Developing Criteria for the Single-Device Track of the Threshold Cryptography Project at NIST

Luís Brandão* and Apostol Vassilev

National Institute of Standards and Technology
(Gaithersburg, USA)

Presentation on July 7, 2020, at the
Online Workshop on Threshold Schemes for NIST-approved Symmetric Block Ciphers in a Single-Device Setting
https://www.esat.kuleuven.be/cosic/events/tis-online-workshop

*Contractor at NIST as a Foreign Guest Researcher, employed by Strativia.

Outline

Goals of this presentation:

- Update on the Threshold Cryptography project
- Overview the new NISTIR 8214A (roadmap to criteria)
- Goals and pointers for structured feedback

Outline 1

1. Intro: NIST Crypto Standards and the Threshold approach
2. The Threshold Cryptography Project and the new NISTIR
3. Testing and Validation
4. Topics for a structured discussion
5. Concluding remarks

NIST: Laboratory → division → groups

Information Technology Laboratory (ITL):
advancing measurement science, standards, and technology through research and development in information technology, mathematics, and statistics.

NIST: Laboratory → division → groups

Information Technology Laboratory (ITL):
advancing measurement science, standards, and technology through research and development in information technology, mathematics, and statistics.

→ **Computer Security Division (CSD):** Cryptographic Technology; Secure Systems and Applications; Security Components and Mechanisms; Security Engineering and Risk Management; Security Testing, Validation and Measurement.

NIST: Laboratory → division → groups

Information Technology Laboratory (ITL):
advancing measurement science, standards, and technology through research and development in information technology, mathematics, and statistics.

→ **Computer Security Division (CSD):** Cryptographic Technology; Secure Systems and Applications; Security Components and Mechanisms; Security Engineering and Risk Management; Security Testing, Validation and Measurement.

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

→ **Security Testing, Validation and Measurement (STVM):** validate cryptographic algorithm implementations, cryptographic modules, [...] develop test suites and test methods; provide implementation guidance [...]

NIST: Laboratory → division → groups

Information Technology Laboratory (ITL):
advancing measurement science, standards, and technology through research and development in information technology, mathematics, and statistics.

→ **Computer Security Division (CSD):** Cryptographic Technology; Secure Systems and Applications; Security Components and Mechanisms; Security Engineering and Risk Management; Security Testing, Validation and Measurement.

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

→ **Security Testing, Validation and Measurement (STVM):** validate cryptographic algorithm implementations, cryptographic modules, [...] develop test suites and test methods; provide implementation guidance [...]

- Documents: FIPS, SP 800, NISTIR.
- International cooperation: government, industry, academia, standardization bodies.

FIPS = Federal Information Processing Standards; SP 800 = Special Publications in Computer Security; NISTIR = NIST Internal or Interagency Report.

NIST standardizes cryptographic primitives

Traditional focus on "basic" primitives:

NIST standardizes cryptographic primitives

Traditional focus on “basic” primitives:

- Block ciphers (e.g., AES, FIPS 197)
- Cipher modes of operation (SP 800-38 series)
- DRBGs (SP 800-90 series) and crypto key generation (SP 800-133)
- Hash functions (e.g., SHA2, FIPS 180-4; SHA3, FIPS 202)
- Signatures (FIPS 186-5), primitives for pair-wise key agreement (SP 800-56)

(Not an exhaustive list; Further details in “NIST Cryptographic Standards and Guidelines Development Program Briefing Book”)

NIST standardizes cryptographic primitives

Traditional focus on "basic" primitives:

- Block ciphers (e.g., AES, FIPS 197)
- Cipher modes of operation (SP 800-38 series)
- DRBGs (SP 800-90 series) and crypto key generation (SP 800-133)
- Hash functions (e.g., SHA2, FIPS 180-4; SHA3, FIPS 202)
- Signatures (FIPS 186-5), primitives for pair-wise key agreement (SP 800-56)

(Not an exhaustive list; Further details in "NIST Cryptographic Standards and Guidelines Development Program Briefing Book")

NIST standardizes cryptographic primitives

Traditional focus on "basic" primitives:

- Block ciphers (e.g., AES, FIPS 197)
- Cipher modes of operation (SP 800-38 series)
- DRBGs (SP 800-90 series) and crypto key generation (SP 800-133)
- Hash functions (e.g., SHA2, FIPS 180-4; SHA3, FIPS 202)
- Signatures (FIPS 186-5), primitives for pair-wise key agreement (SP 800-56)

(Not an exhaustive list; Further details in "NIST Cryptographic Standards and Guidelines Development Program Briefing Book")

Some guidance on Cryptography Standards:

- NISTIR 7977 (2016): NIST Cryptographic Standards and Guidelines Development Process
  Formalizes several **principles** to follow: transparency, openness, balance, integrity, technical merit, usability, global acceptability, continuous improvement, innovation and intellectual property (and overarching considerations)
- SP 800-175: Guideline for Using Cryptographic Standards in the Federal Government
- FIPS 140-3: Security Requirements for Cryptographic Modules

NIST standardizes cryptographic primitives

Traditional focus on "basic" primitives:

- Block ciphers (e.g., AES, FIPS 197)
- Cipher modes of operation (SP 800-38 series)
- DRBGs (SP 800-90 series) and crypto key generation (SP 800-133)
- Hash functions (e.g., SHA2, FIPS 180-4; SHA3, FIPS 202)
- Signatures (FIPS 186-5), primitives for pair-wise key agreement (SP 800-56)

(Not an exhaustive list; Further details in "NIST Cryptographic Standards and Guidelines Development Program Briefing Book")

Some guidance on Cryptography Standards:

- NISTIR 7977 (2016): NIST Cryptographic Standards and Guidelines Development Process
  Formalizes several **principles** to follow: transparency, openness, balance, integrity, technical merit, usability, global acceptability, continuous improvement, innovation and intellectual property (and overarching considerations)
- SP 800-175: Guideline for Using Cryptographic Standards in the Federal Government
- FIPS 140-3: Security Requirements for Cryptographic Modules

Development of new standards

Several methods to develop cryptography standards:

- Internal or interagency developed techniques
- Adoption of external standards
- Open call, competition, "competition-like"

Development of new standards

Several methods to develop cryptography standards:

- Internal or interagency developed techniques
- Adoption of external standards
- Open call, competition, "competition-like"

Examples of ongoing standardization projects:

- **Post-quantum cryptography:** signatures, public-key encryption, key encapsulation
- **Lightweight cryptography:** ciphers, authenticated encryption, hash functions
- **Threshold Cryptography:** threshold schemes for cryptographic primitives

Development of new standards

Several methods to develop cryptography standards:

- Internal or interagency developed techniques
- Adoption of external standards
- Open call, competition, “competition-like”

Examples of ongoing standardization projects:

- **Post-quantum cryptography:** signatures, public-key encryption, key encapsulation
- **Lightweight cryptography:** ciphers, authenticated encryption, hash functions
- **Threshold Cryptography:** threshold schemes for cryptographic primitives

This presentation:

- Threshold Cryptography project $\rightarrow$ “Single-device” track
- Loose use of “new standards” (may mean recommendations, guidelines, reference definitions, etc.) across various types of documentation. No promise implied.

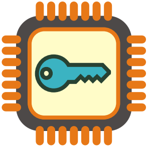
Beyond defining basic crypto primitives?

Security often hinges on a good application of cryptography

Beyond defining basic crypto primitives?

Security often hinges on a good application of cryptography

Specially relevant: key-based cryptographic primitives



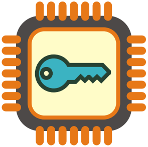
openclipart.org/detail/101407

Beyond defining basic crypto primitives?

Security often hinges on a good application of cryptography

Specially relevant: **key**-based cryptographic primitives

Security relies on:



openclipart.org/detail/101407

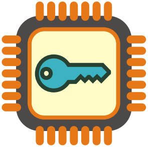
- secrecy, correctness, availability ... of cryptographic **keys**
- **implementations** that use **keys** to operate an algorithm
- **operators** to decide when/where to apply the algorithms

Beyond defining basic crypto primitives?

Security often hinges on a good application of cryptography

Specially relevant: **key**-based cryptographic primitives

Security relies on:



- secrecy, correctness, availability ... of cryptographic **keys**
- **implementations** that use **keys** to operate an algorithm
- **operators** to decide when/where to apply the algorithms

Some things can go wrong!

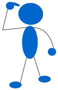
Crypto can be affected by vulnerabilities!

- Attacks can exploit differences between ideal vs. real **implementations**
- **Operators** of cryptographic implementations can go rogue

Crypto can be affected by vulnerabilities!

- Attacks can exploit differences between ideal vs. real **implementations**
- **Operators** of cryptographic implementations can go rogue

How to address single-points of failure?

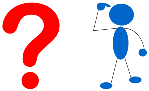


*question-2.html *4296.html
* = clker.com/clipart-

Crypto can be affected by vulnerabilities!

- Attacks can exploit differences between ideal vs. real **implementations**
- **Operators** of cryptographic implementations can go rogue

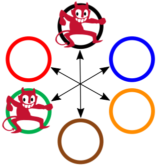
How to address single-points of failure?



*question-2.html *4296.html

* = clker.com/clipart-

The threshold approach



The red dancing devil is from clker.com/clipart-13643.html

At a high-level:

use redundancy & diversity

to mitigate the *compromise*

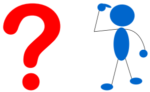
of up to a threshold number

(f-out-of-n) of components

Crypto can be affected by vulnerabilities!

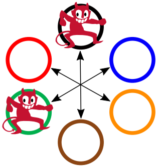
- Attacks can exploit differences between ideal vs. real **implementations**
- **Operators** of cryptographic implementations can go rogue

How to address single-points of failure?



*question-2.html *4296.html
* = clker.com/clipart-

The threshold approach



The red dancing devil is from clker.com/clipart-13643.html

At a high-level:

use redundancy & diversity to mitigate the *compromise* of up to a threshold number (f-out-of-n) of components

Two main platforms:

- **Single-device:** components (e.g., wires in a circuit) within a device
- **Multi-party:** distributed computation across separate devices

Threshold properties

- **withstands** up to f *compromised* components;
- **needs** the participation of at least k *un*compromised components;
- **prevents** the bits of the secret key from being in one place;
- **enhances** resistance against side-channel attacks; ...

Threshold properties

- **withstands** up to f *compromised* components;
- **needs** the participation of at least k *uncompromised* components;
- **prevents** the bits of the secret key from being in one place;
- **enhances** resistance against side-channel attacks; ...

Example: 3-out-of-3 enciphering:

- **Availability:** 3 nodes needed to encipher
- **Key secrecy:** okay while 1 share is secret

clker.com/clipart-encryption.html

Threshold properties

- **withstands** up to f *compromised* components;
- **needs** the participation of at least k *un*compromised components;
- **prevents** the bits of the secret key from being in one place;
- **enhances** resistance against side-channel attacks; ...

Example: 3-out-of-3 enciphering:

- **Availability:** 3 nodes needed to encipher $(k = 3,\ f = 0)$
- **Key secrecy:** okay while 1 share is secret

clker.com/clipart-encryption.html

(Each security property has its own k and f)

Threshold properties

- **withstands** up to f *compromised* components;
- **needs** the participation of at least k *<u>un</u>compromised* components;
- **prevents** the bits of the secret key from being in one place;
- **enhances** resistance against side-channel attacks; ...

Example: 3-out-of-3 enciphering:

- **Availability:** 3 nodes needed to encipher $(k = 3, \ f = 0)$
- **Key secrecy:** okay while 1 share is secret $(k = 1, \ f = 2)$

clker.com/clipart-encryption.html

(Each security property has its own k and f)

But "k-out-of-n" or "f-out-of-n" is not a sufficient characterization for a comprehensive security assertion

Threshold properties

- **withstands** up to f *compromised* components;
- **needs** the participation of at least k *uncompromised* components;
- **prevents** the bits of the secret key from being in one place;
- **enhances** resistance against side-channel attacks; ...

Example: 3-out-of-3 enciphering:

- **Availability:** 3 nodes needed to encipher ($k = 3$, $f = 0$)
- **Key secrecy:** okay while 1 share is secret ($k = 1$, $f = 2$)

clker.com/clipart-encryption.html

(Each security property has its own k and f)

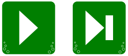
But "k-out-of-n" or "f-out-of-n" is not a sufficient characterization for a comprehensive security assertion

Security depends on system model (e.g., rejuvenations, ...), attack model (e.g., attack surface, ...), ...

Characterizing threshold schemes

Characterizing threshold schemes

To reflect on a threshold scheme, start by characterizing **4 main features**:

- Kinds of threshold 
- Executing platform 
- Communication interfaces 
- Setup and maintenance 

The cliparts are from openclipart.org/detail/$*$, with $* \in \{71491, 190624, 101407, 161401, 161389\}$

Characterizing threshold schemes

To reflect on a threshold scheme, start by characterizing **4 main features**:

- Kinds of threshold 
- Executing platform 
- Communication interfaces 
- Setup and maintenance  

The cliparts are from openclipart.org/detail/$*$, with $* \in \{71491, 190624, 101407, 161401, 161389\}$

Each feature spans distinct options that affect security in different ways.

Characterizing threshold schemes

To reflect on a threshold scheme, start by characterizing **4 main features**:

- Kinds of threshold 
- Executing platform 
- Communication interfaces 
- Setup and maintenance  

The cliparts are from openclipart.org/detail/$*$, with $* \in \{71491, 190624, 101407, 161401, 161389\}$

Each feature spans distinct options that affect security in different ways.

A characterization provides a better context for security assertions.

Characterizing threshold schemes

To reflect on a threshold scheme, start by characterizing **4 main features**:

- Kinds of threshold 
- Communication interfaces 
- Executing platform 
- Setup and maintenance  

The cliparts are from openclipart.org/detail/∗, with $* \in \{71491, 190624, 101407, 161401, 161389\}$

Each feature spans distinct options that affect security in different ways.

A characterization provides a better context for security assertions.

But there are other factors ...

Another model

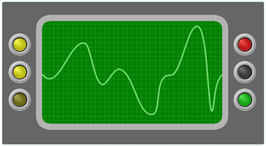
What if all nodes are compromised (e.g., leaky) from the start?

Another model

What if all nodes are compromised (e.g., leaky) from the start?

Threshold scheme may still be effective,
if it increases the cost of exploitation!

(e.g., if exploiting a leakage vulnerability requires exponential number of traces for high-order Differential Power Analysis)



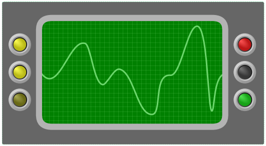
openclipart.org/detail/172330

Another model

What if all nodes are compromised (e.g., leaky) from the start?

Threshold scheme may still be effective,
if it increases the cost of exploitation!

(e.g., if exploiting a leakage vulnerability requires exponential number of traces for high-order Differential Power Analysis)



openclipart.org/detail/172330

Challenge questions:

- Which models are realistic / match state-of-the-art attacks?
- What concrete parameters (e.g., k, n) thwart real attacks?

Outline 2

The Threshold Cryptography Project at NIST

https://csrc.nist.gov/Projects/Threshold-Cryptography/ threshold-crypto@nist.gov

Scope: standardization of threshold schemes for cryptographic primitives

The Threshold Cryptography Project at NIST

https://csrc.nist.gov/Projects/Threshold-Cryptography/ threshold-crypto@nist.gov

Scope: standardization of threshold schemes for cryptographic primitives

Milestones:

- NISTIR 8214: Threshold Schemes for Cryptographic Primitives: Challenges and Opportunities in Standardization and Validation of Threshold Cryptography
- NTCW 2019: NIST Threshold Cryptography Workshop 2019
- NISTIR 8214A: NIST Roadmap Toward Criteria for Threshold Schemes for Cryptographic Primitives

The Threshold Cryptography Project at NIST

https://csrc.nist.gov/Projects/Threshold-Cryptography/ threshold-crypto@nist.gov

Scope: standardization of threshold schemes for cryptographic primitives

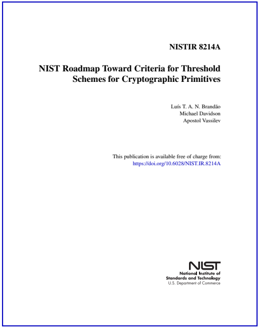
Milestones:

- NISTIR 8214: Threshold Schemes for Cryptographic Primitives: Challenges and Opportunities in Standardization and Validation of Threshold Cryptography
- NTCW 2019: NIST Threshold Cryptography Workshop 2019
- NISTIR 8214A: NIST Roadmap Toward Criteria for Threshold Schemes for Cryptographic Primitives

Main points:

- Two tracks: single-device (this presentation) and multi-party
- Need to engage with stakeholders
- Need to define criteria for possible calls/evaluation of threshold schemes
- Need to characterize threshold schemes and models

NISTIR 8214A: A roadmap toward criteria


NISTIR 8214A

NIST Roadmap Toward Criteria for Threshold Schemes for Cryptographic Primitives

Luís T. A. N. Brandão
Michael Davidson
Apostol Vassilev

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8214A

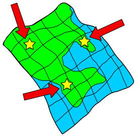
NIST
National Institute of Standards and Technology
U.S. Department of Commerce

NISTIR 8214A: NIST Roadmap Toward Criteria for Threshold Schemes for Cryptographic Primitives

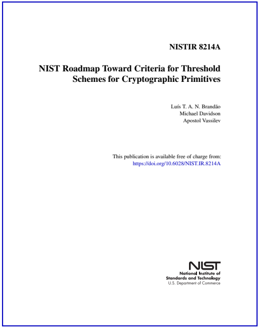
(Title changed since draft “Towards NIST Standards for Threshold Schemes for Cryptographic Primitives: A Preliminary Roadmap”)

NISTIR 8214A: A roadmap toward criteria

1. **Coordinates** (domains, primitives, modes, features)
2. **Features** (Security, configurability, validation, modularity)
3. **Phases** (of the development process)
4. **Collaboration** (useful feedback from stakeholders)



clker.com/clipart-15840.html



NISTIR 8214A

NIST Roadmap Toward Criteria for Threshold Schemes for Cryptographic Primitives

Luís T. A. N. Brandão
Michael Davidson
Apostol Vassilev

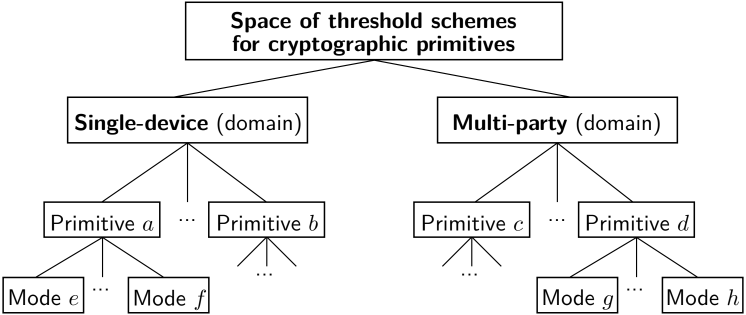
This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8214A

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

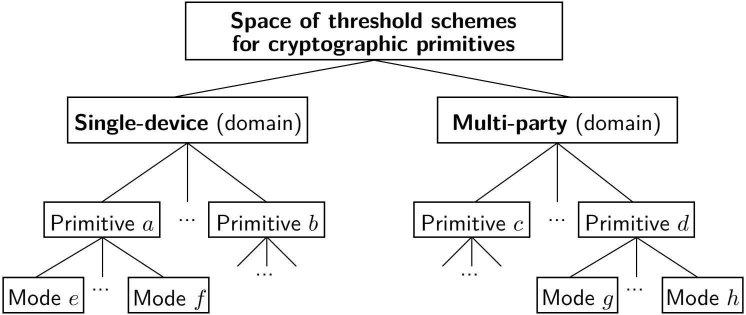
NISTIR 8214A: NIST Roadmap Toward Criteria for Threshold Schemes for Cryptographic Primitives

(Title changed since draft "Towards NIST Standards for Threshold Schemes for Cryptographic Primitives: A Preliminary Roadmap")

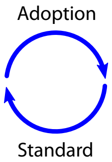
Mapping the space of potential "schemes"



Mapping the space of potential “schemes”



- “*Not every conceivable possibility is suitable for standardization*”.
- We find useful to hear stakeholders’ insights, to “*focus on where there is a high need and high potential for adoption*”.
- *Best practices*; *minimum defaults*; *interoperability*; *innovation*.



Single Device track

This presentation is focused on the single-device domain/track:

- (Typically) rigid configuration of components
- Strictly defined physical boundaries
- Dedicated communication network

Current focus of single-device track is on block-ciphers:

Single Device track

This presentation is focused on the single-device domain/track:

- (Typically) rigid configuration of components
- Strictly defined physical boundaries
- Dedicated communication network

Current focus of single-device track is on block-ciphers:

- Less complex: AES threshold circuit design against leakage.
- More complex: AES threshold circuit against combined attacks.
- Research interest: other lightweight crypto primitives.

Single Device track

This presentation is focused on the single-device domain/track:

- (Typically) rigid configuration of components
- Strictly defined physical boundaries
- Dedicated communication network

Current focus of single-device track is on block-ciphers:

- Less complex: AES threshold circuit design against leakage.
- More complex: AES threshold circuit against combined attacks.
- Research interest: other lightweight crypto primitives.

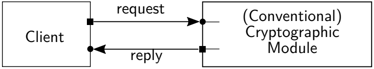
Modularity is an important consideration:

- Useful gadgets: secret-sharing, distributed/correlated RNG, ...
- Non-linear part (S-Box) and linear parts may be treated differently ...

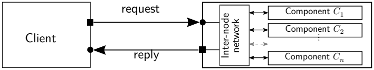
Threshold interface modes (in the perspective of the client)

Threshold interface modes (in the perspective of the client)

Input/Output interface: client communication with the module / threshold entity?



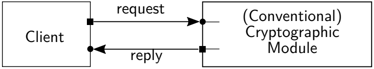
Conventional (non-threshold)



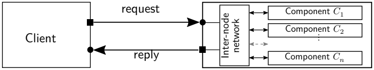
Not-shared-IO

Threshold interface modes (in the perspective of the client)

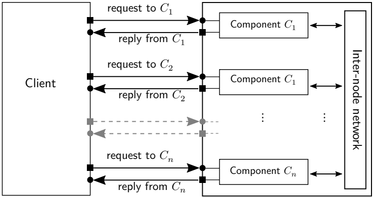
Input/Output interface: client communication with the module / threshold entity?



Conventional (non-threshold)



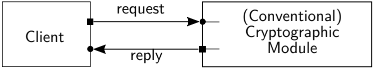
Not-shared-IO



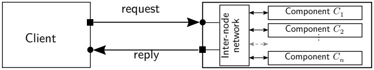
Shared-IO

Threshold interface modes (in the perspective of the client)

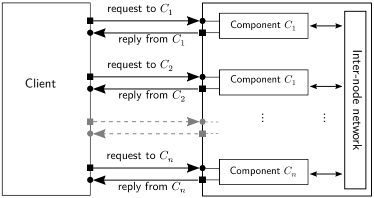
Input/Output interface: client communication with the module / threshold entity?



Conventional (non-threshold)



Not-shared-IO



Shared-IO

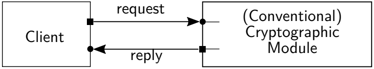
(Shared-I and Shared-O are other modes where only the **i**nput and only the **o**utput are shared, respectively)

Example (how relevant in the single-device setting?):

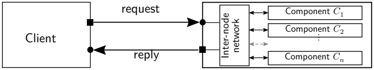
- Shared-Output: enhance secrecy of the output of a decryption process?

Threshold interface modes (in the perspective of the client)

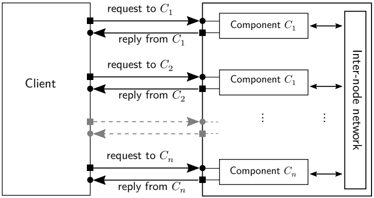
Input/Output interface: client communication with the module / threshold entity?



Conventional (non-threshold)



Not-shared-IO



Shared-IO

(Shared-I and Shared-O are other modes where only the **i**nput and only the **o**utput are shared, respectively)

Example (how relevant in the single-device setting?):

- Shared-Output: enhance secrecy of the output of a decryption process?

Auditability: can the client prove (or be convinced) the operation was thresholdized?

We welcome feedback toward defining **criteria**

Some relevant aspects (from Section 6.1 of NISTIR 8214A):

1. Definition of system model and threat model
2. Description of characterizing features
3. Analysis of efficiency and practical feasibility
4. Existence of open-source reference implementations
5. Concrete benchmarking (threshold vs. conventional; different platforms)
6. Detailed description of operations
7. Example application scenarios
8. Security analysis
9. Automated testing and validation of implementations
10. Disclosure and licensing of intellectual property

We welcome feedback on any of these items, not only in abstract but also about concrete published works / proposals.

Development process

A sequence of phases:

1. **Devise criteria for standardization**
2. **Calls for contributions**
3. **Evaluation of threshold schemes**
4. **Publish standards**

(Each phase will be open to public feedback. Some Threshold Cryptography workshops along the way?)

Development process

A sequence of phases:

1. **Devise criteria for standardization**
2. **Calls for contributions**
3. **Evaluation of threshold schemes**
4. **Publish standards**

(Each phase will be open to public feedback. Some Threshold Cryptography workshops along the way?)

Note: Here, "Standards" is used loosely and does not intend to imply FIPS.

Final formats may include addenda or reference to other standards, implementation/validation guidelines, reference definitions, ...

Outline 3

The validation challenge

- **Validation** means checking that a specific implementation of a cryptographic primitive specified in a standard satisfies a set of security assertions.
- The NIST Cryptographic Algorithm Validation Program (CAVP) covers algorithm/scheme implementations
- CAVP is a prerequisite for the Cryptographic Module Validation Program (CMVP, a.k.a. FIPS 140-2/3).

The validation challenge

- **Validation** means checking that a specific implementation of a cryptographic primitive specified in a standard satisfies a set of security assertions.
- The NIST Cryptographic Algorithm Validation Program (CAVP) covers algorithm/scheme implementations
- CAVP is a prerequisite for the Cryptographic Module Validation Program (CMVP, a.k.a. FIPS 140-2/3).

Why is it relevant?

- **Required by law in the US.** Crypto primitives used in federal systems must be NIST-approved and their implementations must be validated.
- **Voluntary adoption.** CMVP/CAVP validations are voluntarily outside the US Federal Government, adopted by industries (e.g., Financial industry) and countries (e.g., Canada).

The validation challenge

- **Validation** means checking that a specific implementation of a cryptographic primitive specified in a standard satisfies a set of security assertions.
- The NIST Cryptographic Algorithm Validation Program (CAVP) covers algorithm/scheme implementations
- CAVP is a prerequisite for the Cryptographic Module Validation Program (CMVP, a.k.a. FIPS 140-2/3).

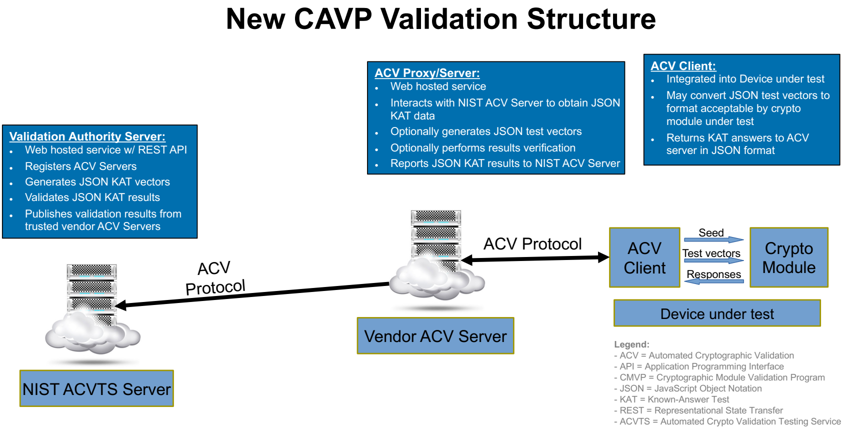
Why is it relevant?

- **Required by law in the US.** Crypto primitives used in federal systems must be NIST-approved and their implementations must be validated.
- **Voluntary adoption.** CMVP/CAVP validations are voluntarily outside the US Federal Government, adopted by industries (e.g., Financial industry) and countries (e.g., Canada).

A dual perspective:

- Devise standards of testable and validatable threshold schemes
- Devise testing and validation for standardized threshold schemes

The Automated Cryptographic Algorithm Validation



Computer-based testing and validation

10

Outline 4

Deployment context

- **Conceivable attack types.**

clker.com/clipart-10778

- Active vs. passive
- Static vs. adaptive
- Stealth vs. detected
- Invasive (physical) vs. non-invasive
- Side-channel vs. communication interfaces
- Parallel vs. sequential (wrt attacking nodes)

Deployment context

- **Conceivable attack types.**

clker.com/clipart-10778

- Active vs. passive
- Static vs. adaptive
- Stealth vs. detected
- Invasive (physical) vs. non-invasive
- Side-channel vs. communication interfaces
- Parallel vs. sequential (wrt attacking nodes)

A threshold scheme **improving** security against an attack in an application **may be powerless or degrade** security for another attack in another application

Deployment context

- **Conceivable attack types.**

clker.com/clipart-10778

- Active vs. passive
- Static vs. adaptive
- Stealth vs. detected
- Invasive (physical) vs. non-invasive
- Side-channel vs. communication interfaces
- Parallel vs. sequential (wrt attacking nodes)

A threshold scheme **improving** security against an attack in an application **may be powerless or degrade** security for another attack in another application

Two starting points:

1. Passive: AES threshold circuit design against leakage
2. Active: AES threshold circuit against combined attacks

Baseline scenarios for threshold circuit design

Resistance against side-channel attacks

Assume passive adversary that does not interfere with the computation.

- Main property of interest: confidentiality of the key (prevent leakage)
- What number of traces (e.g., power-analysis) is it reasonable to assume the adversary can collect?
- What are suitable models of leakage (noisy, wire-probing, ...)?

Baseline scenarios for threshold circuit design

Resistance against side-channel attacks

Assume passive adversary that does not interfere with the computation.

- Main property of interest: confidentiality of the key (prevent leakage)
- What number of traces (e.g., power-analysis) is it reasonable to assume the adversary can collect?
- What are suitable models of leakage (noisy, wire-probing, ...)?

Resistance against combined attacks (side-channel and fault injection)

- Main property of interest: confidentiality of the key (prevent leakage)
- Also of interest — integrity nuances: error detection, error correction
- What kinds of fault-injection (controlled vs. random bit in a wire, ...)
- Against what kind of interferences is the threshold approach useful (e.g., varying power supply, temperature other environmental conditions)?

More questions for each scenario

Useful feedback now — potential to shape the criteria:

(From Section 7.2 of NISTIR 8214A)

1. Enumerate and define the desirable **properties** (e.g., uniformity and non-completeness) that are possible to achieve in threshold circuit designs.

2. Identify useful **construction paradigms** for threshold circuit design and the **gadgets** that are useful to implement them.

3. Indicate the models/**conditions** under which the threshold schemes may enable a higher resistance to side-channel and/or fault attacks (e.g., quantifying the increase in the number of traces required for a successful differential power analysis attack).

4. Indicate possible **parameters** (e.g., masking order and number of shares) for realistic implementations of threshold circuit designs.

Other relevant aspects of feedback

- **Motivation/applicability:** real-world applications, deployment settings
- **Concrete protocols/algorithms:** comparison of state-of-the-art references
- **Reference implementations:** feasibility, benchmarks, open source, ...
- **Intellectual property:** information on known patents, licenses, ...

The modularity challenge

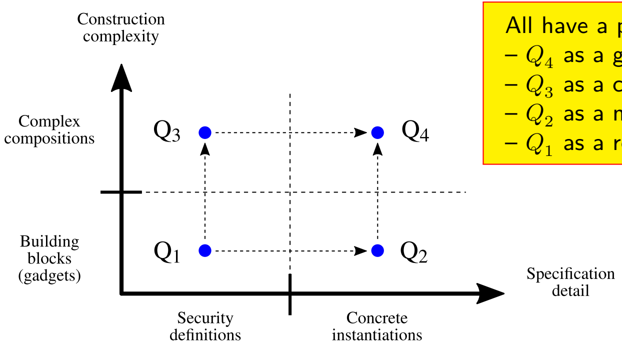
Inter-play between:

- security definitions vs. concrete instantiations
- building blocks vs. complex compositions

The modularity challenge

Inter-play between:

- security definitions vs. concrete instantiations
- building blocks vs. complex compositions



All have a place in the process:
– Q_4 as a goal;
– Q_3 as a criterion;
– Q_2 as a module;
– Q_1 as a reference definition.

Example *gadgets*: secret sharing; distributed/correlated randomness; ...

Outline 5

Concluding remarks

- Feedback from stake-holders is essential ... will help devise criteria.
- Very diverse threshold space ... need rationale to select what to focus on.
- Want to focus on well-understood, robust threshold schemes and models.
- Automated validation is to be considered part of the development process.
- Process with openness, transparency, scrutiny, technical merit, trust, ...

- Project webpage: https://csrc.nist.gov/Projects/Threshold-Cryptography
- Project email adress: threshold-crypto@nist.gov
- NISTIR 8214: https://csrc.nist.gov/publications/detail/nistir/8214/final
- NISTIR 8214A: https://csrc.nist.gov/publications/detail/nistir/8214a/final
- TC-forum: https://list.nist.gov/tc-forum

- Project webpage: https://csrc.nist.gov/Projects/Threshold-Cryptography
- Project email adress: threshold-crypto@nist.gov
- NISTIR 8214: https://csrc.nist.gov/publications/detail/nistir/8214/final
- NISTIR 8214A: https://csrc.nist.gov/publications/detail/nistir/8214a/final
- TC-forum: https://list.nist.gov/tc-forum

Word cloud based on the NISTIR 8214

Thank you for your attention

- Project webpage: https://csrc.nist.gov/Projects/Threshold-Cryptography
- Project email adress: threshold-crypto@nist.gov
- NISTIR 8214: https://csrc.nist.gov/publications/detail/nistir/8214/final
- NISTIR 8214A: https://csrc.nist.gov/publications/detail/nistir/8214a/final
- TC-forum: https://list.nist.gov/tc-forum

Word cloud based on the NISTIR 8214

Presentation at the Online Workshop on Threshold Schemes for NIST-approved Symmetric Block Ciphers in a Single-Device Setting

July 7, 2020 @ Virtual

luis.brandao@nist.gov, apostol.vassilev@nist.gov