Side Channel Attack on GIFT COFB

Attack on Lightweight Cryptography

CS6630 Secure Processor Microarchitecture Project

Arun Krishna AMS (EE19B001)

Yogasanthoshi (CS19B049)

Background

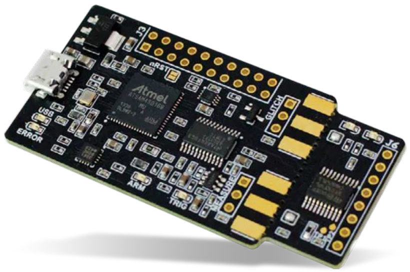
- GIFT-COFB is an Authenticated Encryption with Associated Data (AEAD) scheme, based on
  - GIFT lightweight block cipher and,
  - COFB lightweight AEAD operating mode.

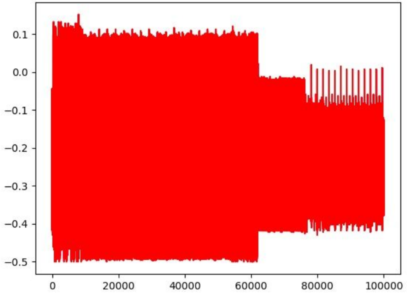
- Authenticated Encryption with Associated Data (AEAD) scheme:
  - Provides both confidentiality and authenticity.

- Power Side Channel Attacks:
  - Recover secret information from hardware by processing the power consumption of the device.

Our work & Overall results

- Power traces from ChipWhisperer Nano running software implementation of GIFT-COFB collected
- Performed two different implementations of CPA and DOM attacks
- Observed both fail to obtain secret key





COFB Encryption

- The encryption algorithm takes as input
  - An encryption key $K \in \{0, 1\}^{128}$
  - A nonce $N \in \{0, 1\}^{128}$ (non-repeating)
  - Associated data and message $A, M \in \{0, 1\}^*$
- Generates the following output:
  - Ciphertext $C \in \{0, 1\}^{|M|}$
  - Tag $T \in \{0, 1\}^{128}$
- Underlying block cipher: GIFT 128 block cipher

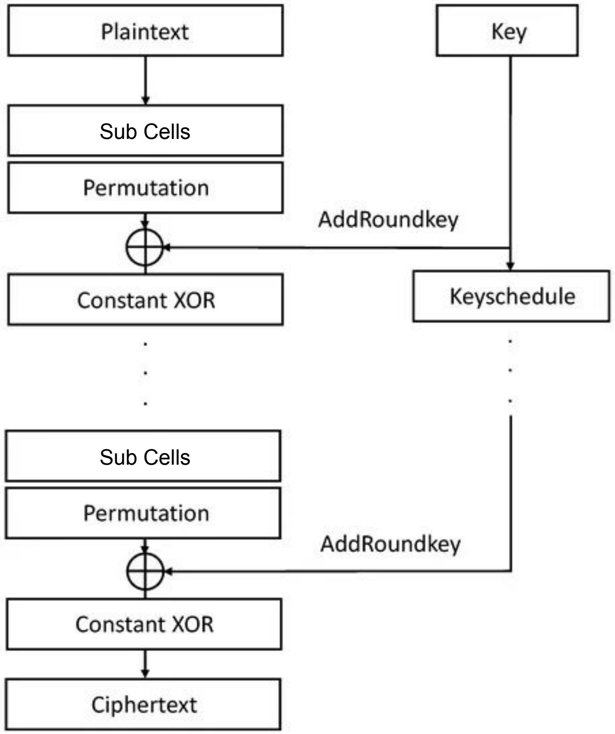
COFB Decryption with Verification

- Decryption of a ciphertext-tag pair (C, T)
  - An encryption key $K \in \{0, 1\}^{128}$
    - Symmetric Key algorithm
  - A nonce $N \in \{0, 1\}^{128}$
  - Associated data and ciphertext $A, C \in \{0, 1\}^*$
  - Tag $T \in \{0, 1\}^{128}$

- It generates the following output:
  - Message $M \in \{0, 1\}^{|M|} \cup \{\perp\}$

Block Cipher GIFT-128

- GIFT-128 is an 128-bit Substitution-Permutation network (SPN) based block cipher with a key length of 128-bit.
- 40 Identical rounds. Each round consists of
  - SubCells Operation,
  - Permutation of Bits,
  - Add Round Key
  - Key-Schedule and Add-Round constant



State Representation

$$\textbf{PlainText :} \begin{pmatrix} B_0 & B_1 & B_2 & B_3 \\ B_4 & B_5 & B_6 & B_7 \\ B_8 & B_9 & B_{10} & B_{11} \\ B_{12} & B_{13} & B_{14} & B_{15} \end{pmatrix} \longrightarrow \boxed{\text{Encryption}} \longrightarrow \textbf{CipherText :} \begin{pmatrix} C_0 & C_1 & C_2 & C_3 \\ C_4 & C_5 & C_6 & C_7 \\ C_8 & C_9 & C_{10} & C_{11} \\ C_{12} & C_{13} & C_{14} & C_{15} \end{pmatrix}$$

$$S = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} \leftarrow \begin{bmatrix} B_0 & \| & B_1 & \| & B_2 & \| & B_3 \\ B_4 & \| & B_5 & \| & B_6 & \| & B_7 \\ B_8 & \| & B_9 & \| & B_{10} & \| & B_{11} \\ B_{12} & \| & B_{13} & \| & B_{14} & \| & B_{15} \end{bmatrix}$$

$$KS = \begin{bmatrix} W_0 & \| & W_1 \\ W_2 & \| & W_3 \\ W_4 & \| & W_5 \\ W_6 & \| & W_7 \end{bmatrix} \leftarrow \begin{bmatrix} B_0 \| B_1 & \| & B_2 \| B_3 \\ B_4 \| B_5 & \| & B_6 \| B_7 \\ B_8 \| B_9 & \| & B_{10} \| B_{11} \\ B_{12} \| B_{13} & \| & B_{14} \| B_{15} \end{bmatrix}$$

Subcell Operation

$$
\begin{aligned}
S_1 &\leftarrow S_1 \oplus (S_0 \mathbin{\&} S_2) \\
S_0 &\leftarrow S_0 \oplus (S_1 \mathbin{\&} S_3) \\
S_2 &\leftarrow S_2 \oplus (S_0 \mid S_1) \\
S_3 &\leftarrow S_3 \oplus S_2 \\
S_1 &\leftarrow S_1 \oplus S_3 \\
S_3 &\leftarrow \sim S_3 \\
S_2 &\leftarrow S_2 \oplus (S_0 \mathbin{\&} S_1) \\
\{S_0, S_1, S_2, S_3\} &\leftarrow \{S_3, S_1, S_2, S_0\},
\end{aligned}
$$

Bit Permutation Operation

Index	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
S_0	29	25	21	17	13	9	5	1	30	26	22	18	14	10	6	2
S_1	30	26	22	18	14	10	6	2	31	27	23	19	15	11	7	3
S_2	31	27	23	19	15	11	7	3	28	24	20	16	12	8	4	0
S_3	28	24	20	16	12	8	4	0	29	25	21	17	13	9	5	1

Index	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S_0	31	27	23	19	15	11	7	3	28	24	20	16	12	8	4	0
S_1	28	24	20	16	12	8	4	0	29	25	21	17	13	9	5	1
S_2	29	25	21	17	13	9	5	1	30	26	22	18	14	10	6	2
S_3	30	26	22	18	14	10	6	2	31	27	23	19	15	11	7	3

$$b_{P(i)} \leftarrow b_i, \; i \in \{0, \ldots, n-1\}$$

Add Round Key & Constant Operation

Add Round Key:

$$U \leftarrow W_2 \| W_3, \; V \leftarrow W_6 \| W_7.$$

$$RK = U \| V.$$

$$S_2 \leftarrow S_2 \oplus U,$$
$$S_1 \leftarrow S_1 \oplus V.$$

$$S = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} \leftarrow \begin{bmatrix} B_0 & \| & B_1 & \| & B_2 & \| & B_3 \\ B_4 & \| & B_5 & \| & B_6 & \| & B_7 \\ B_8 & \| & B_9 & \| & B_{10} & \| & B_{11} \\ B_{12} & \| & B_{13} & \| & B_{14} & \| & B_{15} \end{bmatrix}$$

$$KS = \begin{bmatrix} W_0 & \| & W_1 \\ W_2 & \| & W_3 \\ W_4 & \| & W_5 \\ W_6 & \| & W_7 \end{bmatrix} \leftarrow \begin{bmatrix} B_0 \| B_1 & \| & B_2 \| B_3 \\ B_4 \| B_5 & \| & B_6 \| B_7 \\ B_8 \| B_9 & \| & B_{10} \| B_{11} \\ B_{12} \| B_{13} & \| & B_{14} \| B_{15} \end{bmatrix}$$

Add Round Constant:

$$S_3 \leftarrow S_3 \oplus \texttt{0x800000XY},$$

$$\text{byte } \texttt{XY} = 00c_5c_4c_3c_2c_1c_0$$

COFB Authenticated Encryption Mode

COFB has the following building blocks:

- Key and Block cipher: E_K
- Padding Function $\mathsf{Pad}(x)$
- Feedback Function G

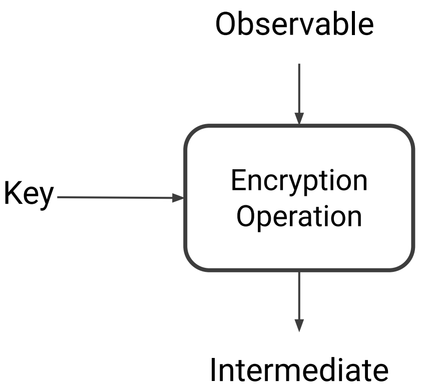
Algorithm $\mathsf{COFB}\text{-}\mathcal{E}_K(N, A, M)$

$Y[0] \leftarrow E_K(N),\ L \leftarrow \mathsf{Trunc}_{n/2}(Y[0])$
$(A[1], \ldots, A[a]) \xleftarrow{n} \mathsf{Pad}(A)$
if $M \neq \epsilon$ **then**
  $(M[1], \ldots, M[m]) \xleftarrow{n} \mathsf{Pad}(M)$
for $i = 1$ **to** $a - 1$
  $L \leftarrow 2 \cdot L$
  $X[i] \leftarrow A[i] \oplus G \cdot Y[i-1] \oplus L \| 0^{n/2}$
  $Y[i] \leftarrow E_K(X[i])$
if $|A| \bmod n = 0$ **and** $A \neq \epsilon$ **then** $L \leftarrow 3 \cdot L$
else $L \leftarrow 3^2 \cdot L$
if $M = \epsilon$ **then** $L \leftarrow 3^2 \cdot L$
$X[a] \leftarrow A[a] \oplus G \cdot Y[a-1] \oplus L \| 0^{n/2}$
$Y[a] \leftarrow E_K(X[a])$
for $i = 1$ **to** $m - 1$
  $L \leftarrow 2 \cdot L$
  $C[i] \leftarrow M[i] \oplus Y[i+a-1]$
  $X[i+a] \leftarrow M[i] \oplus G \cdot Y[i+a-1] \oplus L \| 0^{n/2}$
  $Y[i+a] \leftarrow E_K(X[i+a])$
if $M \neq \epsilon$ **then**
  if $|M| \bmod n = 0$ **then** $L \leftarrow 3 \cdot L$
  else $L \leftarrow 3^2 \cdot L$
  $C[m] \leftarrow M[m] \oplus Y[a+m-1]$
  $X[a+m] \leftarrow M[m] \oplus G \cdot Y[a+m-1] \oplus L \| 0^{n/2}$
  $Y[a+m] \leftarrow E_K(X[a+m])$
  $C \leftarrow \mathsf{Trunc}_{|M|}(C[1] \| \ldots \| C[m])$
  $T \leftarrow \mathsf{Trunc}_{\tau}(Y[a+m])$
else $C \leftarrow \epsilon,\ T \leftarrow \mathsf{Trunc}_{\tau}(Y[a])$
return (C, T)

Attack Design

Side Channel attacks require:

- Observable
- Intermediate

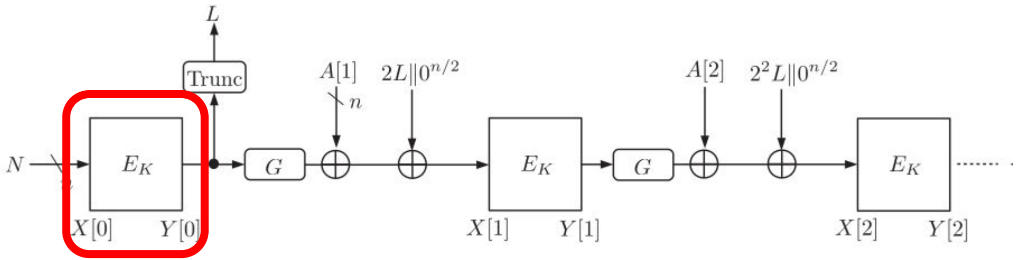


The nonce is encrypted to yield the first internal chaining value

Encryption on Nonce is the first block encryption component and is our point of attack

Algorithm COFB-$\mathcal{E}_K(N, A, M)$

1. $\boxed{Y[0] \leftarrow E_K(N),}$ $L \leftarrow \mathsf{Trunc}_{n/2}(Y[0])$
2. $(A[1], \ldots, A[a]) \xleftarrow{n} \mathsf{Pad}(A)$
3. **if** $M \neq \epsilon$ **then**
4. $\quad (M[1], \ldots, M[m]) \xleftarrow{n} \mathsf{Pad}(M)$
5. **for** $i = 1$ **to** $a - 1$
6. $\quad L \leftarrow 2 \cdot L$



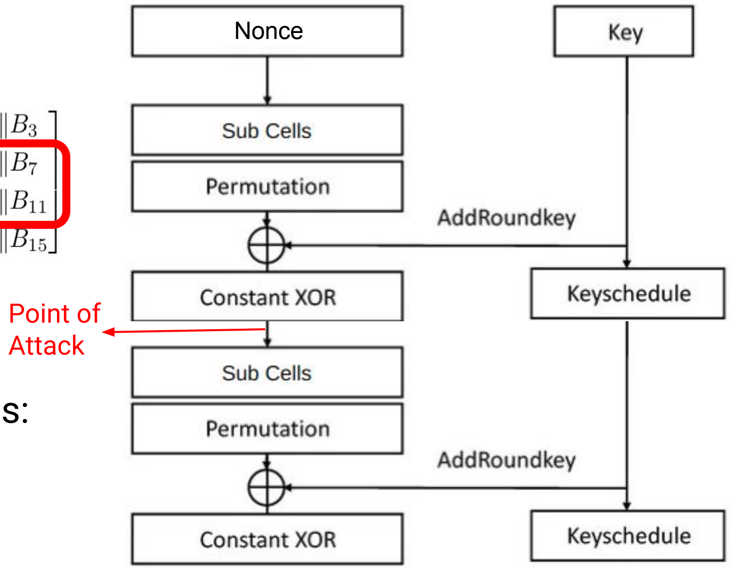
Attack Design: 1

Keys retrieved:

$$KS = \begin{bmatrix} W_0 & \| & W_1 \\ W_2 & \| & W_3 \\ W_4 & \| & W_5 \\ W_6 & \| & W_7 \end{bmatrix} \leftarrow \begin{bmatrix} B_0\|B_1 & \| & B_2\|B_3 \\ B_4\|B_5 & \| & B_6\|B_7 \\ B_8\|B_9 & \| & B_{10}\|B_{11} \\ B_{12}\|B_{13} & \| & B_{14}\|B_{15} \end{bmatrix}$$

Number of possible guesses:

- 256 per byte of key
- 2048 for all 8 bytes

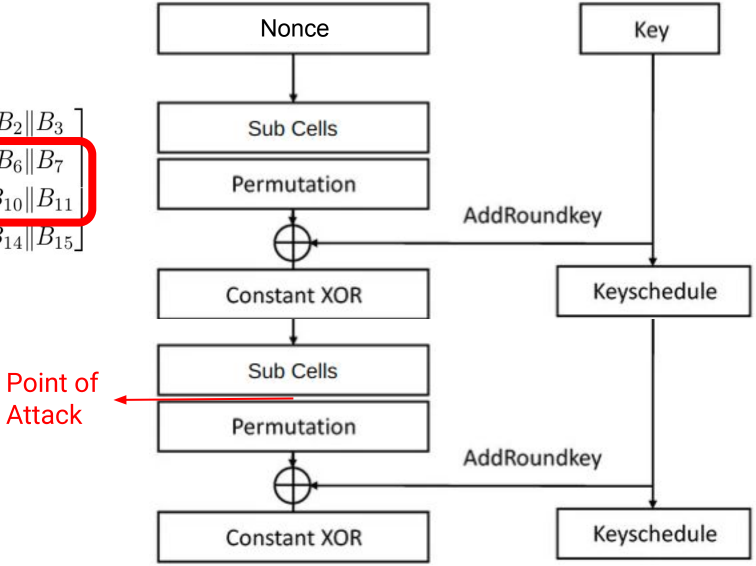



```
S[2] ^= S[0] & S[1]

T = S[0]
S[0] = S[3]
S[3] = T

# ===PermBits=== #
S[0] = rowperm(S[0],0,3,2,1)
S[1] = rowperm(S[1],1,0,3,2)
S[2] = rowperm(S[2],2,1,0,3)
S[3] = rowperm(S[3],3,2,1,0)

# ===AddRoundKey=== #

#keyindex = 4: gives W2[0]
if key_index == 4:
    intermediate = ((S[2] >>  0) & 0b11111111) ^ keyguess
#keyindex = 5: gives W2[1]
elif key_index == 5:
    intermediate = ((S[2] >>  8) & 0b11111111) ^ keyguess
#keyindex = 6: gives W3[0]
elif key_index == 6:
    intermediate = ((S[2] >>  16) & 0b11111111) ^ keyguess
#keyindex = 7: gives W3[1]
elif key_index == 7:
    intermediate = ((S[2] >>  24) & 0b11111111) ^ keyguess
#keyindex = 12: gives W6[0]
elif key_index == 12:
    intermediate = ((S[1] >>  0) & 0b11111111) ^ keyguess
#keyindex = 13: gives W6[1]
elif key_index == 13:
    intermediate = ((S[1] >>  8) & 0b11111111) ^ keyguess
#keyindex = 14: gives W7[0]
elif key_index == 14:
    intermediate = ((S[1] >>  16) & 0b11111111) ^ keyguess
#keyindex = 15: gives W7[1]
elif key_index == 15:
    intermediate = ((S[1] >>  24) & 0b11111111) ^ keyguess

return intermediate
```

Attack Design: 2

Keys retrieved:

$$KS = \begin{bmatrix} W_0 & \| & W_1 \\ W_2 & \| & W_3 \\ W_4 & \| & W_5 \\ W_6 & \| & W_7 \end{bmatrix} \leftarrow \begin{bmatrix} B_0\|B_1 & \| & B_2\|B_3 \\ B_4\|B_5 & \| & B_6\|B_7 \\ B_8\|B_9 & \| & B_{10}\|B_{11} \\ B_{12}\|B_{13} & \| & B_{14}\|B_{15} \end{bmatrix}$$




```
S[3] ^= 0xffffffff
S[2] ^= S[0] & S[1]

T = S[0]
S[0] = S[3]
S[3] = T

# ===PermBits=== #
S[0] = rowperm(S[0],0,3,2,1)
S[1] = rowperm(S[1],1,0,3,2)
S[2] = rowperm(S[2],2,1,0,3)
S[3] = rowperm(S[3],3,2,1,0)


#Add Round Constant operation
S[3] ^= 0x80000000 ^ GIFT_RC[1]

#Add Round Key Operation - For the byte from S1 and S2
S1dash = ((S[1]>>(key_index*8)) & 0xff) ^ keyguess1
S2dash = ((S[2]>>(key_index*8)) & 0xff) ^ keyguess2

#Choosing only the byte from S0 and S3 that is being a
S0dash = (S[0]>>(key_index*8)) & 0xff
S3dash = (S[3]>>(key_index*8)) & 0xff

#Part of Subcell Operation to increase nonlinearity
S1dashdash = S1dash ^ (S0dash & S2dash)
S0dashdash = S0dash ^ (S1dashdash & S3dash)

#Returen Immediate Value
intermediate = S0dashdash
return intermediate
```

Attack Design: 2

Nonce $\rightarrow$

SubCell + Permbits Round 1

$$(S'_0, S'_1, S'_2, S'_3)$$

$\rightarrow$

Add Round Key & constant operation Round 1

$$S''_0 \leftarrow S'_0$$
$$S''_1 \leftarrow S'_1 \oplus W_6||W_7,$$
$$S''_2 \leftarrow S'_2 \oplus W_2||W_3,$$
$$S''_3 \leftarrow S'_3 \oplus 0\text{x}80000001$$
$$(S''_0, S''_1, S''_2, S''_3)$$

$\rightarrow$

SubCell Round 2

$$S_1 \leftarrow S''_1 \oplus \left(S''_0 \& S''_2\right)$$
$$S_0 \leftarrow S''_0 \oplus \left(S_1 \& S''_3\right)$$

Number of possible guesses:
256*256 = 65536 for two byte of key
4*256*256 = 262144 for all 8 bytes

Work Split up

Arun Krishna: Hardware setup, Capturing Power Trace, CPA design

Yogasanthoshi: DOM, CPA Design

Drive Link (containing codes, presentation video): https://drive.google.com/drive/folders/1vmNPdr0VXJ3OygLR3y7RHXDDJ5G2sD2X?usp=sharing

Thank You !!