Leakage-Resilient Symmetric Key Cryptography

Sougata Mandal

Institute for Advancing Intelligence, TCG CREST
&
Ramakrishna Mission Vivekananda Educational and Research Institute

About Me

- B.Sc in Mathematics from University of Calcutta.
- M.Sc in Pure Mathematics from University of Calcutta.
- M.Tech in Cryptology and Security from ISI Kolkata.
- JRF in Computer Science from RKMVERI and TCG CREST.
- Thesis Topic: Leakage-Resilient Symmetric Key Cryptography.

Courses Taken

Course Name	Marks Obtained
Graph Theory and Matroids	63
Advanced Symmetric Key Cryptology	65
Stocastic Process	71
Algebra and Its Application	78
Research Methodology	86
Advanced Quantum Information and Cryptology	90
Advanced Cryptology	83
Design and Analysis of Algorithms	92

Books Read

[1] Oded Goldreich: Foundation of Cryptography- A Primer

[2] Jonathan Katz, Yehuda Lindell: Introduction to Modern Cryptography

[3] Dan Boneh, Victor Shoup: A Graduate Course in Applied Cryptography

Papers Read

[1] Olivier Pereira, François-Xavier Standaert, Srinivas Vivek: Leakage-Resilient Authentication and Encryption from Symmetric Cryptographic Primitives. CCS 2015: 96-108.

[2] Francesco Berti, François Koeune, Olivier Pereira, Thomas Peters, François-Xavier Standaert: Ciphertext Integrity with Misuse and Leakage: Definition and Efficient Constructions with Symmetric Primitives. AsiaCCS 2018: 37-50

[3] Francesco Berti, Olivier Pereira, Thomas Peters, François-Xavier Standaert: On Leakage-Resilient Authenticated Encryption with Decryption Leakages. IACR Trans. Symmetric Cryptol. 2017(3): 271-293 (2017).

[4] Francesco Berti, Chun Guo, Olivier Pereira, Thomas Peters, François-Xavier Standaert: TEDT, a Leakage-Resist AEAD Mode for High Physical Security Applications. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2020(1): 256-320 (2020).

[5] Eik List: TEDT2 - Highly Secure Leakage-Resilient TBC-Based Authenticated Encryption. LATINCRYPT 2021: 275-295.

Papers Read

[6] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, David J. Wu: Exploring Crypto Dark Matter: - New Simple PRF Candidates and Their Applications. TCC 2018: 699-729.

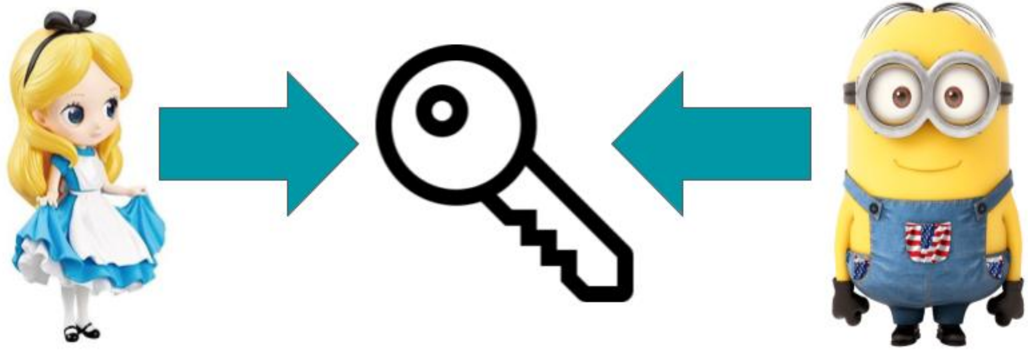
[7] Markus Grassl, Brandon Langenberg, Martin Roetteler, Rainer Steinwandt: Applying Grover's Algorithm to AES: Quantum Resource Estimates. PQCrypto 2016: 29-43.

[8] Akinori Hosoyamada, Tetsu Iwata: 4-Round Luby-Rackoff Construction is a qPRP. ASIACRYPT 2019: 145-174.

[9] Orr Dunkelman, Nathan Keller, Eyal Ronen, Adi Shamir: Quantum Time/Memory/Data Tradeoff Attacks. IACR Cryptol. ePrint Arch. 2021: 1561 (2021) 2020.

[10] Gregor Leander, Alexander May: Grover Meets Simon - Quantumly Attacking the FX-construction. ASIACRYPT 2017: 161-178.

Symmetric Key Cryptography

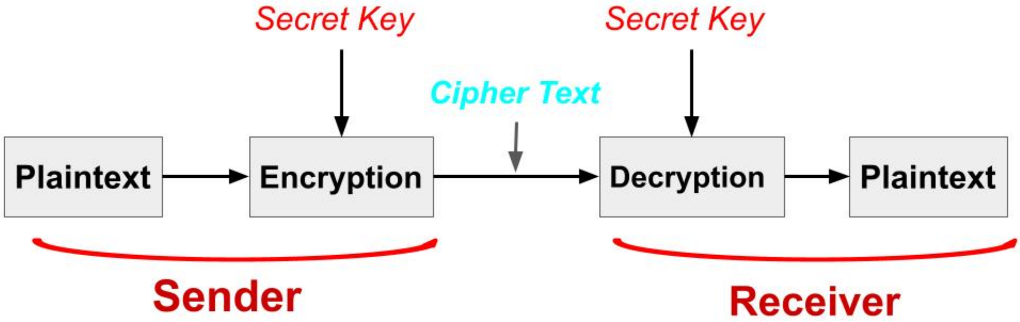


Kerckhoffs's principle

The principle says that a cryptosystem should be secure, even if everything about the system, except the key, is public knowledge.

Data Confidentiality

Encryption-Decryption



Encryption

m_i

$c_{b,i}$, (s.t. $c_{0,i} \leftarrow E(k, m_i)$ and $c_{1,i} \xleftarrow{\$} \mathcal{C}$)

$\widehat{b} \in \{0,1\}$

$b \xleftarrow{\$} \{0,1\}$

Win if $b = \widehat{b}$

Data Integrity

Message Authentication Code (MAC)

- Used for message **integrity**.

Message Authentication Code (MAC)

- Used for message **integrity**.
- $MAC = (S, V)$.

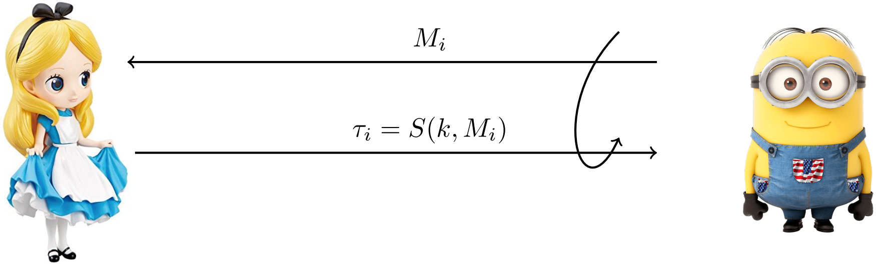
Message Authentication Code (MAC)

- Used for message **integrity**.
- $MAC = (S, V)$.
- **Signing** algorithm: $S(k, m) = t$, for sender.

Message Authentication Code (MAC)

- Used for message **integrity**.
- $MAC = (S, V)$.
- **Signing** algorithm: $S(k, m) = t$, for sender.
- **Verification** algorithm: $V(k, m, t) = accept/reject$, for receiver.

Secure MAC



Secure MAC

M_i

$\tau_i = S(k, M_i)$

(M, τ)

Win if (M, τ) is fresh and valid

Basic Cryptographic Primitives

Pseudo Random Function (PRF)

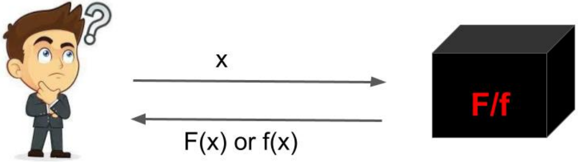
- $F : \mathcal{M} \times \mathcal{K} \to \mathcal{C}$, where $\mathcal{M} := \{0,1\}^m$, $\mathcal{K} := \{0,1\}^k$ and $\mathcal{C} := \{0,1\}^n$

Pseudo Random Function (PRF)

- $F : \mathcal{M} \times \mathcal{K} \to \mathcal{C}$, where $\mathcal{M} := \{0,1\}^m$, $\mathcal{K} := \{0,1\}^k$ and $\mathcal{C} := \{0,1\}^n$
- $\mathsf{f} \xleftarrow{\$} \mathsf{Func}[\mathcal{M}, \mathcal{C}]$, where $\mathsf{Func}[\mathcal{M}, \mathcal{C}]$ is the set of all functions $\mathsf{g} : \mathcal{M} \to \mathcal{C}$

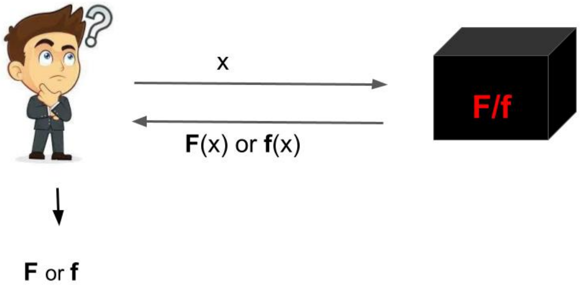
Pseudo Random Function (PRF)

- $F : \mathcal{M} \times \mathcal{K} \to \mathcal{C}$, where $\mathcal{M} := \{0,1\}^m$, $\mathcal{K} := \{0,1\}^k$ and $\mathcal{C} := \{0,1\}^n$
- $\mathsf{f} \xleftarrow{\$} \mathsf{Func}[\mathcal{M}, \mathcal{C}]$, where $\mathsf{Func}[\mathcal{M}, \mathcal{C}]$ is the set of all functions $\mathsf{g} : \mathcal{M} \to \mathcal{C}$



Pseudo Random Function(PRF)

- $F : \mathcal{M} \times \mathcal{K} \to \mathcal{C}$, where $\mathcal{M} := \{0,1\}^m$, $\mathcal{K} := \{0,1\}^k$ and $\mathcal{C} := \{0,1\}^n$
- $f \xleftarrow{\$} \mathsf{Func}[\mathcal{M},\mathcal{C}]$, where $\mathsf{Func}[\mathcal{M},\mathcal{C}]$ is the set of all functions $g : \mathcal{M} \to \mathcal{C}$

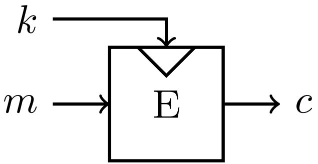


Block Cipher

- $E : \mathcal{M} \times \mathcal{K} \to \mathcal{C}$, where $\mathcal{M} := \{0,1\}^n$, $\mathcal{K} := \{0,1\}^k$ and $\mathcal{C} := \{0,1\}^n$

Block Cipher

- $E : \mathcal{M} \times \mathcal{K} \to \mathcal{C}$, where $\mathcal{M} := \{0,1\}^n$, $\mathcal{K} := \{0,1\}^k$ and $\mathcal{C} := \{0,1\}^n$

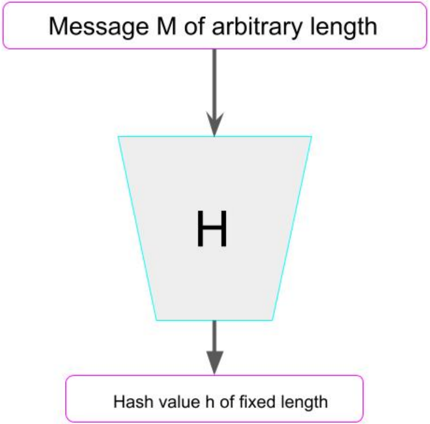


Hash Function

- $H : \{0,1\}^* \rightarrow \{0,1\}^n$.

Hash Function

- $H : \{0,1\}^* \to \{0,1\}^n$.



Security of Hash

- **Collision resistant:** Hard to find x_1 and x_2 such that $H(x_1) = H(x_2)$.

Security of Hash

- **Collision resistant:** Hard to find x_1 and x_2 such that $H(x_1) = H(x_2)$.
- **Second pre-image resistant:** Given $y = H(x)$, for any $x \xleftarrow{\$} \{0,1\}^*$, hard to find z such that $H(z) = y$.

Security of Hash

- **Collision resistant:** Hard to find x_1 and x_2 such that $H(x_1) = H(x_2)$.
- **Second pre-image resistant:** Given $y = H(x)$, for any $x \xleftarrow{\$} \{0,1\}^*$, hard to find z such that $H(z) = y$.
- **Range-oriented pre-image resistant:** Given $y \xleftarrow{\$} \{0,1\}^n$, hard to find x such that $H(x) = y$.

Theoretical Secure Primitives

Physical Attacks

Physical Attack: due to Power-Leakage

- Calculate $(m^k \mod n)$, for any message m and key k.

Physical Attack: due to Power-Leakage

- Calculate ($m^k \mod n$), for any message m and key k.
- Use Square and Multiply.

Physical Attack: due to Power-Leakage

- Calculate $(m^k \mod n)$, for any message m and key k.
- Use Square and Multiply.
- Number of multiplications = number of 1's.

Physical Attack: due to Power-Leakage

- Calculate ($m^k \mod n$), for any message m and key k.
- Use Square and Multiply.
- Number of multiplications = number of 1's.
- Multiplication takes more time and power than squaring.

Physical Attack: due to Power-Leakage

- Calculate ($m^k \mod n$), for any message m and key k.
- Use Square and Multiply.
- Number of multiplications = number of 1's.
- Multiplication takes more time and power than squaring.
- Observing power and time pattern guess the key.

Physical Attacks

- Power analysis:

Physical Attacks

- Power analysis:
  - Simple Power Analysis (SPA)
  - Differential Power Analysis (DPA)

Physical Attacks

- Power analysis:
    - Simple Power Analysis (SPA)
    - Differential Power Analysis (DPA)
- Timing attack.

Physical Attacks

- Power analysis:
  - Simple Power Analysis (SPA)
  - Differential Power Analysis (DPA)
- Timing attack.
- Deep-learning-based side-channel attack.

Physical Attacks

- Power analysis:
  - Simple Power Analysis (SPA)
  - Differential Power Analysis (DPA)
- Timing attack.
- Deep-learning-based side-channel attack.
- Optical side-channel attack.

Physical Attacks

- Power analysis:
    - Simple Power Analysis (SPA)
    - Differential Power Analysis (DPA)
- Timing attack.
- Deep-learning-based side-channel attack.
- Optical side-channel attack.
- Cache side-channel attack.

Physical Attacks

- Power analysis:
  - Simple Power Analysis (SPA)
  - Differential Power Analysis (DPA)
- Timing attack.
- Deep-learning-based side-channel attack.
- Optical side-channel attack.
- Cache side-channel attack.
- Allocation-based side channels.

Leakage Property

- Information leakage about input.

Leakage Property

- Information leakage about input.
- All internal calculated values.

Leakage Property

- Information leakage about input.
- All internal calculated values.
- Example: One bit of the secret key.

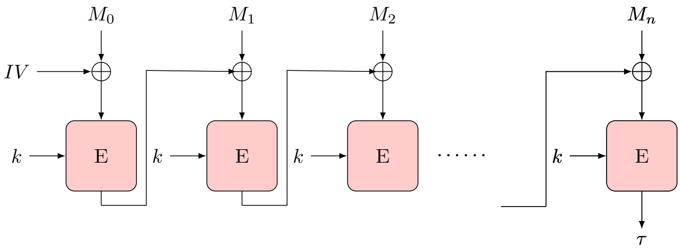
Leakage Property

- Information leakage about input.
- All internal calculated values.
- Example: One bit of the secret key.
- Can we assume bound on leakage?

Leakage Property

- Information leakage about input.
- All internal calculated values.
- Example: One bit of the secret key.
- Can we assume bound on leakage?

CBC-MAC



If E is secure block cipher then CBC-MAC is a secure MAC.

CBC-MAC with Unbounded Leakage

M

$\tau = S(k, M)$

k

CBC-MAC with Unbounded Leakage

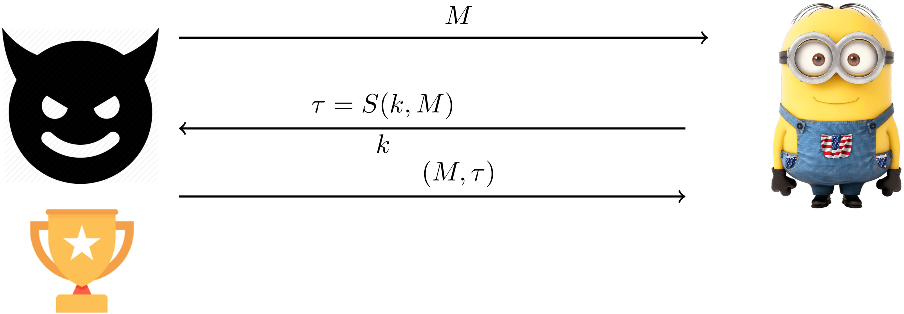
M

$\tau = S(k, M)$

k

(M, τ)

CBC-MAC with Unbounded Leakage



Countermeasure

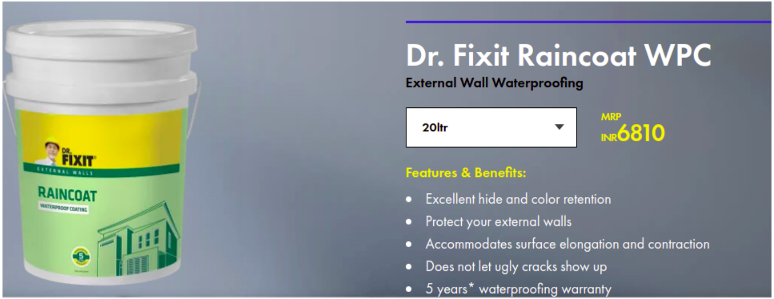
- Leak free implementation.

Countermeasure

- Leak free implementation.
- Leak nothing.

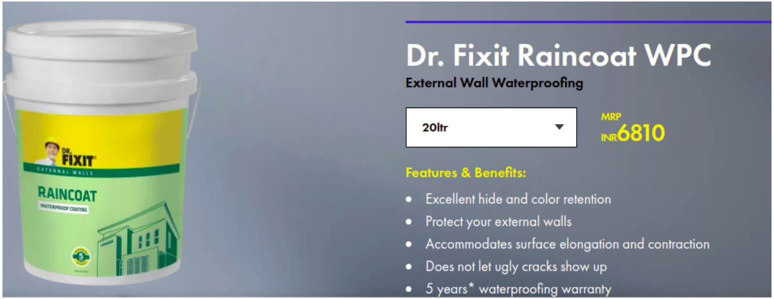
Problem

- Very expensive.



Problem

- Very expensive.



- Minimal use.

Design Rationale

- Problem with CBC-MAC: same key for each message block.

Design Rationale

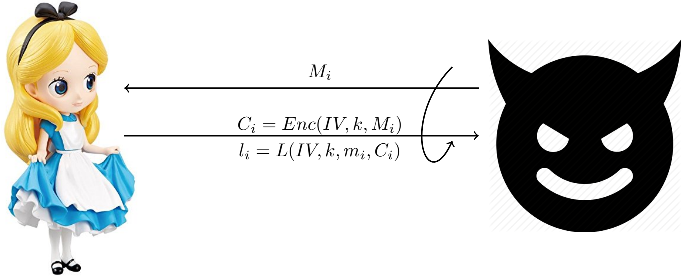
- Problem with CBC-MAC: same key for each message block.

- Different key for each message block.

Design Rationale

- Problem with CBC-MAC: same key for each message block.
- Different key for each message block.
- Minimal use of leak-free component.

L-R Encryption



L-R Encryption

M_i

$C_i = Enc(IV, k, M_i)$

$l_i = L(IV, k, m_i, C_i)$

(IV, C)

L-R Encryption

M_i

$C_i = Enc(IV, k, M_i)$

$l_i = L(IV, k, m_i, C_i)$

(IV, C)

C is fresh and valid

L-R Encryption

$b \xleftarrow{\$} \{0,1\}$

$\xleftarrow{\quad m_i \quad}$

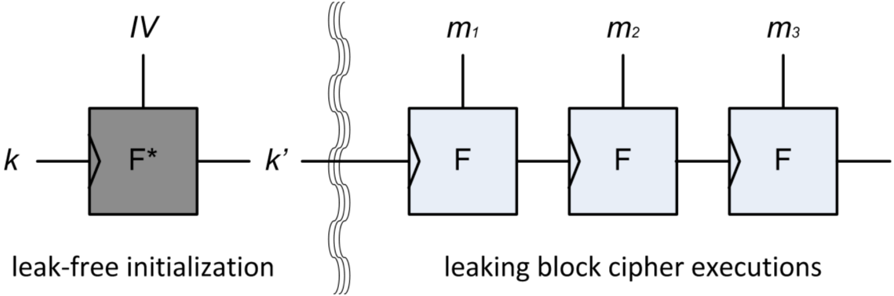
$c_{b,i},\ (\text{s.t. } c_{0,i} \leftarrow E(k, m_i) \text{ and } c_{1,i} \xleftarrow{\$} \mathcal{C}) \longrightarrow$

$\xleftarrow{\quad \widehat{b} \in \{0,1\} \quad}$

Win if $b = \widehat{b}$

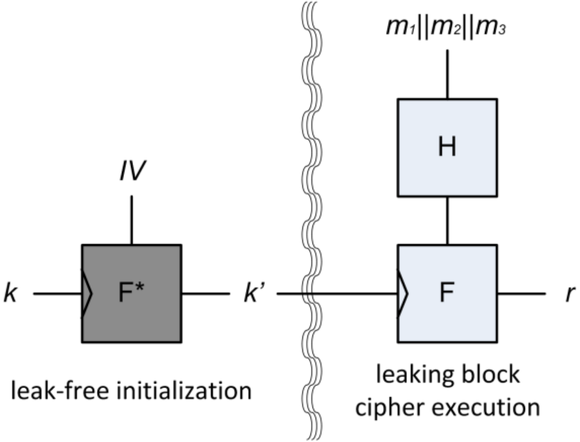
Pereira et al. [CCS'15]

Re-Keying L-R MAC



Pereira et al. [CCS'15]

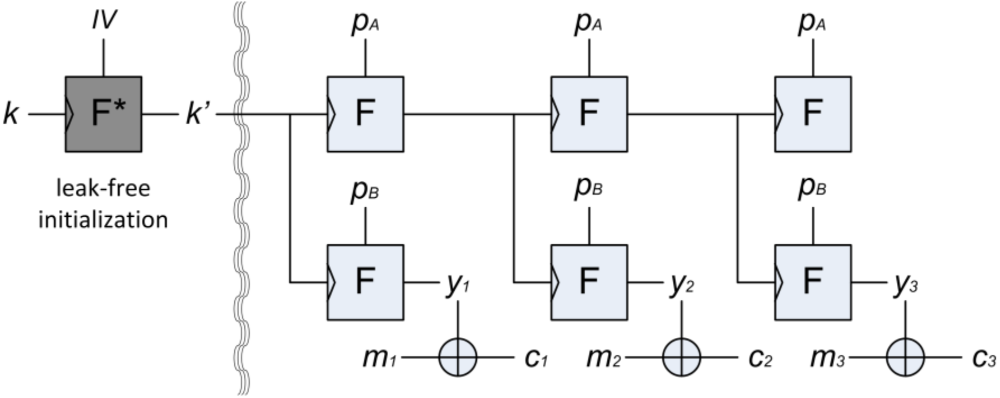
Hash then MAC paradigm



What about confidentiality?

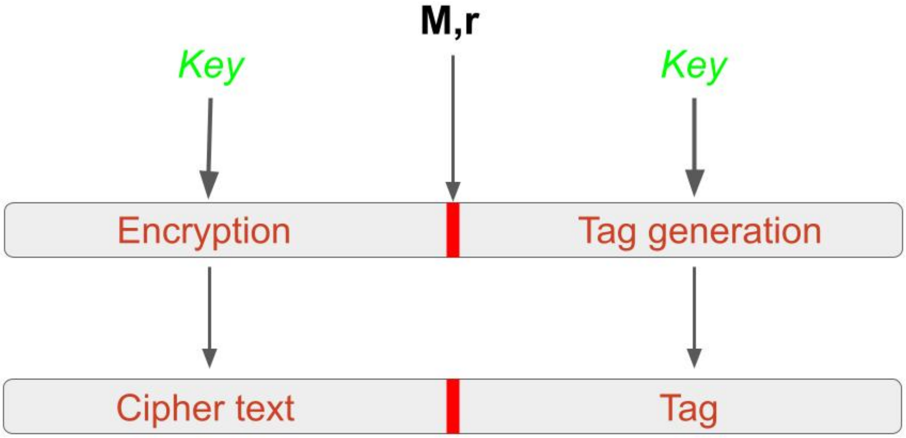
Pereira et al. [CCS'15]

L-R encryption scheme



Can we achieve both **integrity** and **confidentiality**?

Authenticated Encryption



Berti et al. [AsiaCCS'18]

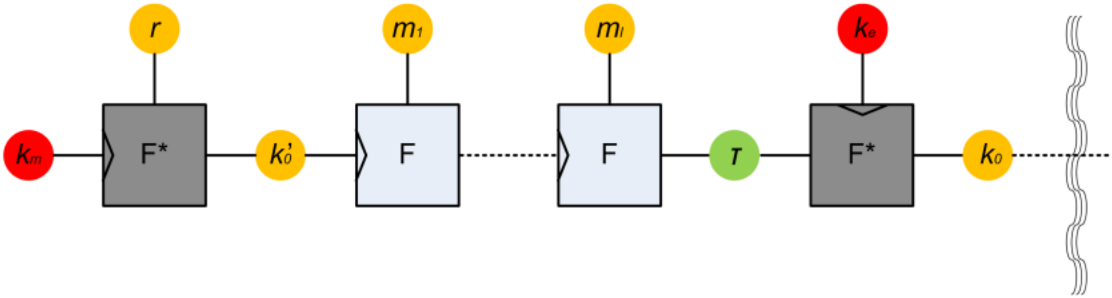
- Coin Misuse-Resistant (M-R) Authenticated Encryption (AE).

Berti et al. [AsiaCCS'18]

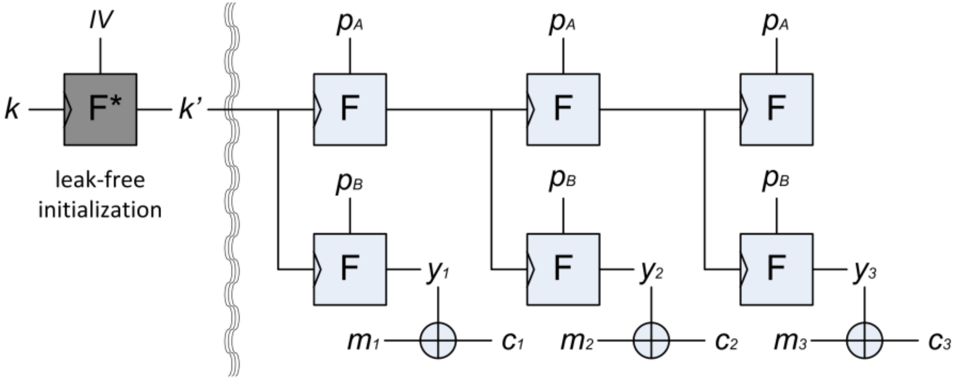
- Coin Misuse-Resistant (M-R) Authenticated Encryption (AE).

- New scheme PSV-AE.

PSV-MAC



PSV-Enc


IV
k
F*
k'
leak-free
initialization
pA
F
pB
F
y1
m1
c1
pA
F
pB
F
y2
m2
c2
pA
F
pB
F
y3
m3
c3

Security and way forward

- PSV-AE is M-R AE.

Security and way forward

- PSV-AE is M-R AE.
- Use two different key.

Security and way forward

- PSV-AE is M-R AE.
- Use two different key.
- Simple Power Analysis (**SPA**) attack on PSV-AE.

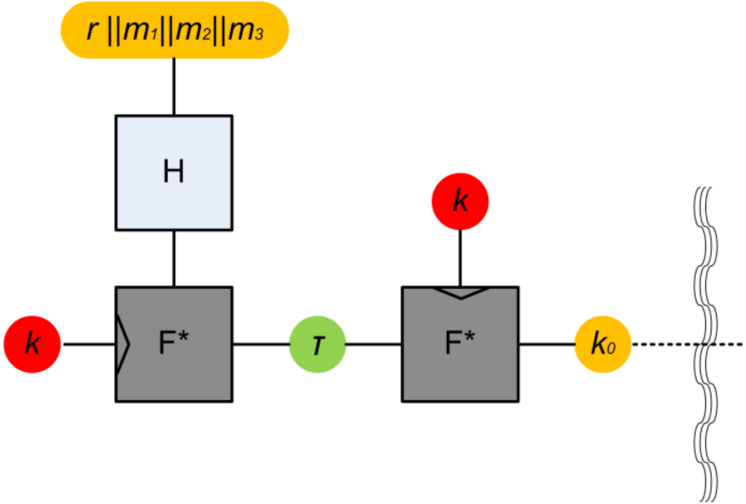
Security and way forward

- PSV-AE is M-R AE.
- Use two different key.
- Simple Power Analysis (**SPA**) attack on PSV-AE.
- Cipher text Integrity with Misuse and Leakage (**CIML**) security notion.

Security and way forward

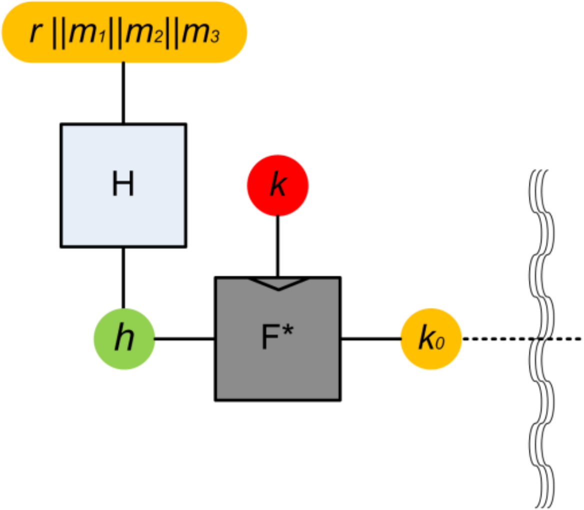
- PSV-AE is M-R AE.
- Use two different key.
- Simple Power Analysis (**SPA**) attack on PSV-AE.
- Cipher text Integrity with Misuse and Leakage (**CIML**) security notion.
- AE scheme DTE with **single key**.

DTE



Can we reduce the number of **leak-free** component?

DCE



Berti et al. [TCS'17]

- Leakage during decryption.

Berti et al. [TCS'17]

- Leakage during decryption.
- CIML2: CIML with **Decryption Leakage** (Confidentiality).

Berti et al. [TCS'17]

- Leakage during decryption.
- CIML2: CIML with **Decryption Leakage** (Confidentiality).
- DTE is **not** CIML2 secure.

Berti et al. [TCS'17]

- Leakage during decryption.
- CIML2: CIML with **Decryption Leakage** (Confidentiality).
- DTE is **not** CIML2 secure.
- DTE2: DTE with leak free **permutation**.

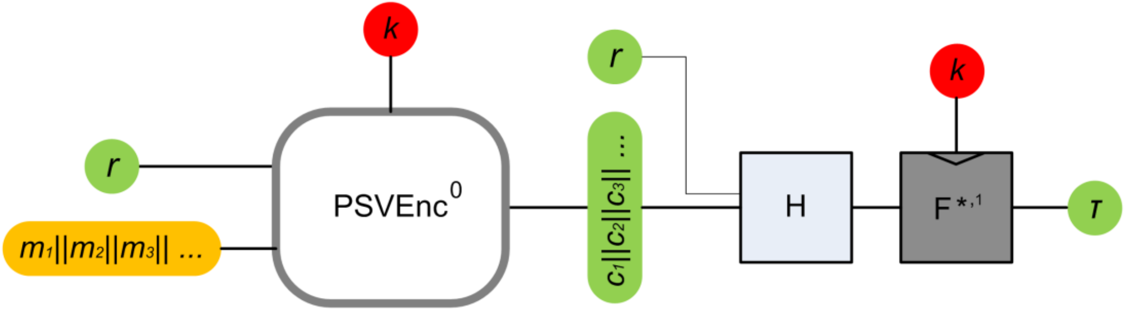
Berti et al. [TCS'17]

- Leakage during decryption.
- CIML2: CIML with **Decryption Leakage** (Confidentiality).
- DTE is **not** CIML2 secure.
- DTE2: DTE with leak free **permutation**.
- EavDL: **Eavesdropper** security with decryption leakage (Indistinguishability).

Berti et al. [TCS'17]

- Leakage during decryption.
- CIML2: CIML with **Decryption Leakage** (Confidentiality).
- DTE is **not** CIML2 secure.
- DTE2: DTE with leak free **permutation**.
- EavDL: **Eavesdropper** security with decryption leakage (Indistinguishability).
- EDT: CIML2 and EavDL secure Authenticated Encryption scheme.

EDT



Berti et al. [TCHES'20]

- EDT can not handle **Associated Data**

Berti et al. [TCHES'20]

- EDT can not handle **Associated Data**

- Security degradation in multi-user settings.

Berti et al. [TCHES'20]

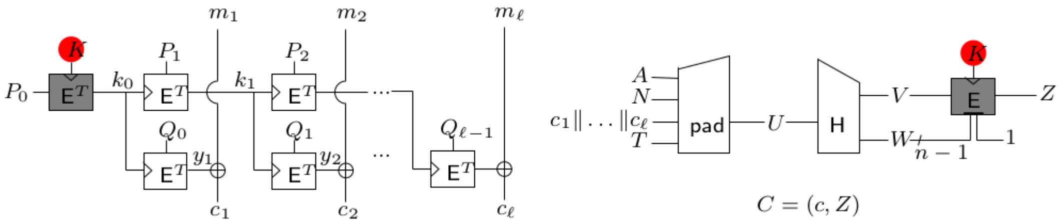
- EDT can not handle **Associated Data**
- Security degradation in multi-user settings.
- muCIML2: CIML2 in multi-user settings.

Berti et al. [TCHES'20]

- EDT can not handle **Associated Data**
- Security degradation in multi-user settings.
- muCIML2: CIML2 in multi-user settings.
- TEDT: muCIML2 secure **A**uthenticated **E**ncryption with **A**ssociated **D**ata(AEAD).

TEDT

$$P_i(N) = N\|[i]_{\frac{n}{4}-1}\|0, \quad Q_i(N) = N\|[i]_{\frac{n}{4}-1}\|1$$



$$C = (c, Z)$$

List [LATINCRYPT'21]

- Replace hash function of TEDT by "Naito's MDPH".

List [LATINCRYPT'21]

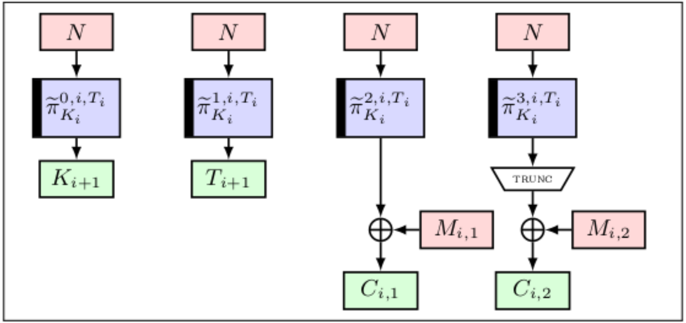
- Replace hash function of TEDT by "Naito's MDPH".

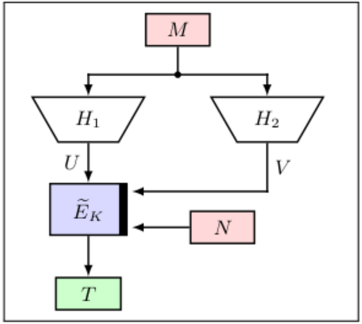
- Nonce used in "Tag generation".

List [LATINCRYPT'21]

- Replace hash function of TEDT by "Naito's MDPH".
- Nonce used in "Tag generation".
- Achieves **B**eyond **B**irthday **B**ound (BBB) security under leakage assumption.

TEDT2


N
N
N
N
$\widetilde{\pi}_{K_i}^{0,i,T_i}$
$\widetilde{\pi}_{K_i}^{1,i,T_i}$
$\widetilde{\pi}_{K_i}^{2,i,T_i}$
$\widetilde{\pi}_{K_i}^{3,i,T_i}$
K_{i+1}
T_{i+1}
TRUNC
$M_{i,1}$
$M_{i,2}$
$C_{i,1}$
$C_{i,2}$


M
H_1
H_2
U
V
$\widetilde{E}_K$
N
T

Future scope

- Analysis of security of various modes under leakage assumption.

Future scope

- Analysis of security of various modes under leakage assumption.
- Finding suitable hash function and other primitives for designing leakage resilient schemes.

Future scope

- Analysis of security of various modes under leakage assumption.
- Finding suitable hash function and other primitives for designing leakage resilient schemes.
- Efficiency in multi-user scenario.

Future scope

- Analysis of security of various modes under leakage assumption.
- Finding suitable hash function and other primitives for designing leakage resilient schemes.
- Efficiency in multi-user scenario.
- Analysis of security under leakage assumption of stateless and stateful schemes.

thank you!