03 Cryptography

Engr 399/599: Hardware Security
Grant Skipper, Ph.D.
Indiana University

Adapted from: Mark Tehranipoor of University of Florida

Course Website

engr599.github.io

Write that down!

SIDE QUEST: LASERS!

Laser Fault Injection (LFI)

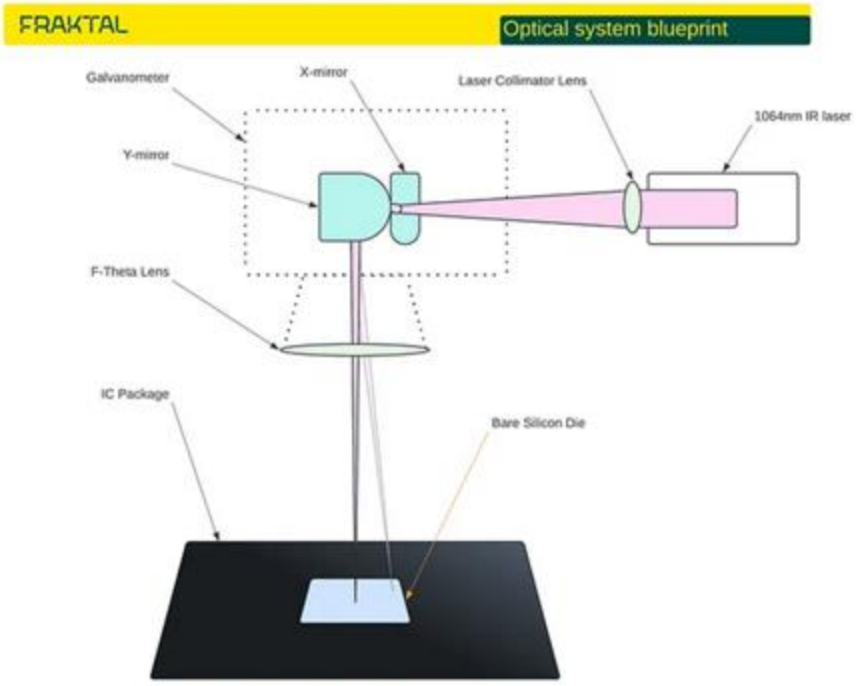
What is Fault Injection?

What is LFI?

Pros/Cons

Fun thing:

- https://github.com/fraktalcyber/lfi-rig
- https://blog.fraktal.fi/laser-fault-injection-for-the-masses-1860afde5a26
- https://www.youtube.com/watch?v=4ts3wNRt18g



Optical system blueprint

DES (Data Encryption Standard)

Background and History of DES (1)

- Early 1970's - NBS (Nat'l Bureau of Standards) recognized general public's need for a secure crypto system

  NBS – part of US gov't / Now: NIST – Nat'l Inst. of Stand's & Technology

  – "Encryption for the masses" [A. Striegel]
  – Existing US gov't crypto systems were not meant to be made public
    - E.g. DoD, State Dept.
  – Problems with proliferation of commercial encryption devices
    - Incompatible
    - Not extensively tested by independent body

Background and History of DES (2)

- 1972 - NBS calls for proposals for a *public* crypto system
  - Criteria:
    - Highly secure / easy to understand / publishable / available to all / adaptable to diverse app's / economical / efficient to use / able to be validated / exportable
    - In truth: Not *too* strong (for NSA, etc.)
- 1974 – IBM proposed its Lucifer
  - DES *based* on it
  - Tested by NSA (Nat'l Security Agency) and the general public
- Nov. 1976 – DES adopted as US standard for *sensitive but unclassified* data / communication
  - Later adopted by ISO (Int'l Standards Organization)
  - Official name: DEA - Data Encryption Algorithm / DEA-1 abroad

Overview of DES

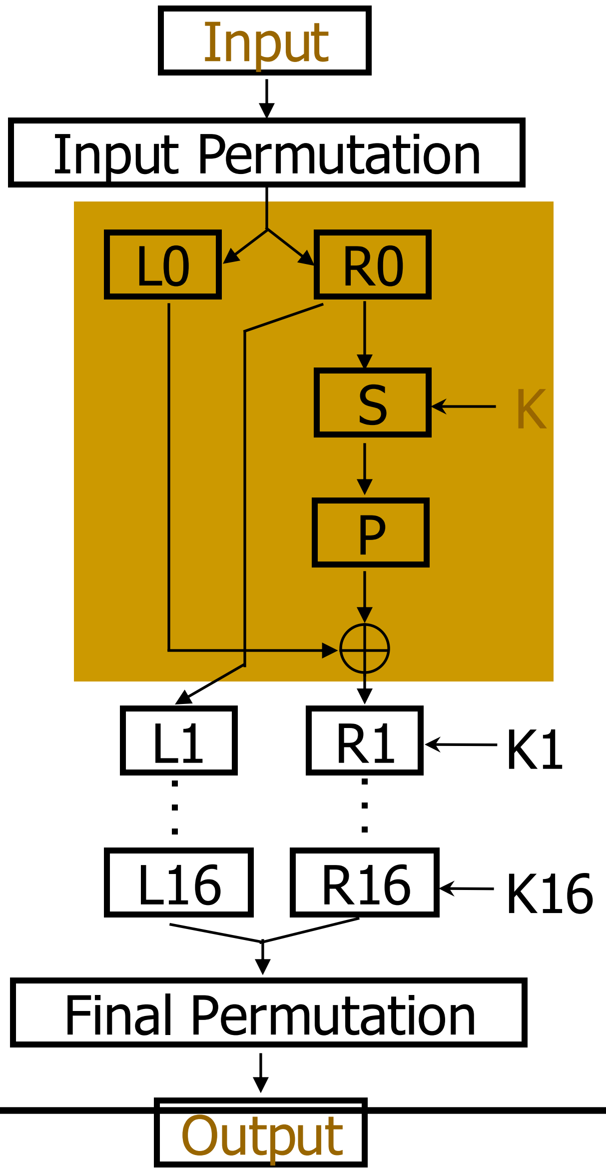
- DES - a block cipher
  - a product cipher
  - 16 rounds (iterations) on the input bits (of P)
    - substitutions (for confusion) and permutations (for diffusion)
  - Each round with a *round key*
    - Generated from the user-supplied key
- Easy to implement in S/W or H/W

- There are 72,000,000,000,000,000 (72 quadrillion) or more possible encryption keys that can be used.
- For each given message, the key can be chosen at random from among this enormous number of keys.

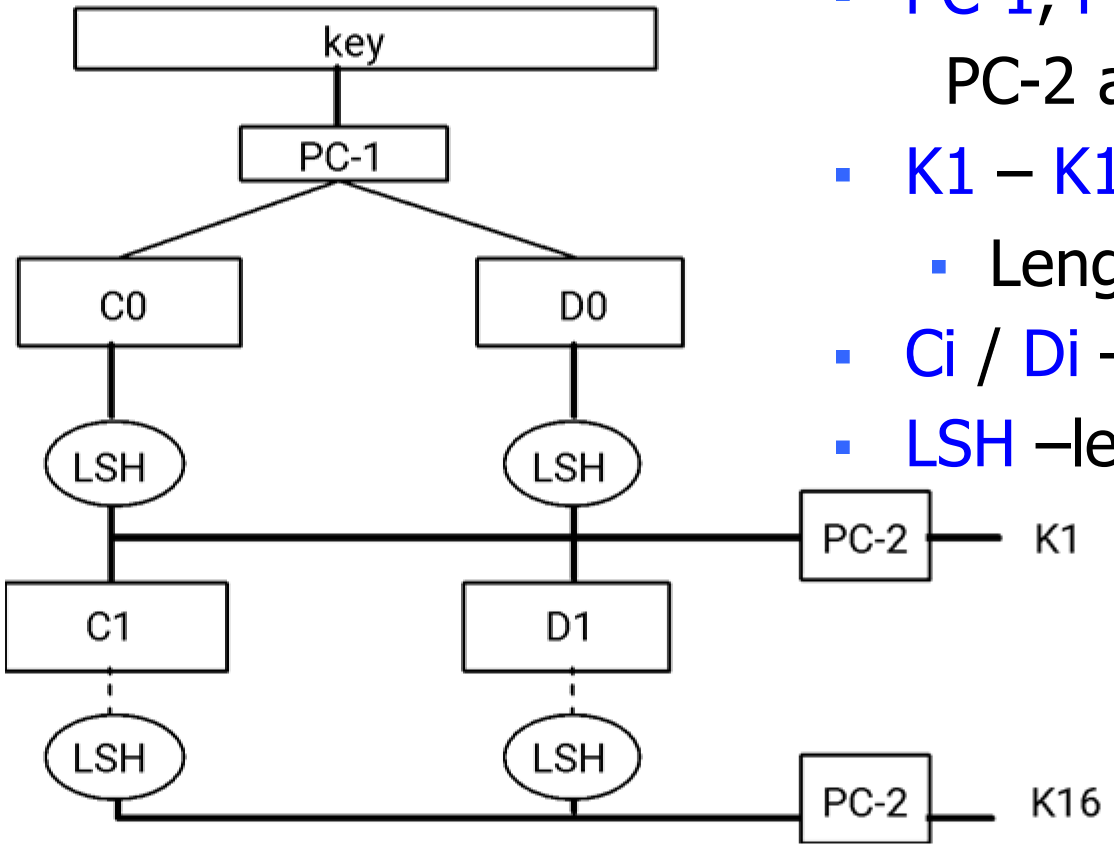
Basic Structure

- Input: 64 bits (a block)
- Li/Ri– left/right half of the input block for iteration i (32 bits) – subject to substitution S and permutation P
- K - user-supplied key
- Ki - round key:
  - 56 bits used +8 unused
    (unused for E but often used for error checking)
- Output: 64 bits (a block)
- Note: Ri becomes L(i+1)
- All basic op's are simple logical ops
  - Left shift / XOR

[Fig. – cf. J. Leiwo]



Generation of Round Keys



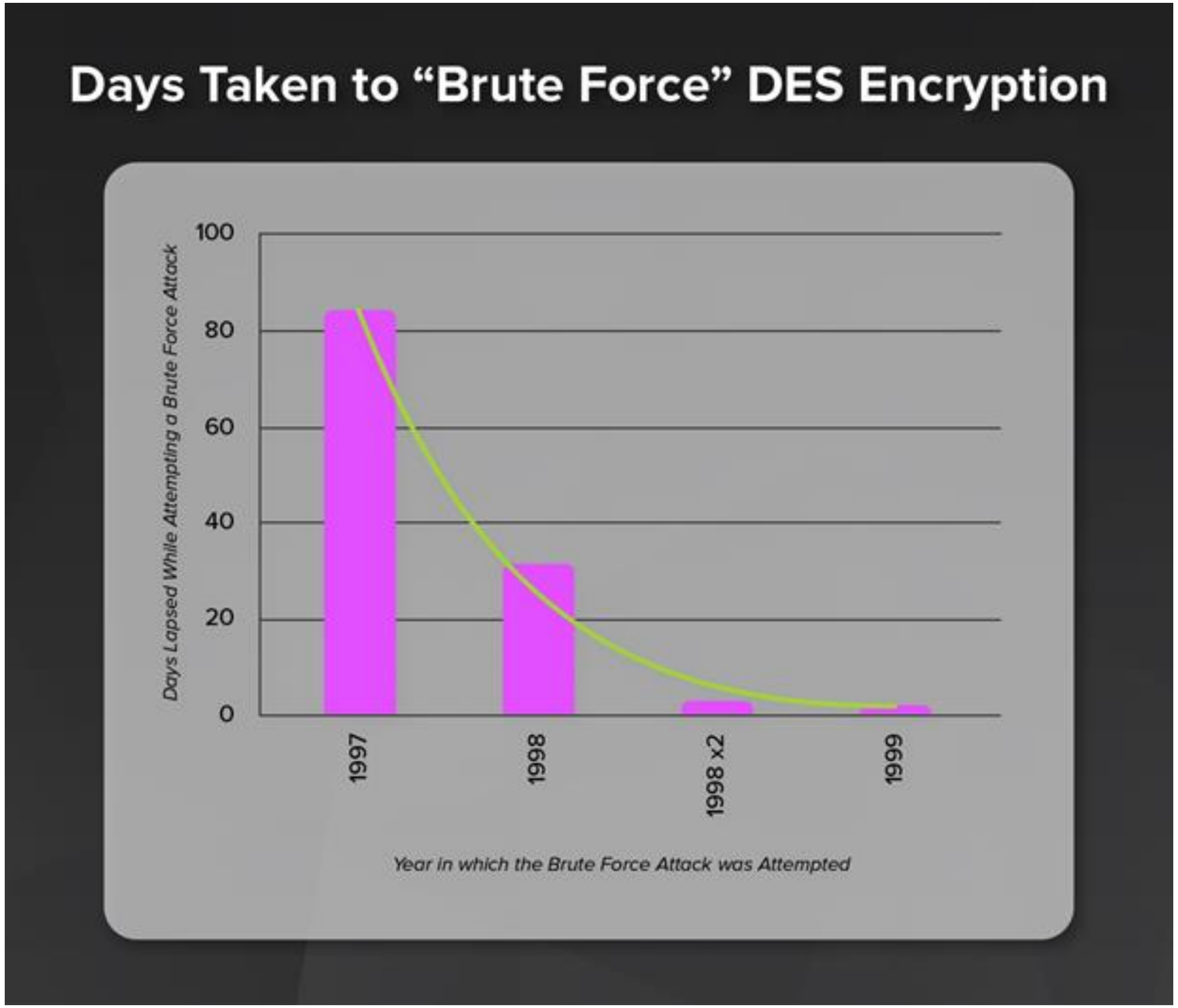
- key – user-supplied key (input)
- PC-1, PC-2 – permutation tables
  PC-2 also extracts 48 of 56 bits
- K1 – K16 – round keys (outputs)
  - Length(Ki) = 48
- Ci / Di – confusion / diffusion (?)
- LSH –left shift (rotation) tables

[Fig: cf. Barbara Endicott-Popovsky, U. Washington]

Problems with DES

- Diffie, Hellman 1977 prediction: "In a few years, technology would allow DES to be broken in days."
- Key length is fixed (= 56)
  - 2^{56} keys ~ 10^{15} keys
  - "Becoming" too short for faster computers
    - 1997: 3,500 machines – 4 months
    - 1998: special "DES cracker" h/w – 4 days
- Design decisions not public
  - Suspected of having backdoors
    - Speculation: To facilitate government access?

Problems with DES



Double and Triple DES

- Double DES:
  - Use double DES encryption

    C = E(k2, E(k1, P))
  - Expected to multiply difficulty of breaking the encryption
    - Not true!
      - In general, 2 encryptions are not better than one

        [Merkle, Hellman, 1981]
      - Only doubles the attacker's work

Double and Triple DES (2)

- Triple DES:
  - Is it C = E(k3, E(k2, E(k1, P)) ?
  - Not soooo simple!

Double and Triple DES (3)

- Triple DES: *Is it C=E(k3, E(k2, E(k1, P))?*
  - Tricks used:

    D not E in the 2nd step, k1 used twice (in steps 1 & 3)
  - It is:

    C = E(k1, D(k2, E(k1, P))

    and

    P = D(k1, E(k2, D(k1, C))
  - Doubles the effective key length
    - 112-bit key is quite strong
      - Even for today's computers
      - For all feasible known attacks

Security of DES

- So, is DES insecure?
- No, not yet
  - 1997 attack required a lot of cooperation
  - The 1998 special-purpose machine is still very expensive
  - Triple DES still beyond the reach of these 2 attacks
- But ...
  - In 1995, NIST (formerly NBS) began search for new strong encryption standard

The AES Contest (1)

- 1997 – NIST calls for proposals NIST (Nat'l Institute of Standards and Technology)
  - Criteria:
    - Unclassifed code
    - Publicly disclosed
    - Royalty-free worldwide
    - Symmetric block cipher for 128-bit blocks
    - Usable with keys of 128, 192, and 256 bits
- 1998 – 15 algorithms selected

The AES Contest (2)

- 1999 – 5 finalists [cf. J. Leiwo]
  - MARS by IBM
  - RC6 by RSA Laboratories
  - Rijndael (RINE-dahl) by Joan Daemen and Vincent Rijmen
  - Serpent by Ross Anderson, Eli Biham and Lars Knudsen
  - Twofish by Bruce Schneier, John Kelsey, Doug Whiting, Dawid Wagner, Chris Hall and Niels Ferguson
- Evaluation of finalists
  - Public and private scrutiny
  - Key evaluation areas:

    security / cost or efficiency of operation /

    ease of software implementation

The AES Contest (3)

- 2001- … and the winner is …

  Rijndael (RINE-dahl)

  Authors: Vincent Rijmen + Joan Daemen (Dutchmen)

- Adopted by US gov't as

  Federal Info Processing Standard 197 (FIPS 197)

Overview of Rijndael/AES

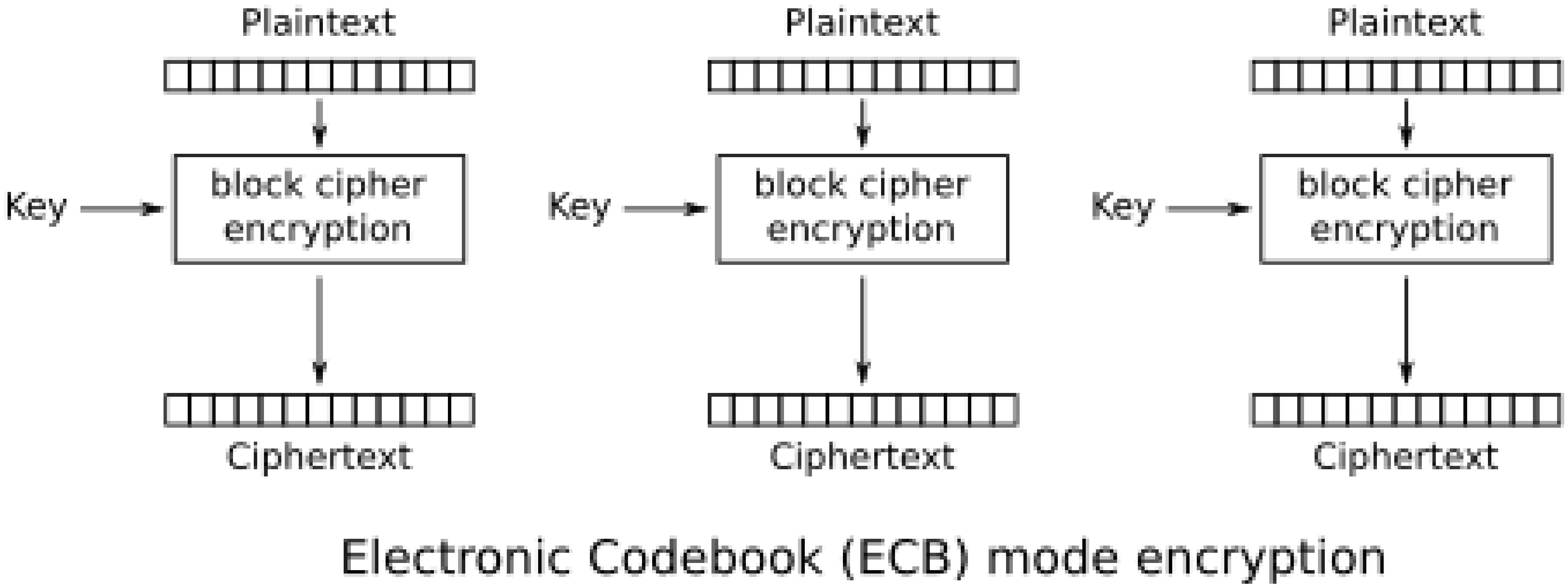
- Similar to DES – cyclic type of approach
  - 128-bit blocks of P
  - # of iterations based on key length
    - 128-bit key => 9 "rounds" (called rounds, not cycles)
    - 192-bit key => 11 rounds
    - 256-bit key => 13 rounds
- Basic ops for a round:
  - Substitution – byte level (confusion)
  - Shift row (transposition) – depends on key length (diff.)
  - Mix columns – LSH and XOR (confusion +diffusion)
  - Add subkey – XOR used (confusion)

Strengths of AES

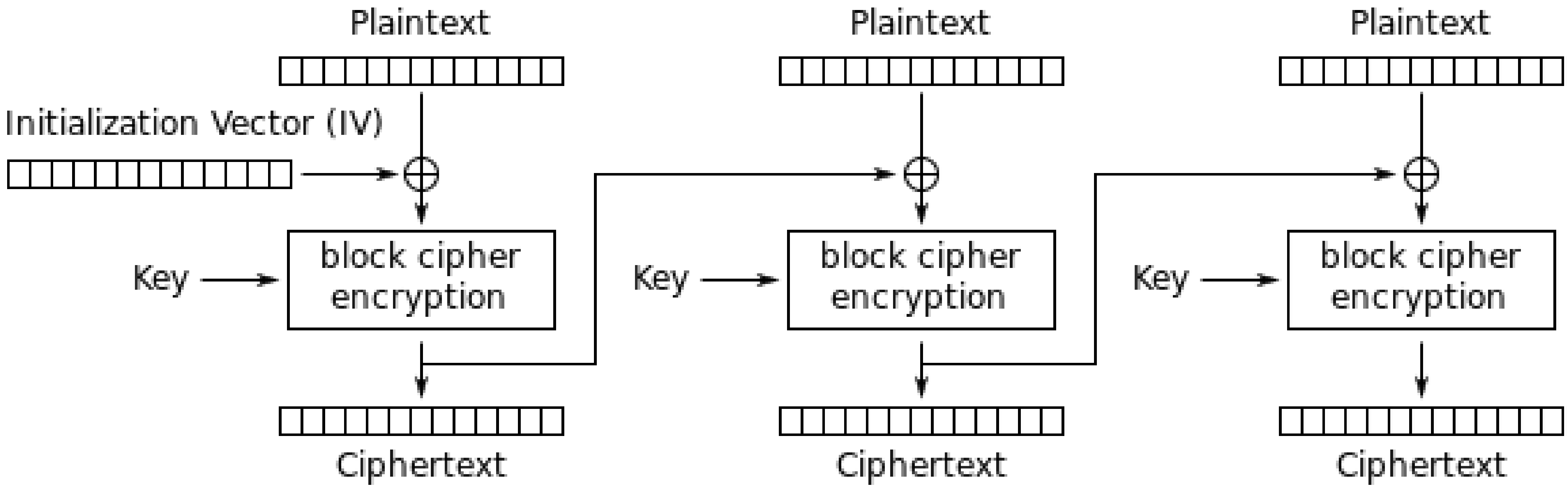
- Extensive cryptanalysis by US gov’t and independent experts
- Dutch inventors have no ties to NSA or other US gov’t bodies (less suspicion of trapdoor)
- Solid math basis
  - Despite seemingly simple steps within rounds

- TODO LEAD INTO NEXT SLIDES.

AES-Electronic Codebook (ECB)

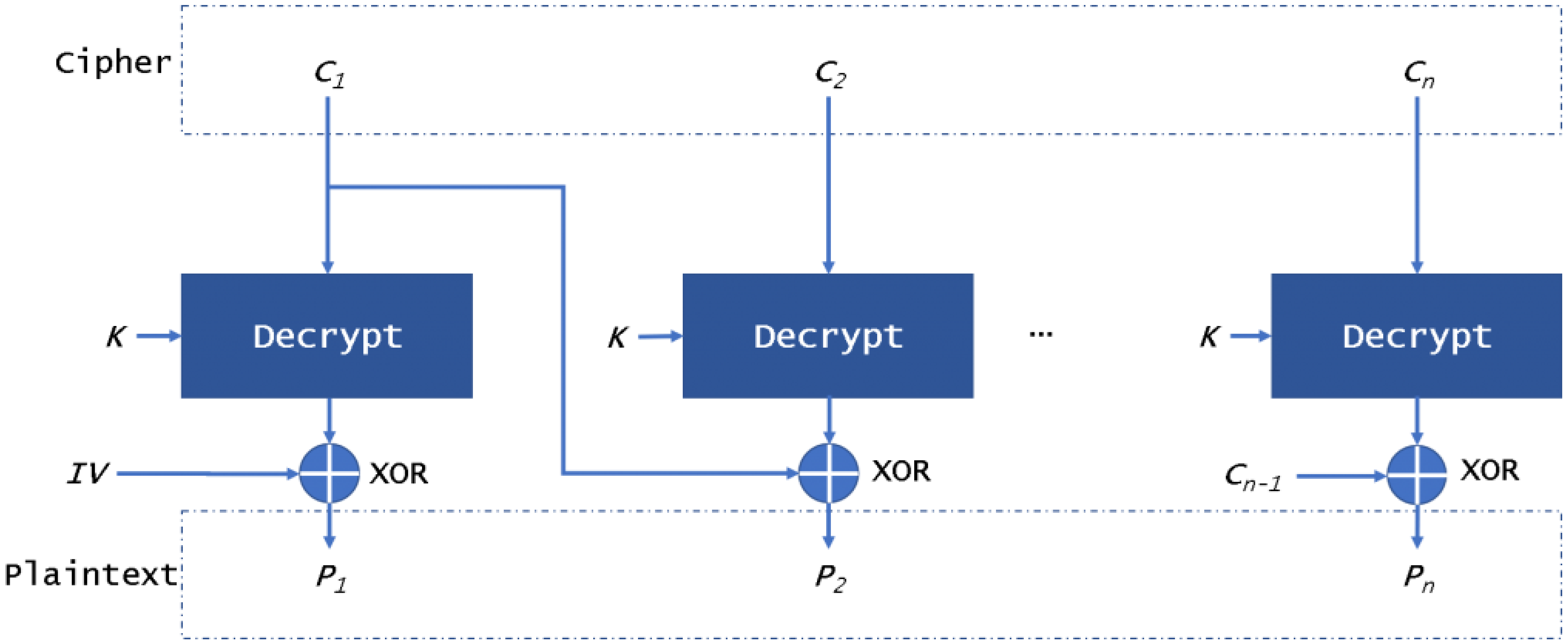


AES-Cipher Block Chaining (CBC)



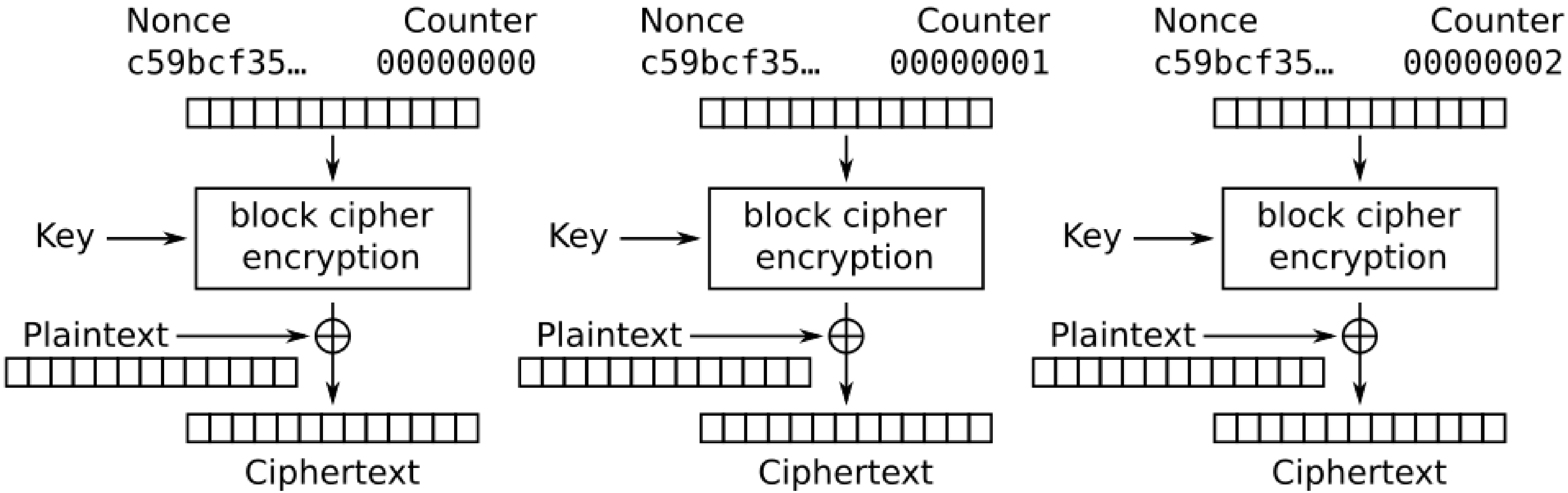
Cipher Block Chaining (CBC) mode encryption

AES-Cipher Block Chaining (CBC)



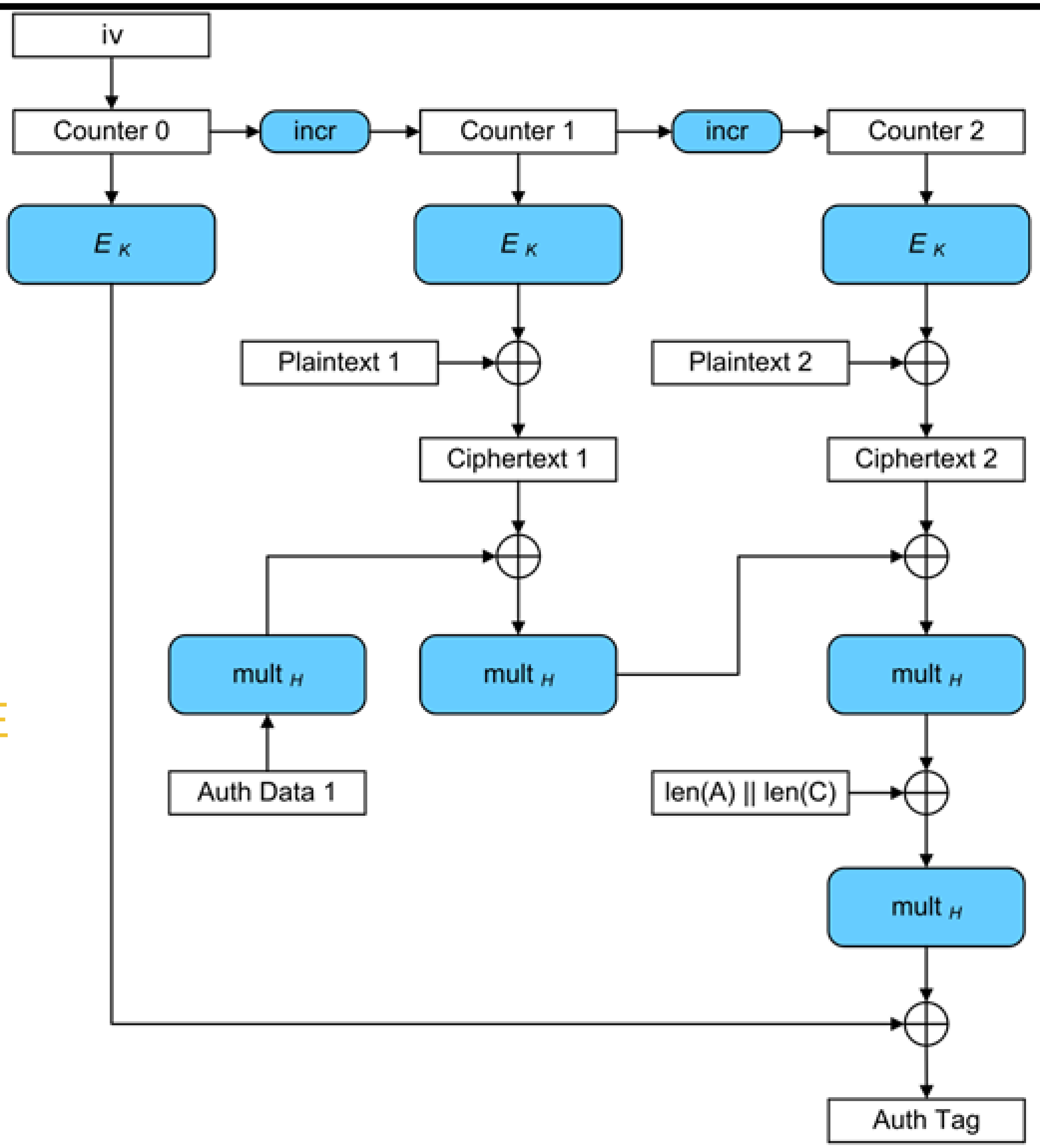
CBM Attack! Cipher Block Malleability! Happens Today!
AES CTR also affected, but less so.
Is GCM affected?

AES-Counter Mode (CTR)



Counter (CTR) mode encryption

AES-Galois Counter Mode (GCM)



GOLDEN RULE:

ALWAYS AUTHENTICATE

AUTHENTICATE BEFORE USE

Comparison of DES & AES (1)

	DES	AES
Date	1976	1999
Block size [bits]	64	128
Key length [bits]	56 (effect.)	128, 192, 256, or more
Encryption Primitives	substitution, permutation	substitution, shift, bit mixing
Cryptographic Primitives	confusion, diffusion	confusion, diffusion
Design	open	open
Design Rationale	closed	open
Selection process	secret	secret, but accepted public comments
Source	IBM, enhanced by NSA	independent Dutch cryptographers

Comparison of DES & AES (2)

- Weaknesses in AES?
  - 20+ yrs of experience with DES eliminated fears of its weakness (intentional or not)
    - Might be naïve…
  - Experts pored over AES for 2-year review period

Comparison of DES & AES (3)

- Longevity of AES?
  - DES is nearly 40 yrs old (1976)
    - DES-encrypted message can be cracked in days
  - Longevity of AES more difficult to answer
    - Can extend key length to > 256 bits (DES: 56)
      - 2 * key length => 4 * number of keys
    - Can extend number of rounds (DES: 16)
  - Extensible AES seems to be significantly better than DES, but..
    - Human ingenuity is unpredicatble!

    => Need to incessantly search for better and better encryption algorithms

Motivation for PKE (1)

- So far - cryptosystems with secret keys
- Problems:
  - A lot of keys
    - $o(n^2)$ keys for n users (n * (n-1) /2 keys)

      — if each must be able to communicate with each
  - Distributing so many keys securely
  - Secure storage for the keys
    - User with n keys can't just memorize them
- Can have a system with significantly fewer keys?

  Yes!

Motivation for PKE (2)


Whitfield Diffie

- 1976 — Diffie and Hellman — new kind of cryptosystem:

  public key cryptosystem = asymmetric cryptosystem

  - Key pairs: < $k_{PRIVATE}$, k_{PUBLIC}>
  - Each user owns one private key
  - Each user shares the corresponding public key with n-1 remaining users => n users share each public key
  - Only 2n keys for n users $2n = n * (1 + n * 1/n)$
    - » Since public key is shared by n people: 1 "owner" + (n-1) others = n
    - » 1/n since each part "owns" 1/n of the public key
    - Even if each communicates with each
    - Reduction from $o(n^2)$ to $o(n)$!
    - n key pairs are:

      <k_{PRIV-1}, k_{PUB-1} >, <k_{PRIV-2}, k_{PUB-2}>, ..., <k_{PRIV-n}, k_{PUB-n}>


Martin E. Hellman

Characteristics of PKE (1)

- PKE requirements
  1. It must be computationally easy to encipher or decipher a message given the appropriate key
  2. It must be computationally infeasible to derive k_{PRIV} from k_{PUB}
  3. It must be computationally infeasible to determine k_{PRIV} from a chosen plaintext attack

[cf. Barbara Endicott-Popovsky, U. Washington]

Characteristics of PKE (2)

- Key pair characteristics
  - One key is inverse of the other key of the pair
    - i.e., it can undo encryption provided by the other:
      - $D(k_{PRIV}, E(k_{PUB}, P)) = P$
      - $D(k_{PUB}, E(k_{PRIV}, P)) = P$
  - One of the keys can be public since each key does only half of E "+" D
    - As shown above – need both E and D to get P back

Characteristics of PKE (3)

- Two E/D possibilities for key pair $<k_{PRIV}, k_{PUB}>$
  - $P = D(k_{PRIV}, E(k_{PUB}, P))$
    - User encrypts msg with k_{PUB} (k_{PUB}" "locks")
    - Recipient decrypts msg with k_{PRIV} (k_{PRIV} "unlocks")

  OR

  - $P = D(k_{PUB}, E(k_{PRIV}, P))$ (e.g., in RSA)
    - User encrypts msg with k_{PRIV} (k_{PRIV} "locks")
    - Recipient decrypts msg with key k_{PUB} (k_{PUB} "unlocks")

- Do we still need symmetric encryption (SE) systems?
  - Yes, PKEs are 10,000+ times (!) slower than SEs
    - PKEs use exponentiation – involves multiplication and division
    - SEs use bit operations (add,XOR< substitute, shift)–much faster

RSA Encryption (1)

- RSA = Rivest, Shamir, and Adelman (MIT), 1978
- **RSA** is one of the first practical public-key cryptosystems and is widely used for secure data transmission.
- Underlying hard problem:
  - Number theory – determining prime factors of a given (large) number (ex. factoring of small #: 5 ☐ 5, 6 ☐ 2 *3)
  - Arithmetic modulo n
- How secure is RSA?
  - So far remains secure (after all these years...)
  - ~~Will quantum computing break it? TBD~~

RSA Encryption (2)

- In RSA:
  $P = E(D(P)) = D(E(P))$ (order of D/E does not matter)
  - More precisely: $P = E(k_E, D(k_D, P)) = D(k_D, E(k_E, P))$
- Encryption: $C = P^e \bmod n$ $K_E = e$
  - Given C, it is very difficult to find P without knowing K_D
- Decryption: $P = C^d \bmod n$ $K_D = d$

Post-Quantum Cryptography (PQC)

Quantum computers are rapidly being developed and will eventually be deployed

Quantum computers threaten cryptographic algorithms due to their speed

PQC algorithms in development to prepare for a quantum future.

Products TODAY are being made with PQC in mind!

Quantum Computer

https://www.csoonline.com/article/3562701/chinese-researchers-break-rsa-encryption-with-a-quantum-computer.html

New 2024: China factors 22-bit RSA Integer with Quantum Computer

Google's "Willow" Quantum Computer

Latest PQC Algorithms - NIST

1. FIPS 203 - Module-Lattice-Based Key-Encapsulation Mechanism (CRYSTALS-KYBER)

1. FIPS 204 - Module-Lattice-Based Digital Signature (CRYSTALS-DILITHIUM)

1. FIPS 205 - Stateless Hash-Based Digital Signature (SPHINCS+)

FIPS - Federal Information Processing Standards

Stateful Hash-Based Signatures

Stateful hash-based signatures same goal as PQC algorithms to prevent quantum attacks, difference is the underlying mathematical structure (state vs. stateless)

1. Leighton-Micali Signature System (LMS)
2. eXtended Merkle Signature Scheme (XMSS)

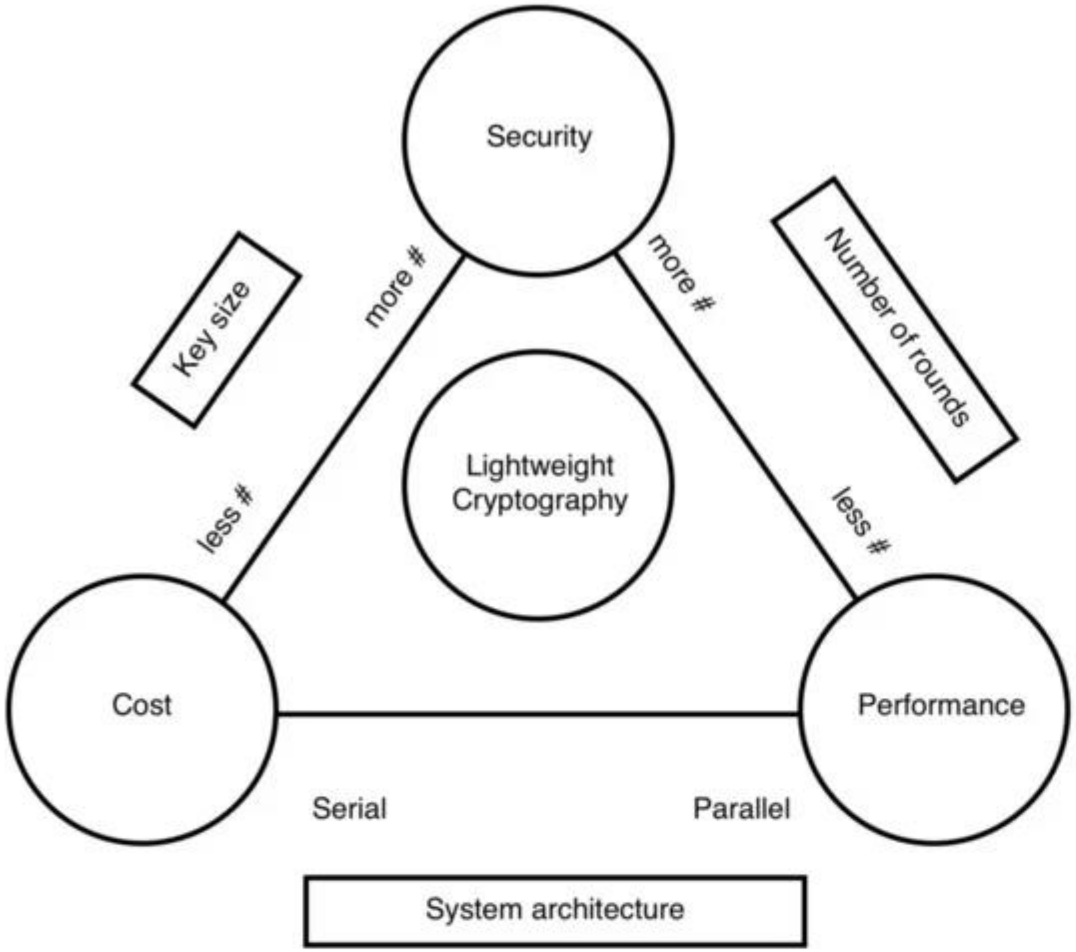
Lightweight Cryptography (LWC)

Most Internet of Things (IoT) devices, embedded systems, or power-restricted devices in the past and to this day transmitted data unencrypted…

These devices cannot afford power, performance or area of conventional cryptographic algorithms

Lightweight Cryptography being developed to help reduce the “cost” of conventional algorithms -> NIST SP 800-232: Ascon

LWC TRADE OFFS!!!



LWC Timeline

NIST are the facilitators of the standardization process, but…

Open to anyone to help with the standardization process

- Algorithm Submissions
- Comments
- Testing/Attacking
- Standardization Criteria

All work is done publicly

Profile Comments

57 Submissions for Round 1

Comments on Bleep64 Algorithm

Date	Event
July 20-21, 2015	First Lightweight Cryptography Workshop at NIST
August 11, 2016	(Draft) NIST IR 8114 is published.
October 17-18, 2016	Second Lightweight Cryptography Workshop at NIST
October 31, 2016	End of public comment period to Draft NISTIR 8114 Public comments received (August 11 - October 31,2016)
March 28, 2017	NIST IR 8114, *Report on Lightweight Cryptography* is published.
April 26, 2017	(Draft) Profiles for Lightweight cryptography standardization process is published.
June 16, 2017	Public comments received (April 26 - June 16, 2017)
May 14, 2018	(Draft) Submission Requirements and Evaluation Criteria for the Lightweight cryptography standardization process is published.
May 14, 2018	Federal Register Notice is published.
June 28, 2018	End of public comment period to the submission requirement. Public comments received (May 14-June 28, 2018).
August 27, 2018	Federal Register Notice is published
August 27, 2018	Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process is published.
January 4, 2019	Early submission deadline for early feedback
February 25, 2019	Submission deadline
March 29, 2019	Amendment Deadline
April 18, 2019	Announcement of the Round 1 Candidates
August 30, 2019	Announcement of Round 2 Candidates
October 7, 2019	NIST IR 8268, *Status Report on the First Round of the NIST Lightweight Cryptography Standardization Process* is published
November 4-6, 2019	Third Lightweight Cryptography Workshop at NIST
October 19-21, 2020	Fourth Lightweight Cryptography Workshop (virtual)
March 29, 2021	Announcement of Finalist Candidates
July 21, 2021	NIST IR 8369, *Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process* is published.
May 9-11, 2022	Fifth Lightweight Cryptography Workshop (virtual)
February 7, 2023	Announcement: NIST Selects Ascon for Standardization
June 16, 2023	NIST IR 8454, *Status Report on the Final Round of the NIST Lightweight Cryptography Standardization Process* is published
June 21-22, 2023	Sixth Lightweight Cryptography Workshop (virtual)
November 8, 2024	NIST SP 800-232 (initial public draft), *Ascon-Based Lightweight Cryptography Standards for Constrained Devices: Authenticated Encryption, Hash, and Extendable Output Functions* is published.

Homomorphic Encryption

- What if we did not need to decrypt our messages?
- Homomorphic Encryption:
  - Client encrypts data
  - Server receives encrypted data.
  - Server operates on the encrypted data.
  - Client receives mutated enc. data from server.
  - Client decrypts data to see a new message!
- Is this useful?