NTT

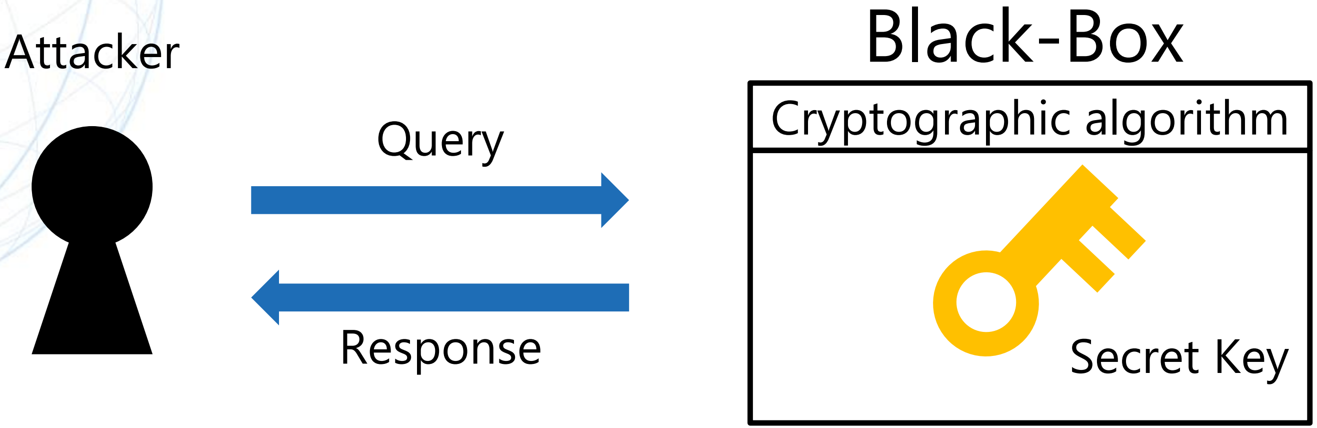
A Modular Approach to the Incompressibility of Block-Cipher-Based AEADs

Akinori Hosoyamada[1] Takanori Isobe[2,3,4]
Yosuke Todo[1] Kan Yasuda[1]

[1] NTT Social Informatics Laboratories
[2] University of Hyogo [3] NICT [4] PRESTO

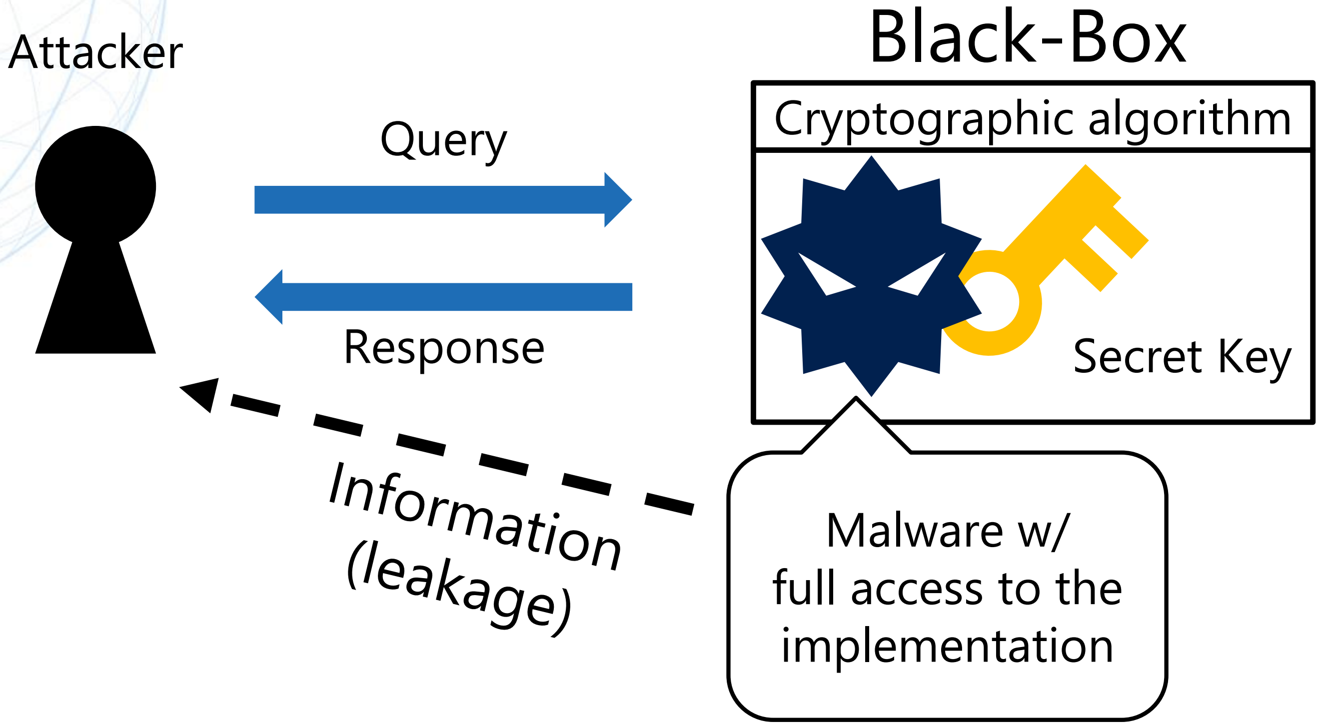
Backgrounds

Black-Box Model



No direct access to the implementation of the algorithm

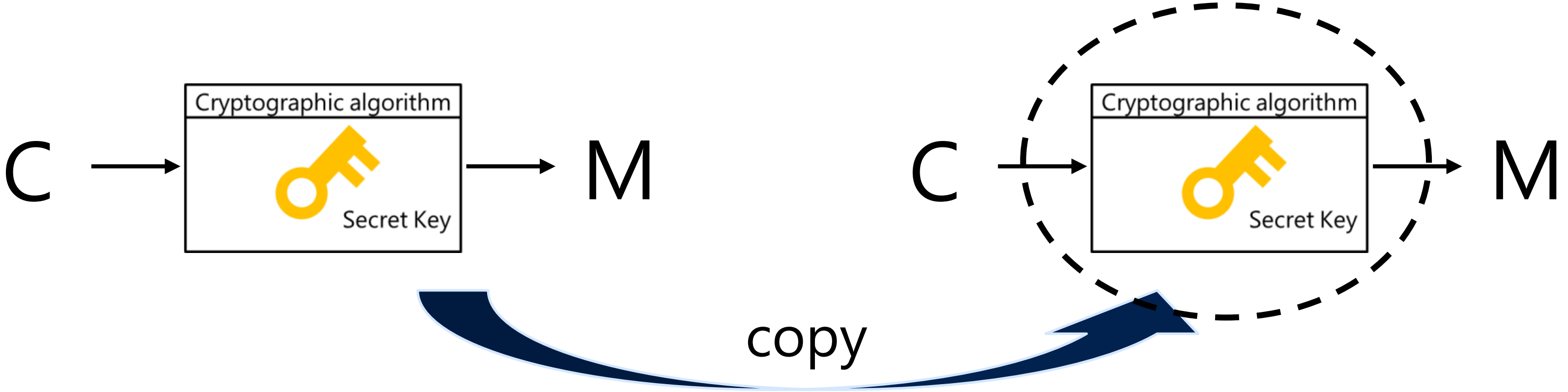
Real-World Threats



White-Box Cryptography [Chow et al. 2002]

- Technique to protect data against attackers who may have **full direct** access to implementations of cryptographic algorithms

- Requirement: Resistance against **key extraction** and **code lifting**
  - Key extraction…an attack to recover the secret key
  - Code lifting…an attack to copy the entire implementation



Incompressibility [Delerablée et al. 2013]

- Security notion against code lifting
- Hardness of compressing cryptographic implementations while keeping functionality
  - An encryption algorithm E_K is compiled to a large (e.g.,10GB) program **P**$[E_K]$
  - **P**$[E_K]$ is *incompressible* if, even if **P**$[E_K]$ is given, it's hard to build a smaller program that is functionally equivalent to **P**$[E_K]$
  - Incompressible ⇒ hard for malwares to leak useful information
  - Many variants exist (ENC-COM [Fouque et al. 2016], SPACE-Hardness [Bogdanov-Isobe 2015], etc.)
- Achievable without relying on special secure hardware
  - There exist high demands for software-only solutions in various scenarios (e.g., cloud-based payment services) [Bogdanov et al. 2016]

Motivation of Research

NTT

- There exist secure & efficient incompressible BCs, but no modes of operation to convert them into incompressible AEADs
  - GCM [MV04] is *not* incompressible even if instantiated with an incompressible BC: Once the hashing key for GHASH is leaked, universal forgery is possible
  - Similar attacks also work for CCM [WHF02], OCB [KR11], GCM-SIV [GL15],...
- There is no incompressible AE scheme achieving both of confidentiality & authenticity (w/o special hardware, when ∃ leakage)
  - Can't we reduce incompressibility-like security notions of an AEAD mode to those of BCs?
- New security notions are necessary
  - For both of BCs and AEADs...because existing security notions do not seem suitable for reductions from AEADs to BCs
  - Authenticity notions achieved so far in the white-box setting are only (a kind of) universal unforgeability, much weaker than black-box model

Results

- New white-box security notions for AEAD/BC/PRF/etc.
- A weak variant of public indifferentiability implies reduction
- SIV w/ Sponge & CTR is a white-box secure AEAD mode of BCs
  - Secure up to $2^{n/4}$ black-box queries (n : block length of BC)
- New white-box-secure 256-bit block cipher, “SPACE256-16”
  - Variant of SPACE-16 [Bogdanov-Isobe 2015]
  - We conjecture it is secure (w.r.t. our new incompressibility security notion)
- Model & Assumption
  - Malwares can be detected if they consume lots of computational resources / send huge data outside
  - No assumptions on hardware

New Security Notions

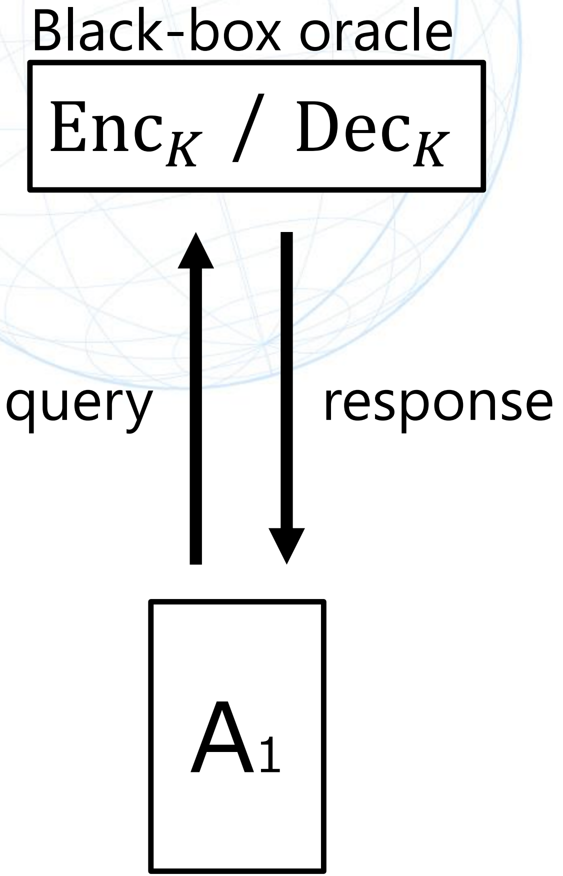
How to Define Security Notions for AEADs?

NTT

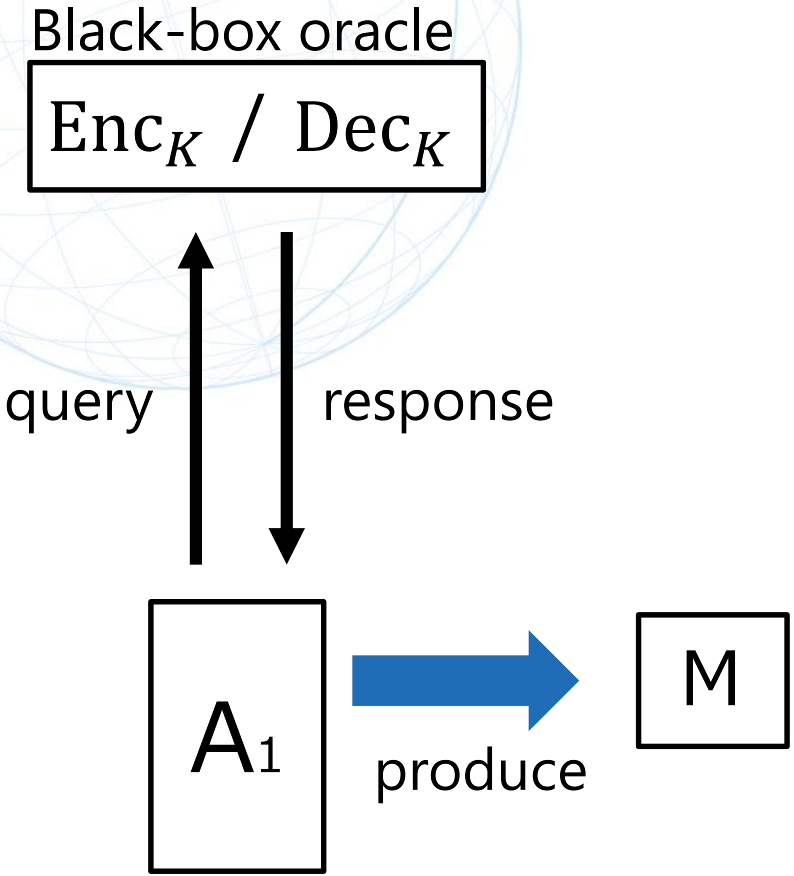
- Real-Ideal distinguishing games
  - Like various conventional AEAD security notions
- Limits on the amount of leakage / malware running time
  - expecting malware can be detected if the leakage / running time are large
- Security after code lifting (= after malware stops)
  - no security is guaranteed during code lifting
- Leakage means nonce-repeat
  - Malware may leak valid plaintext-ciphertext pairs under some nonces that haven't been queried by the attacker, so nonce-misuse resistance is necessary
- We cannot prohibit the attacker to forward outputs from the encryption oracle to the decryption oracle

We extend the **Pseudo Random Injection (PRI) security** [RS06]
Ideal object: Random Injection (and its inverse)

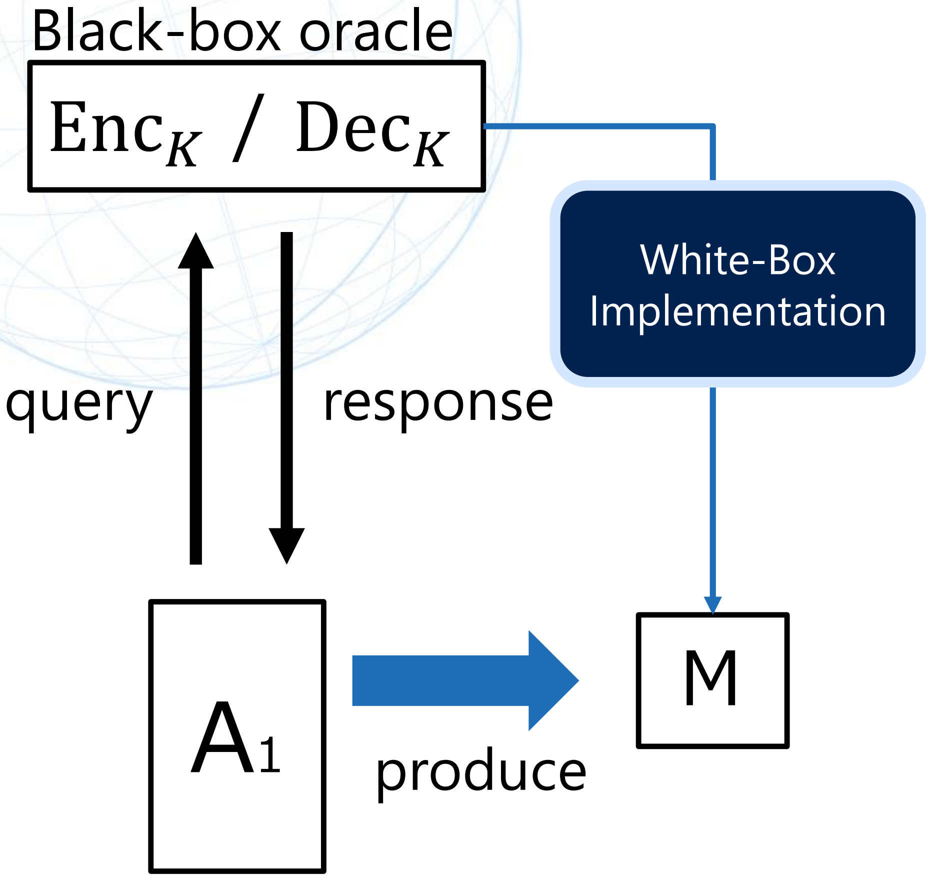
Real World



Real World

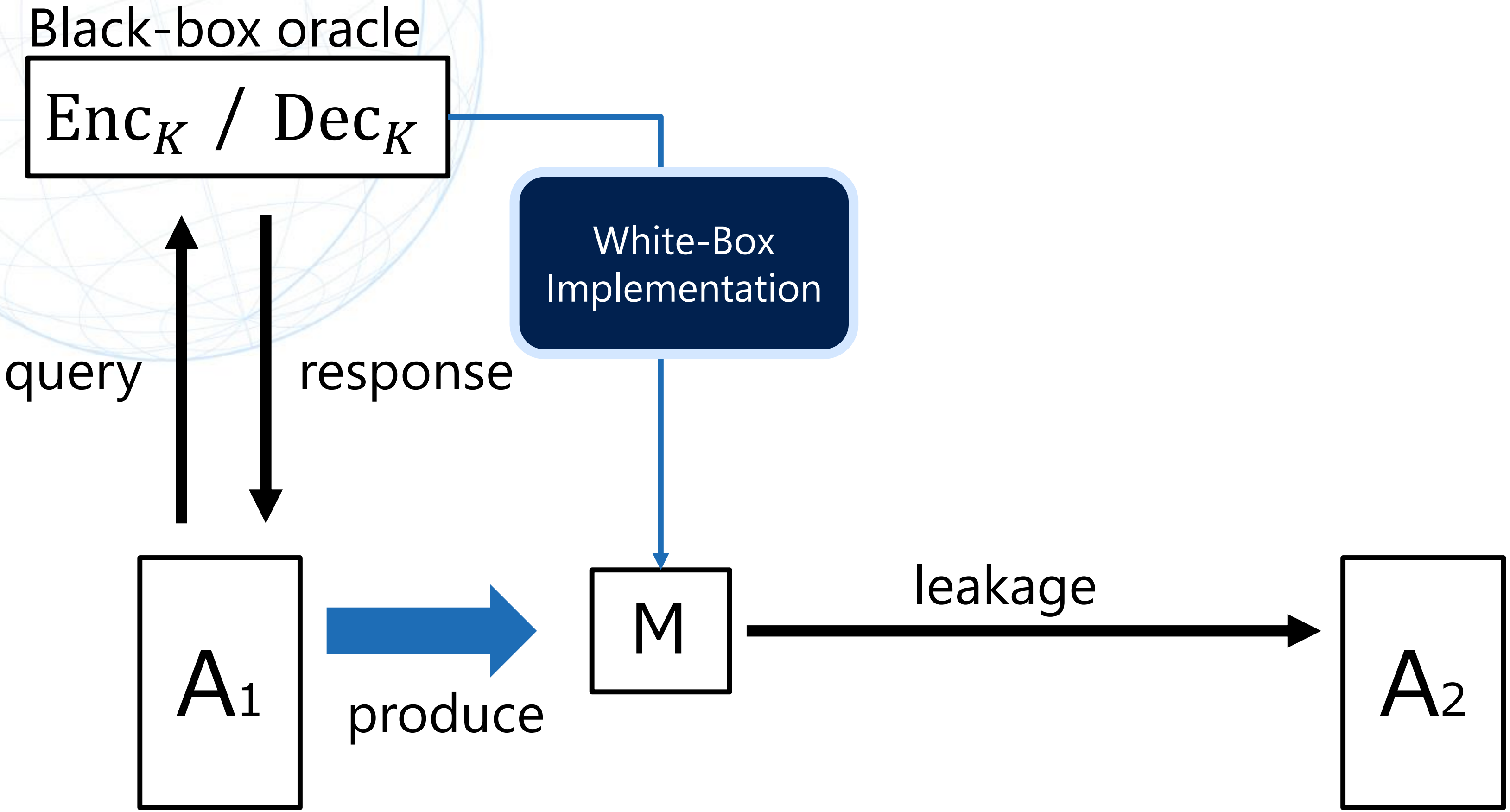

Black-box oracle
Enc_K / Dec_K
query
response
A1
produce
M

Real World



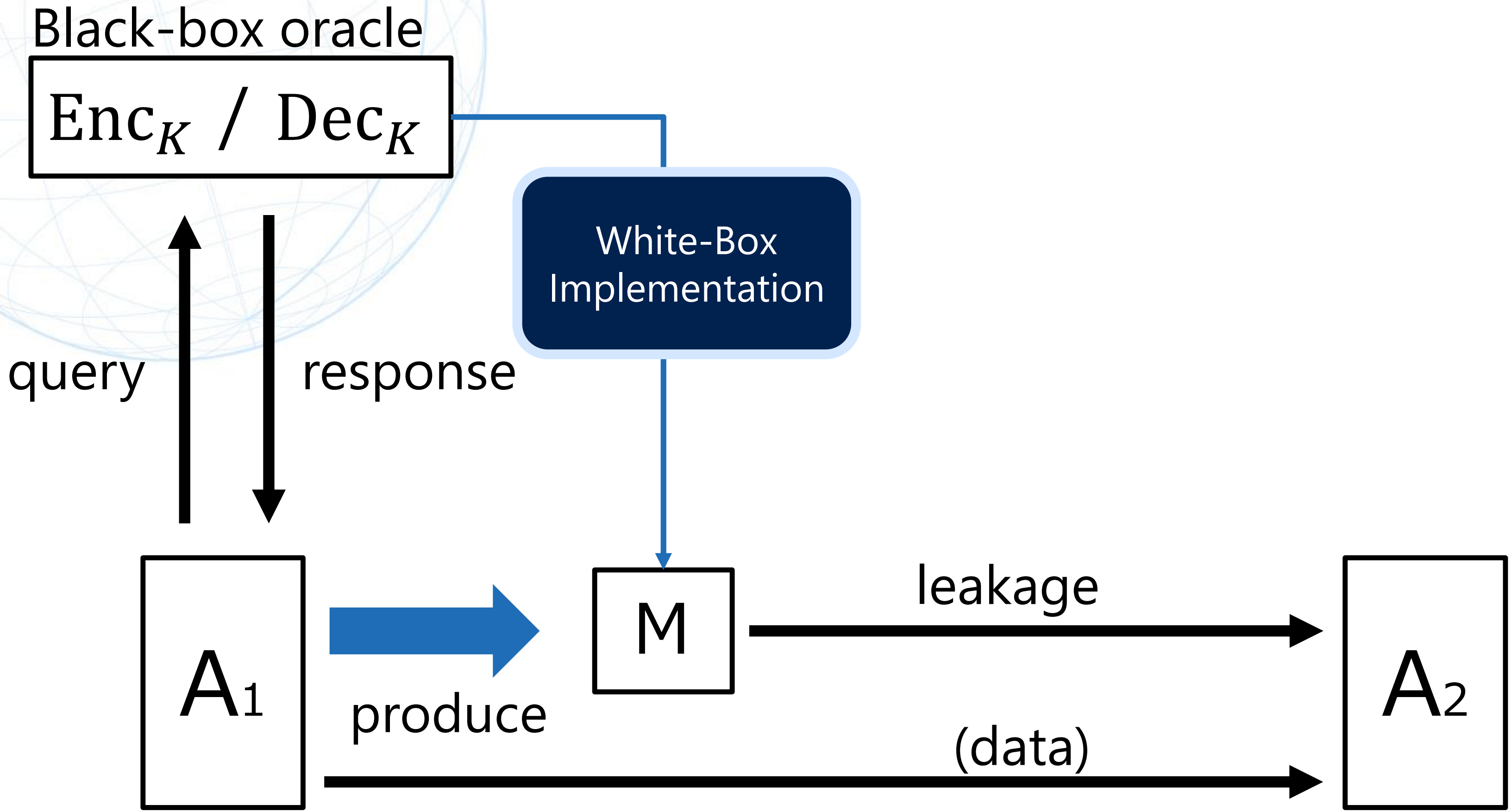
Black-box oracle
Enc_K / Dec_K
White-Box Implementation
query
response
A1
produce
M

Real World

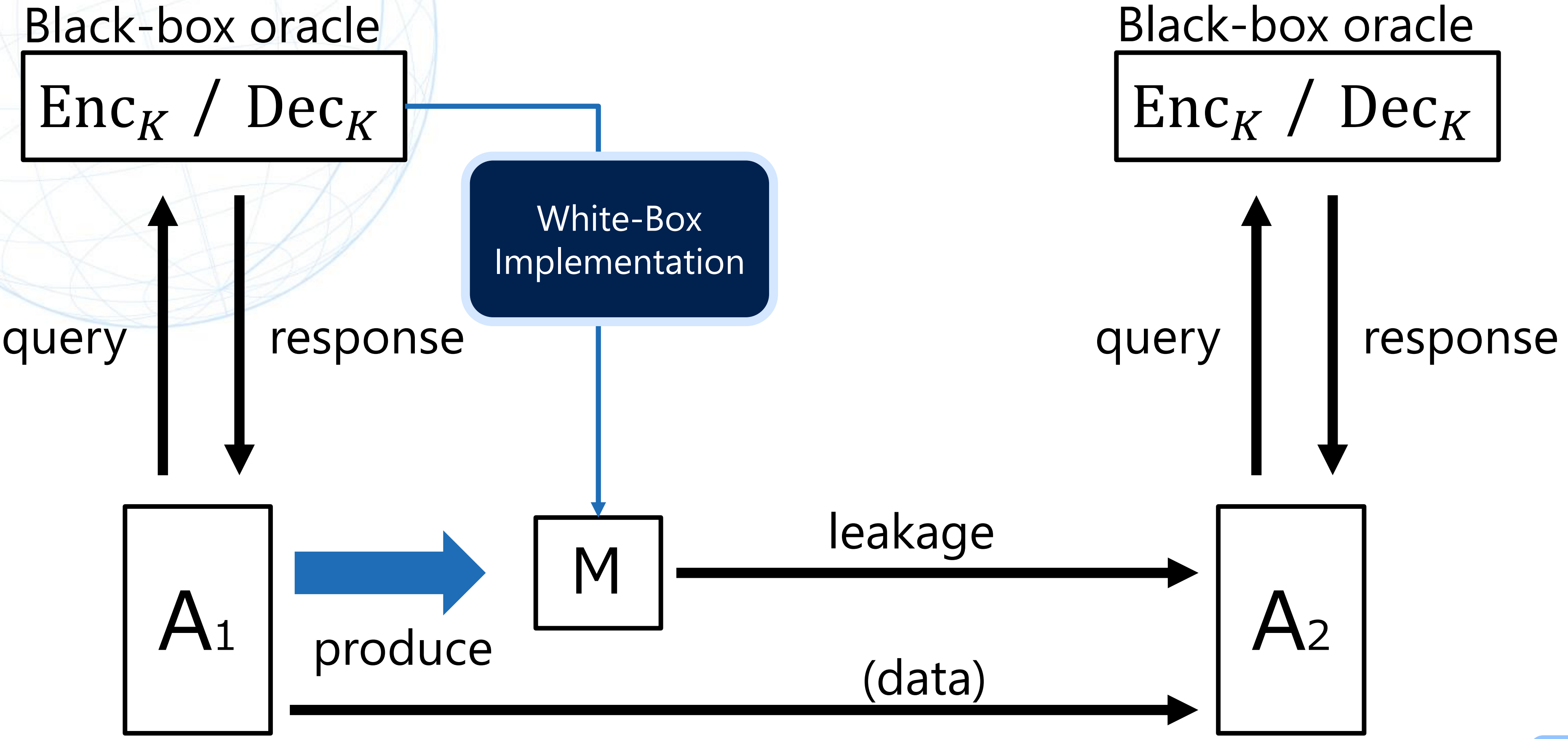


Black-box oracle
Enc_K / Dec_K
query
response
White-Box Implementation
A1
produce
M
leakage
A2

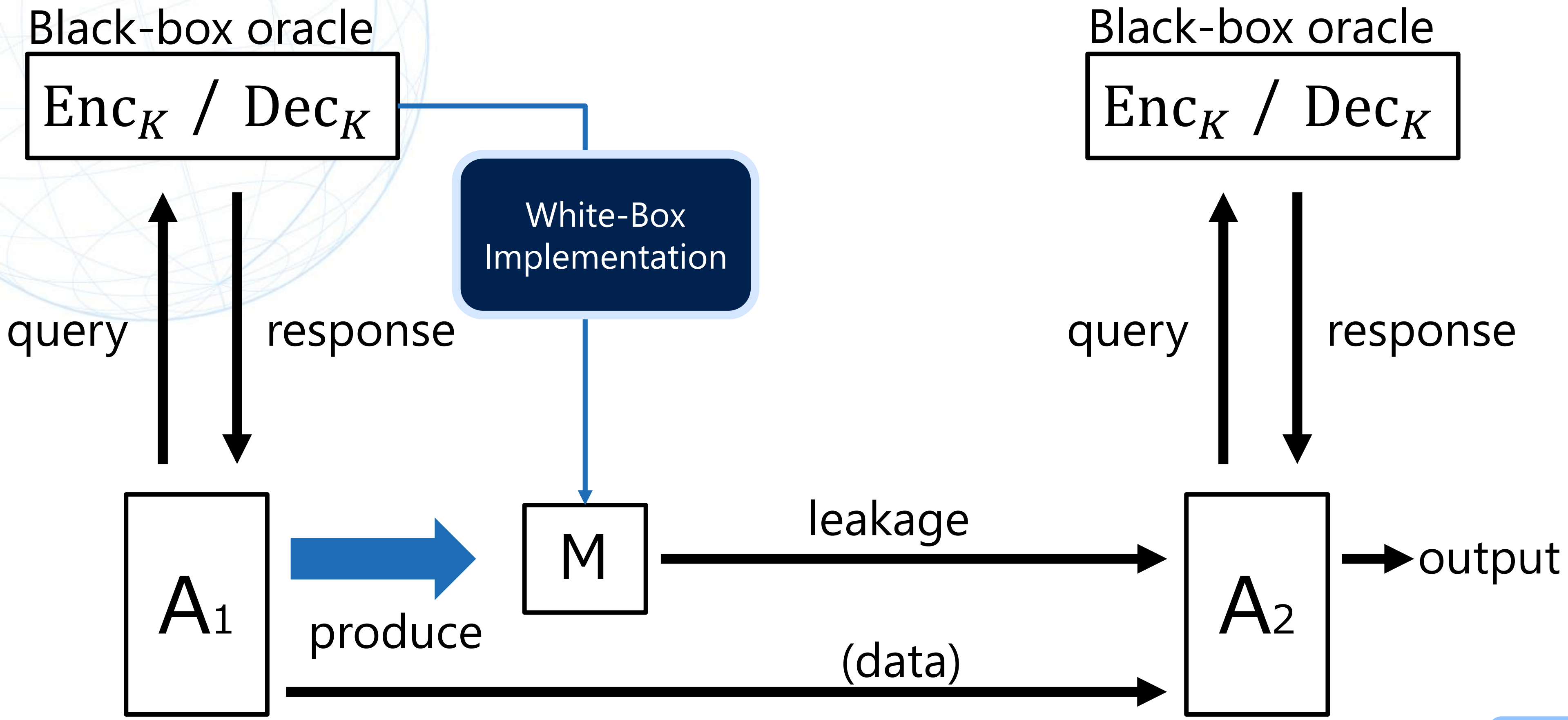
Real World


Black-box oracle
Enc_K / Dec_K
White-Box Implementation
query
response
A1
produce
M
leakage
(data)
A2

Real World

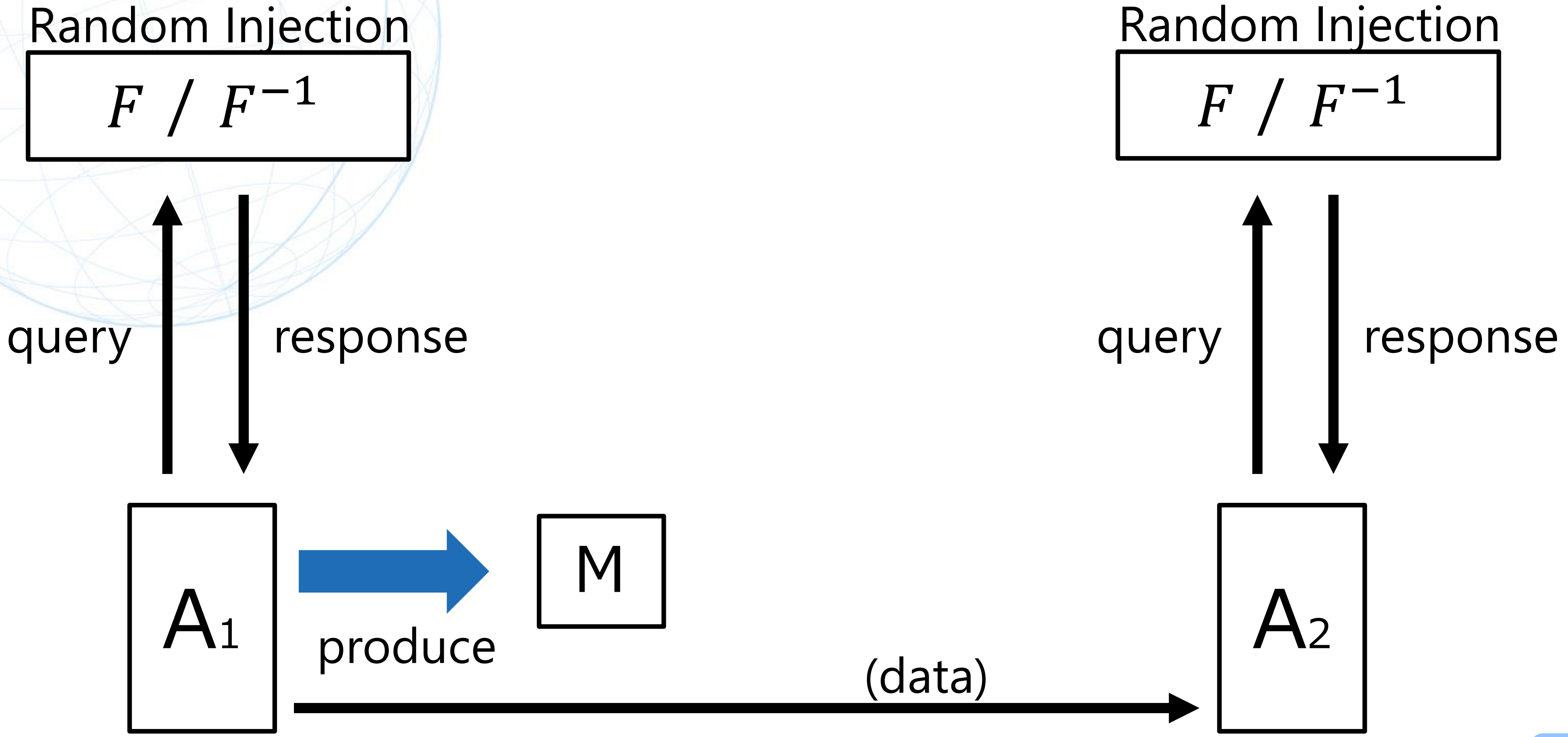

Black-box oracle
Enc_K / Dec_K
Black-box oracle
Enc_K / Dec_K
White-Box Implementation
query
response
query
response
A1
produce
M
leakage
(data)
A2

Real World



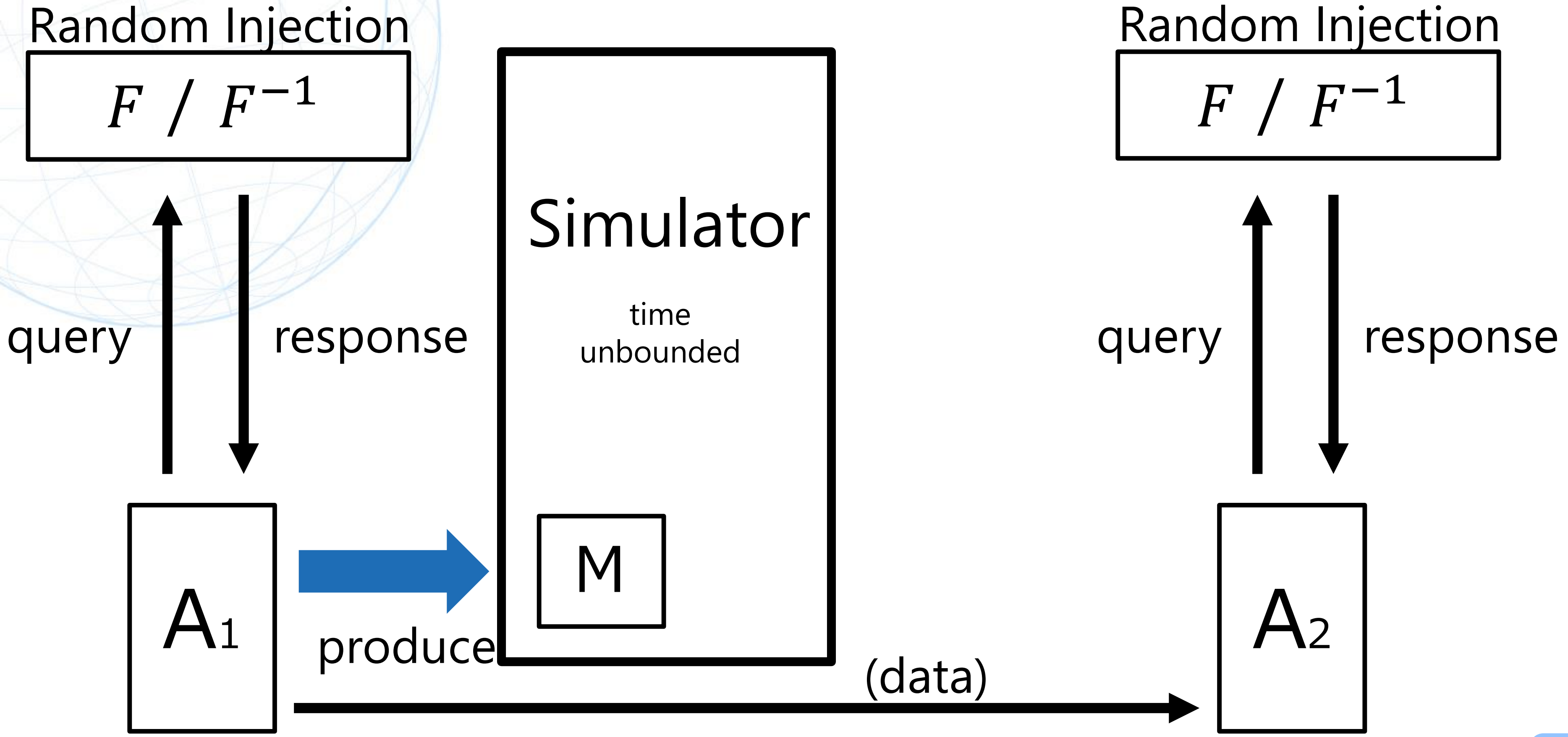
Black-box oracle
Enc_K / Dec_K
query
response
White-Box Implementation
A_1
produce
M
leakage
(data)
Black-box oracle
Enc_K / Dec_K
query
response
A_2
output

Ideal World

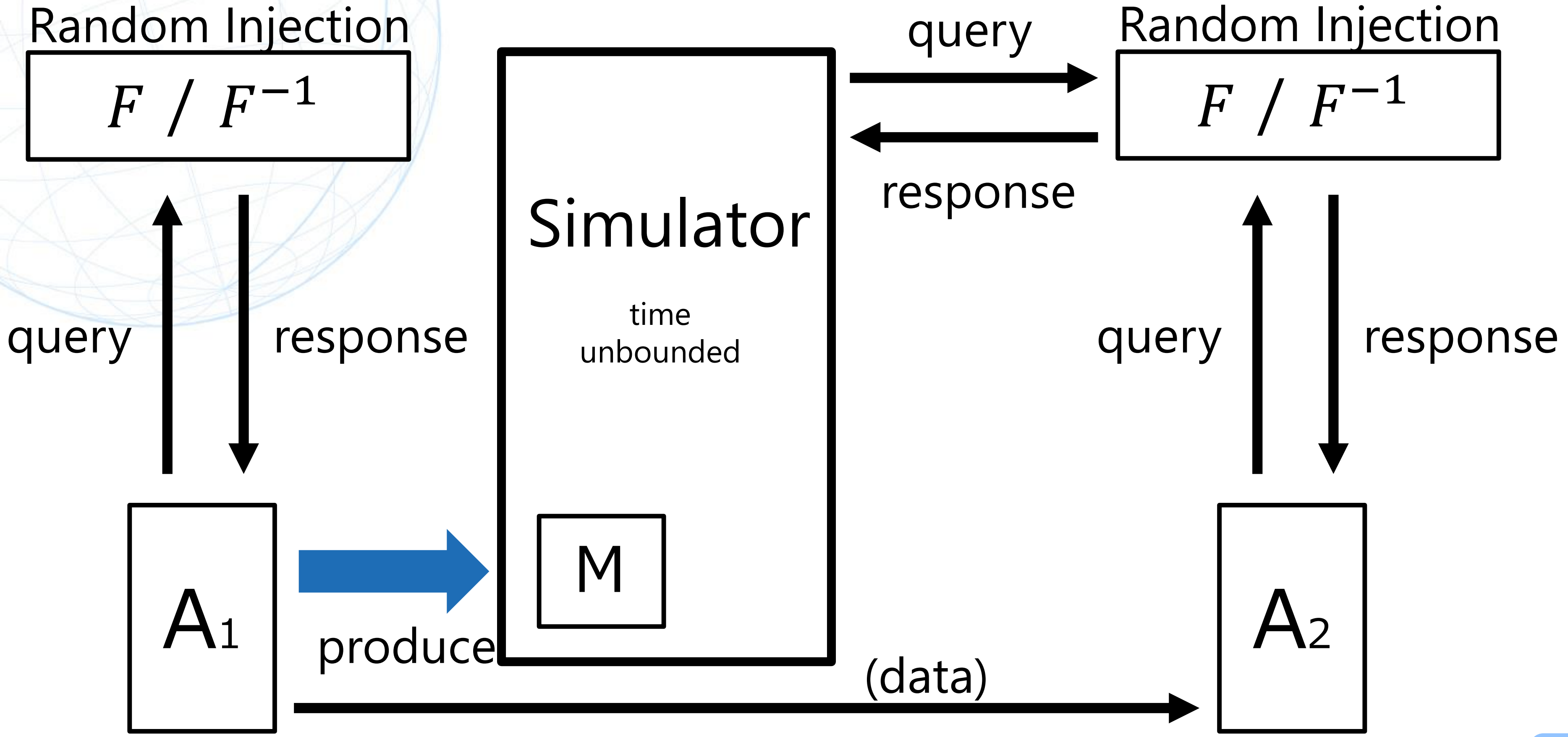


Random Injection
F / F^{-1}
query
response
Random Injection
F / F^{-1}
query
response
A1
produce
M
(data)
A2

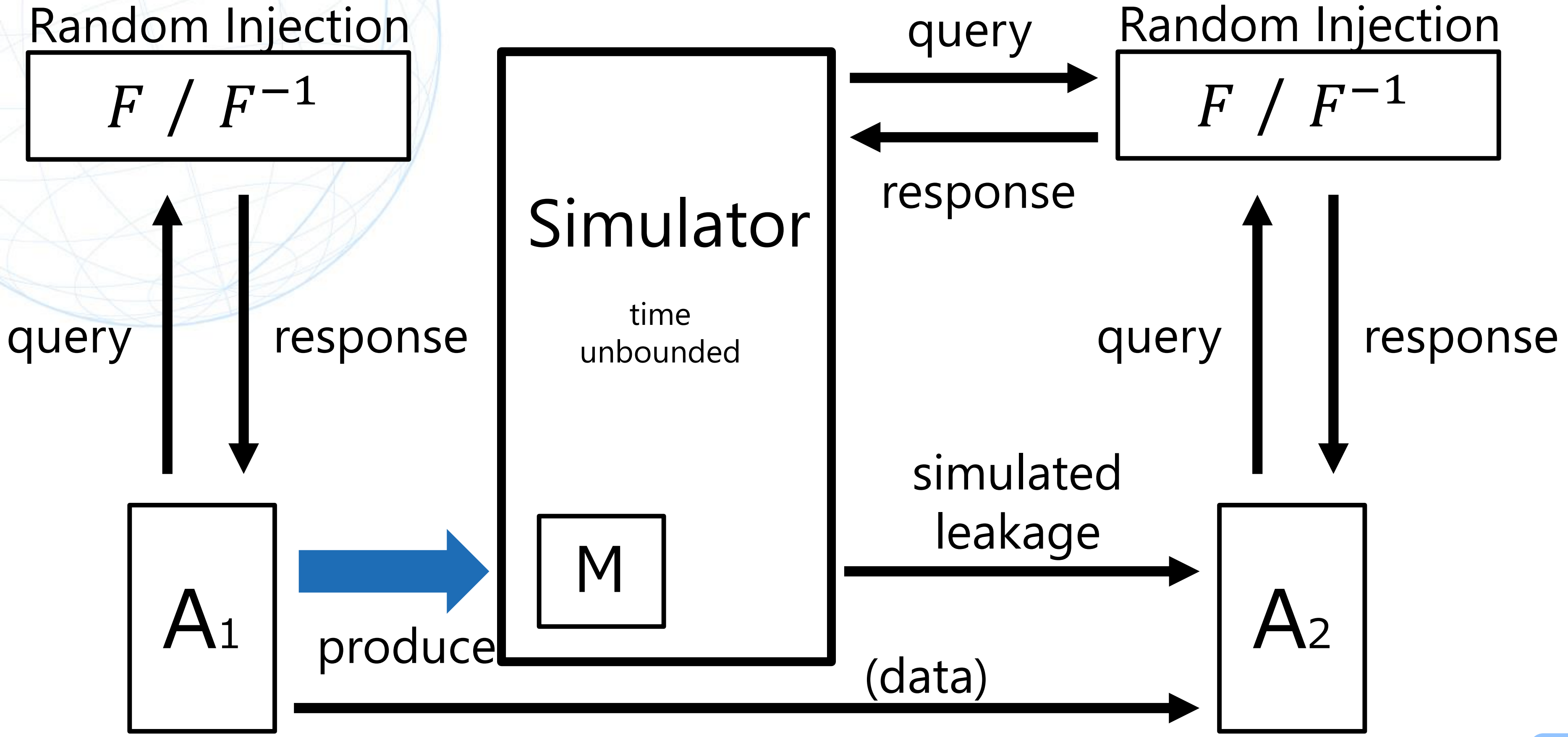
Ideal World


Random Injection
F / F^{-1}
query
response
Simulator
time
unbounded
M
A1
produce
(data)
Random Injection
F / F^{-1}
query
response
A2

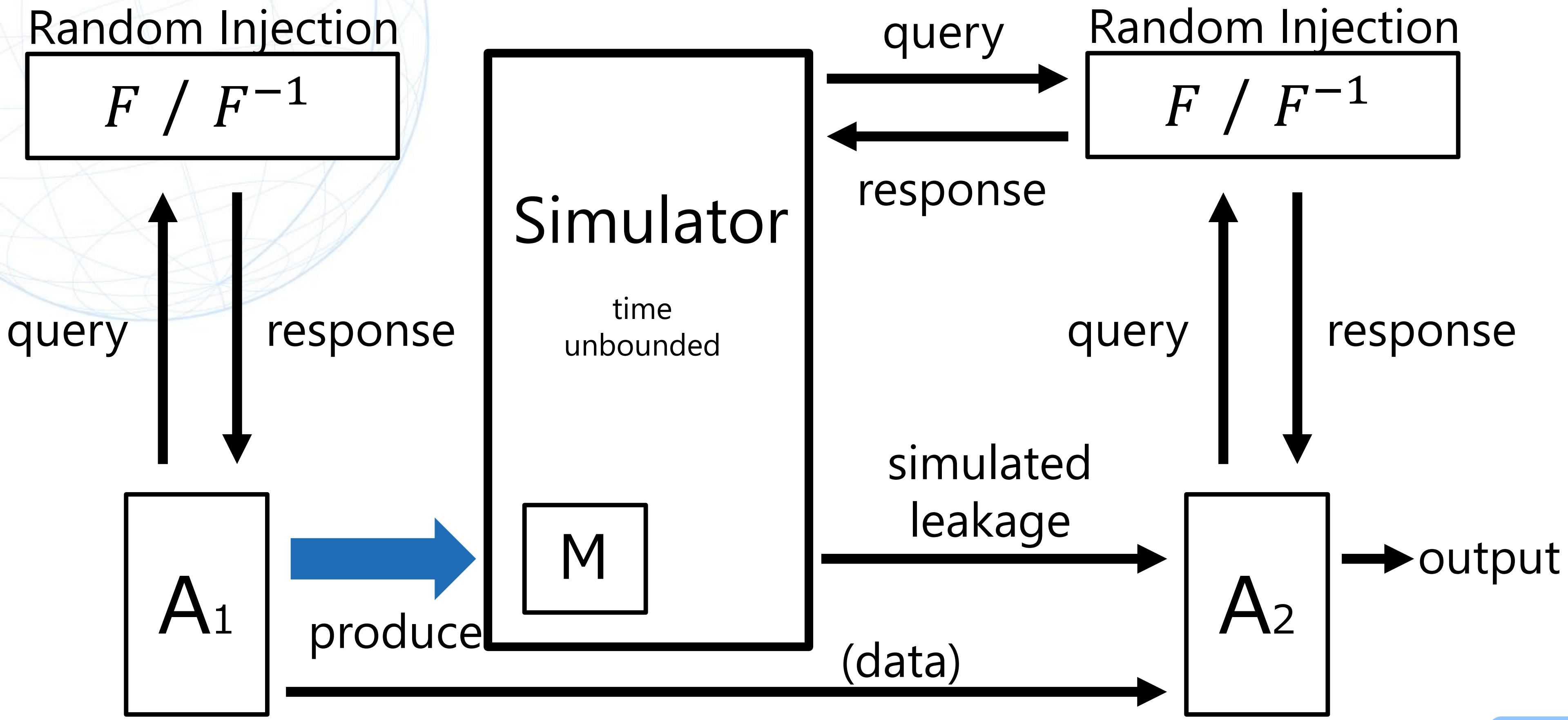
Ideal World


Random Injection
$F \ / \ F^{-1}$
query
response
Simulator
time
unbounded
M
A1
produce
query
Random Injection
$F \ / \ F^{-1}$
response
query
response
(data)
A2

Ideal World


Random Injection
F / F^{-1}
query
response
Simulator
time
unbounded
M
Random Injection
F / F^{-1}
query
response
query
response
simulated
leakage
A1
produce
(data)
A2

Ideal World


Random Injection
F / F−1
query
response
A1
produce
Simulator
time
unbounded
M
query
response
Random Injection
F / F−1
simulated
leakage
query
response
A2
output
(data)

Security Notion for AEADs : whPRI

We say an AEAD scheme is *whPRI-secure* if

- For any "efficient" A = (A1,A2),
- there exists a time-unbounded simulator S making "reasonable amount of" queries to F / F^{-1} s.t.

$$Adv_S^{whPRI}(A) \coloneqq \Pr[\text{Game-Real} \Rightarrow 1] - \Pr[\text{Game-Ideal} \Rightarrow 1]$$

is small

Interpretation of whPRI-security

∃ simulator s.t. Adv becomes small

⬇

λ-bit leakage by malware can contain
only λ-bit information of valid plaintext-ciphertext pairs

⬇

<u>The implementation is incompressible</u>
privacy & authenticity lost by λ-bit leakage is only λ-bit

Security Notion for other schemes

- Security notions for other schemes (BCs, keyed functions,…) are similarly defined
- The new notion for BCs : “whPRP” (extension of PRP security)

- **Conjecture**: The SPACE cipher [Bogdanov-Isobe 2015] is whPRP-secure w.r.t. some reasonable parameters
- **Our Goal**: Reduce whPRI-security of an AEAD mode of BCs to whPRP-security of a BC (e.g., SPACE)

(Some variants of) Indifferentiability Implies Reductions

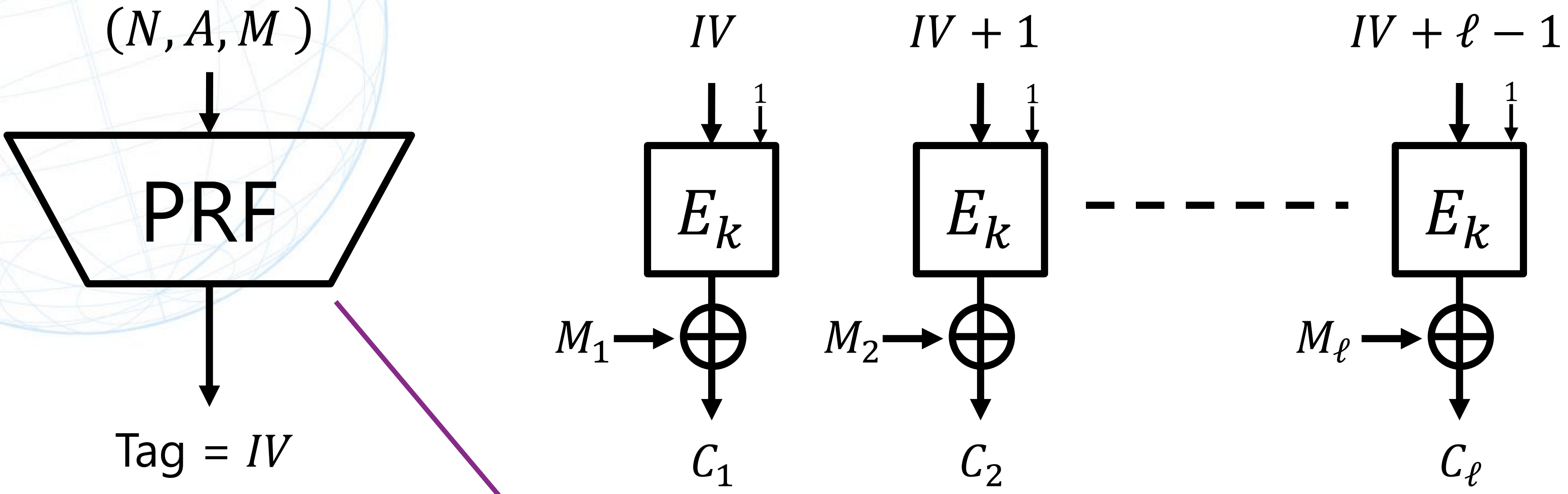
- The structure of a scheme is indifferentiable from the random object (when the primitive is ideally random)

  ⇒ ∃ reductions between new white-box security notions

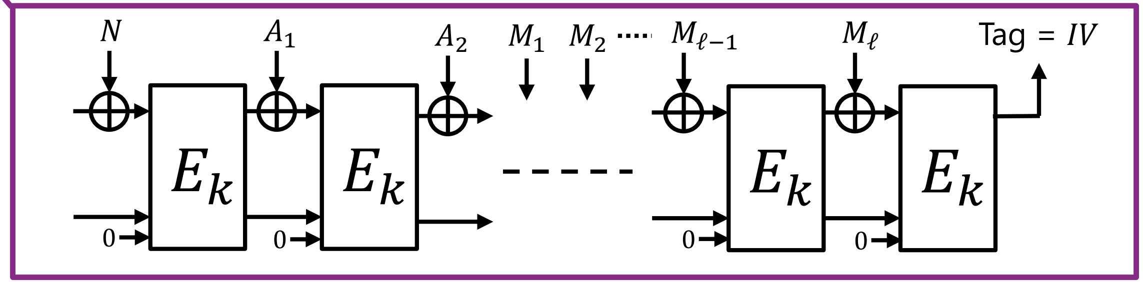
- In fact some weaker variants of indifferentiability are sufficient to show reductions
  - public indifferentiability [Dodis et al. 2009] [Yoneyama et al. 2009]
  - "weak public indifferentiability" (new!)

Incompressible AEAD Mode of BCs & New Incompressible BC

SIV [RS06] with Sponge [BDPV08] Based PRF + CTR



The structure is public indifferentiable from a random injection!



256-bit Block Variant of SPACE Cipher

NTT

- (Very roughly,) the mode is secure up to $2^{n/4}$ black-box queries
- SPACE cipher seems to be whPRP-secure, but n=128
  - The security of the resulting AEAD is only up to 2^{32} complexity

- SPACE256-16 : a new 256-bit block cipher
  - Based on SPACE(-16) [Bogdanov-Isobe 2015]
  - The resulting AEAD becomes secure up to 2^{64} complexity
  - We conjecture it is secure if {query, malware time} $\ll 2^{64}$ & leakage $\ll 2^{20}$

- The resulting AEAD (w/ SPACE256-16) is practical : ≈530 cycle/B
  - Intel platform, 1KB message
  - Not so much worse than raw SPACE-16 (305.11 cycle/B) [Bogdanov et al. 2016]

Summary

Summary

- New white-box security notions for AEAD/BC/PRF/etc.
- A weak variant of public indifferentiability implies reduction
- SIV w/ Sponge & CTR is a white-box secure AEAD mode of BCs
  - Secure up to $2^{n/4}$ black-box queries (n : block length of BC)
- New white-box-secure 256-bit block cipher, "SPACE256-16"
  - Variant of SPACE(-16)
  - We conjecture it is secure (w.r.t. our new incompressibility security notion)
- Model & Assumption
  - Malwares can be detected if they consume lots of computational resources / send huge data outside
  - No assumptions on hardware

Thank you!