Compilation of Feedback to NIST-MPTC Call 2021a

NIST Multi-Party Threshold Cryptography Project[1]

Updated: August 25, 2022

On July 02, 2021, the NIST multi-party threshold cryptography project issued a call for feedback on selected topics of criteria for multi-party threshold schemes, as it may be useful to support a future call for proposals of those schemes. For that purpose, the call provided a brief note about each of the following six selected topic: (i) scope of proposals; (ii) idealization of security; (iii) security vs. adversaries; (iv) system model; (v) threshold profiles; (vi) building blocks. This document compiles the original call (4 pages), the optional templates provided for feedback, and a printout of the six received comments (along with some related emails).

Contents

[1] Webpage: https://csrc.nist.gov/projects/threshold-cryptography; email address: threshold-MP-call-2021a@nist.gov.

Item 1: Call 2021a for feedback (2021-July-02)

Call 2021a for Feedback on Criteria for Threshold Schemes

NIST Multi-party Threshold Cryptography

2021-July-02: https://csrc.nist.gov/projects/threshold-cryptography

Please send comments to threshold-MP-call-2021a@nist.gov by September 13, 2021.

In a **multi-party threshold scheme**, the secret key needed to operate a cryptographic primitive is "secret shared" across n parties. The operation (e.g., signing, decryption) is then distributively executed, while the key remains secret even if f (the threshold number of) parties are corrupted.[1]

The NIST multi-party threshold cryptography (MPTC) project has received useful feedback on threshold schemes. This has included various comments about draft NIST internal reports (IR) 8214 and 8214A, and presentations in the NTCW 2019 and MPTS 2020 workshops. Currently, a new IR, 8214B, is being prepared to specify **criteria** to support a future call for proposals of **multi-party threshold schemes**. The subsequent evaluation of those proposals will serve as a basis to support the development of **guidelines and recommendations** (G&R) about threshold schemes. A draft of IR 8214B, to be open for public comments, is planned for late 2021.

In advance to draft IR 8214B, this document is an earlier **call for feedback on selected topics of criteria**. The following paragraphs, with some level of informality in the midst of nuanced notions in multi-party / distributed systems, advance a preparatory positioning about six topics. These are open to improvement based on feedback, which is welcome from expert stakeholders. It is particularly useful to hear about the benefits of being more stringent or loose in any of the topics. It is also useful to hear about reference approaches to thresholdization of concrete primitives. For timely consideration, comments should be compiled in a **p**ortable **d**ocument **f**ormat (PDF) file, with up to six pages, letter size, and sent by email, by September 13, 2021. Voluntary template files for feedback are attached in .odt (**o**pen **d**ocument **t**ext) and .tex (La**TeX**) formats.

1. Scope of proposals. The future call for proposals will be intended to gather expert submissions of concrete threshold schemes for primitives that are *interchangeable* (in the sense of IR 8214A, Section 2.4) with[2] ECDSA, EdDSA, RSA signing/decryption, RSA keygen, AES, and ECC-based key agreement.[3] After an evaluation period, and possibly various stages for tweaks,

[1] In this document, f denotes the *corruption* threshold (maximum number of tolerated corruptions) with respect to key hiding. Different properties can have different associated thresholds. In complement, a *participation* threshold denotes the minimum number k (with $k > f$) of parties needed to generate the intended output, e.g., a "3-out-of-5 threshold signature" can be such that any subset of $k = 3$ honest parties can generate a signature, whereas $k - 1$ cannot.

[2] **Legend:** ECDSA: **E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm. EdDSA: **E**dwards **C**urve **D**igital **S**ignature **A**lgorithm. RSA: **R**ivest-**S**hamir-**A**dleman. Keygen: **key-gen**eration. ECC: **E**lliptic **C**urve **C**ryptography. AES: **A**dvanced **E**ncryption **S**tandard. PreQC: **pre-q**uantum **c**ryptography; PQC: **p**ost-**q**uantum **c**ryptography.

[3] Proposals of distributed (threshold) keygen for ECC-based schemes and AES can be associated to the proposals of corresponding threshold signing/encryption/decryption. Proposals of RSA keygen will be considered separately from proposals of RSA signing/decryption, although still taking into account the need for interoperability.

Item 1: Call 2021a for feedback (2021-July-02)

new proposals, and filtration of suitable approaches, the NIST MPTC project may produce differentiated G&R on threshold schemes for each primitive, and possibly also focused on useful building blocks. To better serve the evaluation process and the elaboration of G&R documentation, the future call is expected to request that submitted proposals include a reference open-source implementation, and a disclosure of known applicable patent claims. Additional notes:

- **Interchangeability:** In the scope of ECC-based signatures, it is particularly relevant to analyze the applicability of, and tradeoffs between, probabilistic and deterministic threshold schemes with interchangeable signature verification, possibly leading to differentiated G&R.
- **PreQC:** The utility of threshold schemes for preQC primitives may be affected in the future by advances in quantum computation, possible deprecation of existing standards, and developments of new PQC standards.
- **PQC:** While the present scope does not include the thresholdization of primitives in evaluation by the NIST-PQC project, the experience to be gained with the current scope should be useful for future considerations about threshold PQC. Also, post-quantum security can be considered in gadgets and in the thresholdization of AES.

2. Idealization of security. A proposal of threshold scheme must be supported on a **simulation**-based and/or a **game**-based security formulation. This entails defining an ideal **functionality** (as in the ideal-real simulation paradigm, e.g., in the universal composability framework) and/or an idealized adversarial **game** (or set of games). The proposal must discuss whether/which known useful properties are not captured by the idealized security formulation.[4] The proposal must include a proof ("security proof") that, in a suitable adversarial context (see item 3), the proposed threshold scheme satisfies the proposed security formulation, e.g., by showing "emulation" of the ideal functionality, or by showing that a non-negligible adversarial advantage in the game implies breaking an assumption. The analysis must identify the required cryptographic assumptions and any possibly-idealized trusted components in the setup or operations. It must also consider the (in)security consequences of foreseen real instantiations of the setup and ideal components.

3. Security vs. adversaries.

Security with respect to how an adversary corrupts up to f parties:

(a) **Active.** Proposed threshold schemes must aim for **active security** (i.e., against active corruptions, which enable corrupted parties to "maliciously" deviate from the protocol), as opposed to *passive* only.

(b) **Adaptive.** There is a strong preference for considering threshold schemes that achieve **adaptive security** (i.e., against adaptively chosen corruptions), as compared to *static* only,

[4] For example, even though availability is a generically desirable property, a security formulation may on purpose specify that an adversary is allowed to abort protocol executions, so that the formulated security notion is achievable. As another example (now of an unsuitable formulation), a sole requirement of hiding and binding for a commitment scheme would not suffice for a use (e.g., committing bids in an auction) that would also require a non-malleability property.

Item 1: Call 2021a for feedback (2021-July-02)

with respect to critical safety properties (e.g., unforgeability). There is also a need for **practical feasibility**. Feedback is welcome on: (i) security formulations and reference approaches that simultaneously enable both qualities; and/or (ii) possible acceptable tradeoffs.

(c) **Proactive.** Threshold schemes should be compatible with mechanisms of **proactive** (and reactive) recovery, which attempt to recover possibly corrupted parties back to an uncorrupted state. This is especially important to better handle a persistent **mobile** adversary than continuously attempts to corrupt more parties. With respect to refreshing secret shares, the solutions can be based on a modularized phase of secret-resharing (see item 6), though also specifying the needed conditions (e.g., requirement of some initial/final agreement by a qualified quorum) for its integration.

To achieve security against the mentioned types of corruption, proposals of threshold schemes can consider security formulations with reduced liveness/availability, such as "**security with abort**".[5] This compromise is known to be necessary in some settings, depending on f/n and on the assumptions about the communication network (e.g., about the synchrony and reliability of channels). Still, when possible, there is value in attaining liveness/availability features, such as enabled by **identifiable abort**, **robustness**, **fairness** or even **guaranteed output delivery**. The pertinence of some of these termination options can also depend on the system model, including on how concurrent operations are handled, on who are the beneficiaries of the output (e.g., compare the parties in a threshold keygen vs. the client in a threshold signing).

4. System model. A proposal of threshold scheme must strive for a clear description that facilitates understanding various options across possible deployment scenarios.

(a) **Participants.** There is a **threshold entity** composed on n "parties". On the onset, all parties "know who" are the n parties, namely agreeing on n identifiers (possibly public keys to support authenticated channels).[6] For some operations, such as threshold keygen, the *beneficiaries* of the computation are the parties themselves, who end with a new (secret sharing) state that may require agreement (possibly in a sense of "security with **unanimous** abort"), and/or an administrator (e.g., who needs to accept a new public key). For other operations, such as threshold signing, the beneficiaries can be an external client, who initiates the computation with a request, and intends to determine an output. The client may or may not be aware of (and be able to interact distinctively based on) the n-party threshold composition (see "shared-I/O" interfaces[7] in Section 2.3 of IR 8214A). The possibility of **concurrent** execution requests must be considered. A baseline description can assume that there is a

[5] With "security with unanimous abort" the honest parties agree on whether or not there was an abort. This can be useful, for example, for a threshold keygen or secret resharing where the honest parties should collectively agree on having achieved a new secret-shared state. With "selective abort" (non-unanimous) some parties might be unaware that others have aborted. The suitability of the latter version needs to be carefully considered.

[6] The suitability of keys needs to be confirmed, locally or interactively, possibly using zero-knowledge proofs.

[7] These define whether or not a client can separately send/receive input/output shares to/from each party.

Item 1: Call 2021a for feedback (2021-July-02)

(possibly malicious) **proxy** that can: intermediate the communication between clients and the threshold entity, and authorize requested operations (e.g., the signing of a message).

(b) **Distributed systems and communication.** The description can decouple (i) classical distributed-systems' building blocks (e.g., consensus, reliable broadcast) from the (ii) essential cryptographic operations of the secure multiparty computation over (or of) a secret-shared key, as long as the interface and rules for composition are clearly specified. The specification of instantiations of the former (i), making use of weaker resources (e.g., enabling broadcast based on point-to-point channels), can be provided by reference to existing specifications & open-source implementations, while evaluating the impact of those replacements. A baseline description can make strong assumptions about the communication network, including synchrony and reliability of transmission. However, the proposal must discuss the pitfalls of deployment in environments with weaker guarantees (e.g., with asynchronous and unreliable channels), and possible mitigations. Different threshold schemes may be better suited to different communication environments, namely across the possible guarantees (and lack thereof) in terms of **synchrony**, **broadcast**, and **reliability**. It is important to understand how security guarantees break across these environments. The protocol can be described with various phases (e.g., offline, online, secret resharing), possibly with differentiated requirements.

5. Threshold profiles. For each primitive (see item 1) for thresholdization, it may be useful to consider differentiated solutions across various threshold parametrizations. For f/n: (i) S2PC ($(f, n) = (1, 2)$); (ii) honest majority ($f < n/2$); (iii) two-thirds honest majority ($f < n/3$); (iv) dishonest majority ($f \geq n/2$). For standalone n: "two" ($n = 2$); "small" ($3 \leq n \leq 8$); "medium" ($9 \leq n \leq 64$); and "large" ($n > 64$). The notion of "threshold profile" may be used to identify parametrization ranges. A threshold scheme proposal can focus on a single threshold profile or on several. The proposal must discuss the diversity of thresholds associated with various security properties. Future G&R may consider a suite of threshold schemes to cover various profiles. There is value in identifying motivating applications for adoption of threshold schemes in each profile.

6. Building blocks. Some building blocks (sometimes called gadgets) can be useful across various threshold schemes. A notable building block is Shamir **secret sharing** (and Lagrange interpolation), either in the clear or homomorphically (e.g., "in the exponent"). Other secret sharing variants may also be useful. Other examples of gadgets are **garbled circuits**, **oblivious transfer**, **commitments**, **secret resharing** (possibly for new values f and n), **multiplicative-to-additive share conversion**, **additively homomorphic encryption**, some **zero-knowledge proofs**, **consensus** and **broadcast**. To the extent possible, proposals of threshold schemes should modularize the description of gadgets. This means that a high-level description of the threshold scheme uses references to the interface and security properties of the gadgets, but not necessarily to low-level details. Then, a lower level description can be made for one (or more) possible instantiation of each needed gadget. While some future G&R documents may focus on gadgets, the decision to do so within the MPTC project will be subordinate to their utility for concrete threshold schemes. The upcoming call for proposals might also call for separate proposals of properly motivated gadgets.

Item 2: Printout of the .odp template for providing feedback

Comments in Reply to the NIST MPTC Call 2021a for Feedback on Criteria for Threshold Schemes

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

Month day, 2021

[[REMOVE THIS PORTION: This is a suggested but not mandatory template. Once filled, up to six pages, export to PDF and send by email to threshold-MP-call-2021a@nist.gov with the subject "MPTC Call 2021a: Public Comments on Criteria".]]

1. Scope of proposals
<Comments go here>

2. Idealization of security
<Comments go here>

3. Security vs. adversaries
<Comments go here>

4. System model
<Comments go here>

5. Threshold profiles
<Comments go here>

6. Building blocks
<Comments go here>

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.

Item 3: Printout of the .tex template for providing feedback

```
%% TEMPLATE FOR COMMENTS IN REPLY TO THE NIST MPTC Call 2021a for feedback
%% 2021-06-25 (LB): THIS IS A SUGGESTED BUT NOT MANDATORY TEMPLATE
%% Once filled with comments under the various paragraph headers, compile to a PDF file, with up
to six pages, and send to threshold-MP-feedback-2021a@nist.gov via en email with the subject
"Public Feedback on MP Threshold Criteria".



\documentclass[12pt,letterpaper]{article}
\usepackage[margin=.86in]{geometry}
\usepackage{iftex}
\ifPDFTeX\usepackage[utf8]{inputenc}
\else\usepackage{fontspec}\fi
\usepackage{adjustbox}
\setlength{\parindent}{0in}\setlength{\parskip}{1em}
\usepackage[bottom,hang,flushmargin]{footmisc}
\usepackage{fancyhdr,lastpage}
\renewcommand{\headrulewidth}{0pt}
\pagestyle{fancy}\setlength{\headheight}{14.5pt}
\cfoot{\scalebox{.82}{Page \thepage\ of \pageref{LastPage}}}
\usepackage{color}
\usepackage[colorlinks,allcolors=blue]{hyperref}
\usepackage{bookmark}
\newcommand{\bkmpar}[1]{\vskip.5em\noindent\pdfbookmark[1]{#1}{#1}\textbf{#1}\vskip0pt}
\def\authorA{FirstA LastA} % Fill in with name of author (A)
\def\affilA{Fill in the affiliations and other possible disclaimers.}
\def\authorB{FirstB LastB} % Fill in with name of author (B)
\def\affilB{Fill in the affiliations and other possible disclaimers.}
\def\authorC{FirstC LastC} % Fill in with name of author (C)
\def\affilC{Fill in the affiliations and other possible disclaimers.}
%% Add more definitions as suitable: \authorD, \affilD, ...
\def\ourdate{Month day, 2021}  %% fill in with the appropriate month and day



\begin{document}

\begin{center}
{\bfseries\large Comments in Reply to the NIST MPTC Call 2021a\\[1ex]
for Feedback on Criteria for Threshold Schemes}

\authorA\footnote{\affilA} $\cdot$
\authorB\footnote{\affilB} $\cdot$
\authorC\footnote{\affilC} %%% add more if suitable

\ourdate
\end{center}

\bkmpar{1. Scope and process}
% Comments go here

\bkmpar{2. Idealization of security}
% Comments go here

\bkmpar{3. Security vs. adversaries}
% Comments go here

\bkmpar{4. System model}
% Comments go here

\bkmpar{5. Threshold profiles}
% Comments go here

\bkmpar{6. Building blocks}
% Comments go here

\bkmpar{Other comments}
% Comments go here

\end{document}

```

Item 4: Feedback from Arpita Patra and Nigel Smart

From: Nigel Smart
Cc: Arpita Patra

On the Call 2021a for Feedback on Criteria for Threshold Schemes...

Point 1)

Item 5: You focus on "threshold" adversaries, but you could think about more general access structures and not be tied into threshold only.

e.g. t<n/2 could be generalized to Q2 structures
t<n/3 could be generalized to Q3 structures

This might be quite relevant for gov applications which require a more complex authorization structure then just a threshold.

Point 2)

You also do not mention the t<n/3 (resp t<n/4) bounds for
ASYNCHRONOUS protocols.

For Async protocols for robust computation the usual t<n/2 becomes t<n/3 and the usual t<n/3 becomes t<n/4. The references for these are.....

t<n/3 :
Michael Ben-Or, Boaz Kelmer, Tal Rabin: Asynchronous Secure Computations with Optimal Resilience (Extended Abstract). PODC 1994: 183-192
https://dl.acm.org/doi/10.1145/197917.198088

t <n/4 :
Michael Ben-Or, Ran Canetti, Oded Goldreich: Asynchronous secure computation. STOC 1993: 52-61

Point 3)

You might want to distinguish between malicious+robust computation [as above] vs active-with-abort style computation. For information theoretic constructs you need t<n/3 for active-with-abort security, vs t<n/2 for normal synchronous information theoretic protocols.

Yours

Arpita an Nigel

--
Prof. Nigel Smart
imec-COSIC - KU Leuven

https://homes.esat.kuleuven.be/~nsmart

Item 5: Feedback from Tore Frederiksen

Comments in Reply to the NIST MPTC Call 2021a for Feedback on Criteria for Threshold Schemes

Tore Frederiksen[1] ·

July 8, 2021

1. Scope and process

No comments.

2. Idealization of security

No comments.

3. Security vs. adversaries

Section 3 describes the desired security models in relation to the adversary and their power. Concretely a strong preference is expressed towards adaptive security. Adaptive security is clearly more desirable than static security. This is in particular true in a real-world setting, where "real" adversaries should generally be considered adaptive. However, achieving efficient adaptive security, in conjunction with other strong security requirements, might be really hard, if not impossible. An indication of this is given by the comparative lack of recent research in the area of adaptively secure protocols, in contrast to statically secure ones. This both holds in relation to full protocols, but even in the setting of building blocks such as garbled circuits and oblivious transfers. Furthermore, there seem to have been, both an academic and commercial, acceptance that "static security is good enough in practice, even though the real world is technically adaptive". This can for example be seen from a recent real-world-aimed academic paper by Chen *et al.* [2].

4. System model

No comments.

5. Threshold profiles

Section 5 discusses the different threshold profile concrete protocols could take. it is suggested that it is useful to have solutions for both the 2-party setting, honest majority, two-thirds honest majority and dishonest majority. We note, that in general, most recent threshold cryptography research has focused on the dishonest majority case (including the two-party case) [7, 2, 1, 5, 6, 8]. Furthermore, the recent works focusing on the dishonest majority case generally do so, not for reasons of robustness, but rather as a means of increasing efficiency [4, 3]. That is, cases with two-thirds honest majority to improve robustness has generally not been considered in contemporary research of threshold public key schemes. With this in mind, it might be worthwhile considering

[1]Security Lab, Alexandra Institute, Aarhus, Denmark, tore.frederiksen@alexandra.dk

Item 5: Feedback from Tore Frederiksen

to limit the recommended scope of submissions. Section 5 also considers the amount of parties to be involved with the computation, concretely mentioning settings for "two", "small", "medium" and "large" amount of parties. We note that most research has either focused on the "two" [7, 5], "small" [3, 8, 4, 3] or "large" [6, 2] cases, leaving the "medium" case out. Furthermore one can argue that the "two" and "small" cases are suitable for execution by somewhat trusted and highly reliable servers. This is in contrast to the "large" case which allows for execution by untrusted clients (e.g. in the blockchain setting), while still being able to achieve reasonable security. Thus the space where the "medium" case would fall seems hard to find.

Furthermore since it is suggested that it may be useful for solutions to achieve flexibility in regards to the threshold profile, it might even be impossible to construct candidates fulfilling all the requirements while achieving general flexibility.

6. Building blocks

While the idea of compossible building blocks is really nice, it seems hard to achieve without any formal description/standard, formalizing the interface of these building blocks. Perhaps defining specific interfaces (e.g. in the UC model) of each of the most commonly used primitives might yield an easier comparison of solutions and allowing for easier description and deployment in the future.

Other comments

In general there are a lot of different vectors of security and renationalization when it comes to threshold cryptography. The call for feedback has a lot of flexibility on almost all possible parameters. While this will allow for more schemes being suitable for submission, this could also imply that each future submission will reflect one specific setting, making it hard to compare.

Please note that the comments in this document are primarily in the setting of key generation for threshold public keys systems. However, they also hold for the symmetric case, assuming it is based on standard MPC techniques, as both key generation for threshold public keys systems and symmetric threshold cryptography (based on already existing standards) use many of the same underlying techniques and tools.

References

[1] Megan Chen, Ran Cohen, Jack Doerner, Yashvanth Kondi, Eysa Lee, Schuyler Rosefield, and Abhi Shelat. Multiparty generation of an RSA modulus. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 64–93. Springer, 2020.

[2] Megan Chen, Carmit Hazay, Yuval Ishai, Yuriy Kashnikov, Daniele Micciancio, Tarik Riviere, Abhi Shelat, Muthuramakrishnan Venkitasubramaniam, and Ruihan Wang. Diogenes:

Item 5: Feedback from Tore Frederiksen

Lightweight scalable RSA modulus generation with a dishonest majority. *IACR Cryptol. ePrint Arch.*, 2020:374, 2020.

[3] Anders P. K. Dalskov, Claudio Orlandi, Marcel Keller, Kris Shrishak, and Haya Shulman. Securing DNSSEC keys via threshold ECDSA from generic MPC. In Liqun Chen, Ninghui Li, Kaitai Liang, and Steve A. Schneider, editors, *Computer Security - ESORICS 2020 - 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14-18, 2020, Proceedings, Part II*, volume 12309 of *Lecture Notes in Computer Science*, pages 654–673. Springer, 2020.

[4] Ivan Damgård, Thomas Pelle Jakobsen, Jesper Buus Nielsen, Jakob Illeborg Pagter, and Michael Bæksvang Østergård. Fast threshold ECDSA with honest majority. In Clemente Galdi and Vladimir Kolesnikov, editors, *Security and Cryptography for Networks - 12th International Conference, SCN 2020, Amalfi, Italy, September 14-16, 2020, Proceedings*, volume 12238 of *Lecture Notes in Computer Science*, pages 382–400. Springer, 2020.

[5] Jack Doerner, Yashvanth Kondi, Eysa Lee, and Abhi Shelat. Secure two-party threshold ECDSA from ECDSA assumptions. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*, pages 980–997. IEEE Computer Society, 2018.

[6] Jack Doerner, Yashvanth Kondi, Eysa Lee, and Abhi Shelat. Threshold ECDSA from ECDSA assumptions: The multiparty case. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 1051–1066. IEEE, 2019.

[7] Tore Kasper Frederiksen, Yehuda Lindell, Valery Osheter, and Benny Pinkas. Fast distributed RSA key generation for semi-honest and malicious adversaries. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 331–361. Springer, 2018.

[8] Rosario Gennaro and Steven Goldfeder. One round threshold ECDSA with identifiable abort. *IACR Cryptol. ePrint Arch.*, 2020:540, 2020.

Item 5: Feedback from Tore Frederiksen

From: Brandao, Luis (IntlAssoc)
Sent: July 14, 2021
To: Tore Kasper Frederiksen
Subject: Re: Feedback on criteria for threshold schemes

Hi Tore,

Glad to hear from you.

Thank you for your early comments about the MP-threshold Call2021.

Two notes:

- Adaptive security. Your comment about static/adaptive touches indeed one important point we've been considering (and on which we hope to get ample feedback). My understanding is that many solutions proven secure in a static setting are proposed with an implicit understanding that no obvious "critical-safety" issue exists if the adversary is adaptive. For example, maybe a lack of adaptive security (technically, the environment being able to distinguish between the ideal and the real world) is simply because the protocol does not emulate some "less important" property (e.g., deniability of execution) of the defined ideal functionality. However, if, for example, a statically secure threshold signature scheme becomes forgeable under an adaptive attack, then that is a substantially different case of concern.

- Threshold profiles. Thanks for pointing out different perspectives across threshold ranges. It is acceptable to consider solutions that are specific to just one profile. For the "two" and "small" threshold profiles, by "somewhat trusted and highly reliable servers" did you intend to convey "passive / semi-honest" (as opposed to active/malicious)? For these cases we would also like to consider the malicious setting.

Thank you again for your valuable feedback,
Luís
–
Luís Brandão
Foreign Guest Researcher at NIST (Contractor via Strativia)

From: Tore Kasper Frederiksen
Sent: July 14, 2021
To: Brandao, Luis (IntlAssoc)
Subject: Re: Feedback on criteria for threshold schemes

Hi Luis,

You are every welcome. We are happy to supply feedback to the valuable standardization work done by NIST.
About your two comments:
1. I completely agree with your observation that it only makes sense to go for static security if it is reasonable to assume that there is no concrete break of security if used adaptively.

Item 5: Feedback from Tore Frederiksen

2. By somewhat trusted I did not mean passive/semi honest. I meant it more as an external/real world approach to security. If you pay a company to host part of an MPC computation you generally trust that they behave, contrary to someone who is taking part by running some python script on their laptop.

I hope this makes sense.

Best, Tore

From: Brandao, Luis (IntlAssoc)
Sent: July 21, 2021
To: Tore Kasper Frederiksen
Subject: Re: Feedback on criteria for threshold schemes

Hi Tore,

Thank you for the clarification!

1. Indeed. It's a point about which it will be useful and interesting to consider diverse feedback.

2. I agree there are important use-cases where a company providing MPC-as-a-service has to gain in providing a good secure service based on reliable servers and communication network. Ideally, some threshold schemes can be tailored to work very efficiently and effectively in that setting, while providing proper fallback guarantees for when the deployment setting turns out to be worst (e.g., malicious and asynchronous).

Regards,
Luís

Item 6: Feedback from Ran Canetti

From: Ran Canetti
Sent: September 7, 2021
To: Brandao, Luis (IntlAssoc)
Subject: Re: Call 2021a for Feedback on Criteria for Threshold Schemes

Dear Luis:

Thanks again for drawing my attention to the call for feedback and for adding me to the mailing list. I do think that the draft IR to be released later this year should be unequivocal about requiring analysis that considers adaptive corruptions.

Regarding use of the random oracle model:
The simple answer to you question is that indeed the protocols that I had in mind when mentioning the ROM in my previous note use the programmable ROM - but I dont think that this fact, in and of itself, should "disqualify" them. (Specifically, two examples I had in mind were Nielsen's non-committing encryption from Crypto 02 and the recent threshold ECDSA paper that I coauthor (CGGMP20) - but these two protocols use the ROM in very different ways: in Nielsen's work the use of the ROM is essential, whereas in the ECDSA work the ROM is used for components that have nothing to do with adaptive security. Also see below.)

In any case, regardless of what I had in mind, it will probably be a good idea to be a bit more explicit in the upcoming IR re how security analysis in the ROM will be treated. This of course applies to any aspect of the security analysis, not just to the handling of adaptive corruptions.

My personal take is that, while it is of course better to have security analysis that does not use the ROM (or other over-idealized models of cryptographic primitives), security analysis in the ROM does have significant value, since it compartmentalizes the "piece that is left un-analyzed" and provides some guarantee as to the soundness of overall structure of the protocol. In particular, experience shows that oftentimes it is possible to later "make ends meet" by either finding a concrete notion of security for the hash function that suffices for the application, or reformulating the security requirement, or both. Still, it would be good to ask that:
(a) the use of the ROM is minimal - components that do not need the use of the ROM are modeled and analyzed in the standard model
(b) the act of replacing the abstract ROM with an actual hash function does not introduce new vulnerabilities --- especially in cases where multiple primitives are analyzed separately in the ROM. (Indeed - schemes that use the programmable ROM often tend to be more susceptible to having issues with this aspect. But, as was demonstrated in a number or works - eg, the wonderful-worlds paper of Camenisch etal from Eurocrypt 18 - this is not necessarily the case.)

BTW, an unrelated comment: It may also be good to ask that the security analysis explicitly specify not only the "adversary model" and the "system model" , but also the "protocol environment" - namely any assumptions or expectations regarding the "calling protocols" - namely the components that provide the inputs to (and read the outputs of) the analyzed protocol. Are there any assumptions/ expectations as to the structure/distribution of the inputs? Are there assumptions/expectations on what information about the input/outputs are leaked by the calling protocol to adversarial entities? etc.
(Of course, security that's guaranteed for "any environment" is always best, but there may well be situations where proviging security only wrt restricted classes of environments might be both meaningful and allow for more efficient solutions.)

Item 6: Feedback from Ran Canetti

Hope this helps,
Ran

From: Brandao, Luis (IntlAssoc)
Sent: September 7, 2021
To: Ran Canetti
Subject: Re: Call 2021a for Feedback on Criteria for Threshold Schemes

Dear Ran,

Thank you for your comments and examples about the importance of security against adaptive corruptions. Your feedback is much appreciated!

[...]

A curiosity: when you mention efficient-adaptively-secure solutions in the RO model, do you think these require programmable random oracles, or would non-programmable RO's suffice?

One concrete intention with the suggested questions in the email (in complement to the actual call 2021a) is to motivate that (ask whether it is reasonable that) protocols that have so far only been analyzed under static corruptions get a possible new look (i.e., new security analysis) to check whether they (or slight adjustments thereof) may be suitable for deployment in a setting of adaptive corruptions.

Thanks again.
Regards, Luís

From: Ran Canetti
Sent: September 4, 2021
To: Brandao, Luis (IntlAssoc)
Subject: Re: Fw: Call 2021a for Feedback on Criteria for Threshold Schemes

Dear Luis,

[...]

Regarding security against adaptive corruptions: I dont think that the suggested questions below are sufficient. In fact, *I would strongly advise against considering any protocol that does not provide a guarantee of security against adaptive corruptions.* Here is why:

(a) We do have very reasonable and efficient protocols for threshold schemes that provide security against adaptive corruptions. Certainly, in the RO model it is possible to obtain adaptive security with very little overhead - for protocols that are designed right. So there is really no excuse for not providing provable adaptive security.

(b) Adaptive attacks are a real concern. Furthermore - adaptive security is a prerequisite for

Item 6: Feedback from Ran Canetti

meaningful proactive security: Proactive refreshes are meaningless unless they consider adaptive corruptions.

Let me highlight this issue via the following example: If a protocol is not required to provide security against adaptive corruptions, then there is nothing that requires the protocol to ever instruct a party to erase local data. That is, take any protocol, and remove from it all the instructions to locally erase data. If your notion of security does not consider adaptive corruptions then the new protocol will be just as secure as the original one. This, of course, is highly counterintuitive since we know that judicious erasures of local data are oftentimes crucial for providing real-life security against attacks.

Hope this helps,

Ran

Item 7: Feedback from Samuel Ranellucci

unbound security

Comments on "Call 2021a for Feedback on Criteria for Threshold Schemes"

Samuel Ranellucci

Unbound security uses MPC to protect cryptographic keys. We currently support all the algorithms that are described in the document "Call 2021a for Feedback on Criteria for Threshold Schemes". This document is very professional and well thought out. We thank NIST for their hard work.

Input enforcement: In many cases, a naive threshold implementation of a primitive is insufficient to provide real world security. Let us take as an example AES. Suppose that an entity wanted to replace their AES implementation with an MPC solution that uses two servers. A naive solution would be to split the key into two shares and then to run a secure MPC protocol where each party provides a share of the key and a share of the message. Unfortunately, such a solution does not provide full security in case one of the parties is corrupt. An adversary that provides bad shares for the AES key can mount a related key attack. More precisely, during threshold decryption of a ciphertext c, instead of providing his share s of the key k during the MPC protocol, the adversary can provide $s \oplus \Delta$. The result is that the adversary gets to learn the decryption of c under the key $k \oplus \Delta$. To improve threshold AES decryption, the parties can precompute shares of the expanded key. However, if we apply this optimization and the adversary can provide incorrect shares of the expanded key, then he can mount an attack that is significantly more powerful. See the paper "Differential Fault Analysis of AES: Towards Reaching its Limits" for more details. As a result, a secure protocol for threshold cryptography needs to include a method of enforcing the use of the correct key share.

Primitive flexibility: Unbound security provides a software platform for using MPC to do key management. Our software platform can be run on a variety of devices. Our software might need to run on a single thread with a network bandwidth of only a few megabits per second. The protocols might be run on servers, laptops and even phones. In contrast, other groups might want to run their protocols solely on dedicated servers. However, this leads to the following issue.

When we want to select the best protocol, we need to consider the setting. Are the protocols being run on AWS or Azure instances or are they being run on laptops. Protocols that may provide great performance in the first setting may be undesirable in the second setting and vice-versa. This naturally leads to the question of how protocols should be evaluated for efficiency and if it might be necessary to provide multiple standards for the same functionality based on the setting that we will use them. For example, the IKNP based OT-extension protocols might be too expensive when we only have access to a low-bandwidth network. In some cases, a restricted setting requires us to develop custom solutions that require complicated algorithmic improvements that may not be suitable for standardization.

Thus, we believe that a standard should be written to allow flexibility. This can be done by allowing standards to use black boxes for primitives such as Oblivious Transfer,

Unbound Security
9 HaPsagot Street
Petach Tikva, Israel

www.unboundsecurity.com
contact@unboundtech.com
+972-72-277-3437

Item 7: Feedback from Samuel Ranellucci

commitments and garbled circuits without specifying a concrete instance. This would allow the designer the choice to instantiate Oblivious Transfer with either the IKNP-based protocols or newer low-communication variants based on LPN.

Building blocks: Oblivious Transfer, commitments and garbled circuits are critical building blocks of threshold cryptography. A large portion of the threshold schemes mentioned in the document use these primitives as a critical building block. As a result, we believe that these building blocks should be the first target of standardization.

Systematic evaluation: Ultimately, when NIST will work towards standardizing MPC protocols it should provide methods to evaluate protocols in a fair and systematic way. First, before making concrete proposals, NIST should write a detailed explanation on how it will evaluate different candidates. Second, to ensure that all reference implementations can be easily compared, we recommend that NIST provide basic networking code that will need to be used by all candidates to demonstrate the efficiency of the protocols. Similarly, NIST may also want to code for basic primitives such as OT and commitments to fairly evaluate different protocols.

Standardizing MPC is a challenging task. The document "Call 2021a for Feedback on Criteria for Threshold Schemes" is an excellent step towards that goal. We thank NIST for this initiative and look forward to the next iteration of this project.

Item 8: Feedback from Dan Boneh and Chelsea Komlo

Feedback on NIST Criteria for Threshold Schemes

Dan Boneh, Chelsea Komlo

September 2021

NIST recently put out a call for feedback on criteria for threshold schemes. This document outlines three points that, if added, would strengthen the applicability of the resulting standards.

1 Comment One: Privacy versus Accountability

The standard notion of a threshold signature scheme in the literature is meant to be *private* — a signature reveals nothing about which t signers out of the set of all possible n signers participated to produce the signature. Even the threshold t is not revealed by the signature. While this is desirable in some settings, it means that signers are not *accountable* for messages they sign.

Accountability is a required in many settings where threshold signatures are used. Most notably, in financial applications, if a transaction is incorrectly authorized, an organization should be able to inspect the signature and learn which signers participated in the rogue authorization process. In such settings, a private threshold signature often cannot be used.

> Suggestion: the criteria for threshold signature schemes could call out for two flavors of threshold signatures: *private threshold signatures* (PTS), where the signature hides the threshold t and hides the signing set, and *accountable threshold signatures* (ATS) [3, 4, 1, 2, 7], where a signature can be securely traced to the set of signers.

Note that an ATS scheme can be trivially constructed by concatenating t non-threshold signatures from the signing parties, and embedding t within the public key along with a set of n public keys, one for each authorized signer. However, there are ATS schemes that are much better than this trivial scheme in that signatures are shorter and verification is faster [4, 1, 2, 7].

2 Comment Two: Support for Threshold PRFs Beyond AES

Simplicity in cryptographic constructions is critical to minimize the risk of security-critical bugs in implementations, as well as to lower the barrier to understanding and implementing the scheme for practitioners. Towards this end, we recommend widening the scope of proposals for the building blocks necessary to instantiate threshold PRFs.

Threshold PRFs have a range of practical use cases. For example, threshold (O)PRFs can be used as a building block for distributed one-time password systems and anonymous token issuance. In this setting, partitioning a secret key among a threshold number of parties is desirable for distribution of trust and redundancy purposes, without requiring the reconstruction of the secret key at a single location.

The current scope of proposals for threshold primitives includes AES, which can be made into a threshold PRF via multiparty computation. However, there are many simpler ways to construct a threshold PRF. One example is a threshold PRF built from the XOR of several non-threshold PRF such as AES [5]. The resulting threshold PRF is far more efficient than the one obtained from MPC applied to AES.

Item 8: Feedback from Dan Boneh and Chelsea Komlo

Another simplified approach to threshold PRF can be realized by an algebraic PRF [6]. Let F be the PRF function, k be the input, and x be the secret key. An algebraic PRF (via hashed Diffie-Hellman) is then $F(k, x) := H(x)^k$, where $H : \mathcal{X} \to \mathbb{G}$ is a random oracle. Here $\mathbb{G}$ is a group where DDH is hard, and $\mathcal{X}$ is the domain of the PRF. A threshold variant of this PRF simply applies Shamir secret sharing to the secret key k.

> Suggestion: the criteria for threshold PRF schemes could allow for PRFs that are not AES. This could result in simpler and more efficient threshold PRF constructions compared to AES-MPC.

3 Comment Three: More General Access Structures

The proposed threshold profiles currently focus on threshold access structures, but it may be desirable to support more general access structures. For example, instead of requiring a threshold number of signers from one set, it may be desirable to require t_1 users from one set and t_2 users from another set.

> Suggestion: the proposed threshold profiles could optionally require support for monotone access structures beyond a simple t-out-of-n design.

References

[1] M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 390–399, 2006.

[2] D. Boneh, M. Drijvers, and G. Neven. Compact multi-signatures for smaller blockchains. In *ASIACRYPT '18*, volume 11273, pages 435–464. Springer, 2018.

[3] K. Itakura, K; Nakamura. A public-key cryptosystem suitable for digital multisignatures. *NEC research and development*, 1983.

[4] S. Micali, K. Ohta, and L. Reyzin. Accountable-subgroup multisignatures: Extended abstract. In *Proceedings of the 8th ACM Conference on Computer and Communications Security*, CCS '01, page 245–254, New York, NY, USA, 2001. Association for Computing Machinery.

[5] S. Micali and R. Sidney. A simple method for generating and sharing pseudo-random functions, with applications to clipper-like escrow systems. In *CRYPTO '95*, volume 963 of *LNCS*, pages 185–196. Springer, 1995.

[6] M. Naor, B. Pinkas, and O. Reingold. Distributed pseudo-random functions and kdcs. In *EUROCRYPT '99*, volume 1592 of *LNCS*, pages 327–346. Springer, 1999.

[7] J. Nick, T. Ruffing, and Y. Seurin. MuSig2: Simple Two-Round Schnorr Multi-Signatures. Cryptology ePrint Archive, Report 2020/1261, 2020.

Item 9: Feedback from Jakob Pagter

From: Jakob Pagter
Sent: September 13, 2021
To: threshold-MP-call-2021a
Subject: Feedback on Criteria for Threshold Schemes

Regarding:
https://csrc.nist.gov/CSRC/media/Projects/threshold-cryptography/documents/MPTC-call2021a-feedback.pdf

Overall we believe that this scope of requirements is reasonable.

Regarding adaptive security we think this is a relevant perspective, but it has not yet received a lot of attention from the research community, so any strong requirements for active security will likely slow down the usage of standardised MPC. Also, practical models should be taken into account; for example it seems unrealistic that an actual adversary can choose whether to compromise a backend MPC node or a mobile node. Finally, for 2-party MPC, adaptive security probably should be ignored, as the adversary can be prevented from having any meaningful information on which party to corrupt.

A perspective which is not well covered in the document is that of thresholds for non-interactive protocols. For instance, for EdDSA and ECDSA, it must not be possible for a dishonest party to re-use the same offline data twice with different subsets of players as this would allow exfiltration of the private key by effectively signing the same message twice with the same nonce.

Best,
Jakob

--
Jakob Pagter, CTO
Sepior

From: Brandao, Luis (IntlAssoc)
Sent: September 13, 2021
To: Jakob Pagter
Subject: Re: Feedback on Criteria for Threshold Schemes

Dear Jakob,

Thank you for your feedback, namely your comments about security types, practical models, and thresholds in non-interactive setting.

Item 9: Feedback from Jakob Pagter

These comments are useful for the ongoing process of consideration of threshold schemes by the MPTC project.

We'll followup sometime later with a public compilation of the received feedback and some reply comments.

One clarification question: your first sentence mentioned "adaptive" and "active" security in separate parts. When saying "any strong requirements for active security will likely ..." did you mean adaptive+active together, our simply active (regardless of being "adaptive" or "static")?

Regards, Luís

--Luís Brandão
Foreign Guest Researcher at NIST (Contractor via Strativia)

From: Jakob Pagter
Sent: September 15, 2021
To: Brandao, Luis (IntlAssoc)
Subject: Re: Feedback on Criteria for Threshold Schemes

Hi Luis,

It should say: “any strong requirements for adaptive”. Hope that makes it clearer :)

Sorry for the confusion.

/Jakob