A NIST Call for Threshold-Friendly & Quantum-Resistant Fully-Homomorphic Encryption (FHE) Schemes

Cryptographic **T**echnology **G**roup
National **I**nstitute of **S**tandards and **T**echnology (NIST)

*Presented at the 6th HomomorphicEncryption.org Standards Meeting
March 23, 2023 @ Seoul (South Korea)

Suggested reading: NISTIR 8214C ipd: *NIST First Call for Multi-Party Threshold Schemes (Initial Public Draft)* [Jan. 2023]

* Luís Brandão: At NIST as a Foreign Guest Researcher (non-employee), Contractor from Strativia. Expressed opinions are from the speaker and should not be construed as official NIST views. Joint work with René Peralta. Minor editorial updates on 2023-March-28.

Outline

1. **Introduction:** NIST/PEC/Threshold
2. **The "Threshold" Call and FHE**
3. **Concluding remarks**

(Slides will be made publicly available)

FHE = fully-homomorphic encryption. NIST = National Institute of Standards and Technology. PEC = privacy-enhancing cryptography.

Outline

1. **Introduction:** NIST/PEC/Threshold
2. **The "Threshold" Call and FHE**
3. **Concluding remarks**

FHE = fully-homomorphic encryption. NIST = National Institute of Standards and Technology. PEC = privacy-enhancing 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.*

Modern/advanced cryptography

Tradition: for long, NIST has had standards for building blocks for "traditional" data security.

	Traditional	
Data status	*At rest* or *In transit*	
Operation being secured	Storage or Communication	
Example **crypto** primitives	Encryption, Signatures, Hashing	
NIST crypto standards today?	**Yes**	

Modern/advanced cryptography

Tradition: for long, NIST has had standards for building blocks for "traditional" data security.

	Traditional	**Advanced**
Data status	*At rest* or *In transit*	*In use*
Operation being secured	Storage or Communication	**Computation**
Example **crypto** primitives	Encryption, Signatures, Hashing	MPC, HE, ZKP
NIST crypto standards today?	**Yes**	**No**

Legend: HE = homomorphic encryption; MP = multi-party; MPC = (secure) MP computation; ZKP = zero-knowledge proof

Modern/advanced cryptography

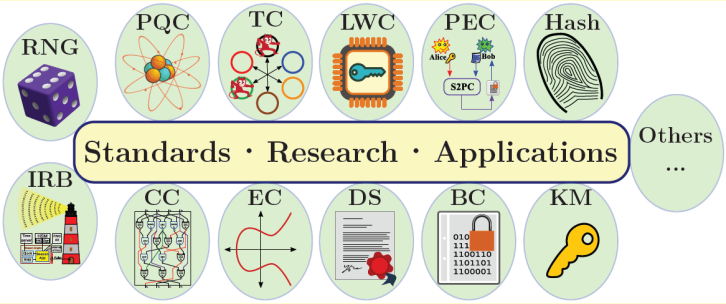
Tradition: for long, NIST has had standards for building blocks for "traditional" data security.

	Traditional	**Advanced**
Data status	*At rest* or *In transit*	*In use*
Operation being secured	Storage or Communication	**Computation**
Example **crypto** primitives	Encryption, Signatures, Hashing	MPC, HE, ZKP
NIST crypto standards today?	**Yes**	**No**

Legend: HE = homomorphic encryption; MP = multi-party; MPC = (secure) MP computation; ZKP = zero-knowledge proof

Modernization: the "Call for MP Threshold Schemes" is connected to **advanced cryptography**.

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**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. **TC** = [Multi-Party] **T**hreshold **C**rypto).

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

Privacy-Enhancing Cryptography (PEC): NIST project

- A project in the **NIST Cryptographic Technology Group**
- **PEC: 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

Privacy-Enhancing Cryptography (PEC): NIST project

- A project in the **NIST Cryptographic Technology Group**
- **PEC: cryptography** (that can be) used to **enhance privacy**.

[emphasis on non-standardized tools]

Goals:

1. Accompany the progress of **emerging *PEC 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

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	**Fu**nctional **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-enhnacing cryptography. Symm./pub.: symmetric-key or public-key based.

Privacy-Enhancing Cryptography (PEC): NIST project

- A project in the **NIST Cryptographic Technology Group**
- **PEC: 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**ultiparty **C**omputation	**F**ully **H**omomorphic **E**ncryption	**P**rivate **S**et **I**ntersection	**G**roup and **R**ing **S**ignatures	**Fu**nctional **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-enhnacing cryptography. Symm./pub.: symmetric-key or public-key based.

Multi-Party Threshold Cryptography: NIST project

Cryptographic primitives:


Signing

Encryption

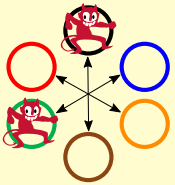

KeyGen


Hashing

etc.

Threshold schemes (for cryptographic primitives):

1. Split (**secret-share**) the secret/private-key across multiple parties.
2. Use **MPC** to perform needed operation (with split key), e.g., decrypt.
   (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

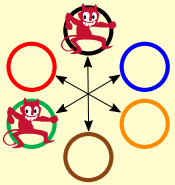
Encryption


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., decrypt.
   (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 **(never reconstructed)**: avoids single-point of failure.

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

Why care about threshold schemes?

Strong feasibility result (theory): can be applied to any cryptographic primitive.

But, in practice, some primitives are ***threshold-friendlier**** than others.

(* i.e., informally, easier in practice to thresholdize, or amenable to more efficient threshold schemes)

Why care about threshold schemes?

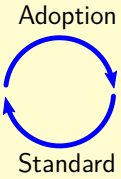
Strong feasibility result (theory): can be applied to any cryptographic primitive.

But, in practice, some primitives are ***threshold-friendlier**** than others.

(* i.e., informally, easier in practice to thresholdize, or amenable to more efficient threshold schemes)

- Standards "should" focus on high need and potential for **adoption**
- **Threshold friendliness:** desirable feature → improves **adoptability**

  (e.g., determ. vs. prob. threshold EdDSA/Schnorr signatures [NISTIR 8214B ipd])



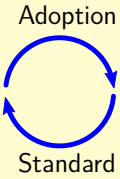
Why care about threshold schemes?

Strong feasibility result (theory): can be applied to any cryptographic primitive.

But, in practice, some primitives are ***threshold-friendlier**** than others.

(* i.e., informally, easier in practice to thresholdize, or amenable to more efficient threshold schemes)

- Standards "should" focus on high need and potential for **adoption**
- **Threshold friendliness:** desirable feature → improves **adoptability**
  (e.g., determ. vs. prob. threshold EdDSA/Schnorr signatures [NISTIR 8214B ipd])



How to explore the threshold space?:

- applicable to a **wide scope** of primitives
- which brings **added complexity**

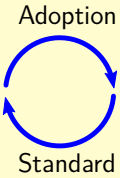
Why care about threshold schemes?

Strong feasibility result (theory): can be applied to any cryptographic primitive.

But, in practice, some primitives are ***threshold-friendlier**** than others.

(* i.e., informally, easier in practice to thresholdize, or amenable to more efficient threshold schemes)

- Standards "should" focus on high need and potential for **adoption**
- **Threshold friendliness:** desirable feature → improves **adoptability**
  (e.g., determ. vs. prob. threshold EdDSA/Schnorr signatures [NISTIR 8214B ipd])



How to explore the threshold space?:

- applicable to a **wide scope** of primitives
- which brings **added complexity**

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

Outline

1. **Introduction:** NIST/PEC/Threshold
2. **The "Threshold" Call and FHE**
3. **Concluding remarks**

FHE = fully-homomorphic encryption. NIST = National Institute of Standards and Technology. PEC = privacy-enhancing cryptography.

NIST Call for Multi-Party Threshold Schemes

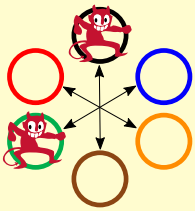
NISTIR 8214C ipd (initial public **draft**) — public comments till **2023-April-10**

Calling for threshold schemes for diverse primitives:

- **Cat1: Selected NIST-standardized primitives**

  EdDSA, ECDSA, RSA, AES, ECC-KE, ...

- **Cat2: Primitives not specified by NIST**
  - Interest in **threshold friendliness** and **quantum resistance**
  - Interest in advanced features, from PEC “tools”: **FHE**, IBE, ZKP, ...



AES = Advanced Encryption Standard. EC = Elliptic curve. ECC-KE = EC cryptography (based) key-exchange. EdDSA = Edwards-Curve digital signature algorithm. ECDSA = EC digital signature algorithm. FHE = Fully-homomorphic encryption. IBE = Identity-based encryption. NIST = National Institute of Standards and Technology. PEC = Privacy-enhancing cryptography. RSA = Rivest–Shamir–Adleman. ZKP = Zero-knowledge proof.

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 a desired combination 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.

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 a desired combination 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.

Category Cat2 of the NIST “Threshold” Call

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

Note: While TF-QR is a desired combination 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.

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 a desired combination 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.

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 a desired combination 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.

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 a desired combination 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.

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 **k**ey-**a**greement (KA)	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 a desired combination 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.

Confidence and expectations about FHE subcategory (C2.6.1)

- 5+ years of “HomomorphicEncryption.Org Standardization” (HES)
- Community efforts like HES are very useful $\Rightarrow$ FHE is in this call.

Confidence and expectations about FHE subcategory (C2.6.1)

- 5+ years of "HomomorphicEncryption.Org Standardization" (HES)
- Community efforts like HES are very useful $\Rightarrow$ FHE is in this call.

Welcome (and needed) interaction?

1. **Feedback about the call:** [initial comments by **April 10th, 2023**]
   a. positioning of FHE as an advanced primitive
   b. benchmarking use-cases vs. types of FHE and their thresholdizability

Confidence and expectations about FHE subcategory (C2.6.1)

- 5+ years of "HomomorphicEncryption.Org Standardization" (HES)
- Community efforts like HES are very useful $\Rightarrow$ FHE is in this call.

Welcome (and needed) interaction?

1. **Feedback about the call:** [initial comments by **April 10th, 2023**]
   a. positioning of FHE as an advanced primitive
   b. benchmarking use-cases vs. types of FHE and their thresholdizability
2. **Submissions of concrete FHE schemes and their threshold schemes:**
   – specified w/ concrete parameters, implemented (open source), reproducible, ...
3. **Public scrutiny of submitted schemes:**
   – will impact subsequent recommendations (processes and guidance)

Example FHE use-case

The draft call (§A.6.1) exemplifies one FHE use-case: AES oblivious evaluation

(AES = Advanced Encryption Standard)

Example FHE use-case

The draft call (§A.6.1) exemplifies one FHE use-case: AES oblivious evaluation

(AES = Advanced Encryption Standard)

1. Client FHE-encrypts a plaintext message
2. Server with AES-key homomorphically-evaluates the AESenciphering
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

The draft call (§A.6.1) exemplifies one FHE use-case: AES oblivious evaluation

(AES = Advanced Encryption Standard)

1. Client FHE-encrypts a plaintext message
2. Server with AES-key homomorphically-evaluates the AESenciphering
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.

What can conceivably be thresholdized (§A.6.2)?

- 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”

Example items of wanted feedback

(Things to consider when finalizing the call)

- Benchmarking use-cases across types of FHE
  - E.g., Boolean circuits; arithmetic circuits (large modulus); approximate computations; ...
- Use-cases for which primitives to thresholdize? (e.g., beyond keygen and decryption)
- Which FHE schemes are likely to be useful/ready to submit
- Expected security, in comparison with NIST-selected PQC primitives.

It is useful to hear these things publicly, from stakeholders.

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

- (Optional) early public abstract: 3 months after final call
- (Optional) preliminary submission to check completeness: $\approx$ 45 days before deadline
- Package-submission deadline: $\approx$ 5 months after final call

Tentative timeframes (subject to change)

Outline

1. **Introduction:** NIST/PEC/Threshold
2. **The "Threshold" Call and FHE**
3. **Concluding remarks**

FHE = fully-homomorphic encryption. NIST = National Institute of Standards and Technology. PEC = privacy-enhancing cryptography.

Assorted brief notes

The call covers other technicalities:

- Requirements about system model and security formulation
- Feedback: does FHE deserve some exception or add-on?
- ZKPs for FHE are also mentioned

More about the process:

- A submission can jointly cover a family of schemes
- How will this community compose teams for submission?
- Would a further clarification session/alignment be useful before the final call?

Concluding remarks

Intended progress

1. **Feedback** that helps improves the final call version, facilitating good submissions.
2. **Submissions** of FHE schemes along with their threshold schemes.
3. **Public analysis** clarifying for technical recommendations (and subsequent processes).

Concluding remarks

Intended progress

1. **Feedback** that helps improves the final call version, facilitating good submissions.
2. **Submissions** of FHE schemes along with their threshold schemes.
3. **Public analysis** clarifying for technical recommendations (and subsequent processes).

Ohter notes useful notes to recall:

- $\{$"threshold is useful" and "FHE $\in$ PEC" $\} \Rightarrow$ FHE subcategory in the threshold call.
- Not a competition for a selection, but rather a gathering of reference material.
- Work developed with other SDOs and in community efforts is also welcome.

Thank you for your attention! Questions?

A NIST Call for Threshold-Friendly & Quantum-Resistant Fully-Homomorphic Encryption (FHE) Schemes

Presented at the 6th HomomorphicEncryption.org Standards Meeting

luis.brandao@nist.gov — March 23, 2023 @ Seoul (South Korea)

- **NISTIR 8214C ipd:** *NIST First Call for Multi-Party Threshold Schemes (Initial Public Draft)*
- **Public comments:** send via email nistir-8214C-comments@nist.gov, by April 10th, 2013
- **PEC Website:** https://csrc.nist.gov/projects/pec
- **PEC-Forum:** https://list.nist.gov/PEC-forum
- **MPTC Website:** https://csrc.nist.gov/projects/threshold-cryptography
- **MPTC-Forum:** https://list.nist.gov/MPTC-forum