Universidad Nacional Autónoma de México

Facultad de Ciencias

Criptografía y Seguridad

Tarea 7: Tarea DES y AES

Fecha de entrega: 21/11/2023

Equipo:

Criptonianos

Acosta Arzate Rubén - 317205776

Bernal Marquez Erick - 317042522

Deloya Andrade Ana Valeria - 317277582

Marco Antonio Rivera Silva - 318183583

1. Sistema de Cifrado *Camellia*

En el año 2000, Kazumaro Aoki, Masayuki Kanda y Shiho Moriai, un equipo de criptógrafos de **NTT (Nippon Telegraph and Telephone Corporation)** de Japón, junto con Tetsuya Ichikawa, Mitsuru Matsui, Junko Nakajima y Toshio Tokita de **Mitsubishi Electric Corporation**, desarrollaron Camellia contando con la misma interface que AES (Estándar de Cifrado Avanzado).

Camellia es un cifrado de bloque de clave simétrica que opera con bloques de 128 bits, cifrando y descifrando datos en segmentos fijos de tamaño. No sólo es compatible con bloques de 128 bits, sino que también admite claves de longitud de 192 y 256 bits. Camellia es el único cifrado en el mundo con niveles de rendimiento de procesamiento y seguridad equivalentes a AES ofreciéndonos las mismas especificaciones de éste cifrado

1. *Seguridad de alto nivel:* Se considera esencial una evaluación cuantitativa de la seguridad frente a potentes técnicas criptoanalíticas como el criptoanálisis diferencial y el criptoanálisis lineal en el diseño de cualquier nuevo cifrado de bloques. Camelia cumple con hacer complejo el criptoanálisis.

2. *Eficiencia en múltiples plataformas:* Pocos cifrados de bloques de 128 bits son adecuados para la implementación tanto de software como de hardware, sin embargo, el cifrado Camellia cuenta con la característica de su versatilidad en términos de implementación, debido a que puede implementarse con alto rendimiento mediante software en varias plataformas y para el hardware es posible una implementación del tipo compacto, es decir, puede adaptarse eficientemente en dispositivos con limitaciones de espacio, recursos y de bajo consumo de energía. Además de permitir implementaciones de alta velocidad en hardware.

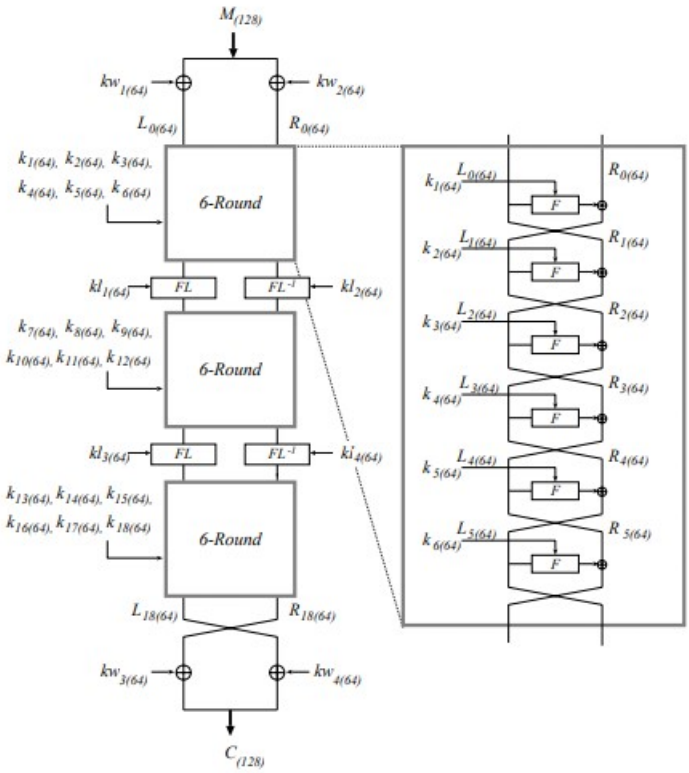
Estructura de Camellia

Este algoritmo de cifrado funciona como un cifrado Feistel, por lo que se divide el bloque de entrada en dos mitades (*L0 y R0*) y realiza varias rondas de operaciones en estas mitades. El número de rondas depende de la longitud de la clave, por lo que si la clave es de 128 bits son 18 rondas y si la clave es de 192 o 256 bits son 24 rondas. Además de que utiliza cuatro "S-boxes" de 8 x 8 bits con transformaciones afines de entrada y salida, así como operaciones lógicas denominadas *FL-function* y FL^{-1}*-function* las cuales son insertadas cada 6 rondas. Estas funciones proporcionan *confusión* y *difusión* en los datos, contribuyendo con esto a la seguridad de nuestro algoritmo.

Una vez finalizadas las rondas, se aplica una *capa de difusión* haciendo uso de una transformación lineal basada en una Matriz de Difusión Máxima (MDS) con 5 ramas. La matriz va a actuar como una función de difusión, mezclando así los bits del bloque de datos de manera extensa.

Además *Camellia* utiliza operaciones de *Key Whitening* al principio y al final del cifrado, donde se realizan operaciones XOR con subclaves obtenidas a partir de la clave principal. Esto con la finalidad de aumentar la seguridad al cifrado.

A continuación se muestra un ejemplo de Camellia para 128 bits con 18 rondas.



Vulnerabilidades de Camellia

Sus principales vulnerabilidades son ataques hacía bloques de cifrado en específico, siendo los más populares en los primeros 8 bloques, sin embargo existen diversos ataques a los que *Camellia* es susceptible, como:

- Differential and Linear Cryptanalysis:

  Analiza las diferencias entre textos cifrados relacionados. Se busca explotar las diferencias estadísticas en el comportamiento del cifrado.

- Truncated Differential Cryptanalysis

  Similar a la differential cryptanalysis, pero se enfoca en un número limitado de rondas del cifrado. Caso similar con *Truncated Linear Cryptanalysis* donde se analiza de manera lineal pero con un numero limitado de rondas.

- Boomerang Attack

  Utiliza textos cifrados que forman un "boomerang", es decir un camino de idea y vuelta en el espacio de texto cifrado, al pasar por dos caminos diferentes de cifrado, esto puede aumentar la probabilidad de encontrar la clave ya que queremos que los datos de entrada originales coincidan con los datos resultantes después de dos procesos de cifrado y descifrado.

- Higher Order Differential Attack

  Parecido a *differential cryptanalysis* pero toomando un enfoque más avanzado que considera diferencias en bloques más grandes y complejos

- Interpolation Attack and Linear Sum Attack

  Se basa en la interpolación polinómica para recuperar información sobre la clave.

- Slide Attack

  Aprovecha patrones repetitivos en el cifrado. Suele ser efectivo contra la estrcutura de Red de Feistel

- Related-key Attack

  Este ataque explota la relación entre claves relacionadas. Si dos claves tienen una relación conocida, se pueden utilizar para comprometer el cifrado.

Aplicaciones

Camellia ha sido implementado en múltiples softwares y ha demostrado tener un buen desempeño como se muestra en la siguiente tabla:

Table 7: Camellia software performance

Processor	Language	Key len. (bits)	Timing[a] Setup[e] ([f])	Timing[a] Enc.[g] ([h])	Dynamic[b] Setup[e]	Dynamic[b] Enc.[g]	Code[c] Setup[e]	Code[c] Enc.[g]	Table[d]
P III[i]	Assembly	128	160 (4.4M)	371 (242M)	28	36	1,046	2,150	8,224
		192	222 (3.2M)	494 (181M)	28	36	1,469	3,323	8,240
		256	226 (3.1M)	494 (181M)	28	36	1,485	3,323	8,240
P II[j]	ANSI C[k]	128	263 (1.1M)	577 (67M)	44	64	1,600	3,733	4,128
Alpha[l]	Assembly	128	118 (5.7M)	339 (252M)	48	48	1,132	3,076	16,528
		192	176 (3.7M)	445 (192M)	48	48	1,668	4,000	16,528
		256	176 (3.7M)	445 (192M)	48	48	1,676	4,000	16,528
		128	158 (4.2M)	326 (262M)	48	48	1,600	2,928	16,512
8051[m]	Assembly	128	0 (0)	10217 (10m)	0	32	0	702	288

Como se mencionó con anterioridad, una característica destacable de *Camellia* es su capacidad para tener implementaciones compactas, haciendo que ésta sea portátil.

Table 8: Hardware evaluation environment (ASIC, FPGA)

Language	(ASIC, FPGA) Verilog-HDL
Simulator	(ASIC, FPGA) Verilog-XL
Design library	(ASIC) Mitsubishi Electric 0.35μ CMOS ASIC library (FPGA) Xilinx XC4000XL series
Login synthesis	(ASIC) Design Compiler version 1998.08 (FPGA) Synplify version 5.3.1 and ALLIANCE version 2.1i

Table 9: Hardware performance (Type 1: Fast implementation [ASIC(0.35μ CMOS)])

Algorithm name	Area [Gate] Enc.&Dec.[a]	Area [Gate] Key expan.[b]	Area [Gate] Total logic[c]	Key setup time [ns]	Critical-path [ns][d]	Throughput [Mb/s]
DES	42,204	12,201	54,405	—	55.11	1161.31
Triple-DES	124,888	23,207	128,147	—	157.09	407.40
MARS	690,654	2,245,096	2,935,754	1740.99	567.49	225.55
RC6	741,641	901,382	1,643,037	2112.26	627.57	203.96
Rijndael	518,508	93,708	612,834	57.39	65.64	1950.03
Serpent	298,533	205,096	503,770	114.07	137.40	931.58
Twofish	200,165	231,682	431,857	16.38	324.80	394.08
Camellia	216,911	55,907	272,819	24.36	109.35	1170.55

2. Ataques al AES.

Ataques Cuánticos

Primero se propuso un nuevo marco para la búsqueda estructurada clásica y cuántica, que permite presentar de manera concisa nuestros algoritmos y calcular sus complejidades. En segundo lugar, se utilizó ese marco para presentar nuevos ataques cuánticos que son versiones cuánticas de las familias de criptoanálisis más eficientes en AES de ronda reducida. Si bien algunas de estas familias no se benefician de una aceleración competitiva o significativa se lograron acelerar dos. Aunque se consideraron varios modelos cuánticos para el atacante, los ataques se pueden ubicar en el modelo Q1, donde el atacante tiene acceso a una computadora cuántica pero está restringido a consultas clásicas de cifrado/descifrado.

Ataque cuadrado cuántico

El ataque cuadrado se propuso y se estudió en AES con 6 rondas como objetivo. Se ha ampliado a 7 rondas para AES-192 y 256. Utiliza un distinguidor integral en AES de 3 rondas, que necesita 256 textos claros elegidos (si un byte toma todos sus 28 valores posibles mientras los demás permanecen constantes, tres rondas después todos los bytes del estado interno estarán equilibrados). Se amplía agregando algunas rondas antes y después, a costa de una mayor complejidad de los datos (232 textos sin formato elegidos) y algunas conjeturas de bytes clave. Esta familia de ataques tiene un rendimiento clásico peor que los ataques de encuentro en el medio más conocidos, pero, sin embargo, es interesante como proporciona baja complejidad. Los ataques de datos bajos son de interés independiente.

Este ataque ya es un ataque cuántico para AES-128 de 6 rondas, en el sentido de que cuesta menos tiempo que la búsqueda exhaustiva de Grover (otro ataque). Se propusieron versiones cuantificadas del ataque cuadrado. Se ejecutan en el modelo Q1, en el que es esencial utilizar la técnica de sumas parciales. Sin embargo no se encontró una manera de hacerlo sin depender de qRAM.

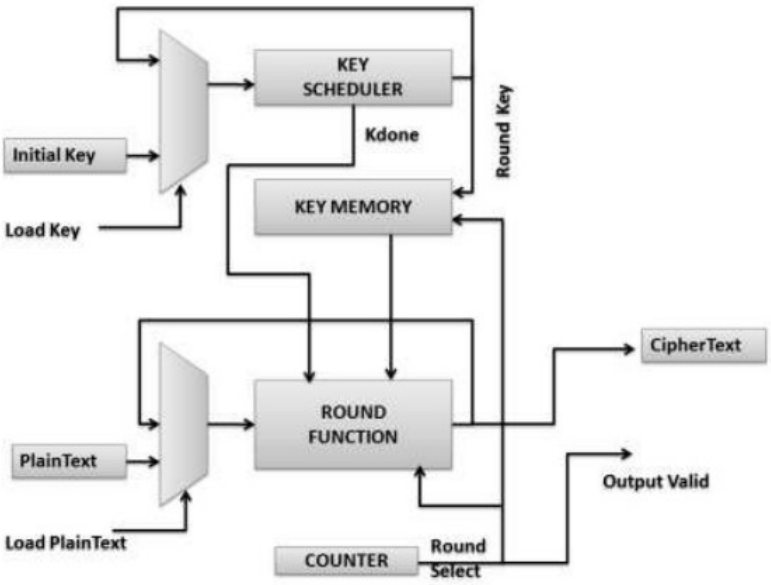
Version	Classical counterpart	Queries	Quantum time	Quantum memory	Classical memory	Grover on keys
6-rd. AES-128	[FKL+00]	2^{35}	2^{44}	2^{25}	2^{36}	$2^{72.2}$
7-rd. AES-256	[DKR97]	2^{37}	2^{121}	negligible	2^{38}	$2^{137.3}$
7-rd. AES-256	[FKL+00]	2^{37}	2^{107}	2^{27}	2^{38}	$2^{137.3}$
7-rd. AES-192	[FKL+00]	2^{37}	$2^{103.4}$	2^{27}	2^{38}	$2^{105.6}$

Ataque de recuperación de llaves

Un ataque de recuperación de claves (key recovery attack en inglés) es un tipo de ataque criptográfico que tiene como objetivo recuperar la clave secreta utilizada en un sistema criptográfico. La clave secreta es un componente crítico en muchos algoritmos criptográficos y su compromiso puede llevar a la revelación de información confidencial o a la posibilidad de realizar operaciones no autorizadas.

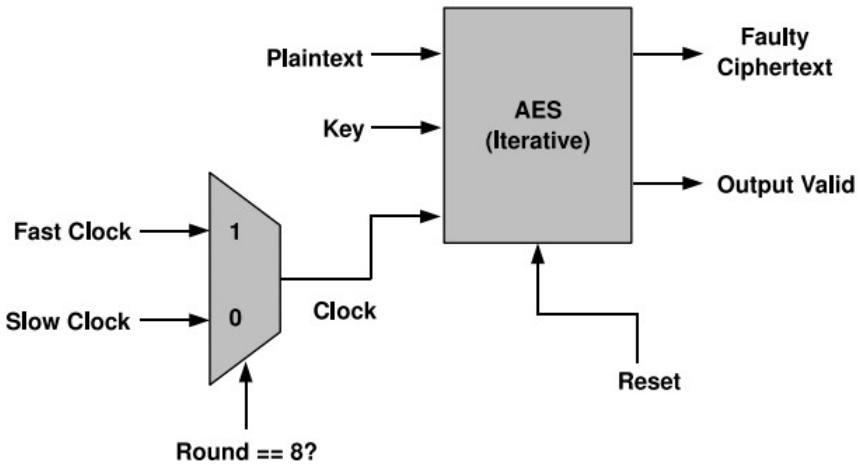
Según el articulo *A Diagonal Fault Attack on the Advanced Encryption Standard* la implementación del ataque fue la siguiente:

Se implementó una arquitectura iterativa de **AES** utilizando *Verilog HD*L. La clave y el texto sin formato se cargan cuando se confirman las señales 'Cargar clave' y 'Cargar texto sin formato'. El primer módulo genera cada clave redonda y la almacena en la Memoria de claves. Una vez finalizada la generación de claves, el Programador de claves afirma la señal *Kdone*. Esto sucede una vez por cada nueva clave cargada. Luego, el módulo de función circular itera 10 veces antes de generar el texto cifrado que se indica mediante la señal de salida válida. El controlador consta de un contador que realiza un seguimiento del número de rondas y también selecciona la clave de ronda específica de la Memoria de claves. El diseño se descarga en una plataforma Xilinx Spartan-3E XC3S500E y opera a una frecuencia máxima de 36 MHz.



La eficacia de los ataques a fallos depende de la capacidad de inducir fallos dentro del hardware. Los métodos propuestos utilizan haces ópticos para inducir fallas dentro de un circuito. Sin embargo, para la experimentación se utilizaron fallas en el reloj como medio para inducir fallas internas. La ventaja de este método es el menor coste de la técnica de inyección de fallos. De este modo, el atacante sólo manipula la señal del reloj.

La idea básica es utilizar un multiplexor para cambiar del reloj de baja frecuencia a un reloj de mayor frecuencia en el comienzo de la octava ronda. Los relojes se generan a partir de un generador de funciones/formas de onda arbitrarias *Agilent 33250A* de 80 Mhz. Por lo tanto, sólo la octava ronda se ejecutará con el reloj más rápido. La frecuencia del reloj más rápido debe ajustarse cuidadosamente desde un generador de señales. Debe ser ligeramente superior a la frecuencia máxima que admite el diseño AES. Nuestros experimentos muestran que existe una relación entre el número de fallas inducidas y la frecuencia del reloj más rápido. Se verificó utilizando el analizador lógico integrado *Xilinx ChipScope Pro* que el número de diagonales defectuosas es directamente proporcional a la frecuencia del reloj.



Referencias

- Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., & Tokita, T. (2001). *Camellia: a 128-Bit block cipher suitable for multiple platforms - Design andAnalysis. En Lecture Notes in Computer Science* (pp. 39-56) https://doi.org/10.1007/3-540-44983-3_4

- Bonnetain, X., Naya-Plasencia, M., & Schrottenloher, A. (2019). *Quantum Security Analysis of AES*. HAL (Le Centre pour la Communication Scientifique Directe). https://doi.org/10.13154/tosc.v2019.i2.55-93

- *Introduction to Camellia - NTT Social Informatics Laboratories. (s.f.). NTT Cryptographic Primitive.* https://info.isl.ntt.co.jp/crypt/eng/camellia/intro.html

- *Voce. (2021, 16 septiembre). Camellia. VOCAL Technologies.* https://vocal.com/cryptography/camellia/