On Devising Recommendations for Multi-Party Threshold Schemes

Presented* in Singapore, on September 12th, 2023 at

DeCompute 2023

* Luís Brandão: At NIST as a Foreign Guest Researcher (non-employee), Contractor from Strativia.

 Joint work with René Peralta.

Outline

1. NIST Intro
2. NIST project on PEC and Threshold Crypto
3. The Threshold Call
4. A process toward recommendations

(Slides will be publicly available)

Crypto = Cryptography. NIST = National Institute of Standards and Technology. PEC = Privacy-Enhancing Crytpography.

Outline

1. NIST Intro
2. NIST project on PEC and Threshold Crypto
3. The Threshold Call
4. A process toward recommendations

Crypto = Cryptography. NIST = National Institute of Standards and Technology. PEC = Privacy-Enhancing Crytpography.

NIST: Laboratories → Divisions → Groups

- **Non-regulatory** federal agency (@ U.S. Dept. Commerce)
- **Mission:** ... innovation ... industrial competitiveness ... measurement science, standards, and technology ... economic security ... quality of life.

NIST name and address plate (source: nist.gov)

Presented at
DeCompute 2023

NIST: Laboratories → Divisions → Groups

- **Non-regulatory** federal agency (@ U.S. Dept. Commerce)
- **Mission:** ... innovation ... industrial competitiveness ... measurement science, standards, and technology ... economic security ... quality of life.

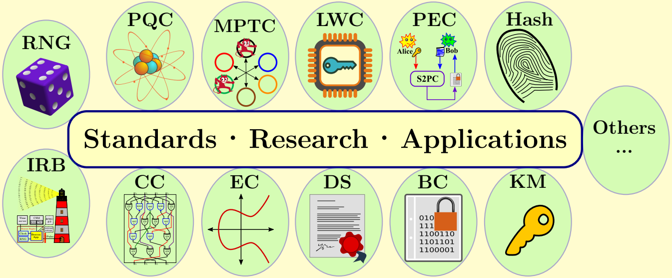
NIST name and address plate (source: nist.gov)

INFORMATION TECHNOLOGY LABORATORY → **Computer Security Division (CSD):**

→ **Cryptographic Technology Group (CTG):** *research, develop, engineer, and produce guidelines, recommendations and best practices for cryptographic algorithms, methods, and protocols.*

Presented at DeCompute 2023

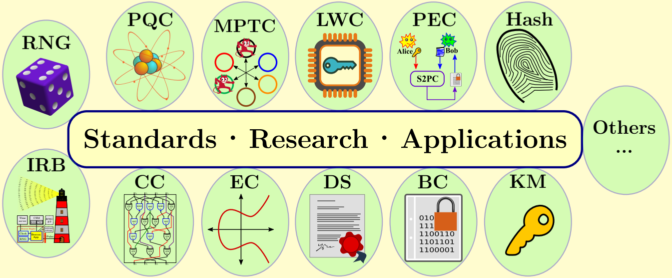
Activities in the “Crypto” Group



Legend: **BC** = **B**lock **C**iphers. **CC** = **C**ircuit **C**omplexity. **Crypto** = **Crypto**graphy. **DS** = **D**igital **S**ignatures. **EC** = **E**lliptic **C**urves. **FIPS** = **F**ederal **I**nformation **P**rocessing **S**tandards. **IR** = **I**nternal or **I**nteragency (denoting that the public NIST report was developed internally at NIST or in an interagency collaboration, respectively. **IRB** = **I**nteroperable **R**andomness **B**eacons. **KM** = **K**ey **M**anagement. **LWC** = **L**ight**w**eight **C**rypto. **PEC** = **P**rivacy-**E**nhancing **C**rypto. **PQC** = **P**ost-**Q**uantum **C**rypto. **RNG** = **R**andom-**N**umber **G**eneration. **SP 800** = **S**pecial **P**ublications in Computer Security. **TC** = [Multi-Party] **T**hreshold **C**rypto).

More details at https://www.nist.gov/itl/csd/cryptographic-technology

Activities in the "Crypto" Group



- **Public documentation:** FIPS; Special Publications (SP 800); NIST Reports (IR).
- **International cooperation:** government, industry, academia, standardization bodies.

Legend: **BC** = **B**lock **C**iphers. **CC** = **C**ircuit **C**omplexity. **Crypto** = **Crypto**graphy. **DS** = **D**igital **S**ignatures. **EC** = **E**lliptic **C**urves. **FIPS** = **F**ederal **I**nformation **P**rocessing **S**tandards. **IR** = **I**nternal or **I**nteragency (denoting that the public NIST report was developed internally at NIST or in an interagency collaboration, respectively. **IRB** = **I**nteroperable **R**andomness **B**eacons. **KM** = **K**ey **M**anagement. **LWC** = **L**ight**w**eight **C**rypto. **PEC** = **P**rivacy-**E**nhancing **C**rypto. **PQC** = **P**ost-**Q**uantum **C**rypto. **RNG** = **R**andom-**N**umber **G**eneration. **SP 800** = **S**pecial **P**ublications in Computer Security. **TC** = [Multi-Party] **T**hreshold **C**rypto).

More details at https://www.nist.gov/itl/csd/cryptographic-technology

Presented at
DeCompute 2023

Some examples of NIST Crypto 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** = **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.

Some examples of NIST Crypto 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 = **Auth**[enticated] **Enc**[ryption] w[ith] **Assoc**[iated] **Data**. **CTG** = **C**ryptographic **T**echnology **G**roup. **LWC** = **L**ightweight **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.

Some examples of NIST Crypto 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** = **C**ryptographic **T**echnology **G**roup. **LWC** = **L**ightweight **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.

Some examples of NIST Crypto Projects

- **PQC:** [standardization] "**post-quantum**" signatures and key-encapsulation
- **LWC:** [standardization] "**lightweight**" auth. enc. w/ assoc. data, 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** = 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.

Outline

1. NIST Intro
2. NIST project on PEC and Threshold Crypto
3. The Threshold Call
4. A process toward recommendations

Crypto = Cryptography. NIST = National Institute of Standards and Technology. PEC = Privacy-Enhancing Crytpography.

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	**S**tructured **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**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)

- 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

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; ...

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**unctio**n**al **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

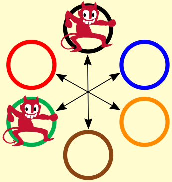
Encryption


KeyGen


Hashing

etc.

Threshold schemes (for cryptographic primitives):



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

Presented at
DeCompute 2023

Multi-Party Threshold Cryptography: NIST project

Cryptographic primitives:


Signing

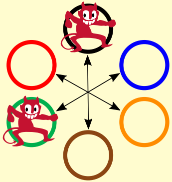
Encryption


KeyGen


Hashing

etc.

Threshold schemes (for cryptographic primitives):

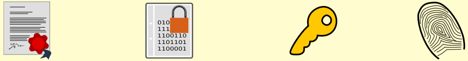


1. Split (**secret-share**) the secret/private-key across multiple parties.
2. Use **MPC** to perform needed operation (with split key), e.g., sign.
   (MPC = secure multiparty computation ... or call it "Threshold Cryptography")

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

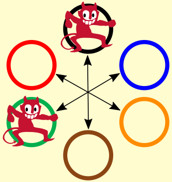
Multi-Party Threshold Cryptography: NIST project

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

Presented at
DeCompute 2023

Outline

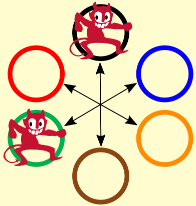
Crypto = Cryptography. NIST = National Institute of Standards and Technology. PEC = Privacy-Enhancing Crytpography.

NIST Call for Multi-Party Threshold Schemes

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

NIST Call for Multi-Party Threshold Schemes

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

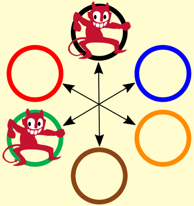


Calling for submissions of threshold schemes

(And gadgets for modular use)

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
DeCompute 2023

Category Cat1 of NIST Call for Multi-Party Threshold Schemes

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

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

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)

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.

Presented at
DeCompute 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
C1.2: **PKE**	RSA decrypt, RSA encrypt (a secret value)	SP 800-56B 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**: **Key-c**onfirmtion. **KDM**: **Key-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.

Presented at
DeCompute 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-curve **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.

Presented at
DeCompute 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
C1.1: **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 **k**ey-**a**greement. **2KE**: **pair**-wise **k**ey-**e**stablisment. **AES**: **A**dvanced **E**ncryption **S**tandard. **CDH**: cofactor **D**iffie–**H**ellman. **ECC**: **E**lliptic-**c**urve **c**ryptography (or, if used as an adjective, **EC**-based). **ECDSA**: **E**lliptic-curve **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.

Presented at
DeCompute 2023

Also to be added to Category Cat1

Primitives specified in NIST draft pubs 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*
- **Upcoming signature standard** (based on Falcon): Upcoming Draft FIPS pub
- **Upcoming XOF standard** (based on ASCON): Upcoming Draft FIPS pub

Legend: DSA = **d**igital **s**ignature **a**lgorithm. FIPS = **F**ederal **I**nformation **P**rocessing **S**tandard [Publication]. KEM = **k**ey-**e**ncapsulation **m**ethod. XOF = **ex**tendable **o**utput **f**unction. ML = **M**odule **L**attice. SLH = **S**tate**L**ess **h**ash.

Presented at
DeCompute 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.

Presented at DeCompute 2023

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.

Presented at DeCompute 2023

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.

Outline

1. NIST Intro
2. NIST project on PEC and Threshold Crypto
3. The Threshold Call
4. A process toward recommendations

Crypto = Cryptography. NIST = National Institute of Standards and Technology. PEC = Privacy-Enhancing Crytpography.

Assorted 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, ...)

Some technical notes

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

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:** it 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-QC requires justification.
7. **Concrete implementation:** e.g., inc. communication (e.g., broadcast? P2P?).

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)

Upcoming: (Sep 26–28) Workshop on Multi-Party Threshold Schemes (MPTS) 2023

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

Thank you for your attention! Questions?

On Devising Recommendations for Multi-Party Threshold Schemes

Presented at DeCompute 2023 | September 12th @ Singapore

Followup comments appreciated: luis.brandao@nist.gov

Threshold Call (Draft)

MPTS 2023 (Sept. 26–28)

MPTC-Forum (email list)

PEC-Forum (email list)

Slide intentionally blank

Example ZKPoKs of interest (related to Cat1)

Related type	Related (sub)sub-category: Primitive	Example ZKPoK (including consistency with public commitments of secret-shares, when applicable)
Signing	C1.1.1/2: EC-signing	of pre-image of deterministic nonce (if applicable)
Keygen	C5.1.1: ECC keygen	of discrete-log (s or d) of pub key Q
|	C5.1.2: RSA keygen	of factors (p, q), or group order ϕ, or decryption key d
|	C5.1.3: AES keygen	of secret key k (with regard to secret-sharing commitments)
PKE	C1.2.1: RSA encryption	of secret plaintext m (encrypted)
|	C1.2.2: RSA decryption	of secret-shared plaintext m (after SSO-threshold decryption)
Symmetric	C1.4.1: AES enciphering	of secret key k (with regard to plaintext/ciphertext pair)
|	C1.4.2: Hashing in KDM	of secret pre-image Z

Source: Table 12 or NISTIR 8214C ipd

Vision: techniques demonstrated for these cases also enable more generic cases

Auxiliary slide

Legend: AES = Advanced Encryption Standard. Cat1 = Category 1. ECC = Elliptic-Curve Cryptography. ipd = initial public draft. keygen = key-generation. KDM = key-derivation mechanism. RSA = Rivest-Shamir Adleman. SSO = Secret-

Some expected revisions in the call

1. In Cat1, add subcategories for the NIST-selected PQC primitives
2. In Cat2, differentiate better the advanced subcats (e.g., what to thresholdize)
3. Clarify scope of “gadgets” subcategory (and how to motivate them)
4. Detail better some logistic requirements (e.g., code licensing)
5. Include LaTeX template for submission

Public comments received in first phase (April 2023)

#	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.

https://csrc.nist.gov/files/pubs/ir/8214/c/ipd/docs/nistir-8214c-ipd-public-feedback.pdf

Brainstorming on crypto standardization

1. On the **timing** & **speed** of processes: what is too soon, too late, too slow, and too fast?
2. What **value** is there in still pursuing new standards for **quantum-breakable** primitives?
3. How to handle the standardization tension between **innovation** and **interoperability**?
4. Which crypto functionalities/features make sense to **prioritize** for standardization?
5. What **synergies** to aim for between academia, industry, gov and standards bodies?

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As there was s
unprocessed data that should have been added to the final page this extra pag
been added to receive it.

If you rerun the document (without altering it) this surplus page will go away,
LaTeX now knows how many pages to expect for this document.