PQC: THE QUANTUM SHIELD

ทำไมข้อมูลที่ปลอดภัยวันนี้ อาจไม่ปลอดภัยในอนาคต?

WHY

ทำไมต้องเตรียมตัว

WHAT

PQC ป้องกันอะไร

HOW

องค์กรเริ่มได้อย่างไร

ภัยจาก Quantum ต้องวางแผนรับมือก่อนเทคโนโลยีพร้อม

ข้อมูลวันนี้ ต้องเป็นความลับอีกกี่ปี?

ข้อมูลแต่ละประเภทมีอายุความลับไม่เท่ากัน

ชั่วคราว

รหัสใช้งานครั้งเดียว
หมดอายุเร็ว

หลายปี

ข้อมูลการเงิน
และข้อมูลลูกค้า

หลายสิบปี

ข้อมูลวิจัย
และเอกสารลับ

อายุความลับของข้อมูล เป็นตัวกำหนดความเร่งด่วน

RSA / ECC มีความเสี่ยงเพราะอะไร?

Quantum + Shor’s Algorithm กระทบคณิตศาสตร์ที่ใช้

RSA

Rivest–Shamir–Adleman
อาศัยการแยกตัวประกอบ

ECC

Elliptic Curve
Cryptography
อาศัย Discrete Logarithm

Quantum

Shor’s Algorithm
แก้ปัญหาเหล่านี้ได้เมื่อมี
เครื่องที่เพียงพอ

ไม่ใช่การเข้ารหัสทุกชนิดจะได้รับผลกระทบเท่ากัน

Harvest Now, Decrypt Later

ดักเก็บข้อมูลวันนี้ ⟶ รอความสามารถถอดรหัสในอนาคต

01

ดักเก็บ

Traffic ที่เข้ารหัส

02

จัดเก็บ

Ciphertext และ Handshake

03

อนาคต

อาจถอดข้อมูลย้อนหลัง

HNDL = Harvest Now, Decrypt Later

ทำไมต้องเริ่มตอนนี้?

ข้อมูลอายุยาว + เวลาเปลี่ยนระบบ + ความไม่แน่นอน

X

ข้อมูลต้องลับอีกกี่ปี

Y

องค์กรต้องใช้เวลากี่ปี
ในการเปลี่ยน

Z

Quantum Threat จะ
มาถึงเมื่อไร

ถ้า X + Y > Z อาจเตรียมตัวไม่ทัน (กรอบคิด Mosca)

PQC คืออะไร?

Post-Quantum Cryptography: คณิตศาสตร์ใหม่บนคอมพิวเตอร์เดิม

PQC

อัลกอริทึมที่ออกแบบให้ต้านทาน Quantum

ไม่ต้องซื้อ

Quantum Computer มาใช้งาน

ต้องรองรับ

Software / Protocol / Hardware

PQC ไม่ใช่ Quantum Key Distribution (QKD)

ทำไม PQC ต่างจาก RSA / ECC?

เลือกปัญหาคณิตศาสตร์ที่ Shor ไม่ได้โจมตีโดยตรง

RSA / ECC

Factorization /
Discrete Log
มี Shor’s Algorithm

ML-KEM / ML-DSA

Module-Lattice Problems
ยังไม่มีการโจมตีที่รู้จักว่าทำลายค่ามาตรฐานได้

SLH-DSA

Hash-Based Signatures
พื้นฐานต่างจาก Lattice

ความเชื่อมั่นมาจาก Cryptanalysis + มาตรฐาน + การใช้ที่ถูกต้อง

RSA และ ECC: ทำไม Quantum จึงเป็นความเสี่ยง?

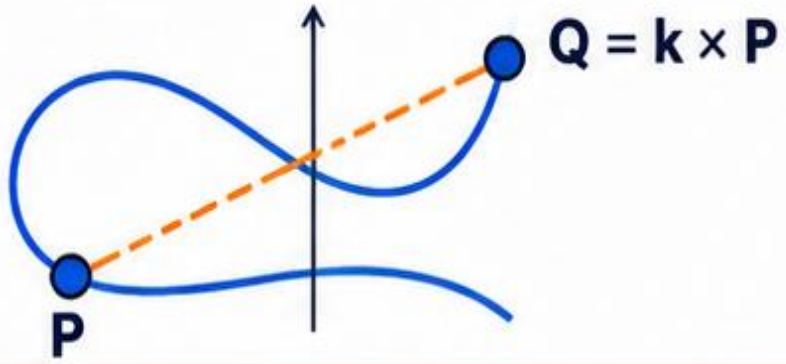
แบบเดิม: ความปลอดภัยพึ่งพาปัญหาคณิตศาสตร์ที่ Shor สามารถโจมตีได้

RSA (Rivest–Shamir–Adleman)



- **ความปลอดภัย:** แยก N กลับเป็น p และ q (การแยกตัวประกอบจำนวนใหญ่)
- **เปรียบเทียบ:** กุญแจที่ล็อกด้วย “ตัวคูณขนาดใหญ่”

ECC (Elliptic Curve Cryptography)



- **ความปลอดภัย:** หาค่า k จาก P และ Q (ปัญหาลอการิทึมบนเส้นโค้ง)
- **เปรียบเทียบ:** เดินบน “เส้นทางบนภูเขา” ที่คำนวณไปง่าย แต่ย้อนกลับหาจุดเริ่มต้นยาก

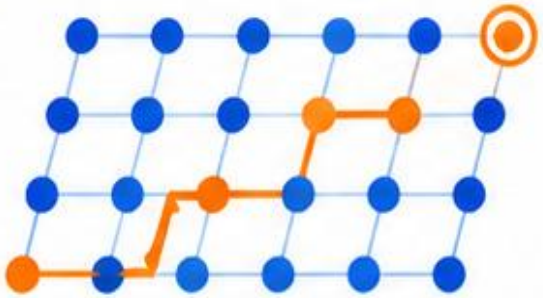
ภาพเปรียบเทียบเพื่อความเข้าใจ ไม่ใช่ขั้นตอนคำนวณจริงของอัลกอริทึม

PQC: ML/Lattice และ Hash–Based ต่างอย่างไร?

แบบใหม่: เปลี่ยนฐานปัญหาคณิตศาสตร์ ไม่ใช่เพียงเพิ่มความยาวกุญแจ

ML (Lattice-based)

เช่น ML-KEM (เข้ารหัส/แลกกุญแจ), ML-DSA (ลายมือชื่อ)



ค้นหา “จุดที่ใกล้ที่สุด” ในตาข่ายจุดจำนวนมาก (Lattice)

- **ความปลอดภัย:** ปัญหา “จุดใกล้ที่สุดในตาข่าย” (เรียนกว่า Lattice Problem)
- **เปรียบเทียบ:** หาทางไปจุดหมายในเมืองที่มีทางแยกจำนวนมาก ซึ่งยังไม่มีวิธีลัดที่มีประสิทธิภาพด้วย Quantum

Hash (Hash-based)

เช่น SLH-DSA (ลายมือชื่อ)



ข้อมูล (เอกสาร) → ค่าลายนิ้วมือ (ความยาวคงที่) → ใช้ยืนยันว่าข้อมูลเดิม ไม่ถูกแก้ไข

- **ความปลอดภัย:** อาศัย “ฟังก์ชันย่อข้อมูล (Hash Function)” ที่ย้อนกลับหาข้อมูลเดิมได้ยากมาก
- **เปรียบเทียบ:** สร้าง “ลายนิ้วมือดิจิทัล” ของข้อมูล เพื่อใช้ยืนยันความถูกต้อง

ภาพเปรียบเทียบเพื่อความเข้าใจ ไม่ใช่ขั้นตอนคำนวณจริงของอัลกอริทึม

Cryptography ทำงาน 3 หน้าที่

อย่าสับสนการสร้างกุญแจ การยืนยันตัวตน และการเข้ารหัส

สร้างกุญแจ

ML-KEM

Key Establishment

ลงลายมือชื่อ

ML-DSA / SLH-DSA

Digital Signature

ปกป้องข้อมูล

AES-256-GCM

Symmetric

Encryption

เปลี่ยน Key Exchange ไม่ได้เปลี่ยน Certificate อัตโนมัติ

ML-KEM: สร้างกุญแจร่วมกัน

Module-Lattice-Based Key-Encapsulation Mechanism

01

KeyGen

ฝ่าย A สร้าง Public / Secret Key

02

Encaps

ฝ่าย B สร้าง Ciphertext + Shared Secret

03

Decaps

ฝ่าย A ได้ Shared Secret เดียวกัน

Shared Secret ⟶ Key Derivation ⟶ AES-GCM

ML-DSA: ลงนามเอกสารดิจิทัล

Module-Lattice-Based Digital Signature Algorithm

01

KeyGen

หน่วยงานสร้างกุญแจลงนาม

02

Sign

ลงนามบนเอกสารด้วย Private Key

03

Verify

ผู้รับตรวจด้วย Public Key

Signature ช่วยตรวจแหล่งที่มาและการแก้ไข ไม่ได้ซ่อนเนื้อหา

SLH-DSA: ลายเซ็นอีกแนวทาง

Stateless Hash-Based Digital Signature Algorithm

01

Hash

ใช้ Cryptographic Hash Functions

02

Sign

สร้างหลักฐานลายเซ็นแบบ Hash-Based

03

Verify

ตรวจสอบเส้นทาง/ค่าที่ผูกกับ Public Key

ต่างพื้นฐานจาก ML-DSA ⟶ ลดการพึ่งพา Algorithm Family เดียว

PQC ใน TLS 1.3: เปลี่ยนส่วนไหน?

Key Establishment ≠ Certificate Authentication ≠ AES

Authentication

Certificate / RSA / ECDSA

ยังต้องมีแผน PQC

Key Establishment

X25519MLKEM768

Hybrid PQC

Data Encryption

AES–256–GCM

ยังใช้ต่อได้

Hybrid KEM ลด HNDL แต่ไม่ทำให้ Certificate เป็น PQC ทันที

HOW

NIST IR 8547: เปลี่ยนผ่านอย่างเป็นระบบ

Transition to Post-Quantum Cryptography Standards (2026)

01

รู้จักระบบ

ค้นหา Algorithm / Dependency

02

ลดความเสี่ยง

เลือกข้อมูลและระบบสำคัญ

03

เปลี่ยนผ่าน

ทดสอบ / วางแผน / ติดตาม

NIST ให้ทิศทางการเปลี่ยนผ่าน ไม่ใช่แค่ชื่อ Algorithm

1 | เริ่มจาก Cryptographic Inventory

ตัวอย่าง: ระบบ VPN ที่ใช้อยู่ทุกวัน

ต้องบันทึก

Asset + Owner

Protocol + Algorithm

ต้องตรวจ

Certificate / Library

HSM / Firmware /

Vendor

ผลลัพธ์

รายการระบบที่เปลี่ยนได้

และระบบที่ติดข้อจำกัด

ตัวอย่าง: VPN Gateway ใช้ ECDH ⟶ ตรวจ Vendor PQC Roadmap

เริ่มจากภาพรวม: Crypto อยู่ที่ไหน?

Cryptographic Inventory = รายการการใช้งาน Cryptography ขององค์กร

TLS / HTTPS

เว็บและ API

VPN / SSH

เชื่อมต่อระยะไกล

PKI / HSM

Certificate และกุญแจ

Code Signing

Software / Firmware

Cloud / SaaS

บริการภายนอก

IoT / OT

อุปกรณ์และโรงงาน

อย่าถามแค่ว่า “รองรับ PQC หรือยัง?” ต้องถามว่า “ใช้ Crypto ที่ไหนและทำหน้าที่อะไร?”

ตัวอย่าง Inventory ที่ทีม IT ทำได้ทันที

เริ่มด้วย Spreadsheet ก่อนก็ได้

ระบบ	ใช้งาน Crypto	เจ้าของ	สิ่งที่ต้องถาม
VPN Gateway	ECDH / IKEv2	Network	Firmware รองรับ PQC?
Document Sign	RSA Signature	Application	HSM / CA รองรับ?
Web Portal	TLS 1.3 / ECDHE	Web Team	Hybrid TLS พร้อม?

นี่คือตัวอย่างข้อมูลสมมติ — ใช้เป็นต้นแบบสำหรับ Workshop ภายในองค์กร

2 | จัดลำดับด้วยความเสี่ยง

ข้อมูลอายุยาวและระบบที่เปลี่ยนยากต้องเริ่มเร็ว

Government Archive

ข้อมูลลับ 20 ปี
HNDL สูง

Core Service

หยุดให้บริการไม่ได้
ต้องทดสอบหนัก

Legacy OT

Firmware อัปเดตยาก
เริ่มคุย Vendor เร็ว

แยก “เริ่มวางแผนก่อน” ออกจาก “Deploy ก่อน”

3 | เลือก Target Architecture

เริ่มจาก Use Case ไม่ใช่เลือก Algorithm ตามชื่อ

TLS / VPN

Hybrid หรือ PQC Key Establishment ตาม Protocol

Digital Signature

ประเมิน ML-DSA / SLH-DSA และ PKI

Data at Rest

ประเมิน Key Management; AES-GCM ยังมีบทบาท

ถาม Vendor: รองรับ Algorithm + Protocol + Version ใดจริง?

4 | ทำ Pilot: ต้องตรวจอะไรบ้าง?

ตัวอย่าง Hybrid TLS 1.3 ในระบบทดสอบ

01

ตั้งค่า

Client / Server รองรับ Hybrid

02

ตรวจจริง

Negotiated Group = X25519MLKEM768

03

ประเมิน

Auth + Security + Performance

เชื่อมต่อสำเร็จ ≠ ใช้ PQC จริง

5 | Pilot ปลอดภัยแค่ไหน?

แยกหลักฐานความมั่นใจออกเป็น 3 ระดับ

ALGORITHM

FIPS 203–205

Cryptanalysis / Standards

IMPLEMENTATION

Library / Validation

Keys / Randomness

SYSTEM

TLS Configuration

Authentication / Monitoring

Pilot ไม่พิสูจน์คณิตศาสตร์ แต่ตรวจว่าระบบใช้ได้ถูกต้อง

6 | เกณฑ์ผ่าน Pilot ที่วัดได้

กำหนด Acceptance Criteria ก่อนทดลอง

SECURITY

Group ถูกต้อง

Certificate ตรวจผ่าน

COMPATIBILITY

Client / Server / Proxy ทำงานร่วมกัน

OPERATIONS

Latency / CPU / Error

Rollback / Monitoring

ส่งมอบ: Test Report + Risk Register + Go / No-Go Decision

7 | Migration Roadmap + NIST Timeline

ระยะเร่งด่วนของสหรัฐฯ ไม่ใช่เส้นตายกฎหมายไทย

01

DISCOVER

Inventory + Data Lifetime

02

PLAN / PILOT

Risk + Vendor + Testing

03

MIGRATE

Rollout + Governance

2030 / 2031 / 2035: ตรวจขอบเขตข้อกำหนดก่อนนำไปใช้

8 | เริ่มได้จริงใน 90 วัน

เป้าหมายคือ “รู้ความเสี่ยงและมีแผน” ไม่ใช่ย้ายทั้งองค์กร

01

วัน 1–30

แต่งตั้ง Owner

เลือก Critical Systems

02

วัน 31–60

Crypto Inventory

Data Lifetime

03

วัน 61–90

Vendor Review

Pilot Plan + Roadmap

ผลลัพธ์: Inventory • Priority List • Pilot Scope • Owner

องค์กรพร้อมหรือยัง?

รู้ว่า Crypto อยู่ที่ไหน • รู้ว่าอะไรเสี่ยง • รู้ว่าจะเริ่มอย่างไร

KNOW

รู้ Cryptography และ Dependencies

PRIORITIZE

เลือกข้อมูลและระบบที่สำคัญ

ACT

เริ่ม Pilot และสร้าง Crypto-agility

อย่ารอ Quantum Computer พร้อม แล้วค่อยเริ่มเตรียมตัว