NIST Call for Multi-Party Threshold Schemes
Brief Notes at ICMC 2023

Presented by Lily Chen[*] at **ICMC 2023**

International Cryptographic Module Conference

September 22nd @ Ottawa, Canada

Expressed opinions are from the speakers/authors and should not be construed as official NIST views.

Slides authored by Luís Brandão[†], based on "*NISTIR 8214C ipd*" (L. Brandão and R. Peralta. January 2023.)

[*] Lily Chen is at NIST. [†] Luís Brandão is at NIST as a Foreign Guest Researcher (non-employee), contractor from Strativia.

Outline

1. NIST Crypto Standardization/Exploratory Projects
2. The "Threshold Call" (at a high level)
3. Subcategories and Submissions

(Slides will be publicly available)

Legend: Crypto = Cryptography. NIST = National Institute of Standards and Technology.

Outline

1. NIST Crypto Standardization/Exploratory Projects
2. The "Threshold Call" (at a high level)
3. Subcategories and Submissions

Legend: Crypto = Cryptography. NIST = National Institute of Standards and Technology.

NIST Crypto Standardization/Exploratory Projects

- **PQC:** [standardization] "**Post-Quantum**" signatures and key-encapsulation
- **LWC:** [standardization] "**LightWeight**" **a**uth. **e**nc. w/ **a**ssoc. **d**ata, and hashing

Legend: **AEAD** = **A**uth[enticated] **E**nc[ryption] w[ith] **A**ssoc[iated] **D**ata. **CTG** = **C**ryptographic **T**echnology **G**roup. **LWC** = **L**ight**w**eight **C**ryptography. **MPTC** = **M**ulti-**P**arty **T**hreshold **C**ryptography. **NIST** = **N**ational **I**nstitute of **S**tandards and **T**echnology. **PEC** = **P**rivacy-**E**nhancing **C**ryptography. **PQC** = **P**ost-**Q**uantum **C**ryptography.

Presented at
ICMC 2023

NIST Crypto Standardization/Exploratory Projects

- **PQC:** [standardization] "**Post-Quantum**" signatures and key-encapsulation
- **LWC:** [standardization] "**LightWeight**" **a**uth. **e**nc. w/ **a**ssoc. **d**ata, and hashing
- **PEC:** [exploratory] "**Privacy-Enhancing**" (advanced) features/functionalities
- **MPTC:** [exploratory] "**Multi-Party Threshold**" schemes for crypto primitives

Legend: **AEAD** = **A**uth[enticated] **E**nc[ryption] w[ith] **A**ssoc[iated] **D**ata. **CTG** = **C**ryptographic **T**echnology **G**roup. **LWC** = **L**ight**w**eight **C**ryptography. **MPTC** = **M**ulti-**P**arty **T**hreshold **C**ryptography. **NIST** = **N**ational **I**nstitute of **S**tandards and **T**echnology. **PEC** = **P**rivacy-**E**nhancing **C**ryptography. **PQC** = **P**ost-**Q**uantum **C**ryptography.

NIST Crypto Standardization/Exploratory Projects

- **PQC:** [standardization] “**Post-Quantum**” signatures and key-encapsulation
- **LWC:** [standardization] “**LightWeight**” **a**uth. **e**nc. w/ **a**ssoc. **d**ata, and hashing
- **PEC:** [exploratory] “**Privacy-Enhancing**” (advanced) features/functionalities
- **MPTC:** [exploratory] “**Multi-Party Threshold**” schemes for crypto primitives
- **...** (various other projects in the NIST “Crypto group” [CTG])

Legend: **AEAD** = Auth[enticated] Enc[ryption] w[ith] Assoc[iated] Data. **CTG** = Cryptographic Technology Group. **LWC** = Lightweight Cryptography. **MPTC** = Multi-Party Threshold Cryptography. **NIST** = National Institute of Standards and Technology. **PEC** = Privacy-Enhancing Cryptography. **PQC** = Post-Quantum Cryptography.

NIST Crypto Standardization/Exploratory Projects

- **PQC:** [standardization] "**Post-Quantum**" signatures and key-encapsulation
- **LWC:** [standardization] "**LightWeight**" **auth. enc.** w/ **assoc. d**ata, and hashing
- **PEC:** [exploratory] "**Privacy-Enhancing**" (advanced) features/functionalities
- **MPTC:** [exploratory] "**Multi-Party Threshold**" schemes for crypto primitives
- **...** (various other projects in the NIST "Crypto group" [CTG])

The "Threshold Call" (from MPTC+PEC): to gather **reference material** for public analysis ... aiming for **recommendations** (in a 1st phase), including about PEC.

Legend: **AEAD** = **Auth**[enticated] **Enc**[ryption] w[ith] **Assoc**[iated] **Data**. **CTG** = **C**ryptographic **T**echnology **G**roup. **LWC** = **L**ight**w**eight **C**ryptography. **MPTC** = **M**ulti-**P**arty **T**hreshold **C**ryptography. **NIST** = **N**ational **I**nstitute of **S**tandards and **T**echnology. **PEC** = **P**rivacy-**E**nhancing **C**ryptography. **PQC** = **P**ost-**Q**uantum **C**ryptography.

Presented at
ICMC 2023

Privacy-Enhancing Cryptography (PEC): NIST Project

Cryptography (that can be) used to **enhance privacy**.

(emphasis on non-standardized tools)

Privacy-Enhancing Cryptography (PEC): NIST Project

Cryptography (that can be) used to **enhance privacy**.
(emphasis on non-standardized tools)

Goals:

1. Accompany the progress of **emerging *PEC tools***.

ZKP	MPC	FHE	PSI	GRS	FnE	PIR	StE
Zero-**K**nowledge **P**roofs	(**S**ecure) **M**ulti**p**arty **C**omputation	**F**ully **H**omomorphic **E**ncryption	**P**rivate **S**et **I**ntersection	**G**roup and **R**ing **S**ignatures	**Fun**ctional **E**ncryption (Inc. ABE & IBE)	**P**rivate **I**nformation **R**etrieval	**St**ructured **E**ncryption (Symm./Pub.)

Legend: ABE: attribute-based encryption. IBE: identity-based encryption. Inc.: including. PEC: privacy-enhancing cryptography. Symm./pub.: symmetric-key or public-key based.

Privacy-Enhancing Cryptography (PEC): NIST Project

Cryptography (that can be) used to **enhance privacy**.
(emphasis on non-standardized tools)

Goals:

1. Accompany the progress of **emerging *PEC tools***.
2. Promote development of PEC **reference material**.

- **PEC tools**
- **STPPA (series of talks)**
- **PEC use-case suite**
- **Threshold schemes**
- **ZKProof collaboration**
- **Encounter metrics**
- **Email list (PEC Forum)**

https://csrc.nist.gov/projects/pec

ZKP	**MPC**	**FHE**	**PSI**	**GRS**	**FnE**	**PIR**	**StE**
Zero-**K**nowledge **P**roofs	(**S**ecure) **M**ultiparty **C**omputation	**F**ully **H**omomorphic **E**ncryption	**P**rivate **S**et **I**ntersection	**G**roup and **R**ing **S**ignatures	**F**u**n**ctional **E**ncryption (Inc. ABE & IBE)	**P**rivate **I**nformation **R**etrieval	**St**ructured **E**ncryption (Symm./Pub.)

Legend: ABE: attribute-based encryption. IBE: identity-based encryption. Inc.: including. PEC: privacy-enhancing cryptography. Symm./pub.: symmetric-key or public-key based.

Privacy-Enhancing Cryptography (PEC): NIST Project

Cryptography (that can be) used to **enhance privacy**.
(emphasis on non-standardized tools)

Goals:

1. Accompany the progress of **emerging *PEC tools***.
2. Promote development of PEC **reference material**.
3. **Exploratory work** to assess potential for recommendations, standardization; ...

- **PEC tools**
- **STPPA (series of talks)**
- **PEC use-case suite**
- **Threshold schemes**
- **ZKProof collaboration**
- **Encounter metrics**
- **Email list (PEC Forum)**

https://csrc.nist.gov/projects/pec

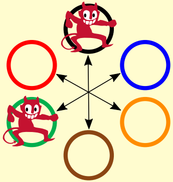
ZKP	**MPC**	**FHE**	**PSI**	**GRS**	**FnE**	**PIR**	**StE**
Zero- **K**nowledge **P**roofs	(**S**ecure) **M**ulti**p**arty **C**omputation	**F**ully **H**omomorphic **E**ncryption	**P**rivate **S**et **I**ntersection	**G**roup and **R**ing **S**ignatures	**F**u**n**ctional **E**ncryption (Inc. ABE & IBE)	**P**rivate **I**nformation **R**etrieval	**St**ructured **E**ncryption (Symm./Pub.)

Legend: ABE: attribute-based encryption. IBE: identity-based encryption. Inc.: including. PEC: privacy-enhancing cryptography. Symm./pub.: symmetric-key or public-key based.

Multi-Party Threshold Cryptography: NIST project

Cryptographic primitives: Signing Encryption KeyGen Hashing etc.

Threshold schemes (for cryptographic primitives):

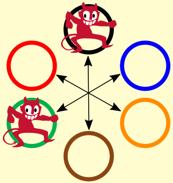


https://csrc.nist.gov/projects/threshold-cryptography

Multi-Party Threshold Cryptography: NIST project

Cryptographic primitives: Signing Encryption KeyGen Hashing etc.

Threshold schemes (for cryptographic primitives):



1. Split (**secret-share**) the secret/private-key across multiple parties.
2. Use **MPC** to perform needed operation (with split key), e.g., sign.
   (MPC = secure multiparty computation ... or call it “Threshold Cryptography”)

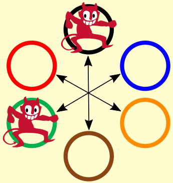
https://csrc.nist.gov/projects/threshold-cryptography

Multi-Party Threshold Cryptography: NIST project

Cryptographic primitives: Signing Encryption KeyGen Hashing etc.

Threshold schemes (for cryptographic primitives):

1. Split (**secret-share**) the secret/private-key across multiple parties.
2. Use **MPC** to perform needed operation (with split key), e.g., sign.
   (MPC = secure multiparty computation ... or call it "Threshold Cryptography")



- **"Threshold" (f):** Operation is secure if number of corrupted parties is $\leq f$.
- **Decentralized** trust about key **(not reconstructed)**: avoids single-point of failure.

https://csrc.nist.gov/projects/threshold-cryptography

Why care about / explore PEC and threshold schemes?

Why care about / explore PEC and threshold schemes?

Attraction: Potential applications and feasibility: threshold crypto; privacy apps; ...

Why care about / explore PEC and threshold schemes?

Attraction: Potential applications and feasibility: threshold crypto; privacy apps; ...

Hesitations / need for recommendations:

- Which *claimed crypto/security* is useful and trustworthy?
- Which primitives are ***threshold-friendlier***? (easier in practice to thresholdize or, amenable to "better" threshold schemes)

Why care about / explore PEC and threshold schemes?

Attraction: Potential applications and feasibility: threshold crypto; privacy apps; ...

Hesitations / need for recommendations:

- Which *claimed crypto/security* is useful and trustworthy?
- Which primitives are ***threshold-friendlier***? (easier in practice to thresholdize or, amenable to "better" threshold schemes)

Goal: Promote *good* **adoptability** (secure, interoperable, best practices; ...)

Presented at **ICMC 2023**

Why care about / explore PEC and threshold schemes?

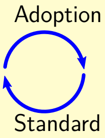
Attraction: Potential applications and feasibility: threshold crypto; privacy apps; ...

Hesitations / need for recommendations:

- Which *claimed crypto/security* is useful and trustworthy?
- Which primitives are ***threshold-friendlier***? (easier in practice to thresholdize or, amenable to "better" threshold schemes)

Goal: Promote *good* **adoptability** (secure, interoperable, best practices; ...)

How to explore the threshold space?



Presented at
ICMC 2023

Why care about / explore PEC and threshold schemes?

Attraction: Potential applications and feasibility: threshold crypto; privacy apps; ...

Hesitations / need for recommendations:

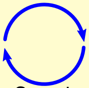
- Which *claimed crypto/security* is useful and trustworthy?
- Which primitives are ***threshold-friendlier***? (easier in practice to thresholdize or, amenable to "better" threshold schemes)

Goal: Promote *good* **adoptability** (secure, interoperable, best practices; ...)

How to explore the threshold space?

Next section: A public **Call** for reference material ... toward **recommendations**

Adoption



Standard

Outline

1. NIST Crypto Standardization/Exploratory Projects
2. The "Threshold Call" (at a high level)
3. Subcategories and Submissions

Legend: Crypto = Cryptography. NIST = National Institute of Standards and Technology.

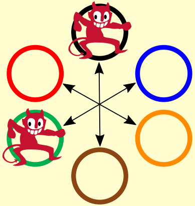
NIST Call for Multi-Party Threshold Schemes

- NISTIR 8214C: Initial public **draft** (**Jan 2023**) ⇒ Revised version (**late 2023**).
- Submission deadline (expected ≈ **2nd-half 2024**)

Presented at
ICMC 2023

NIST Call for Multi-Party Threshold Schemes

- NISTIR 8214C: Initial public **draft** (**Jan 2023**) ⇒ Revised version (**late 2023**).
- Submission deadline (expected ≈ **2nd-half 2024**)



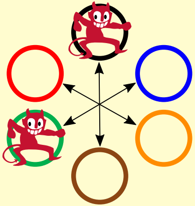
Calling for submissions of threshold schemes

(And gadgets for modular use)

Presented at
ICMC 2023

NIST Call for Multi-Party Threshold Schemes

- NISTIR 8214C: Initial public **draft** (**Jan 2023**) ⇒ Revised version (**late 2023**).
- Submission deadline (expected ≈ **2nd-half 2024**)



Calling for submissions of threshold schemes for:

- **[Cat1] Selected NIST-standardized primitives**
- **[Cat2] Other primitives (including FHE, IBE/ABE, ZKP)**

  (And gadgets for modular use)

FHE = Fully-homomorphic encryption.
IBE/ABE = Identity/Attribute-based encryption.
ZKP = Zero-knowledge proof.

Presented at
ICMC 2023

Notes about the process

- **Setup:** A gathering of reference material (not a competition for a selection).
- **Expected:** The process will clarify relevant system models, best practices, ...
- **Aim:** Devise recommendations about advanced cryptography (PEC + MPTC)
  (Will support future standardization processes.)

  PEC = Privacy-Enhancing Crypto
  MPTC = Multi-Party Threshold Crypto
- **Ample room for participation:** Give feedback → Submit → Analyze
- **It's time:** Consider starting to organize a future submission (team, scope, ...)

Notes about the process

- **Setup:** A gathering of reference material (not a competition for a selection).
- **Expected:** The process will clarify relevant system models, best practices, ...
- **Aim:** Devise recommendations about advanced cryptography (PEC + MPTC)
  (Will support future standardization processes.)

PEC = Privacy-Enhancing Crypto
MPTC = Multi-Party Threshold Crypto

- **Ample room for participation:** Give feedback → Submit → Analyze
- **It's time:** Consider starting to organize a future submission (team, scope, ...)

The call is not aimed at directly selecting a standard, but is part of a longer process toward possible standardization.

Community participation

Various areas / possible synergies:

- Scope of the call is of interest to various crypto communities: MPC, ZKP, FHE, ...
- Work developed with other SDOs and in community efforts is also welcome.

(SDO = Standards Development Organization)

Community participation

Various areas / possible synergies:

- Scope of the call is of interest to various crypto communities: MPC, ZKP, FHE, ...
- Work developed with other SDOs and in community efforts is also welcome.

(SDO = Standards Development Organization)

Some variables:

- How will the community compose teams? (How to avoid effort duplication?)
- How will the scope of the call be covered? (primitives / models / approaches)

Community participation

Various areas / possible synergies:

- Scope of the call is of interest to various crypto communities: MPC, ZKP, FHE, ...
- Work developed with other SDOs and in community efforts is also welcome.

(SDO = Standards Development Organization)

Some variables:

- How will the community compose teams? (How to avoid effort duplication?)
- How will the scope of the call be covered? (primitives / models / approaches)

MPTS 2023: (Sep 26–28) NIST Workshop on **M**ulti-**P**arty **T**hreshold **S**chemes

http://csrc.nist.gov/events/2023/mpts2023

Outline

1. NIST Crypto Standardization/Exploratory Projects
2. The "Threshold Call" (at a high level)
3. Subcategories and Submissions

Legend: Crypto = Cryptography. NIST = National Institute of Standards and Technology.

Category Cat1 of NIST Call for Multi-Party Threshold Schemes

Subcategory: Type
C1.1: **Signing**
C1.2: **PKE**
C1.3: **2KA**
C1.4: **Symmetric**
C1.5: **Keygen**

Category Cat1 of NIST Call for Multi-Party Threshold Schemes

Too many acronyms, we know. (Legend further below)

Subcategory: Type	Families of specifications	NIST references
C1.1: **Signing**	EdDSA sign, ECDSA sign, RSADSA sign	FIPS 186-5 (see also NISTIR 8214B)

Legend: **2KA**: pair-wise key-agreement. **2KE**: pair-wise key-establishment. **AES**: Advanced Encryption Standard. **CDH**: cofactor Diffie–Hellman. **ECC**: Elliptic-curve cryptography (or, if used as an adjective, EC-based). **ECDSA**: Elliptic-curve Digital Signature Algorithm. **EdDSA**: Edwards-curve Digital Signature Algorithm. Elliptic-curve based Key-Establishment. **FIPS**: Federal Information Processing Standard. **KC**: Key-confirmtion. **KDM**: Key-derivation mechanism. **Keygen**: Key-generation. **MQV**: Menezes-Qu-Vanstone. **PKE**: public-key encryption. **RSA**: Rivest–Shamir–Adleman (signature and encryption schemes). **RSADSA**: **RSA** digital signature algorithm. **SP 800**: Special Publication (in Computer Security).
Note: In the 2nd column, each item within a subcategory is itself called a family of specifications, since it may include diverse primitives or modes/variants.

Category Cat1 of NIST Call for Multi-Party Threshold Schemes

Too many acronyms, we know. (Legend further below)

Subcategory: Type	Families of specifications	NIST references
C1.2: **PKE**	RSA decrypt, RSA encrypt (a secret value)	SP 800-56B Rev2

Legend: **2KA**: pair-wise key-agreement. **2KE**: pair-wise key-establisment. **AES**: Advanced Encryption Standard. **CDH**: cofactor Diffie–Hellman. **ECC**: Elliptic-curve cryptography (or, if used as an adjective, EC-based). **ECDSA**: Elliptic-curve Digital Signature Algorithm. **EdDSA**: Edwards-curve Digital Signature Algorithm. Elliptic-curve based Key-Establishment. **FIPS**: Federal Information Processing Standard. **KC**: Key-confirmtion. **KDM**: Key-derivation mechanism. **Keygen**: Key-generation. **MQV**: Menezes-Qu-Vanstone. **PKE**: public-key encryption. **RSA**: Rivest–Shamir–Adleman (signature and encryption schemes). **RSADSA**: **RSA** digital signature algorithm. **SP 800**: Special Publication (in Computer Security).
Note: In the 2nd column, each item within a subcategory is itself called a family of specifications, since it may include diverse primitives or modes/variants.

Presented at
ICMC 2023

Category Cat1 of NIST Call for Multi-Party Threshold Schemes

Too many acronyms, we know. (Legend further below)

Subcategory: Type	Families of specifications	NIST references
C1.4: **Symmetric**	AES encipher/decipher, KDM/KC (for 2KE)	FIPS 197, SP 800-56C Rev2, ...

Legend: **2KA**: **pair**-wise **k**ey-**a**greement. **2KE**: **pair**-wise **k**ey-**e**stablisment. **AES**: **A**dvanced **E**ncryption **S**tandard. **CDH**: **c**ofactor **D**iffie–**H**ellman. **ECC**: **E**lliptic-**c**urve **c**ryptography (or, if used as an adjective, **EC**-based). **ECDSA**: **E**lliptic-**c**urve **D**igital **S**ignature **A**lgorithm. **EdDSA**: **Ed**wards-curve **D**igital **S**ignature **A**lgorithm. **E**lliptic-**c**urve based **K**ey-**E**stablishment. **FIPS**: **F**ederal **I**nformation **P**rocessing **S**tandard. **KC**: **K**ey-**c**onfirmtion. **KDM**: **K**ey-**d**erivation **m**echanism. **Keygen**: **Key**-**gen**eration. **MQV**: Menezes-Qu-Vanstone. **PKE**: **p**ublic-**k**ey **e**ncryption. **RSA**: **R**ivest–**S**hamir–**A**dleman (signature and encryption schemes). **RSADSA**: **RSA** digital signature algorithm. **SP 800**: **S**pecial **P**ublication (in Computer Security).

Note: In the 2nd column, each item within a subcategory is itself called a family of specifications, since it may include diverse primitives or modes/variants.

Category Cat1 of NIST Call for Multi-Party Threshold Schemes

Too many acronyms, we know. (Legend further below)

Subcategory: Type	**Families of specifications**	**NIST references**
C1.1: **Signing**	EdDSA sign, ECDSA sign, RSADSA sign	FIPS 186-5 (see also NISTIR 8214B)
C1.2: **PKE**	RSA decrypt, RSA encrypt (a secret value)	SP 800-56B Rev2
C1.3: **2KA**	ECC-CDH, ECC-MQV	SP 800-56A Rev3
C1.4: **Symmetric**	AES encipher/decipher, KDM/KC (for 2KE)	FIPS 197, SP 800-56C Rev2, ...
C1.5: **Keygen**	ECC keygen, RSA keygen, bitstring keygen	(corresponding references above)

Legend: **2KA**: pair-wise key-agreement. **2KE**: pair-wise key-establishment. **AES**: Advanced Encryption Standard. **CDH**: cofactor Diffie–Hellman. **ECC**: Elliptic-curve cryptography (or, if used as an adjective, EC-based). **ECDSA**: Elliptic-curve Digital Signature Algorithm. **EdDSA**: Edwards-curve Digital Signature Algorithm. Elliptic-curve based Key-Establishment. **FIPS**: Federal Information Processing Standard. **KC**: Key-confirmtion. **KDM**: Key-derivation mechanism. **Keygen**: Key-generation. **MQV**: Menezes-Qu-Vanstone. **PKE**: public-key encryption. **RSA**: Rivest–Shamir–Adleman (signature and encryption schemes). **RSADSA**: **RSA** digital signature algorithm. **SP 800**: Special Publication (in Computer Security). **Note:** In the 2nd column, each item within a subcategory is itself called a family of specifications, since it may include diverse primitives or modes/variants.

Also to be added to Category Cat1

Primitives from NIST draft standards emerging from the PQC and LWC projects:

- **ML-KEM** (based on KYBER) Draft FIPS 203: ***M**odule-**L**attice-Based KEM Standard*
- **ML-DSA** (based on DILITHIUM) Draft FIPS 204: ***M**odule-**L**attice-Based DSA*
- **SLH-DSA** (based on SPHINCS) Draft FIPS 205: ***S**tate**l**ess **H**ash-Based DSA*
- **FN-DSA** (based on Falcon): Upcoming Draft FIPS
- **AEAD and XOF standards** (based on ASCON): Upcoming Special Publication(s)

Legend: AEAD = **A**uthenticated **E**ncryption with **A**ssociated **D**ata. DSA = **D**igital **S**ignature **A**lgorithm. FIPS = **F**ederal **I**nformation **P**rocessing **S**tandard [Publication]. KEM = **K**ey-**E**ncapsulation **M**echanism. ML = **M**odule **L**attice. SLH = **S**tate**L**ess **h**ash. XOF = e**x**tendable **O**utput **F**unction.

Presented at
ICMC 2023

Category Cat2 of the NIST “Threshold” Call

Subcategory: Type
C2.1: **Signing**
C2.2: **PKE**
C2.3: **Key-agreem.**
C2.4: **Symmetric**
C2.5: **Keygen**

Note: While TF-QR is desired for any type of scheme, some examples show just TF to highlight that it is welcome even if not QR.

Legend: agreem. = **agreem**ent. **Keygen** = **key-gen**eration. **PKE** = **p**ublic-**k**ey **e**ncryption. **PRF** = **p**seudo**r**andom **f**unction [family]. **PRP** = **p**seudo**r**andom **p**ermutation [family]. **QR** = **q**uantum **r**esistant. **TF** = **t**hreshold-**f**riendly. **ZKPoK** = **z**ero **k**nowledge **p**roof **o**f **k**nowledge.

Category Cat2 of the NIST "Threshold" Call

TF = **t**hreshold **f**riendly. QR = **q**uantum **r**esistant.

Subcategory: Type	**Example types of schemes**	**Example primitives**
C2.1: **Signing**	TF succinct & verifiably-deterministic signatures	Sign
│	TF-QR signatures	Sign

Note: While TF-QR is desired for any type of scheme, some examples show just TF to highlight that it is welcome even if not QR.

Legend: agreem. = **agreem**ent. **Keygen** = **key-gen**eration. **PKE** = **p**ublic-**k**ey **e**ncryption. **PRF** = **p**seudo**r**andom **f**unction [family]. **PRP** = **p**seudo**r**andom **p**ermutation [family]. **QR** = **q**uantum **r**esistant. **TF** = **t**hreshold-**f**riendly. **ZKPoK** = **z**ero **k**nowledge **p**roof **o**f **k**nowledge.

Category Cat2 of the NIST “Threshold” Call

Subcategory: Type
C2.6: **Advanced**
C2.7: **ZKPoK**
C2.8: **Gadgets**

Note: While TF-QR is desired for any type of scheme, some examples show just TF to highlight that it is welcome even if not QR.

Legend: **agreem.** = **agreem**ent. **Keygen** = **key-gen**eration. **PKE** = **p**ublic-**k**ey **e**ncryption. **PRF** = **p**seudo**r**andom **f**unction [family]. **PRP** = **p**seudo**r**andom **p**ermutation [family]. **QR** = **q**uantum **r**esistant. **TF** = **t**hreshold-**f**riendly. **ZKPoK** = **z**ero **k**nowledge **p**roof **o**f **k**nowledge.

Category Cat2 of the NIST “Threshold” Call

TF = **t**hreshold **f**riendly. QR = **q**uantum **r**esistant.

Subcategory: Type	**Example types of schemes**	**Example primitives**
C2.6: **Advanced**	TF-QR fully-homomorphic encryption	Decryption; Keygen
	TF identity-based and attribute-based encryption	Decryption; Keygens

Note: While TF-QR is desired for any type of scheme, some examples show just TF to highlight that it is welcome even if not QR.

Legend: agreem. = **agreem**ent. **Keygen** = **key-gen**eration. **PKE** = **p**ublic-**k**ey **e**ncryption. **PRF** = **p**seudo**r**andom **f**unction [family]. **PRP** = **p**seudo**r**andom **p**ermutation [family]. **QR** = **q**uantum **r**esistant. **TF** = **t**hreshold-**f**riendly. **ZKPoK** = **z**ero **k**nowledge **p**roof **o**f **k**nowledge.

Category Cat2 of the NIST "Threshold" Call

Subcategory: Type	**Example types of schemes**	**Example primitives**
C2.7: **ZKPoK**	**Z**ero-**k**nowledge **p**roof **o**f **k**nowledge of private key	ZKPoK.Generate

Note: While TF-QR is desired for any type of scheme, some examples show just TF to highlight that it is welcome even if not QR.

Legend: **agreem.** = **agreem**ent. **Keygen** = **key**-**gen**eration. **PKE** = **p**ublic-**k**ey **e**ncryption. **PRF** = **p**seudo**r**andom **f**unction [family]. **PRP** = **p**seudo**r**andom **p**ermutation [family]. **QR** = **q**uantum **r**esistant. **TF** = **t**hreshold-**f**riendly. **ZKPoK** = **z**ero **k**nowledge **p**roof **o**f **k**nowledge.

Category Cat2 of the NIST "Threshold" Call

Subcategory: Type	Example types of schemes	Example primitives
C2.8: **Gadgets**	Garbled circuit (GC)	GC.generate; GC.evaluate

Note: While TF-QR is desired for any type of scheme, some examples show just TF to highlight that it is welcome even if not QR.

Legend: **agreem.** = **agreem**ent. **Keygen** = **key**-**gen**eration. **PKE** = **p**ublic-**k**ey **e**ncryption. **PRF** = **p**seudo**r**andom **f**unction [family]. **PRP** = **p**seudo**r**andom **p**ermutation [family]. **QR** = **q**uantum **r**esistant. **TF** = **t**hreshold-**f**riendly. **ZKPoK** = **z**ero **k**nowledge **p**roof **o**f **k**nowledge.

Category Cat2 of the NIST “Threshold” Call

TF = **t**hreshold **f**riendly. QR = **q**uantum **r**esistant.

Subcategory: Type	**Example types of schemes**	**Example primitives**
C2.1: **Signing**	TF succinct & verifiably-deterministic signatures	Sign
|	TF-QR signatures	Sign
C2.2: **PKE**	TF-QR **p**ublic-**k**ey **e**ncryption (PKE)	Decrypt/Encrypt (a secret value)
C2.3: **Key-agreem.**	TF Low-round multi-party key-agreement	Single-party primitives
C2.4: **Symmetric**	TF blockcipher/PRP	Encipher/decipher
|	TF key-derivation / key-confirmation	PRF and hash function
C2.5: **Keygen**	Any of the above	Keygen
C2.6: **Advanced**	TF-QR fully-homomorphic encryption	Decryption; Keygen
|	TF identity-based and attribute-based encryption	Decryption; Keygens
C2.7: **ZKPoK**	**Z**ero-**k**nowledge **p**roof **o**f **k**nowledge of private key	ZKPoK.Generate
C2.8: **Gadgets**	Garbled circuit (GC)	GC.generate; GC.evaluate

Note: While TF-QR is desired for any type of scheme, some examples show just TF to highlight that it is welcome even if not QR.

Legend: **agreem.** = **agreem**ent. **Keygen** = **key-gen**eration. **PKE** = **p**ublic-**k**ey **e**ncryption. **PRF** = **p**seudo**r**andom **f**unction [family]. **PRP** = **p**seudo**r**andom **p**ermutation [family]. **QR** = **q**uantum **r**esistant. **TF** = **t**hreshold-**f**riendly. **ZKPoK** = **z**ero **k**nowledge **p**roof **o**f **k**nowledge.

Main components of a submission package

Check	#	Item
□	M1	Written specification (S1–S16)
□	M2	Reference implementation (Src1–Src4)
□	M3	Execution instructions (X1–X7)
□	M4	Experimental evaluation (Perf1–Perf5)
□	M5	Additional statements

Main components of a submission package

Check	#	Item
□	M1	Written specification (S1–S16)
□	M2	Reference implementation (Src1–Src4)
□	M3	Execution instructions (X1–X7)
□	M4	Experimental evaluation (Perf1–Perf5)
□	M5	Additional statements

The revised version of the call will detail better each **component**.

A submission package can propose various **objects** (schemes/gadgets).

Each **component** will then map all such **objects**.

Some technical notes

1. **Submission focuses**
2. **Threshold profile**
3. **Active security**
4. **Adaptive security**
5. **Modularity**
6. **Post-vs-Pre quantum crypto**

Some technical notes

1. **Submission focuses:** can specify a family of schemes (in various subcategories).
2. **Threshold profile:** open to choice (number of parties; dishonest proportion; ...)
3. **Active security:** is required, though open to diverse security formulations.
4. **Adaptive security:** at least "argued for" for major safety properties.
5. **Modularity:** modularize gadgets; encouraged proactive resharing module; ...
6. **Post-vs-Pre quantum crypto:** both in scope; pre-quantum needs justification.

Concluding remarks

Selected takeaways

- The “Threshold Call” has a **wide scope** of subcategories for submission
- Enables an **exploration** of advanced cryptography, before promising standards
- The initial process will devise **recommendations** for subsequent processes
- Community **participation** is essential (feedback; submissions; analyses)

Thank you for your attention! Questions?

NIST Call for Multi-Party Threshold Schemes Brief Notes at ICMC 2023

Presented at ICMC 2023 | September 22nd @ Ottawa, Canada

We appreociate followup comments: luis.brandao@nist.gov

Threshold Call (Draft)

MPTS 2023 (Sept. 26–28)

MPTC-Forum (email list)

PEC-Forum (email list)