NIST Projects on Threshold and Privacy-Enhancing Crypto

Opportunities for Post-Quantum Cryptography

Presented* at IHP-PQAC Workshop 2

Emerging Topics in Design and Cryptanalysis of Post-Quantum Schemes

November 07, 2024 | Paris (France)

* **Luís Brandão:** NIST Associate (Foreign Guest Researcher, non-employee), Contractor from Strativia. Expressed opinions are from the speaker. Work by the PEC team is joint with R. Peralta and A. Robinson; Work by the Threshold Crypto team is joint with R. Peralta and M. Davidson.

Legend: NIST = National Institute of Standards and Technology. IHP = Institut Henri Poincaré. PQAC = Post-quantum Algebraic Cryptography.

About this Presentation

1. **Merci** pour l'invitation à présenter à l'atelier IHP-PQAC-W2
   IHP = Institut Henri Poincaré; PQAC = Post-quantum Algebraic Cryptography.
   W2 = Workshop 2: Emerging Topics in Design and Cryptanalysis of Post-Quantum Schemes

2. **Goals:**
   - Convey a perspective from two *exploratory* NIST projects
   - Suggest topics where PQC has a place in this exploration
   - Questions for feedback from the audience

3. The slide-deck will be **publicly available**

Outline

1. NIST Crypto Intro
2. NIST PEC and Threshold Crypto
3. The Threshold Call
4. PQC Opportunities in the Threshold Call
5. Interaction and Feedback

NIST = National Institute of Standards and Technology. PEC = Privacy-Enhancing Cryptography. PQC = Post-Quantum Cryptography.

Outline

NIST = National Institute of Standards and Technology. PEC = Privacy-Enhancing Cryptography. PQC = Post-Quantum Cryptography.

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)

NIST: Laboratories → Divisions → Groups

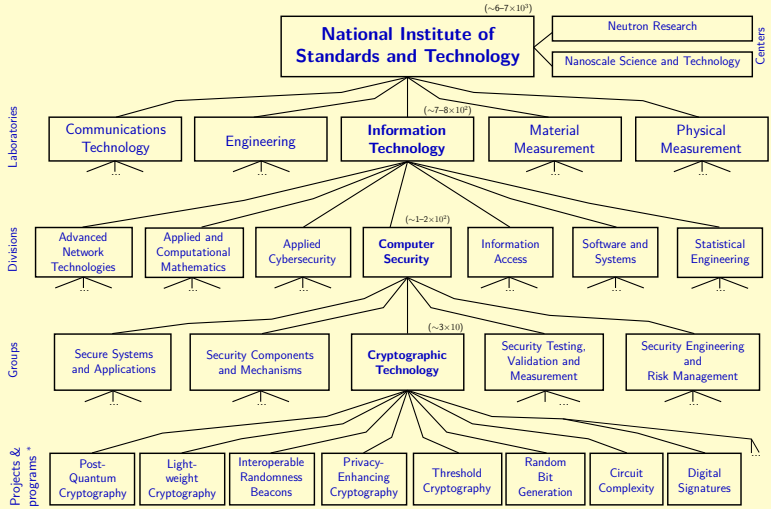
- **Non-regulatory** federal agency (@ U.S. Dept. Commerce)
- **Mission:** ... innovation ... industrial competitiveness ... measurement science, standards, and technology ... economic security ... quality of life.

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.*

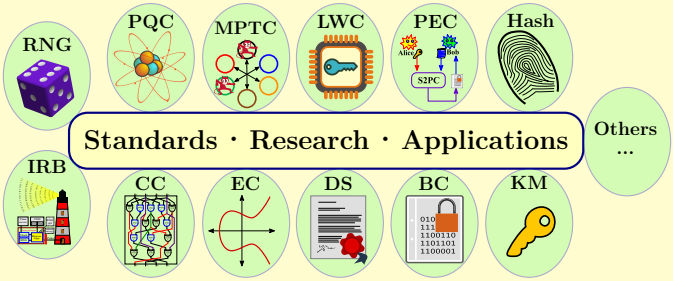
The Crypto Group within the NIST organization



* (Some projects / programs involve various groups, divisions or labs.)

(in parenthesis: estimate (2019) of approximate number of people, inc. employees and associates)

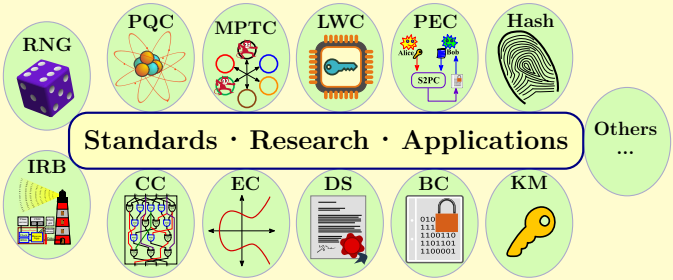
Activities in the “Crypto” Group



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. **MPTC** = **M**ulti-**P**arty **T**hreshold **C**rypto). **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.

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. **MPTC** = **M**ulti-**P**arty **T**hreshold **C**rypto). **LWC** = **L**ightweight **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.

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

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 = **A**uth[enticated] **Enc**[ryption] w[ith] **Assoc**[iated] **D**ata.**LWC** = **L**ightweight **C**ryptography. **MPTC** = **M**ulti-**P**arty **T**hreshold **C**ryptography. **PEC** = **P**rivacy-**E**nhancing **C**ryptography. **PQC** = **P**ost-**Q**uantum **C**ryptography.

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 others https://www.nist.gov/itl/csd/cryptographic-technology

Legend: AEAD = **A**uth[enticated] **E**nc[ryption] w[ith] **A**ssoc[iated] **D**ata. **LWC** = **L**ightweight **C**ryptography. **MPTC** = **M**ulti-**P**arty **T**hreshold **C**ryptography. **PEC** = **P**rivacy-**E**nhancing **C**ryptography. **PQC** = **P**ost-**Q**uantum **C**ryptography.

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 others https://www.nist.gov/itl/csd/cryptographic-technology

This presentation is focused on the Exploratory perspective
(gather **reference material** for public analysis ... to inform subsequent steps)

Legend: AEAD = **A**uth[enticated] **Enc**[ryption] w[ith] **Assoc**[iated] **Data**.**LWC** = **L**ightweight **C**ryptography. **MPTC** = **M**ulti-**P**arty **T**hreshold **C**ryptography. **PEC** = **P**rivacy-**E**nhancing **C**ryptography. **PQC** = **P**ost-**Q**uantum **C**ryptography.

“Exploratory ...” and the prefix “Standard”

- “**Standard<ABC>**”:

“Exploratory ...” and the prefix “Standard”

- “**Standard**<**ABC**>”:
  - **Standard**: a **specification** that can be followed
  - **Standard**ization: a **process**, including prior to deciding what to “standardize”
  - **Standard**(ization) **bodies** / communities
  - **Standard**ization-related activities: research, workshops, write/review standards

Next section: Two **exploratory** projects ...

Preparing ground for future processes that might consider standardization efforts.

Outline

NIST = National Institute of Standards and Technology. PEC = Privacy-Enhancing Cryptography. PQC = Post-Quantum Cryptography.

Privacy-Enhancing Cryptography (PEC): NIST Project

Cryptography usable to **enhance privacy**.

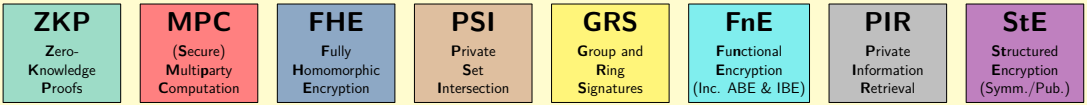
(emphasis on non-standardized tools)

Privacy-Enhancing Cryptography (PEC): NIST Project

Cryptography usable to **enhance privacy**.
(emphasis on non-standardized tools)

Goals:

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



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

Privacy-Enhancing Cryptography (PEC): NIST Project

Cryptography usable 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
Fully-Homomophic Encryption (FHE)
Zero-Knowledge Proof (ZKP)
Multi-Party Computation (MPC)
STPPA (Series of Talks)
PEC Use-Case Suite
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	**Fn**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. PEC: privacy-enhancing cryptography. Symm./pub.: symmetric-key or public-key based.

Privacy-Enhancing Cryptography (PEC): NIST Project

Cryptography usable 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
Fully-Homomophic Encryption (FHE)
Zero-Knowledge Proof (ZKP)
Multi-Party Computation (MPC)
STPPA (Series of Talks)
PEC Use-Case Suite
Encounter Metrics
Email List (PEC Forum)

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

ZKP	MPC	FHE	PSI	GRS	FnE	PIR	StE
Zero-Knowledge Proofs	(Secure) Multiparty Computation	Fully Homomorphic Encryption	Private Set Intersection	Group and Ring Signatures	Functional Encryption (Inc. ABE & IBE)	Private Information Retrieval	Structured Encryption (Symm./Pub.)

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

Multi-Party Threshold Cryptography: NIST project

Cryptographic primitives:


Signing

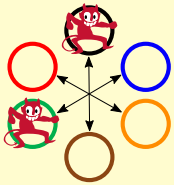
Encryption


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

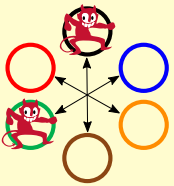

KeyGen


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

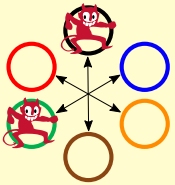
Encryption


KeyGen


Hashing

etc.

Threshold schemes (for cryptographic primitives):



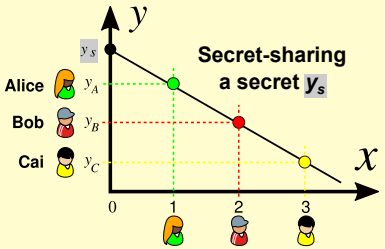
1. Split (**secret-share**) the secret/private-key across multiple parties.
2. Use **MPC** to perform needed operation (with split key), e.g., sign.
   (MPC = secure multiparty computation ... or call it "Threshold Cryptography")

- **"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

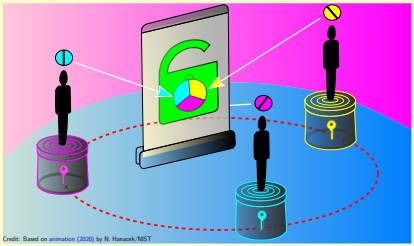
Basics of a Threshold Scheme

Secret-sharing:



Splits the key into secret shares.

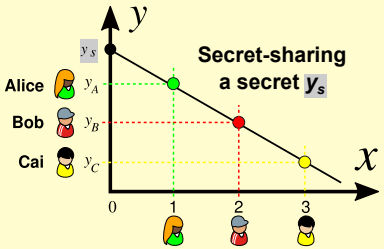
Multi-party computation (MPC)



Operates without recombining the key.

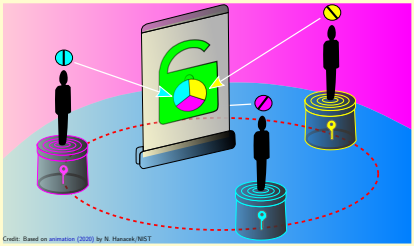
Basics of a Threshold Scheme

Secret-sharing:

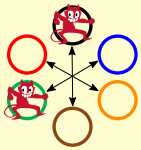


Splits the key into secret shares.

Multi-party computation (MPC)



Operates without recombining the key.

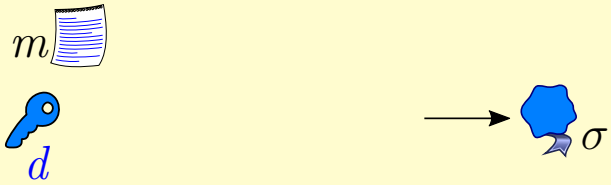


Participation threshold: the operation needs k parties in agreement.

Corruption threshold: system secure even if f parties are malicious.

Simple(st) Example: Threshold n-of-n RSA signatures

Threshold signature:

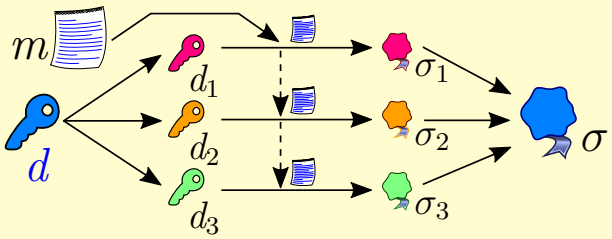


Textbook RSA:

- **Private** signing key: d
- **Priv.** $\phi = (p-1) \times (q-1)$
- **Public:** $N = p \cdot q$; $e =_\phi d^{-1}$
- **Signature:** $\sigma =_N m^d$

Simple(st) Example: Threshold n-of-n RSA signatures

Threshold signature:

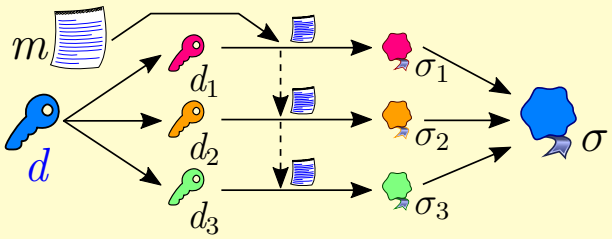


Textbook RSA:

- **Private** signing key: d
- **Priv.** $\phi = (p-1) \times (q-1)$
- **Public:** $N = p \cdot q$; $e =_\phi d^{-1}$
- **Signature:** $\sigma =_N m^d$

Simple(st) Example: Threshold n-of-n RSA signatures

Threshold signature:



Textbook RSA:

- **Private** signing key: d
- **Priv.** $\phi = (p-1) \times (q-1)$
- **Public:** $N = p \cdot q$; $e =_\phi d^{-1}$
- **Signature:** $\sigma =_N m^d$

1. **Secret-share the key** d**:** $d \to d_1, d_2, d_3 : d_1 + d_2 + d_3 = d \bmod \phi$
2. **Produce partial signatures:** $\sigma_i = m^{d_i} \bmod N$, for $i = 1, 2, 3$
3. **Obtain final signature:** $\sigma = \sigma_1 \cdot \sigma_2 \cdot \sigma_3 = m^{d_1+d_2+d_3} = m^d \bmod N$

T. Schemes can get more complicated than T. RSA

- **Threshold EdDSA/Schnorr:** Commitments, ZKPs, ...
- **Threshold ECDSA, distributed RSA KeyGen:** Oblivous transfer, AHE, ...
- **Threshold AES:** Garbled circuits, oblivious transfer, ...
- **Other building blocks:** Reliable broadcast, threshold-friendly hash functions, ...

T. Schemes can get more complicated than T. RSA

- **Threshold EdDSA/Schnorr:** Commitments, ZKPs, ...
- **Threshold ECDSA, distributed RSA KeyGen:** Oblivous transfer, AHE, ...
- **Threshold AES:** Garbled circuits, oblivious transfer, ...
- **Other building blocks:** Reliable broadcast, threshold-friendly hash functions, ...

Other primitives (not standardized by NIST) can be more ***threshold friendly***

(easier in practice to thresholdize, or amenable to "better" threshold schemes)

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:

- Many options (model, assumptions): which are most useful?
- Improving signal-to-noise (identifying sound crypto technologies)
- 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:

- Many options (model, assumptions): which are most useful?
- Improving signal-to-noise (identifying sound crypto technologies)
- Which primitives are ***threshold-friendlier***? (easier in practice to thresholdize or, amenable to "better" threshold schemes)

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

Why care about / explore PEC and threshold schemes?

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

Hesitations / need for recommendations:

- Many options (model, assumptions): which are most useful?
- Improving signal-to-noise (identifying sound crypto technologies)
- 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?

Why care about / explore PEC and threshold schemes?

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

Hesitations / need for recommendations:

- Many options (model, assumptions): which are most useful?
- Improving signal-to-noise (identifying sound crypto technologies)
- 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: A public **Call** for reference material ... toward **recommendations**

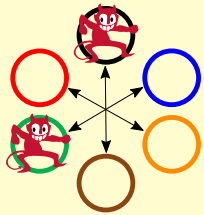
Outline

NIST = National Institute of Standards and Technology. PEC = Privacy-Enhancing Cryptography. PQC = Post-Quantum Cryptography.

The NIST Call for *Multi-Party Threshold Schemes*

NISTIR 8214C ipd (**i**nitial **p**ublic **d**raft) [Jan 2023]. Soon 2nd public draft.

Calling for submissions of threshold schemes for:

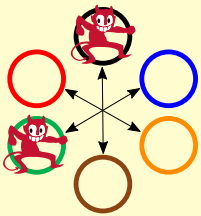


The NIST Call for *Multi-Party Threshold Schemes*

NISTIR 8214C ipd (**i**nitial **p**ublic **d**raft) [Jan 2023]. Soon 2nd public draft.

Calling for submissions of threshold schemes for:

- **[Cat1] Selected NIST-standardized primitives**
  In EdDSA, ECDSA, RSA, AES, KeyGen, ...
- **[Cat2] Other primitives (including FHE, ZKP)**

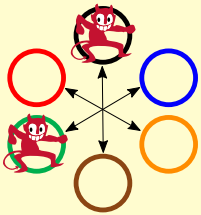


Legend: AES = **A**dvanced **E**ncryption **S**tandard. EC = **E**lliptic curve. FHE = **f**ully-**h**omomorphic **e**ncryption. EdDSA = **Ed**wards-Curve **d**igital **s**ignature **a**lgorithm. ECDSA = **EC** **d**igital **s**ignature **a**lgorithm. RSA = **R**ivest–**S**hamir–**A**dleman. ZKP = **z**ero-**k**nowledge **p**roofs.

The NIST Call for *Multi-Party Threshold Schemes*

NISTIR 8214C ipd (**i**nitial **p**ublic **d**raft) [Jan 2023]. Soon 2nd public draft.

Calling for submissions of threshold schemes for:



- **[Cat1] Selected NIST-standardized primitives**
  In EdDSA, ECDSA, RSA, AES, KeyGen, ...
- **[Cat2] Other primitives (including FHE, ZKP)**
  (And gadgets for modular use)

Legend: AES = **A**dvanced **E**ncryption **S**tandard. EC = **E**lliptic curve. FHE = **f**ully-**h**omomorphic **e**ncryption. EdDSA = **Ed**wards-Curve **d**igital **s**ignature **a**lgorithm. ECDSA = **EC** **d**igital **s**ignature **a**lgorithm. RSA = **R**ivest–**S**hamir–**A**dleman. ZKP = **z**ero-**k**nowledge **p**roofs.

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**

Many acronyms: **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. **FN-DSA**: **F**alcon-Based DSA. **Keygen**: **Key-gen**eration. **MAC**: **M**essage **A**uthentication **C**ode. **ML-DSA**: Module-Lattice-Based DSA. **ML-KEM**: Module-Lattice-Based Key-Encapsulation Method. **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. **SLH-DSA**: Stateless Hash-Based DSA. **SP 800**: **S**pecial **P**ublication (in Computer Security). **XOF**: E**X**tendable **O**utput **F**unction. **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

Subcategory: Type	Families of specifications	NIST references
C1.1: **Signing**	EdDSA sign, ECDSA sign, RSADSA sign ML-DSA sign, SLH-DSA sign, FN-DSA sign	FIPS 186-5 (see also NISTIR 8214B) FIPS 204, 205, 206 (to appear)

Many acronyms: **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. **FN-DSA**: **F**alcon-Based DSA. **Keygen**: **Key-gen**eration. **MAC**: **M**essage **A**uthentication **C**ode. **ML-DSA**: Module-Lattice-Based DSA. **ML-KEM**: Module-Lattice-Based Key-Encapsulation Method. **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. **SLH-DSA**: Stateless Hash-Based DSA. **SP 800**: **S**pecial **P**ublication (in Computer Security). **XOF**: E**X**tendable **O**utput **F**unction. **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

Subcategory: Type	Families of specifications	NIST references
C1.2: **PKE**	RSA decrypt, encrypt (a secret value) ML-KEM decrypt, encrypt (a secret value)	SP 800-56B Rev2 FIPS 203

Many acronyms: **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. **FN-DSA**: **F**alcon-Based DSA. **Keygen**: **Key-gen**eration. **MAC**: **M**essage **A**uthentication **C**ode. **ML-DSA**: Module-Lattice-Based DSA. **ML-KEM**: Module-Lattice-Based Key-Encapsulation Method. **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. **SLH-DSA**: Stateless Hash-Based DSA. **SP 800**: **S**pecial **P**ublication (in Computer Security). **XOF**: E**X**tendable **O**utput **F**unction. **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

Subcategory: Type	Families of specifications	NIST references
C1.4: **Symmetric**	AES & ASCON enc/decipher, MAC/Hash/XOF	FIPS 197, SP 800-56C Rev2, ...

Many acronyms: **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. **FN-DSA**: **F**alcon-Based DSA. **Keygen**: **Key**-**gen**eration. **MAC**: **M**essage **A**uthentication **C**ode. **ML-DSA**: Module-Lattice-Based DSA. **ML-KEM**: Module-Lattice-Based Key-Encapsulation Method. **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. **SLH-DSA**: Stateless Hash-Based DSA. **SP 800**: **S**pecial **P**ublication (in Computer Security). **XOF**: E**X**tendable **O**utput **F**unction. **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

Subcategory: Type	Families of specifications	NIST references
C1.1: **Signing**	EdDSA sign, ECDSA sign, RSADSA sign ML-DSA sign, SLH-DSA sign, FN-DSA sign	FIPS 186-5 (see also NISTIR 8214B) FIPS 204, 205, 206 (to appear)
C1.2: **PKE**	RSA decrypt, encrypt (a secret value) ML-KEM decrypt, encrypt (a secret value)	SP 800-56B Rev2 FIPS 203
C1.3: **2KA**	EC-CDH, EC-MQV	SP 800-56A Rev3
C1.4: **Symmetric**	AES & ASCON enc/decipher, MAC/Hash/XOF	FIPS 197, SP 800-56C Rev2, ...
C1.5: **Keygen**	EC keygen, RSA keygen, bitstring keygen	(corresponding references above)

Many acronyms: **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. **FN-DSA**: **F**alcon-Based DSA. **Keygen**: **Key-gen**eration. **MAC**: **M**essage **A**uthentication **C**ode. **ML-DSA**: Module-Lattice-Based DSA. **ML-KEM**: Module-Lattice-Based Key-Encapsulation Method. **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. **SLH-DSA**: Stateless Hash-Based DSA. **SP 800**: **S**pecial **P**ublication (in Computer Security). **XOF**: E**X**tendable **O**utput **F**unction. **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 Cat2 of the NIST “Threshold” Call

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

Legend: agreem. = **agreem**ent. **Keygen** = **key-gen**eration. **PKE** = **p**ublic-key **e**ncryption. **PRF** = **p**seudorandom **f**unction [family]. **PRP** = **p**seudorandom **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

Category Cat2 of the NIST “Threshold” Call

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

Legend: **agreem.** = **agreem**ent. **Keygen** = **key-gen**eration. **PKE** = **p**ublic-**k**ey **e**ncryption. **PRF** = **p**seudorandom **f**unction [family]. **PRP** = **p**seudorandom **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: **FHE**	TF-QR fully-homomorphic encryption	Decryption; Keygen

Legend: agreem. = **agreem**ent. **Keygen** = **key-gen**eration. **PKE** = **p**ublic-**k**ey **e**ncryption. **PRF** = **p**seudorandom **f**unction [family]. **PRP** = **p**seudorandom **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

Legend: **agreem.** = **agreem**ent. **Keygen** = **key-gen**eration. **PKE** = **p**ublic-**k**ey **e**ncryption. **PRF** = **p**seudorandom **f**unction [family]. **PRP** = **p**seudorandom **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

Legend: agreem. = agreement. Keygen = key-generation. PKE = public-key encryption. PRF = pseudorandom function [family]. PRP = pseudorandom permutation [family]. QR = quantum resistant. TF = threshold-friendly. ZKPoK = zero knowledge proof of knowledge.

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 PRP (e.g., blockcipher)	Encipher/decipher
	TF PRF (e.g., for MAC or KD)	MAC, KDF
|	TF Hash or XOF	Hash function, XOF
C2.5: **Keygen**	Any of the above	Keygen
C2.6: **FHE**	TF-QR fully-homomorphic encryption	Decryption; Keygen
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**seudorandom **f**unction [family]. **PRP** = **p**seudorandom **p**ermutation [family]. **QR** = **q**uantum **r**esistant. **TF** = **t**hreshold-**f**riendly. **ZKPoK** = **z**ero **k**nowledge **p**roof **o**f **k**nowledge.

Components of a submission package

1. **Written specification**
2. **Reference implementation** (open-source)
3. **Evaluation**

Components of a submission package

1. **Written specification**
2. **Reference implementation** (open-source)
3. **Evaluation**

Upcoming revision of the Call:

- It's been taking time (longer than initially projected). Delay well received.
- Expected deadlines: **abstracts** (1st half 2025), **packages** (2nd half 2025).
- Refined submission logistics (e.g., allow multiple crypto-systems per package).
- Revised requirement of open source (allow external dependencies).

Assorted notes

- **Setup:** A gathering of reference material (not a competition for a selection).
- **Interchangeability:** Threshold result usable as if it was conventionally generated. See our IR 8214B (notes on Threshold Schnorr/EdDSA).
- **Expected:** The process will clarify relevant system models, best practices, ...
- **Threshold-friendliness:** a perspective beyond usual efficiency.
- **Aim:** Devise recommendations about advanced cryptography (PEC + MPTC). (Will support future processes.)

  PEC = Privacy-Enhancing Crypto
  MPTC = Multi-Party Threshold Crypto
- **Ample room for participation:** Give feedback → Submit → Analyze.

Some technical notes

1. **Submission focuses**
2. **Threshold profile**
3. **Active security**
4. **Adaptive security**
5. **Modularity**
6. **Concrete implementation**
7. **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:** 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. **Concrete implementation:** communication (broadcast? P2P?), ...
7. **Post-vs-Pre quantum crypto:** both in scope; Pre-QC requires justification.

Outline

NIST = National Institute of Standards and Technology. PEC = Privacy-Enhancing Cryptography. PQC = Post-Quantum Cryptography.

PQC in the Threshold Call Scope

1. PQC versus Pre-QC, for Threshold Scheme and Primitive
2. Subcategories for PQC in the Call
3. Notes on Threshold signatures
4. Notes on ZKPs
5. Notes on FHE
6. More from PEC

Matrix of PQC versus preQC

Are all combinations worth exploring?

Threshold Scheme	Primitive	
	PreQC	PQC
PreQC	?	?
PQC	?	?

Legend: DKG = Distributed Key-Generation. PreQC/PQC= Pre-/Post-Quantum Crypto. QR = Quantum resistance. TS = Threshold Scheme.

Matrix of PQC versus preQC

Threshold Scheme	Primitive	
	PreQC	PQC
PreQC	?	?
PQC	?	?

Are all combinations worth exploring?

Obvious cases:

- **PreQC TS for PreQC Primitive:**
- **PQC TS for PQC Primitive:**

Legend: DKG = Distributed Key-Generation. PreQC/PQC= Pre-/Post-Quantum Crypto. QR = Quantum resistance. TS = Threshold Scheme.

Matrix of PQC versus preQC

Threshold Scheme	Primitive	
	PreQC	PQC
PreQC	?	?
PQC	?	?

Are all combinations worth exploring?

Obvious cases:

- **PreQC TS for PreQC Primitive:** Not add new assumptions.
- **PQC TS for PQC Primitive:** Retain QR for entire operation.

Legend: DKG = Distributed Key-Generation. PreQC/PQC= Pre-/Post-Quantum Crypto. QR = Quantum resistance. TS = Threshold Scheme.

Matrix of PQC versus preQC

Threshold Scheme	Primitive	
	PreQC	PQC
PreQC	?	?
PQC	?	?

Are all combinations worth exploring?

Obvious cases:

- **PreQC TS for PreQC Primitive:** Not add new assumptions.
- **PQC TS for PQC Primitive:** Retain QR for entire operation.

Less obvious (consider carefully):

- **PreQC TS for PQC Primitive:**
- **(Require) PQC TS for PreQC Primitive**:

Legend: DKG = Distributed Key-Generation. PreQC/PQC= Pre-/Post-Quantum Crypto. QR = Quantum resistance. TS = Threshold Scheme.

Matrix of PQC versus preQC

Threshold Scheme	Primitive	
	PreQC	PQC
PreQC	?	?
PQC	?	?

Are all combinations worth exploring?

Obvious cases:

- **PreQC TS for PreQC Primitive:** Not add new assumptions.
- **PQC TS for PQC Primitive:** Retain QR for entire operation.

Less obvious (consider carefully):

- **PreQC TS for PQC Primitive:** If more efficient, if QR is only needed for the output (e.g., a signature), if the parties are assumed PreQC.
- **(Require) PQC TS for PreQC Primitive**:

Legend: DKG = Distributed Key-Generation. PreQC/PQC= Pre-/Post-Quantum Crypto. QR = Quantum resistance. TS = Threshold Scheme.

Matrix of PQC versus preQC

Threshold Scheme	Primitive	
	PreQC	PQC
PreQC	?	?
PQC	?	?

Are all combinations worth exploring?

Obvious cases:

- **PreQC TS for PreQC Primitive:** Not add new assumptions.
- **PQC TS for PQC Primitive:** Retain QR for entire operation.

Less obvious (consider carefully):

- **PreQC TS for PQC Primitive:** If more efficient, if QR is only needed for the output (e.g., a signature), if the parties are assumed PreQC.
- **(Require) PQC TS for PreQC Primitive**: E.g., for Threshold-as-a-service (e.g., for DKG), with PQ parties, where material remains secret-shared.

Legend: DKG = Distributed Key-Generation. PreQC/PQC= Pre-/Post-Quantum Crypto. QR = Quantum resistance. TS = Threshold Scheme.

PQC opportunities across the subcategories

- [C1.1–2] **Signatures/PKE:** QR T-Schemes for NIST standardized PQC
- [C2.1–2] **Signatures/PKE:** T-friendlier than NIST-PQC ones, or new features (e.g., Verifiably deterministic)
- [C1.4] **Symmetric:** QR-MPC for T-Schemes for PQC **symmetric** primitives
- [C2.4] **Symmetric:** QR Threshold/ZKP/MPC/FHE friendly hash function/PRP
- [C2.6] **FHE:** known lattice-based constructions; new constructions?
- [C2.7] **ZKP:** Succinct PQ-ZKPoKs about keys of PQC primitives?
- [C2.8] **Building blocks:** QR constructions.

On Threshold Signatures (PQC)

- **Verifiable determinism?**
  - Randomness Beacon use case. Any PQC-alternatives to Pre-QC RSA and BLS?
  - PQC threshold-friendly PRF for *interchangeable* deterministic signature?
- **MPC-in-the-Head** approach does not appear to be MPC/threshold friendly.
- **T-friendliness:** How are NIST-selected PQC signatures, vs. other PQC ones?
- Compare with "Threshold EdDSA/Schnorr" considerations (NISTIR 8214B):
  - **Security goals:** active + adaptive; strong unforgeability; ...
  - ***Biasiavity*:** usually tied to 2-rounds vs. 3 rounds, rushing adversary, abort conditions.
  - **Setup assumptions:** e.g., PKI established during DKG, broadcast, synchronicity, ...
  - **Termination options:** identifiable abort, robustness,

Interest in Zero-Knowledge Proofs (ZKP)

Focus on **ZKPoK** of secret keys [examples below]. Vision: may enable more generic cases.

Based on Table 12 of NISTIR 8214C ipd (with some adaptations ... more to appear in the final version)

Cat	Related type	Related (sub)sub-category: Primitive	Example ZKPoK (including consistency with public commitments of secret-shares, when applicable)
Cat1	Keygen	1.5.1.1: EC keygen	of discrete-log (s or d) of pub key Q
|	|	1.5.1.2: RSA keygen	of factors (p, q), or group order ϕ, or decryption key d
|	|	1.5.1.3: AES keygen	of secret key k (with regard to secret-sharing commitments)
|	Signature	1.1.1: EdDSA signing	of pre-image μ of deterministic secret nonce r (committed by R)
|	PKE	1.2.1: RSA encryption	of secret plaintext m (encrypted)
|	|	1.2.2: RSA decryption	of secret-shared plaintext m (after SSO-threshold decryption)
|	Symmetric	1.4.1: AES enciphering	of secret key k (with regard to plaintext/ciphertext pair)
|	|	1.4.2: Hashing in KDM	of secret pre-image Z
Cat2	FHE	2.6: FHE encryption	of secret plaintext m (encrypted) also committed in another form
|	|	2.6: FHE evaluation	of correct computation (not necessarily ZK)

Legend: AES = Advanced Encryption Standard. Cat1/2 = Category 1/2. ECC = Elliptic-Curve Cryptography. FHE = Fully-homomorphic encryption. ipd = Initial public draft. keygen = Key-generation. KDM = key-derivation mechanism. RSA = Rivest-Shamir Adleman. SSO = Secret-shared output. NISTIR = NIST Internal[ly developed, public] Report. ZKPoK = Zero-knowledge proof of knowledge.

Example FHE use-case

AES oblivious evaluation

AES = Advanced Encryption Standard
FHE = Fully-Homomorphic Encryption

1. Client FHE-encrypts a plaintext message
2. Server with AES-key homomorphically-evaluates the AES enciphering
3. Client FHE-decrypts the result to obtain the AES-ciphertext

AES is a blockcipher. E.g., AES-128 as Boolean circuit has 6400 ANDs and $\approx$ 22K XOR.

Example FHE use-case

AES oblivious evaluation

AES = Advanced Encryption Standard
FHE = Fully-Homomorphic Encryption

1. Client FHE-encrypts a plaintext message
2. Server with AES-key homomorphically-evaluates the AES enciphering
3. Client FHE-decrypts the result to obtain the AES-ciphertext

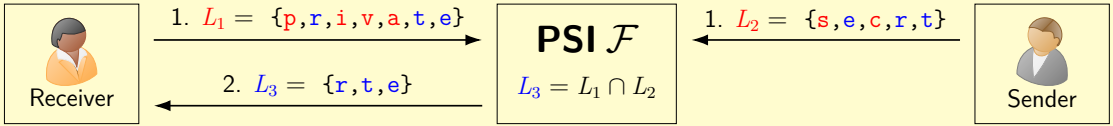
AES is a blockcipher. E.g., AES-128 as Boolean circuit has 6400 ANDs and $\approx$ 22K XOR.

Conceivably thresholdizable?

- FHE-keygen and **FHE-decryption** (with secret-shared FHE decryption key)
- FHE encryption (and decryption) of secret-shared plaintext
- Homomorphic evaluation of "AES-enciphering with secret-shared AES key"

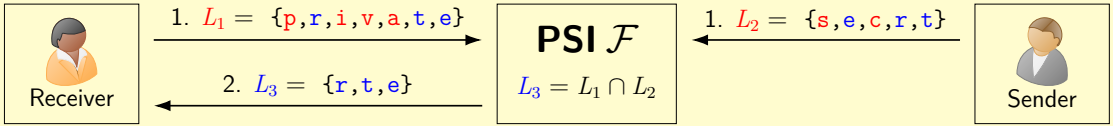
Another PEC example: Private-Set Intersection

Obtain the intersection of two sets, without disclosing the non-intersecting elements.



Another PEC example: Private-Set Intersection

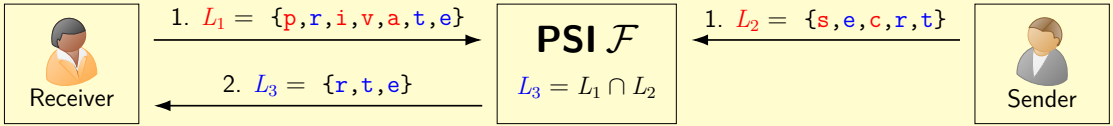
Obtain the intersection of two sets, without disclosing the non-intersecting elements.



- **Insecure:** Usual (non-cryptographer's) intuition: Compare hashes
- **Secure:** Compare (Oblivious) PRF outputs, e.g., via DHKE.

Another PEC example: Private-Set Intersection

Obtain the intersection of two sets, without disclosing the non-intersecting elements.



- **Insecure:** Usual (non-cryptographer's) intuition: Compare hashes
- **Secure:** Compare (Oblivious) PRF outputs, e.g., via DHKE.

Other notes:

- PQC: Oblivious PRF, offline/online optimizations, multi-party, ...
- MPC: Generalize to Circuit-PSI (function applied to the intersection)
- Check "The First PSI day" organized within WPEC 2024.

Outline

NIST = National Institute of Standards and Technology. PEC = Privacy-Enhancing Cryptography. PQC = Post-Quantum Cryptography.

Series of Talks

NIST hosts many talks by external researchers. Virtual attendance allowed.

- **NIST Crypto Reading Club:** crypto-club-questions@nist.gov
  https://csrc.nist.gov/projects/crypto-reading-club
  See also the "Other NIST-hosted Presentations" container.

- **NIST PQC Seminar:** pqc-seminars@nist.gov
  https://csrc.nist.gov/projects/post-quantum-cryptography/workshops-and-timeline/pqc-seminars

- **Special Topics on Privacy and Public Auditability:** pec-stppa@nist.gov
  https://csrc.nist.gov/projects/pec/stppa

To subscribe, check each webpage or contact us.

Learning process

- We'll learn from the filling of the **Threshold Map (submissions)**:
  - T-profiles, practicality, crypto assumptions, PQC-readiness, security formulations (UC/games), security properties (SUF, IND-CPA^{D}, ...).
  - Exploration of threshold schemes $\Rightarrow$ Insights on more generic MPC
  - PQC use cases in the Call $\Rightarrow$ Insights on PQC beyond regular signatures / PKE.
- **Not a competition,** but submissions will influence the developing process:
  - *Apples-to-apples* is not needed (API, metrics), but the Call requests rationale.
- **NIST Seminar series on Threshold Cryptography seminar:**
  - Will start in 2025, around the deadline for submitting abstracts.

The obtained reference material will inform our approach toward recommendations/standards

Cryptographers:

The Crypto Group can host visits

and/or consider integrating a Foreign

Guest Researcher ($\approx$ post-doc) expert on

MPC / FHE / ZKP / Threshold Crypto.

Contact: rene.peralta (at) nist.gov

Thank you for your attention!

- **Questions:** Let's alternate **from** and **to** the audience (next slide)

Threshold Call

MPTC-Forum

PEC-Forum

Brainstorming on Crypto Exploration/Standardization

Photo in 2018

The test of time: Which of today's developing standards will remain, 75 years from now, as building blocks of advanced crypto?

The NIST Stone Test Wall: "*Constructed [in 1948] to study the performance of stone subjected to weathering. It contains 2352 individual samples of stone, of which 2032 are domestic stone from 47 states, and 320 are stones from 16 foreign countries.*"

Brainstorming on Crypto Exploration/Standardization

1. **Timing** & **speed** of processes: what is too soon, too late, too slow, and too fast?

Photo in 2018

The test of time: Which of today's developing standards will remain, 75 years from now, as building blocks of advanced crypto?

The NIST Stone Test Wall: "*Constructed [in 1948] to study the performance of stone subjected to weathering. It contains 2352 individual samples of stone, of which 2032 are domestic stone from 47 states, and 320 are stones from 16 foreign countries.*"

Brainstorming on Crypto Exploration/Standardization

1. **Timing** & **speed** of processes: what is too soon, too late, too slow, and too fast?
2. **Value** in still pursuing new standards for **quantum-breakable** primitives?

Photo in 2018

The test of time: Which of today's developing standards will remain, 75 years from now, as building blocks of advanced crypto?

The NIST Stone Test Wall: "*Constructed [in 1948] to study the performance of stone subjected to weathering. It contains 2352 individual samples of stone, of which 2032 are domestic stone from 47 states, and 320 are stones from 16 foreign countries.*"

Brainstorming on Crypto Exploration/Standardization

1. **Timing** & **speed** of processes: what is too soon, too late, too slow, and too fast?
2. **Value** in still pursuing new standards for **quantum-breakable** primitives?
3. Standardization tension between **innovation** and **interoperability**?

Photo in 2018

The test of time: Which of today's developing standards will remain, 75 years from now, as building blocks of advanced crypto?

The NIST Stone Test Wall: "*Constructed [in 1948] to study the performance of stone subjected to weathering. It contains 2352 individual samples of stone, of which 2032 are domestic stone from 47 states, and 320 are stones from 16 foreign countries.*"

Brainstorming on Crypto Exploration/Standardization

1. **Timing** & **speed** of processes: what is too soon, too late, too slow, and too fast?
2. **Value** in still pursuing new standards for **quantum-breakable** primitives?
3. Standardization tension between **innovation** and **interoperability**?
4. Crypto functionalities/features that should be **prioritized**/**focused on**?

Photo in 2018

The test of time: Which of today's developing standards will remain, 75 years from now, as building blocks of advanced crypto?

The NIST Stone Test Wall: "*Constructed [in 1948] to study the performance of stone subjected to weathering. It contains 2352 individual samples of stone, of which 2032 are domestic stone from 47 states, and 320 are stones from 16 foreign countries.*"

Brainstorming on Crypto Exploration/Standardization

1. **Timing** & **speed** of processes: what is too soon, too late, too slow, and too fast?
2. **Value** in still pursuing new standards for **quantum-breakable** primitives?
3. Standardization tension between **innovation** and **interoperability**?
4. Crypto functionalities/features that should be **prioritized**/**focused on**?
5. **Vision** of future standards/recommendations (5 years, 20 years, 75 year)?

Photo in 2018

The test of time: Which of today's developing standards will remain, 75 years from now, as building blocks of advanced crypto?

The NIST Stone Test Wall: "*Constructed [in 1948] to study the performance of stone subjected to weathering. It contains 2352 individual samples of stone, of which 2032 are domestic stone from 47 states, and 320 are stones from 16 foreign countries.*"