Update On NIST Cryptographic Program

Lily Chen
Manager of Cryptographic Technology Group
Computer Security Division, Information Technology Laboratory, NIST

NIST Cryptography Program

NIST Crypto Standards – Overview[1]

- Crypto standards
  - Public key based
    - Signature (FIPS 186 -4)
    - Key establishment (800-56A/B/C)
  - Tools
    - RNG (800-90A/B/C)
    - KDF (800-108, 800-135)
  - Symmetric key based
    - AES (FIPS 197) TDEA(800-67)
    - Modes of operations (800 38A-38G)
    - SHA-1/2 (FIPS 180) and SHA-3 (Draft FIPS 202)
    - Randomized hash (800-106)
    - HMAC (FIPS 198)
    - SHA3 (FIPS 202)
    - SHA3 derived functions (800-185)
  - Guidelines
    - Hash usage/security (800-107)
    - Transition (800-131A)
    - Key generation (800-133)
    - Key management (800-57)
    - TLS Guidance (800-52)

[1] This is not a complete list

Outline

* Historic Review
* Challenges to Cryptography Standards
* Post-quantum Cryptography
* Lightweight Cryptography
* Code Signing

A Short History of NIST Crypto Standards
- Major Milestones

* FIPS 46 ”Data Encryption Standard (DES)” - 1977
* Public–key Cryptography (FIPS 186, SP 800-56A/56B) – 1990s
* FIPS 197 “Advanced Encryption Standard (AES)” – 2001
* FIPS 202 “SHA-3” (Secure Hash Function 3) – 2015
* Ongoing projects
  * Post-Quantum Cryptography (PQC)
  * Lightweight Cryptography (LWC)
  * Threshold Cryptography
* What is next?

NIST Crypto Standards Approaches

* Cryptographic algorithm competitions
  * Advanced Encryption Standard (AES)
  * Secure Hash Algorithm – 3 (SHA-3)
* Adoption of standards developed in other standards organizations (Diffie-Hellman key agreement in SP 800-56A from X9.42 and X9.63)
  * Some have been revised after adoption based on new results
* Develop new standards
  * In-house development based on well accepted research results (e.g. SP 800-56C)
  * Selected among submissions (e.g. modes of operations in SP 800-38 series)
* Not quite a competition but based on call for submissions (PQC and LWC)

Challenges to Crypto Standardization

* Deal with extremely powerful attack technologies (e.g. quantum computers) and constrained implementation environments (e.g. RFID and sensors in IoT)
* Deprecate weak cryptographic algorithms and methods and assure backward compatibility (e.g. sunset triple DES and PKCS#1 v1.5 padding)
* Handle variations created in practice (e.g. KDFs 800-56C, 800-108, 800-135, …)
  * It has never been easy to find a common ground for standardization
* Emerging technologies constantly demand for new crypto tools
* Resource limits
  * Standards development and maintenance are always costly
  * It takes months or even years to develop or revise a standard

Deal with Quantum Attacks: Post-Quantum Cryptography

Hard Problems and Public Key Cryptography

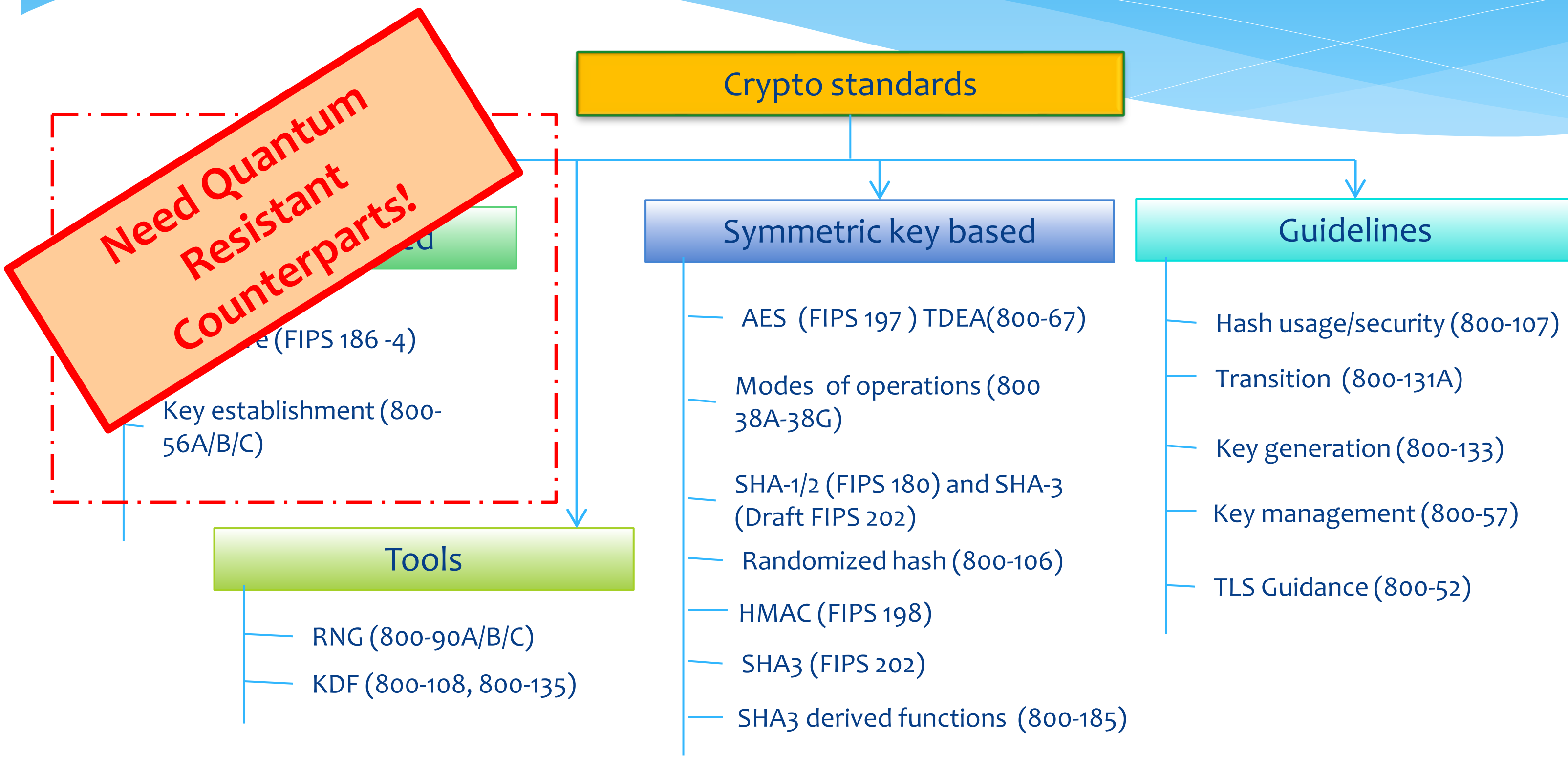
* A problem is hard if no polynomial time algorithm is known to solve it
* The hardness is categorized by computing complexity - generally expressed as a function $n \rightarrow f(n)$, where n is the size of the input, e.g.
  * If $f(n)$ is a polynomial, then the problem is not hard
  * If $f(n) = c \cdot e^{h(n)}$ then, the problem is hard
* Practically, it means that it is infeasible to solve it with the currently available computing resource
* The hardness on certain problems is used as the basic assumptions for some cryptographic schemes, e.g.
  * RSA is based on the hardness of integer factorization, given integer $n \ (= p \cdot q)$ find p and q
  * Diffie-Hellman key agreement is based on the hardness of discrete logarithm problem, given $y \ \in \mathrm{GF}(p)$* and generator g , find x , such that $y = gx \ mod \ p$

Quantum Impact

* Quantum computing changed what we have believed about the hardness of discrete log and factorization problems
  * Using quantum computers, an integer *n* can be factored in polynomial time using Shor's algorithm
  * The discrete logarithm problem can also be solved by Shor's algorithm in polynomial time
* As a result, the public key cryptosystems deployed since the 1980s will need to be replaced
  * RSA signatures, DSA and ECDSA (FIPS 186-4)
  * Diffie-Hellman Key Agreement over finite fields and elliptic curves(NIST SP 800-56A)
  * RSA encryption (NIST SP 800-56B)
* We have to look for quantum-resistant counterparts for these cryptosystems
* Quantum computing also impacted security strength of symmetric key based cryptography algorithms
  * Grover's algorithm can find AES key with approximately $\sqrt{2^n}$ operations where n is the key length
  * Intuitively, we should double the key length, if 2^{64} quantum operations cost about the same as 2^{64} classical operations

NIST

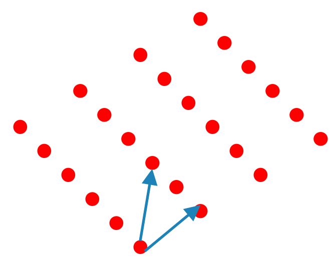
Quantum Impact on NIST Crypto Standards

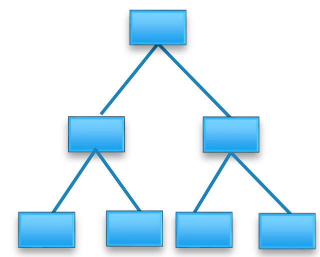


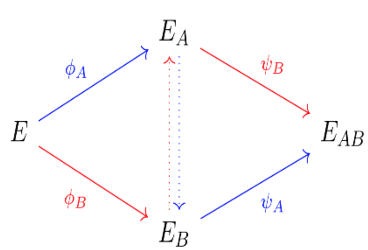
(1) This is not a complete list

Post-Quantum Cryptography (PQC)

- Post-quantum cryptography algorithms are classical cryptographic algorithms which are considered to be able to resist quantum attacks
  - They must be based on hard problems which are still hard even when large scale quantum computers become available
- Some actively researched PQC categories
  - Lattice-based
  - Code-based
  - Multivariate
  - Hash based signatures
  - Isogeny-based schemes







$$
\begin{aligned}
p^{(1)}(x_1, \ldots, x_n) &= \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(1)} \cdot x_i x_j \quad + \quad \sum_{i=1}^{n} p_i^{(1)} \cdot x_i + p_0^{(1)} \\
p^{(2)}(x_1, \ldots, x_n) &= \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(2)} \cdot x_i x_j \quad + \quad \sum_{i=1}^{n} p_i^{(2)} \cdot x_i + p_0^{(2)} \\
&\vdots \\
p^{(m)}(x_1, \ldots, x_n) &= \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(m)} \cdot x_i x_j \quad + \quad \sum_{i=1}^{n} p_i^{(m)} \cdot x_i + p_0^{(m)}
\end{aligned}
$$

NIST

What we have done so far in a long journey

* 2012 – NIST begins PQC project
  * Research and build NIST team
* April 2015 – 1st NIST PQC workshop
* Feb 2016 – NIST Report on PQC (NISTIR 8105)
* Feb 2016 – NIST preliminary announcement of standardization plan
* Dec 2016 – Announcement of Call for Proposals requirements and criteria(Federal Register Notice)
* Nov. 2017 – Received 82 submissions
* Dec. 2017 – Announced the 1st round candidates
* April 2018 – The 1st NIST PQC Standardization Conference
* Jan. 2019 – Announced the 2nd round candidates

NIST

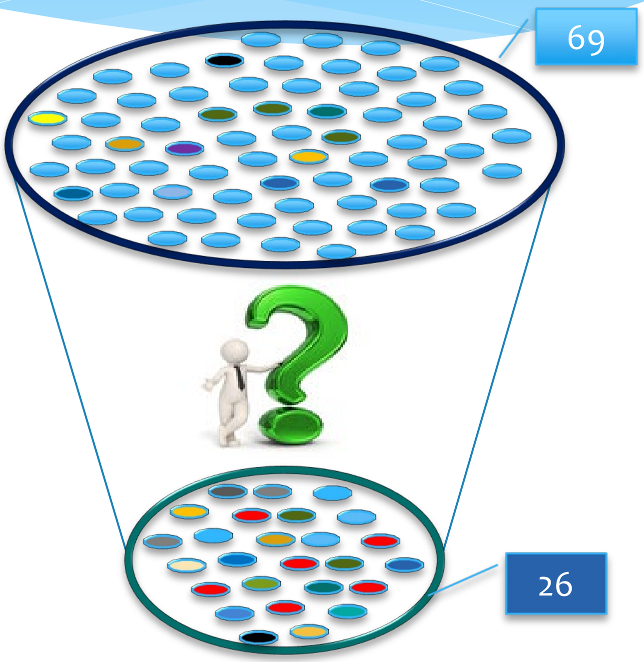
Submissions to NIST Call for Proposals

* 82 total submissions received from 26 Countries, 6 Continents
* 69 accepted as "complete and proper" (5 since withdrawn) in December 2017

	Signatures	KEM/Encryption	Overall
Lattice-based	5	21	26
Code-based	2	17	19
Multi-variate	7	2	9
Stateless Hash-based/Symmetric based	3		3
Other	2	5	7
Total	**19**	**45**	**64**

Selection of second round candidates

* Security
  * Candidates which were broken, significantly attacked, or difficult to establish confidence in their security were left out
  * Candidates which provided clear design rationale and reasonable security proofs to established reasonable confidence in security are advanced
* Performance
  * Candidates with obvious performance or key/signature/ciphertext size issues for existing applications were not advanced - even though they might have been well prepared with good ideas



The 2nd round candidates

KEM/Enc

Lattice –based (9):
Crystals-Kyber; FrodoKEM; LAC; NewHope; NTRU; NTRU Prime; Round 5; Saber; Three Bears

Code –based (7):
Classic McEliece; NTS-KEM; BIKE; HQC; Rollo; LEDAcrypt; RQC

Isogeny –based (1):
SIKE

Signature

Lattice –based (3):
Crystals-Dilithium; Falcon; qTESLA

Symmetric –based (2) :
Sphincs+; Picnic

Multivariate (4):
GeMSS; LUOV; MQDSS; Rainbow

* See NISTIR 8240 for a summary of each of the 2nd round candidates

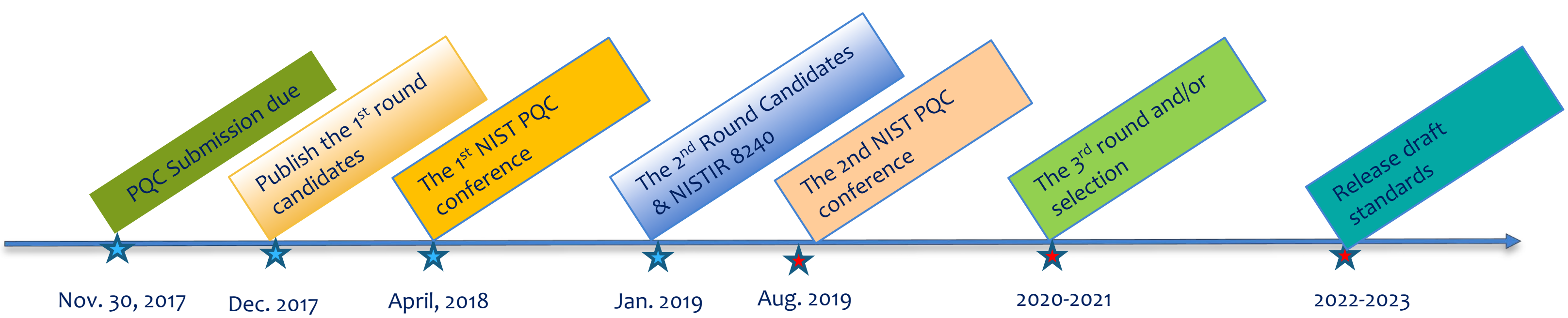
Preparation for Migration

* Enable crypto agility for each function (public key encryption/key encapsulation, signature) when it is possible
* Understand implementation costs and required bandwidth/space for transmitting and storing keys, signatures and ciphertext
* Discuss tradeoff preferences in each application – identify special restrictions, limitations, and show stoppers
* Gain first-hand experience through trial implementations e.g. hybrid mode or dual signatures as a temporary solution
* Do not commit to a specific candidate for long-term products until NIST makes its selection for standardization

NIST

Future plans

* The 2nd PQC Standardization Conference will be held in August 2019
* Spend 12-18 months to analyze and evaluate the 2nd round candidates
* Start a 3rd round and/or select algorithms to standardize 2020-2021
* Release draft standards in 2022-2023 for public comments



Crypto in Constrained Environment: Lightweight Cryptography

Motivation

* Shift from general-purpose computers to dedicated resource-constrained (with limited processing and storage capabilities) devices such as RFID tags, sensor networks, IoT devices.
* New applications (e.g., home automation, smart city technologies, digital assistants, healthcare) that collect, store and process private information (e.g., sleep patterns, heart beat, exercise routines, medical information, location).
* Lack of crypto standards that are suitable for constrained devices.

NIST

Scope and Goal

* Scope: Symmetric key cryptography: Authenticated Encryption with Associated Data (AEAD) and optional hashing functionality.
    * Note that in the current NIST standards, AEAD is achieved through mode of operations such as AES-GCM
    * For hash function, NIST current standards have SHA-2 family and SHA-3 family
* Aim: Developing new guidelines, recommendations and standards for constrained environments when the performance of the current NIST standards is not acceptable.

The Need for Lightweight Crypto

* Performances of NIST standards are not always acceptable.
  * Area requirement of AES is heavy (the choice of 8 bit S-box may not be optimal for area). e.g., combined AES enc/dec implementation is not possible within 512 bytes of ROM and 128 bytes of RAM (Moriai, 2016)
  * Large memory requirements of SHA-2 and SHA-3 families.
* Dedicated algorithms with inherent side channel resistance may provide security and performance advantages over AES.
* Hash functions with smaller internal state size and that can share crypto logic to provide other functionalities are more suitable for constrained devices.
* The needs have been discussed through two workshops in 2015 and 2016 and well studied in NISTIR 8113 Report on Lightweight Cryptography

NIST

Challenges in Standardizing Lightweight Crypto

* The attackers to lightweight cryptography are not lightweight at all
  * They can be as powerful as attackers to any cryptosystems
* It is a new experience to standardize cryptography tools for constrained environment not for general usage
  * Need to draw a line on where they can be used and where they cannot
* Using lightweight crypto may lead to some restrictions such as restrictions on how many bytes of plaintext can be encrypted by the same key
  * The restriction must be practical for most usages
* Lightweight requirements can be very different for applications
  * Need to choose common ones to reduce implementation burden

NIST

Highlight Requirements

* August 27th 2018, Federal Register announcement and the publication of the call of submissions.
  * AEAD
    * In each family, one primary member with key, nonce and tag lengths of at least 128, 96 and 64 bits, respectively
    * Attack complexities at least 2^{112} computation
    * Limits on the input sizes for the primary member shall not be smaller than $2^{50} - 1$ bytes
  * Hash functions
    * Computationally infeasible to find a collision or a (second) preimage
      * Infeasible - with attack complexity at least 2^{112} computations
    * Resistance to length extension attacks
    * The hash output size at least 256 bits

NIST

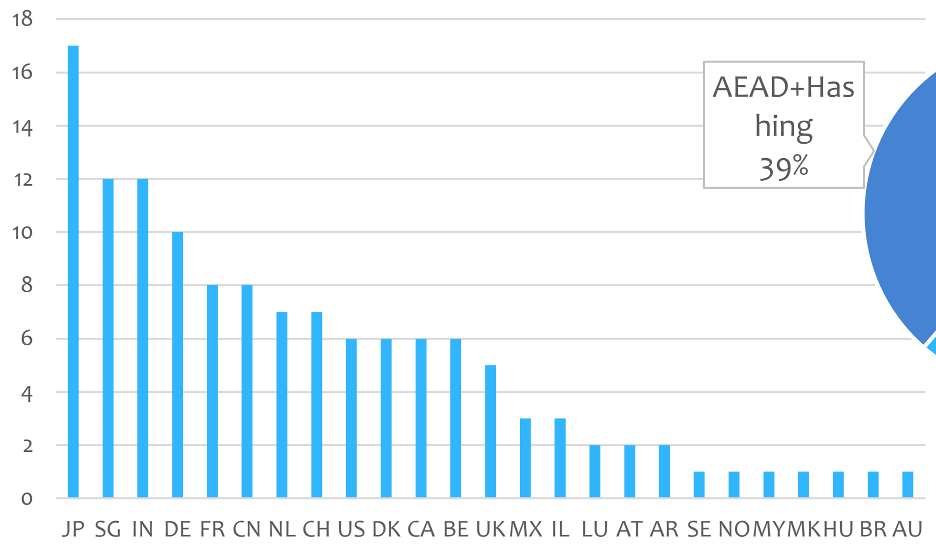
Design requirements

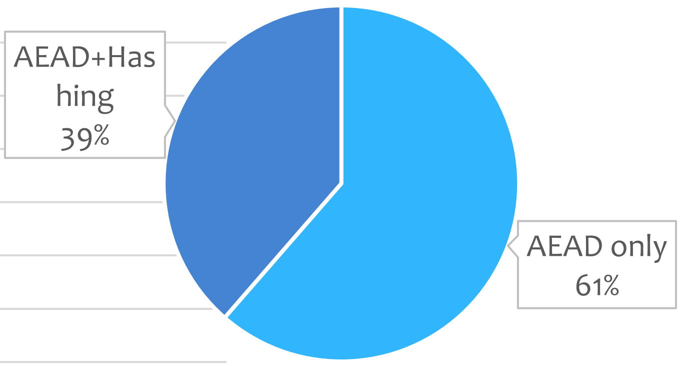
* Perform significantly better in constrained environments (HW and SW platforms) compared to current NIST standards
* Optimized to be efficient for short messages
* Implementations should lend themselves to countermeasures against side channel attacks, fault attacks.
* Designs can make tradeoffs between performance metrics, and submitters are allowed to prioritize certain performance requirements over others

NIST

Submissions and 1st Round

* NIST received 57 total submissions by Feb. 25th 2019 from 25 countries
* 35 AEAD-only submissions, 22 AEAD and hashing functionality
* NIST announced 56 submissions as the first round candidates in April 2019





Next Steps

- First cut and a report justifying the selection (in ~August)
- Candidates will be analyzed based on security, performance and side channel resistance.
- Candidates with significant third-party analysis or leverage components of existing standards will be favored for selection.
- Workshop in November 4-6, 2019
- Standardization in ~2 years* after the public analysis starts.
  - Different from AES/SHA3/PQC – much shorter timeline
  - In favor of submissions with third party analysis (no new design approaches).

* This is a relatively optimistic estimation, when we haven't received submissions. With the amount of submissions, we may justify the timeframe to make sure that we have enough time for thorough analysis.

NIST

Build Trusted Platforms: Code Signing

Code Signing

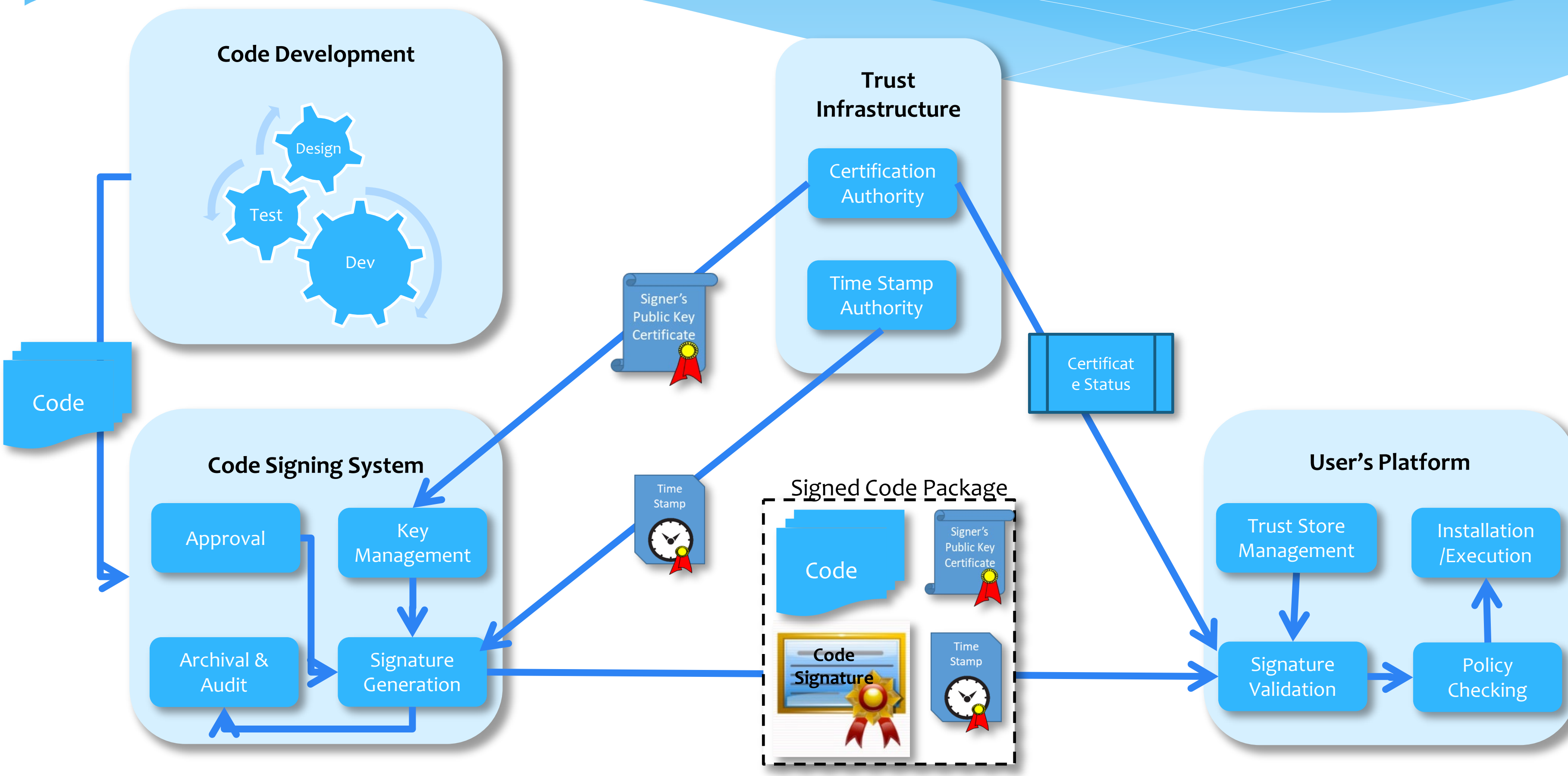
* **Digitally signing software using a key held by the software publisher or developer**
* **Benefits**
  * ***Integrity:*** software cannot been modified after being signed
  * ***Source Authentication:*** Verify that software came from a trusted and known source
  * ***Metadata Assurance:*** Cryptographically bind security-relevant metadata (e.g., version number) to the software package.

NIST

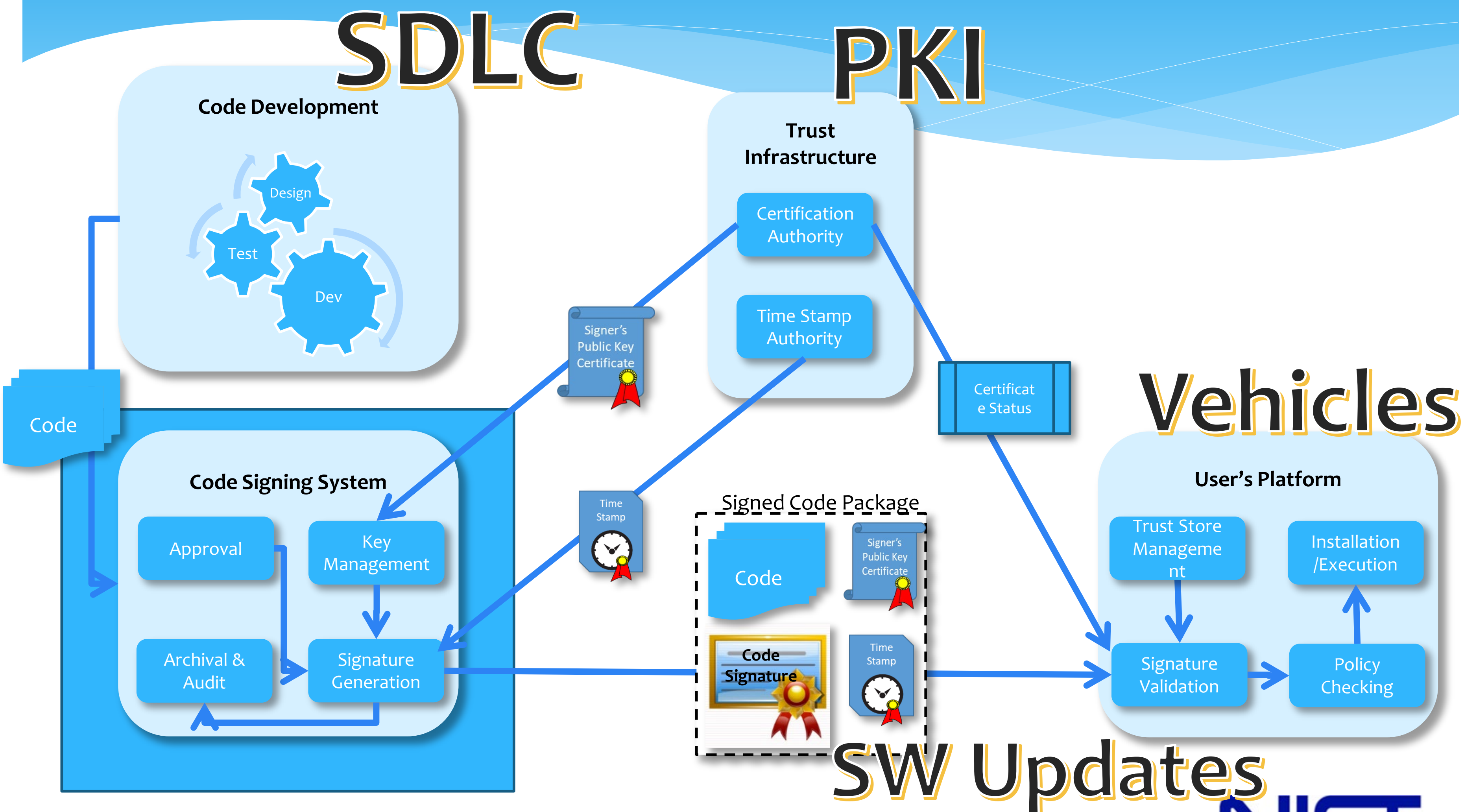
Code Signing Use Cases

* **Authenticate and authorize updates**
  * Firmware
  * Operating Systems
  * Applications and app stores
  * Drivers
* **Verify before execution**
  * Secure Boot/Verified Boot
  * Application Whitelisting

Architecture



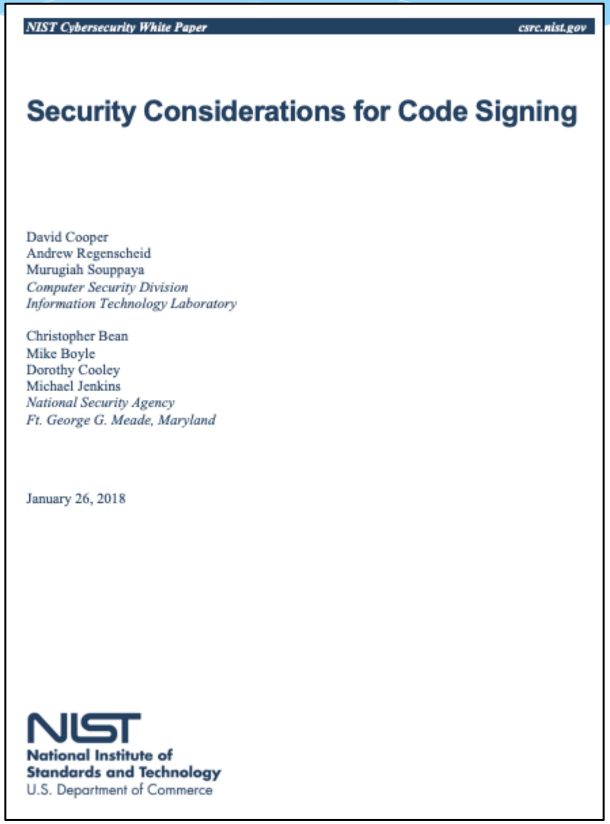
Existing Work



Code Signing Recommendations

- NIST Whitepaper: ***Security Considerations for Code Signing***
- Topics
  - Code signing overview
  - Architectures and use cases
  - Description of roles
  - Major Threats
  - Recommended security practices

https://doi.org/10.6028/NIST.CSWP.01262018


NIST Cybersecurity White Paper *csrc.nist.gov*

Security Considerations for Code Signing

David Cooper
Andrew Regenscheid
Murugiah Souppaya
Computer Security Division
Information Technology Laboratory

Christopher Bean
Mike Boyle
Dorothy Cooley
Michael Jenkins
National Security Agency
Ft. George G. Meade, Maryland

January 26, 2018

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

Threats to Code Signing Systems

* Theft or loss of private signing key
* Issuance of unauthorized code signing certificates
* Signing unauthorized or malicious code
* Use of insecure cryptography
* Poor or insecure trust anchor management

NIST

Recommendations

* Identify and authenticate trusted code signing personals
* Separate roles and require two-party control
* Establish policies and procedures for reviewing, vetting and approving code
* Isolate and protect the Code Signing System
* Utilize auditing and periodically review logs
* Develop revocation/recovery mechanisms for cases of key compromise or unauthorized signing

Secure Updates

* ***Risks of insecure firmware updates***
  * Render systems inoperable if firmware damaged
  * Firmware-level malware can be stealthy, powerful and persistent
* ***Attack vectors***
  * Unauthenticated updates
  * Unprotected flash memory
  * Incorrect hardware configurations (e.g., locks, power transitions)
  * Software vulnerabilities: e.g., buffer overflows, race conditions

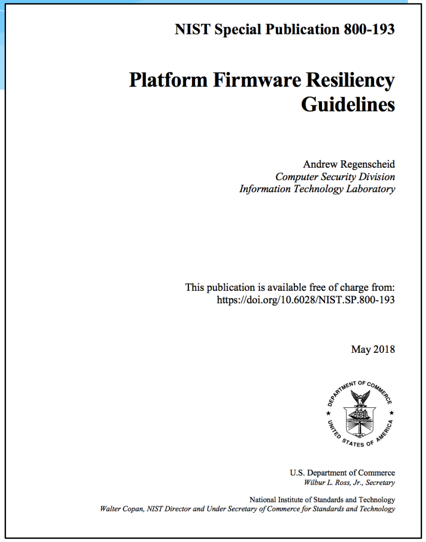
Code signing can authenticate the source and integrity of firmware updates and authorize their installation on platforms

NIST

NIST Guidelines

NIST SP 800-193, *Platform Firmware Resiliency Guidelines*

Purpose: Securing updatable ***firmware*** and ***configuration data*** computer platforms



NIST Special Publication 800-193

Platform Firmware Resiliency Guidelines

Andrew Regenscheid
Computer Security Division
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-193

May 2018

U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Under Secretary of Commerce for Standards and Technology

- **Protect firmware from unauthorized changes**
- **Detect corruption or malicious modification**
- **Recover to a trustworthy state when problems occur**