a proof of

*Dfns Labs. **Email:** {jkatz,chelsea,xm}@dfns.co. Work of Jonathan Katz was not part of his University of Maryland duties or responsibilities.

†University of Waterloo.

‡KU Leuven and Zama. **Email:** nigel.smart@kuleuven.be.

Comment Set #8: From J. Katz, C. Komlo, X. Meng, and N. Smart

adaptive security may be a positive point in favor of a scheme, NIST should consider making adaptive security optional for submissions, and/or allowing submitters to include both a statically secure and an adaptively secure version of a scheme. For schemes that are not proven adaptively secure, submitters should still be encouraged to provide a partial analysis against adaptive attacks and/or to provide an analysis of the scheme against known adaptive attacks.

While proactive security is an important property, mechanisms for key refresh that achieve proactive security are generally orthogonal to the threshold protocols envisioned for subcategories C1.1–C1.4 and C2.1–C2.4. (In particular, key refresh would typically be run in between invocations of, e.g., a threshold signing protocol.) Thus, techniques to achieve key refresh/proactive security may be better as stand-alone submissions in subcategories C1.5/C2.5 (only).

2.3 Concrete Security Bounds

When giving proofs of security, submitters should be encouraged to include concrtete security bounds and set parameters of their schemes accordingly. It should be clearly noted when proofs are only asymptotic and not tight.

2.4 Communication Environments

We have observed that protocol implementers are often unsure about how to implement a "broadcast channel" in practice, and there is no general guidance available about how protocols that rely on a broadcast channel should be implemented in a point-to-point network. We therefore encourage NIST to require submissions to either explicitly state that their security analysis assumes a broadcast channel (and then suggest how such a channel should be implemented), or otherwise provide a protocol specification and proof of security in the point-to-point communication model (i.e., without the assumption of a broadcast channel).

2.5 Threshold Profiles

Line 942 appears to be the only place in the document where the participation threshold k is mentioned. Although in many schemes $k = f + 1$, that need not be the case.

2.6 Modeling State: Deletion and Concurrency

In multi-round protocols, it will be helpful to explicitly describe the state being maintained by the protocol between rounds. As part of this requirement, it would also be useful for schemes to specify when certain values must be deleted (as opposed to assuming the implementer knows when to delete based on context).

Protocols that rely on a unique session identifier (sid) for security, especially in a concurrent setting, should make the sid explicit in the protocol description and state any assumptions made about the sid in the security analysis.

NIST should give explicit recommendations for modeling state-keeping, deletion, and concurrency in order to ensure more consistency across submissions.

3 Other Comments (Typos)

- line 421: missing space ("NISTpublication").

Comment Set #8: From J. Katz, C. Komlo, X. Meng, and N. Smart

- line 957, the first column should be "$< 1/2$" (or "$1/3 \leq f/n < 1/2$").
- line 1312: "bot" should be "not."
- line 1437: Did you mean "symmetric"?
- line 1703: the requirement for p, q to be "randomly generated" does not match what is said in lines 1134–1135 about acceptability of ensuring $p = q = 3 \bmod 4$.

Comment Set #9: From G. Alpáar, L. Botros, A. de la Piedra, and M. Venema

Public comments about the NIST IR 8214C ipd

"NIST First Call for Multi-Party Threshold Schemes" (initial public draft)

Greg Alpár[1,2], Leon Botros[2], Antonio de la Piedra[3], and Marloes Venema[2,4]

[1] Open Universiteit, Heerlen, the Netherlands
[2] Radboud University, Nijmegen, the Netherlands
[3] Kudelski Security Research Team, Cheseaux-sur-Lausanne, Switzerland
[4] University of Wuppertal, Wuppertal, Germany

April 6, 2023

1 Comment on Appendix A.6

Currently, Section 7.2.1 and Appendix 4.6 introduce the notions of identity-based and attribute-based encryption as suitable primitives for thresholdization. However, apart from some high-level requirements, it might not be entirely clear what kinds of schemes this call welcomes. Most notably, we wanted to address this, because there is a body of work in the field of attribute-based encryption that addresses the deployment of multiple parties (herein called "authorities"). However, not all terminology and proof techniques in this particular subfield match those of the multi-party community. Furthermore, this subfield contains several different functionalities and security models, which differ subtly in ways that we do not see in the more general multi-party computation paradigm. We were therefore wondering whether it would be appropriate to include these different functionalities and security models in the call. We explain the nuances among these security models more clearly below.

1.1 Ciphertext-policy attribute-based encryption

Ciphertext-policy attribute-based encryption (CP-ABE) [1] is a type of attribute-based encryption (ABE) [11] that associates the ciphertexts with access structures (or: policies) and the keys with attributes. (In contrast, key-policy ABE (KP-ABE) [5] associates the keys with policies and the ciphertexts with attribute sets.) A key can decrypt a ciphertext if its associated attributes satisfy the access structure of the ciphertext. In this way, we can enforce attribute-based access control on a cryptographic level. Crucially, for this to work securely, if several users have keys for sets that do not satisfy the ciphertext policy, they should not be able to decrypt the ciphertext. This security aspect is also called collusion resistance.

Comment Set #9: From G. Alpáar, L. Botros, A. de la Piedra, and M. Venema

1.2 Multi-authority attribute-based encryption

A type of attribute-based encryption that follows, to some extent, the spirit of multi-party computation is called multi-authority attribute-based encryption (MA-ABE) [2]. Instead of deploying a single authority to generate secret keys for the users, multiple authorities are deployed. By doing this, the system becomes more flexible, as it can naturally support multiple-domain settings, and it is more secure, because there is no single trusted authority. In most multi-authority schemes, each authority in the system manages a unique set of attributes. Furthermore, the security model allows for corruption of authorities, as long as, together, they cannot decrypt the challenge ciphertext. In particular, with multi-authority CP-ABE, we can associate the ciphertext with an access structure using attributes that are managed by different authorities. Essentially, we can ensure in this way that the access structure associated with the ciphertext is determined during encryption.

Compared to more traditional threshold schemes, ABE has various types of functionalities and their associated security models. In fact, some MA-ABE schemes have some interesting properties that seem to more flexible and fine-grained than more traditional thresholdized schemes, but it can also be more restricted. On a high level, we observe that there exist three types of multi-authority ABE: thresholdized, distributed and decentralized. These have various trade-offs in flexibility and security, but each may provide advantages over single-authority ABE (typically at the cost of some efficiency).

Distributed and decentralized ABE. The terms distributed and decentralized typically consider the setting in which the authorities manage unique sets of attributes [13] (although this is not a requirement for the security of such schemes [12]). The goal of distribution and decentralization is to ensure that the authorities do not need to fully trust one another to manage access control securely (e.g., in multiple-domain settings). As long as the malicious authorities do not manage a large enough set of attributes that satisfies the access structure of the ciphertext, it cannot be decrypted.

The difference between decentralized and distributed is in in the level of autonomy and independence that the user has. More specificially, the difference is in whether the decrypting user would need a key from each authority associated with the access structure or simply only keys from those authorities for which they have attributes that satisfy the access structure. For instance, consider the access structure “doctor at Mayo Clinic” OR “doctor at Johns Hopkins Hospital”. We assume that the first attribute is managed by an authority associated with the Mayo Clinic and the second by an authority associated with the Johns Hopkins Hospital. Then, distributed ABE would require the decrypting user to have keys from both hospitals, while decentralized ABE would not require this. Instead, decentralized ABE allows users that have obtained a key for the “doctor” attribute from either of the two hospitals to decrypt. Because of this latter property, it is actually more difficult to achieve decentralization than it

Comment Set #9: From G. Alpáar, L. Botros, A. de la Piedra, and M. Venema

is to achieve distribution. Therefore, relatively few schemes have this property: [7,8,10,4,12].

One of the main reasons why achieving distribution and decentralization is difficult is because it is challenging to achieve collusion resistance across different authorities [2,7]. In particular, users that have keys from different authorities should not be able to combine those keys. For instance, suppose that one user has a key for the attribute "doctor" generated by the Mayo Clinic, and another user (who is not a doctor) has a key for the attribute "employee" generated by the Johns Hopkins Hospital, then they should not be able to decrypt a ciphertext with the policy "doctor at the Mayo Clinic" AND "employee at the Johns Hopkins Hospital" together. Requiring this type of collusion resistance is also one of the reasons why multi-authority schemes are typically much different from their single-authority counterparts: either in their structure [7,10] or in the level of expressivity of the access structures [2,3].

Thresholdized ABE. In addition to the notions of distributed and decentralized ABE (as proposed in [13]), we would also like to propose the notion of thresholdized, which solely considers the setting in which all authorities generate keys for the same attributes, and only the public-key material is shared among authorities. This requires a lower bar of functionality compared to decentralization, but increases the security of ABE compared to its single-authority variants, because the key material is not stored on one server, but rather on multiple servers. For instance, with a t-out-of-n thresholdized ABE scheme, a user would require keys from at least t servers of the n listed servers to decrypt a ciphertext, rather than of one single server. Compared to distributed schemes, it might enable more fine-grained access structures. It might also lower the bar with respect to security compared to decentralized schemes, because we might not require security against collusion among different authorities. (However, note that decentralized ABE implies thresholdized ABE.)

To summarize, thresholdized, distributed and decentralized ABE would distinguish them from one another in the following way. If a multi-authority scheme allows the various authorities to manage different attributes, and requires users to only interact with those authorities that manage their attributes, then we call the scheme decentralized. If users have to interact with all authorities to decrypt (regardless of whether these authorities manage their attributes), then we call the scheme distributed. If the scheme does not allow the authorities to manage different attributes, but requires users to only interact with those authorities that manage their attributes, then we call the scheme thresholdized.

2 Comment on Section 5.6 T6: Building Blocks

In lines 988–994, various examples of building blocks for threshold cryptography constructions are given. We would like to note several building blocks or gadgets related to ABE that could also be included as examples of building blocks.

Comment Set #9: From G. Alpáar, L. Botros, A. de la Piedra, and M. Venema

In particular, ABE constructions use access structures, which are in specific instantiations often represented as access trees [5] and linear secret-sharing scheme (LSSS) matrices [6]. Possibly, ABE constructions can benefit from more efficient representations of such access structures, which may be of interest to the call as well.

3 Comment on Section 3: Call and Scope for Submissions

According to line 390: "Each submission should include a security characterization, a technical description, an open-source reference implementation, and a performance evaluation."

We would like to note that, for attribute-based encryption in particular, existing methodologies for analyzing the performance of pairing-based ABE constructions have been recently proposed [9]. We were wondering if it would be helpful for potential submitters to include it as reference material. Specifically, this methodology was proposed to benchmark and compare schemes more fairly by optimizing them with respect to the same design goal. For example, a scheme that was implemented to optimize the key generation (possibly at the expense of the other algorithms) may perform much differently than a scheme that was optimized for the encryption algorithm. Had they been optimized with respect to the same goal, they might have compared differently. By being transparent about such design goals, it becomes clearer which schemes are the most suitable choices for certain use cases.

References

1. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: S&P. pp. 321–334. IEEE (2007)
2. Chase, M.: Multi-authority attribute-based encryption. In: Vadhan, S.P. (ed.) TCC. LNCS, vol. 4392, pp. 515–534. Springer (2007)
3. Chase, M., Chow, S.S.M.: Improving privacy and security in multi-authority attribute-based encryption. In: Al-Shaer, E., Jha, S., Keromytis, A.D. (eds.) CCS. pp. 121–130. ACM (2009)
4. Datta, P., Komargodski, I., Waters, B.: Decentralized multi-authority abe for nc^1 from computational-bdh. J. Cryptol. **36**(2), 6 (2023)
5. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) CCS. ACM (2006)
6. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. Cryptology ePrint Archive, Report 2006/309 (2006)
7. Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: EUROCRYPT. pp. 568–588. Springer (2011)
8. Okamoto, T., Takashima, K.: Decentralized attribute-based signatures. In: Kurosawa, K., Hanaoka, G. (eds.) PKC. LNCS, vol. 7778, pp. 125–142. Springer (2013)
9. de la Piedra, A., Venema, M., Alpár, G.: ABE squared: Accurately benchmarking efficiency of attribute-based encryption. TCHES **2022**(2), 192–239 (2022)

Comment Set #9: From G. Alpáar, L. Botros, A. de la Piedra, and M. Venema

10. Rouselakis, Y., Waters, B.: Efficient statically-secure large-universe multi-authority attribute-based encryption. In: Böhme, R., Okamoto, T. (eds.) FC. LNCS, vol. 8975, pp. 315–332. Springer (2015)
11. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT. LNCS, vol. 3494, pp. 457–473. Springer (2005)
12. Venema, M.: A practical compiler for attribute-based encryption: New decentralized constructions and more. Cryptology ePrint Archive, Paper 2023/143 (2023)
13. Venema, M., Alpár, G., Hoepman, J.: Systematizing core properties of pairing-based attribute-based encryption to uncover remaining challenges in enforcing access control in practice. Des. Codes Cryptogr. **91**(1), 165–220 (2023)

Comment Set #10: From G. Seghaier, and J. Doget

Public comments about the NIST IR 8214C ipd
"NIST First Call for Multi-Party Threshold Schemes" (initial public draft)

Gilles Seghaier (Astran) – Julien Doget (Astran)

April 10, 2023

1. Generic scope of submissions and organization into two categories (Section 3)

The current comment is related to a Quantum-resistant solution falling in the Category 2 (Cat2) based on AONT, HE and Secret Sharing. We would like to add AONT as a building block (Q1) as stated in the NIST.IR.8214A.

The related solution is falling in the Cat2, for primitives not specified by NIST, but which are friendlier (more amenable) to the threshold paradigm, have enhanced functional features, and are based on different cryptographic assumptions.

The current comment is based on a deployed application making use of threshold schemes, despite lack of standards (or NIST standards). The development of such new standards can promote best practices and interoperability in a field with already concretely demonstrated use cases.

Some parts of the solution are patented but they are available for licensing (and already in use by private and government organizations). Indeed, the first version of this solution is already generally available on a distributed multi-cloud environment.

This architecture relies on a threshold interface that differs with the ones listed and presented in the original NIST.IR.8214A. (section 2.3 Modes of Input/Output interface - Figure2. Several threshold interfaces (and one non-threshold case)).

Usually cloud users assume (erroneously) the cloud providers to be secured, but cloud providers seem to be honest but curious in many situations. In our solution, we focus on cloud storage in terms of confidentiality, integrity and availability, based on a proxy that orchestrates the threshold scheme, but in a different way from the non-threshold scheme presented in the NIST.IR.8214A, in order to preserve data confidentiality against the proxy.

At the moment, confidentiality and integrity can be achieved using classical encryption but at the very high cost of key management client-side. Availability is let to the cloud providers.

We present a novel approach to brings confidentiality, integrity and availability without needs of key management nor any long-term parameters storage.

The solution relies on a multi-cloud approach managed by a proxy storage server. The multi-cloud approach allows redundancy and availability (and performance) and the proxy acts as the orchestrator and a unique entry point to the client.

Technically it relies in one side on secret sharing to bring integrity, availability (k amongst n scheme) and confidentiality with regards to the cloud providers. In the other side it relies on homomorphic encryption to bring confidentiality with regards to the proxy itself.

In details, a proxy server acts as the unique entry point to allow plug and play compatibility with classical storage service (e.g. AWS S3). The role of this proxy is to spread the data over several cloud

Comment Set #10: From G. Seghaier, and J. Doget

providers using a secret sharing scheme (such as Shamir Secret Sharing scheme or AONT-RS) and hide the multi-cloud configuration to the client.

In details, a (k,n) secret sharing scheme is an algorithm which splits the data into n shares in a way such that every subset of k shares permits the reconstruction of the data whereas every subset of l < k shares does not bring any information on the underlying data.

This fragmentation method can be used to overcome some (up to n-k) cloud providers failure or for instance to overcome overseas surveillance by choosing less than k cloud providers in a specific country.

In this scenario, the user sends data through the proxy server which applies the secret sharing and the data is protected against lower than k cloud collusion.

Nevertheless, the proxy still has access to transmitted data. To avoid that we use an innovative combination of AONT and Homomorphic Encryption.

AONT (All Or Nothing Transform) is a transformation which ensures that every subset of the data does not bring any information on the data. A realization of such a transformation (as in [ref AONT-RS]) can rely on a symmetric cipher with a random key, the data is ciphered, the random key is masked by the ciphertext and append to it. In this case one needs the whole data to unmask the key and decipher the data.

Homomorphic encryption is a kind of encryption which allows computation on ciphertext (for example let f(x) and f(y) be the cyphertext of x and y, we can compute f(x+y) = f(x)+f(y) without need of deciphering). Whereas homomorphic encryption schemes are available ([ref BGV] there are very cumbersome to use.
In the following we show how to combine AONT with homomorphic scheme to leverage the cost and we propose a plug and play implementation based on AWS S3.

To bring confidentiality with regards to the proxy, the homomorphic encryption must be done client side. To avoid long term keys, the storage providers must decrypt the data (which remain secure thanks to the secret sharing scheme which has been apply homomorphically).
A common temporary key must be priorly set between the client and all the cloud providers. This key must not be known by the proxy. This key establishment is out of the scope of this document.

Finally:
Client side:
- the data is transformed with an AONT scheme then split into 2 parts, a fixed-length one (e.g. 128 bits) and the remaining data. if the data is smaller than the fixed length, a padding occurs
- the remaining part is sent as-is to the proxy server
- the fixed-length part is encrypted with a HE schemes
- the HE encrypted fixed part is send to the proxy

Proxy side:
- the remaining data is shared to the cloud providers with a secret sharing scheme
- the encrypted data is shared to the cloud providers with a secret sharing scheme applied homomorphically

Providers side:
- the remaining data shares are stored as-is
- the homomorphically encrypted shared are decrypted then stored

Comment Set #10: From G. Seghaier, and J. Doget

An implementation over AWS S3 API is available. AONT scheme is based on AES256, HE is BGV, Secret sharing is RS and SSS, Client-side computation are done by overwriting the Signature Module of she S3 SDK

2. Requirements and recommendations for submissions (Section 4)

Comment regarding Src2. Is licensed as open-source:
Would it be possible to submit solutions that are copyrighted or under a double license mode (copyleft + commercial license)?

3. Technical requirements (Section 5)

5.2.1. T2.1: Participants
As stated in comment on section 3, we would like to present a solution where the client needs to transform the data before requesting or after getting the reply from the proxy. This use case does not fall into any of the existing threshold interfaces mentioned.

4. Primitives and threshold modes in Cat1 (Section 6)

As stated in comment on section 3, would it be possible to consider adding the AONT as a Cat1 primitive?

5. Subcategories in Cat2 (Section 7)

As stated in comment on section 3, would it be possible to consider adding the AONT as a Cat2 C2.8 primitive (gadget non-threshold)?

6. Details of subcategory ___ (Appendix A)

N/A

7. Submission checklists (Appendix B)

N/A

8. Diverse editorial feedback

N/A

9. Other comments

N/A

Comment Set #11: From a. shelat, J. Doerner, E. Lee, and Y. Kondi

From: Shelat, Abhi
Sent: Monday, April 10, 2023
To: nistir-8214C-comments
Cc: Jack Doerner; Eysa Lee; Yashvanth Kondi
Subject: Feedback on NIST IR 8214C ipd

Hello NIST committee,

Overall we are very excited by NIST Internal Report NIST IR 8214C ipd, entitled "NIST First Call for Multi-Party Threshold Schemes." It is a very well written document that clearly enumerates the concerns that a standardization body should raise for this problem domain.

Below, we only have 2 small comments:

(1) Line 659–662: " The code (and its instructions) should be designed to allow for a compilation and execution of the submitted implementation on top of a Linux Ubuntu Desktop 22.04.1 long-term support (LTS) operating system running installed in platform PF1, without requiring software download from external sources."

We would appreciate clarification on this minor point. While certain compilers or interpreters (gcc, python) may be available by default on a 22.04LTS image, others such as rustc, golang etc, as well as some system libraries such as openssl, etc. may not be available by default. Does this mean that the project should include all of its dependencies, including compiler and external libs? Or do you intend for a more general setup than is suggested by these lines?

Additionally, do you have any guidance on using “nightly” features of certain programming languages, e.g., Rust, versus the declared “stable” versions of those languages?

(2) Line 695: "KAT set: A set of “known answer-test” (KAT) values, to aid in correctness verification of the execution of the protocol."

A threshold signing application may require each of the participants to use many random bits during any operation. It is not clear how to specify a KAT for a more complicated, multi-party, multi-round protocol.

We look forward to preparing a submission later this year in collaboration with several other groups who have similar opinions about threshold schemes.

Best,

abhi

Comment Set #12: From T. Ruffing

From: Tim Ruffing
Sent: Tuesday, April 11, 2023
To: nistir-8214C-comments
Subject: Feedback about the draft

Hello,

Sorry for being a few hours late. I hope my feedback is still useful.

Robustness:

I think the call should also consider robustness, i.e., the ability to finish a threshold protocol in the presence of an attacker controlling the network and/or a number of participants. I consider robustness a security property (exactly because it holds in the presence of an attacker), and I feel it's very relevant for threshold cryptography.
Since robustness is "functionality/correctness against an attacker", some form of robustness will be required in every application, though the specific degree depends on the application, e.g., it's a bit less important for a one-time key generation, and it's more important for a threshold signing or decryption scheme that is supposed to run every few minutes.

So proposals should include a description of their robustness guarantees, and how they break down if various assumptions (number of honest parties, network assumptions such as broadcast) are violated. "Breaking down" can also be "soft", e.g., it may be possible for an attacker to delay the protocol but not prevent it from running entirely. Assumptions for robustness may (and usually will) differ from assumptions for other security properties, e.g., robustness is not possible against a malicious network/coordinator, and the maximum tolerable number of malicious nodes may be different for robustness than for other properties.

Note: Maybe a better name can be found. The term "robustness" is very generic. Even though this is a common term for threshold primitives, this term has also been used in the cryptography literature to denote entirely different properties. But I don't have a concrete suggestion that I think is better than "robustness". One candidate is "liveness" but I don't think it's good. It applies rather to systems. (For example, a robust threshold signing protocol can help ensure that a system using this protocol is live.)

Broadcast:

It's good that the draft mentions broadcast as possible requirement. Maybe it should emphasize more that there are many different forms and flavors of reliable broadcast (found in the distributed systems literature), and proposals should specify the exact properties that except from a broadcast mechanism. I think my comment applies more or less to network assumption in general (e.g., also to synchrony assumptions or message delivery guarantees), but things are particularly complex when it comes to broadcast.

Best,
Tim Ruffing