Security of Encryption Modes and an Exposition of Proof Techniques

Bart Mennink
Radboud University (The Netherlands)

WCC 2024

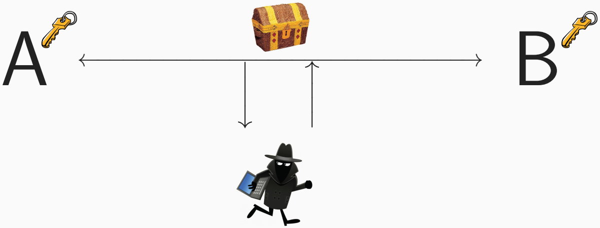
June 21, 2024

Keyed Symmetric Cryptography

General Setting

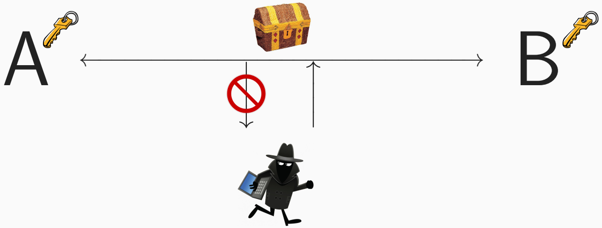
- Two parties, Alice and Bob, communicate over a public channel
    - They have agreed on a joint key and use it to transmit data

General Setting



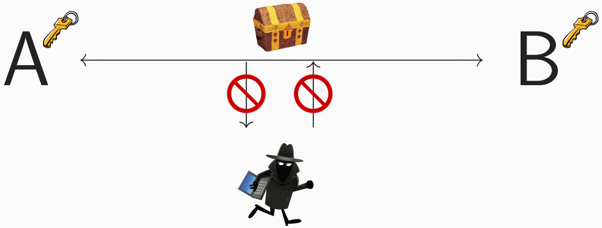
- Two parties, Alice and Bob, communicate over a public channel
  - They have agreed on a joint key and use it to transmit data
- A malicious party, Eve, may try to exploit/disturb/... the communication
- In symmetric cryptography, we are concerned with two main security properties:

General Setting



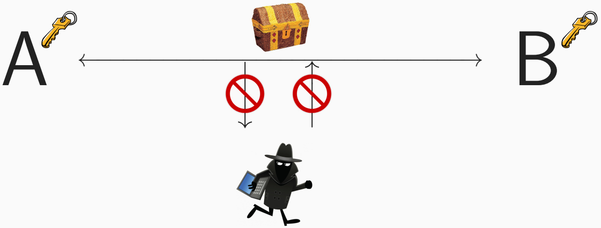
- Two parties, Alice and Bob, communicate over a public channel
    - They have agreed on a joint key and use it to transmit data
- A malicious party, Eve, may try to exploit/disturb/... the communication
- In symmetric cryptography, we are concerned with two main security properties:
    - **Confidentiality (or data privacy):** Eve cannot learn anything about data

General Setting



- Two parties, Alice and Bob, communicate over a public channel
  - They have agreed on a joint key and use it to transmit data
- A malicious party, Eve, may try to exploit/disturb/... the communication
- In symmetric cryptography, we are concerned with two main security properties:
  - **Confidentiality (or data privacy):** Eve cannot learn anything about data
  - **Authenticity:** Eve cannot manipulate the data

General Setting



- Two parties, Alice and Bob, communicate over a public channel
  - They have agreed on a joint key and use it to transmit data
- A malicious party, Eve, may try to exploit/disturb/... the communication
- In symmetric cryptography, we are concerned with two main security properties:
  - **Confidentiality (or data privacy):** Eve cannot learn anything about data
  - **Authenticity:** Eve cannot manipulate the data

In this presentation I will focus on confidentiality

One-Time Pad Encryption

Encryption:

$$
\begin{array}{rccccccccccccccc}
M = & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
K = & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \bigoplus
\end{array}
$$

One-Time Pad Encryption

Encryption:

$$
\begin{array}{rccccccccccccccc}
M = & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
K = & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \bigoplus \\
\hline
C = & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0
\end{array}
$$

One-Time Pad Encryption

Encryption:

$$
\begin{array}{rccccccccccccccccc}
M = & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & \\
K = & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \bigoplus \\
\hline
C = & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 &
\end{array}
$$

Decryption:

$$
\begin{array}{rccccccccccccccccc}
C = & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & \\
K = & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \bigoplus
\end{array}
$$

One-Time Pad Encryption

Encryption:

$$\begin{array}{rccccccccccccccccl} M = & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & \\ K = & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \bigoplus \\ \hline C = & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & \end{array}$$

Decryption:

$$\begin{array}{rccccccccccccccccl} C = & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & \\ K = & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \bigoplus \\ \hline M = & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & \end{array}$$

Properties of One-Time Pad

- One-time pad is a type of stream encryption

Properties of One-Time Pad

- One-time pad is a type of stream encryption
- Perfect secrecy (against an attacker that has no knowledge about the key)
  - Given C, an attacker correctly guesses M with probability $1/2^{|K|}$

Properties of One-Time Pad

- One-time pad is a type of stream encryption
- Perfect secrecy (against an attacker that has no knowledge about the key)
  - Given C, an attacker correctly guesses M with probability $1/2^{|K|}$
- Key must be as long as the plaintext!

Stream Encryption

Properties of One-Time Pad

- One-time pad is a type of stream encryption
- Perfect secrecy (against an attacker that has no knowledge about the key)
  - Given C, an attacker correctly guesses M with probability $1/2^{|K|}$
- Key must be as long as the plaintext!

Stream Ciphers

- Generate long keystream Z from short key K

Properties of One-Time Pad

- One-time pad is a type of stream encryption
- Perfect secrecy (against an attacker that has no knowledge about the key)
  - Given C, an attacker correctly guesses M with probability $1/2^{|K|}$
- Key must be as long as the plaintext!

Stream Ciphers

- Generate long keystream Z from short key K
- Much more practical!

Properties of One-Time Pad

- One-time pad is a type of stream encryption
- Perfect secrecy (against an attacker that has no knowledge about the key)
  - Given C, an attacker correctly guesses M with probability $1/2^{|K|}$
- Key must be as long as the plaintext!

Stream Ciphers

- Generate long keystream Z from short key K
- Much more practical!
- Security degrades:
  1. Key guessing still succeeds with probability $1/2^{|K|}$ but now with shorter key
  2. The stream cipher mechanism is another focal point of attack

Stream Cipher: Vigenère ($\approx$ 1553, Wikipedia)

$K \longrightarrow$ the UHU stic stream cipher $\longrightarrow Z = K \| K \| K \| \cdots$

Stream Cipher: Vigenère ($\approx$ 1553, Wikipedia)

$K \longrightarrow$ [the UHU stic stream cipher] $\longrightarrow Z = K \| K \| K \| \cdots$

- Key guessing:
  - Exhaustive key search succeeds with probability $\mathbf{Pr}\,(\text{success}) = 1/2^{|K|}$

K → the UHU stic stream cipher → $Z = K \| K \| K \| \cdots$

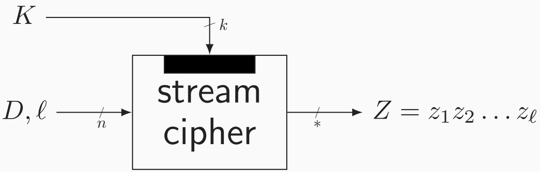
- Key guessing:
  - Exhaustive key search succeeds with probability $\mathbf{Pr}\left(\text{success}\right) = 1/2^{|K|}$
- Ciphertext Only Attack:
  - Long ciphertexts leak info via letter frequencies

- Key guessing:
  - Exhaustive key search succeeds with probability $\mathbf{Pr}(\text{success}) = 1/2^{|K|}$
- Ciphertext Only Attack:
  - Long ciphertexts leak info via letter frequencies
- Known Plaintext Attack:
  - Knowledge of short plaintext sequence reveals full keystream

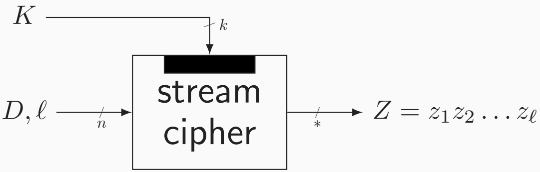
- Key guessing:
  - Exhaustive key search succeeds with probability $\mathbf{Pr}\,(\text{success}) = 1/2^{|K|}$
- Ciphertext Only Attack:
  - Long ciphertexts leak info via letter frequencies
- Known Plaintext Attack:
  - Knowledge of short plaintext sequence reveals full keystream

We need something more sophisticated!

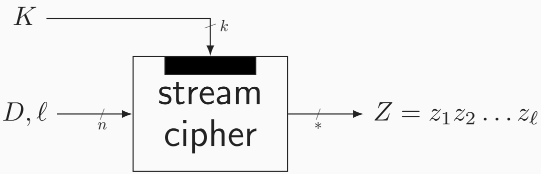
How to Model Security?



- Using key K, diversifier D, and length ℓ, keystream Z of length ℓ is generated

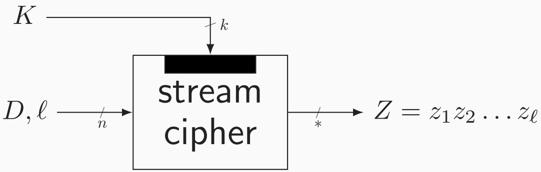


- Using key K, diversifier D, and length ℓ, keystream Z of length ℓ is generated
- The diversifier must be different for each message that is transmitted



- Using key K, diversifier D, and length ℓ, keystream Z of length ℓ is generated
- The diversifier must be different for each message that is transmitted
- Example: data streams, e.g., pay TV and telephone, often split data in relatively short, numbered, frames. The frame number may serve as diversifier:

$$C_i = M_i \oplus \mathsf{SC}(K, i, |M_i|)$$

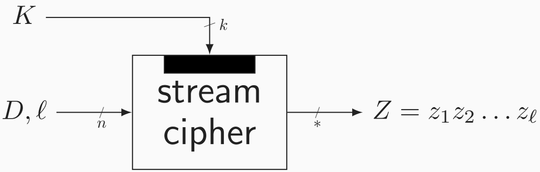


- Using key K, diversifier D, and length ℓ, keystream Z of length ℓ is generated
- The diversifier must be different for each message that is transmitted
- Example: data streams, e.g., pay TV and telephone, often split data in relatively short, numbered, frames. The frame number may serve as diversifier:

$$C_i = M_i \oplus \mathsf{SC}(K, i, |M_i|)$$

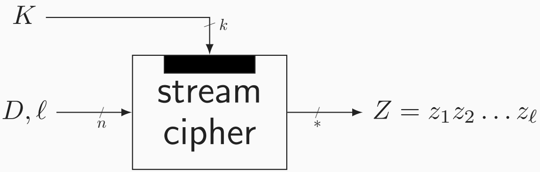
When is a stream cipher strong enough?

Stream Cipher Security, Intuition (1/3)



- Kerckhoffs principle: security should be based on secrecy of K
- Thus: attacker knows the algorithm SC

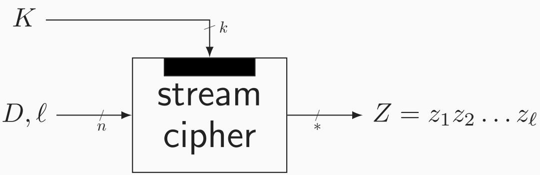
Stream Cipher Security, Intuition (1/3)



- Kerckhoffs principle: security should be based on secrecy of K
- Thus: attacker knows the algorithm SC

- Attacker can also learn some amount of input-output combinations of SC_K
- Intuitively, these data do not expose any irregularities (except for repetition)

Stream Cipher Security, Intuition (1/3)



- Kerckhoffs principle: security should be based on secrecy of K
- Thus: attacker knows the algorithm SC

- Attacker can also learn some amount of input-output combinations of SC_K
- Intuitively, these data do not expose any irregularities (except for repetition)
- SC_K should behave like a random oracle

Random Oracle

- A database of input-output tuples
- Initially empty

D	Z
$\dots$	$\dots$
$\dots$	$\dots$
$\dots$	$\dots$
$\dots$	$\dots$

Random Oracle

- A database of input-output tuples
- Initially empty
- New query (D, ℓ):
  - If D is not in the database:
  - If D is in the database,

D	Z
$\ldots$	$\ldots$
$\ldots$	$\ldots$
$\ldots$	$\ldots$
$\ldots$	$\ldots$

Random Oracle

- A database of input-output tuples
- Initially empty
- New query (D, ℓ):
  - If D is not in the database:
    - generate ℓ random bits Z
    - add (D, Z) to the list
    - return Z
  - If D is in the database,

D	Z
$\dots$	$\dots$
$\dots$	$\dots$
$\dots$	$\dots$
$\dots$	$\dots$

Intermezzo: Random Oracle

Random Oracle

- A database of input-output tuples
- Initially empty
- New query (D, ℓ):
  - If D is not in the database:
    - generate ℓ random bits Z
    - add (D, Z) to the list
    - return Z
  - If D is in the database,

D	Z
1100	101011101010101
$\ldots$	$\ldots$
$\ldots$	$\ldots$
$\ldots$	$\ldots$

Random Oracle

- A database of input-output tuples
- Initially empty
- New query (D, ℓ):
  - If D is not in the database:
    - generate ℓ random bits Z
    - add (D, Z) to the list
    - return Z
  - If D is in the database,

D	Z
1100	101011101010101
1111010101101101	110101
...	...
...	...

Intermezzo: Random Oracle

Random Oracle

- A database of input-output tuples
- Initially empty
- New query (D, ℓ):
  - If D is not in the database:
    - generate ℓ random bits Z
    - add (D, Z) to the list
    - return Z
  - If D is in the database,

D	Z
1100	101011101010101
1111010101101101	110101
001000011100	10101101011101010101011
...	...

D	Z
1100	1010111010101 01
111101010110110 1	110101
001000011100	10101101011101010101 1
$\dots$	$\dots$

Random Oracle

- A database of input-output tuples
- Initially empty
- New query (D, ℓ):
  - If D is not in the database:
    - generate ℓ random bits Z
    - add (D, Z) to the list
    - return Z
  - If D is in the database, look at corresponding Z:
    - If $|Z| \geq \ell$:
    - If $|Z| < \ell$:

Intermezzo: Random Oracle

D	Z
1100	101011101010101
1111010101101101	110101
001000011100	1010110101110101011
...	...

Random Oracle

- A database of input-output tuples
- Initially empty
- New query (D, ℓ):
  - If D is not in the database:
    - generate ℓ random bits Z
    - add (D, Z) to the list
    - return Z
  - If D is in the database, look at corresponding Z:
    - If $|Z| \geq \ell$: return first ℓ bits of Z
    - If $|Z| < \ell$:

Random Oracle

- A database of input-output tuples
- Initially empty
- New query (D, ℓ):
  - If D is not in the database:
    - generate ℓ random bits Z
    - add (D, Z) to the list
    - return Z
  - If D is in the database, look at corresponding Z:
    - If $|Z| \geq \ell$: return first ℓ bits of Z
    - If $|Z| < \ell$:

D	Z
1100	101011101010101
1111010101101101	110101
001000011100	1010110101110101011
...	...

D	Z
1100	101011101010101
1111010101101101	110101
001000011100	10101101011101010101011
...	...

Random Oracle

- A database of input-output tuples
- Initially empty
- New query (D, ℓ):
  - If D is not in the database:
    - generate ℓ random bits Z
    - add (D, Z) to the list
    - return Z
  - If D is in the database, look at corresponding Z:
    - If $|Z| \geq \ell$: return first ℓ bits of Z
    - If $|Z| < \ell$: generate $\ell - |Z|$ random bits Z', append Z' to Z, return $Z \| Z'$

D	Z
1100	101011101010101
1111010101101101	1101011101111101101
001000011100	10101101011101010101011
...	...

Random Oracle

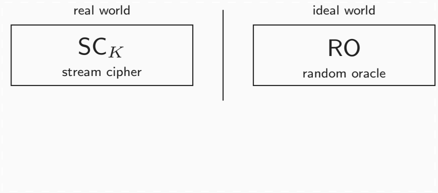
- A database of input-output tuples
- Initially empty
- New query (D, ℓ):
  - If D is not in the database:
    - generate ℓ random bits Z
    - add (D, Z) to the list
    - return Z
  - If D is in the database, look at corresponding Z:
    - If $|Z| \geq \ell$: return first ℓ bits of Z
    - If $|Z| < \ell$: generate $\ell - |Z|$ random bits Z', append Z' to Z, return $Z \| Z'$

D	Z
1100	101011101010101
11110101011101101	1101011101111101101
001000011100	1010110101110101011
...	...

Random Oracle

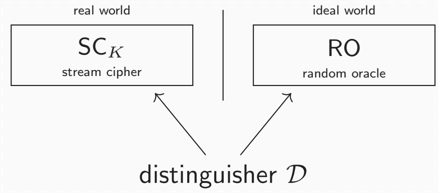
- A database of input-output tuples
- Initially empty
- New query (D, ℓ):
  - If D is not in the database:
    - generate ℓ random bits Z
    - add (D, Z) to the list
    - return Z
  - If D is in the database, look at corresponding Z:
    - If $|Z| \geq \ell$: return first ℓ bits of Z
    - If $|Z| < \ell$: generate $\ell - |Z|$ random bits Z', append Z' to Z, return $Z \| Z'$
    - update (D, Z) in the list

Stream Cipher Security, Intuition (2/3)



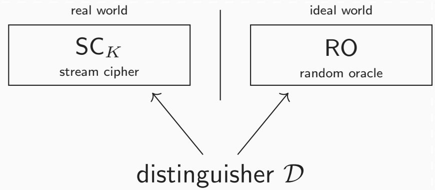
- We thus want to "compare" SC_K with a random oracle RO

Stream Cipher Security, Intuition (2/3)



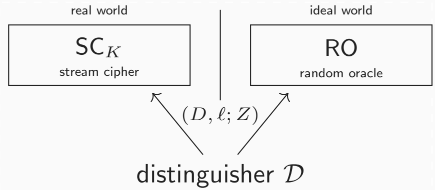
- We thus want to "compare" SC_K with a random oracle RO
- We model a distinguisher $\mathcal{D}$ that is given oracle access to either of the worlds

Stream Cipher Security, Intuition (2/3)



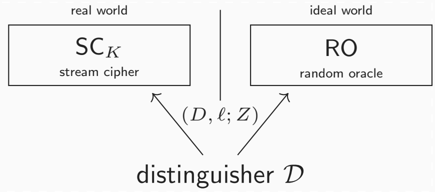
- We thus want to "compare" SC_K with a random oracle RO
- We model a distinguisher $\mathcal{D}$ that is given oracle access to either of the worlds
  - We toss a coin:
    - head: $\mathcal{D}$ is given oracle access to SC_K
    - tail: $\mathcal{D}$ is given oracle access to RO
  - $\mathcal{D}$ does a priori not know which oracle it is given access to

Stream Cipher Security, Intuition (2/3)



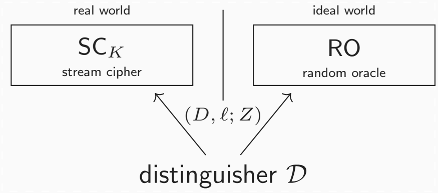
- We thus want to "compare" SC_K with a random oracle RO
- We model a distinguisher $\mathcal{D}$ that is given oracle access to either of the worlds
  - We toss a coin:
    - head: $\mathcal{D}$ is given oracle access to SC_K
    - tail: $\mathcal{D}$ is given oracle access to RO
  - $\mathcal{D}$ does a priori not know which oracle it is given access to
  - $\mathcal{D}$ can now make queries (D, ℓ) to receive Z

Stream Cipher Security, Intuition (2/3)



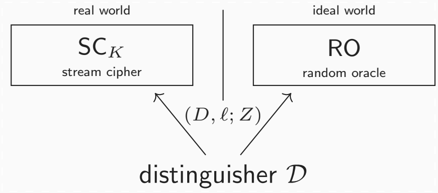
- We thus want to "compare" SC_K with a random oracle RO
- We model a distinguisher $\mathcal{D}$ that is given oracle access to either of the worlds
  - We toss a coin:
    - head: $\mathcal{D}$ is given oracle access to SC_K
    - tail: $\mathcal{D}$ is given oracle access to RO
  - $\mathcal{D}$ does a priori not know which oracle it is given access to
  - $\mathcal{D}$ can now make queries (D, ℓ) to receive Z
  - At the end, $\mathcal{D}$ has to guess the outcome of the toss coin (head/tail)

Stream Cipher Security, Intuition (3/3)



- Denote $\mathcal{D}$'s success probability in correctly guessing head/tail by $\mathbf{Pr}\,(\text{success})$

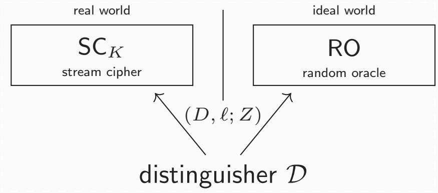
Stream Cipher Security, Intuition (3/3)



- Denote $\mathcal{D}$'s success probability in correctly guessing head/tail by $\mathbf{Pr}\,(\text{success})$
- $\mathcal{D}$ can always guess and succeeds with probability $\geq 1/2$, so we scale the probability to $\mathcal{D}$'s advantage:

$$\mathbf{Adv}(\mathcal{D}) = 2 \cdot \mathbf{Pr}\,(\text{success}) - 1$$

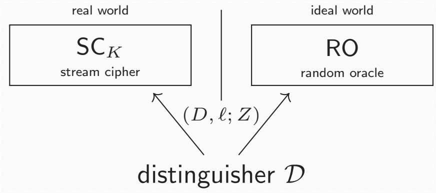
Stream Cipher Security, Intuition (3/3)



- Denote $\mathcal{D}$'s success probability in correctly guessing head/tail by $\mathbf{Pr}\left(\text{success}\right)$
- $\mathcal{D}$ can always guess and succeeds with probability $\geq 1/2$, so we scale the probability to $\mathcal{D}$'s advantage:

$$\begin{aligned}\mathbf{Adv}(\mathcal{D}) &= 2 \cdot \mathbf{Pr}\left(\text{success}\right) - 1 \\ &= \mathbf{Pr}\left(\mathcal{D}^{\mathsf{SC}_K} \text{ returns head}\right) - \mathbf{Pr}\left(\mathcal{D}^{\mathsf{RO}} \text{ returns head}\right)\end{aligned}$$

Stream Cipher Security, Intuition (3/3)

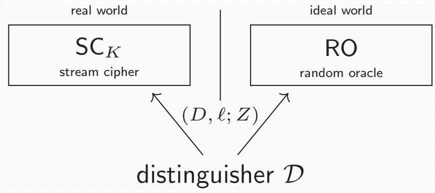


- Denote $\mathcal{D}$'s success probability in correctly guessing head/tail by $\mathbf{Pr}\left(\text{success}\right)$
- $\mathcal{D}$ can always guess and succeeds with probability $\geq 1/2$, so we scale the probability to $\mathcal{D}$'s advantage:

$$\begin{aligned}\mathbf{Adv}(\mathcal{D}) &= 2 \cdot \mathbf{Pr}\left(\text{success}\right) - 1 \\ &= \mathbf{Pr}\left(\mathcal{D}^{\mathsf{SC}_K} \text{ returns head}\right) - \mathbf{Pr}\left(\mathcal{D}^{\mathsf{RO}} \text{ returns head}\right)\end{aligned}$$

- $\mathcal{D}$ is limited by certain constraints

Stream Cipher Security, Intuition (3/3)

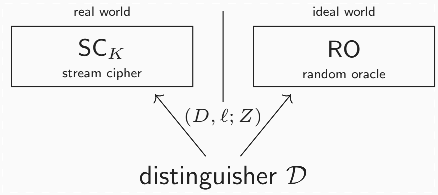


- Denote $\mathcal{D}$'s success probability in correctly guessing head/tail by $\mathbf{Pr}\,(\text{success})$
- $\mathcal{D}$ can always guess and succeeds with probability $\geq 1/2$, so we scale the probability to $\mathcal{D}$'s advantage:

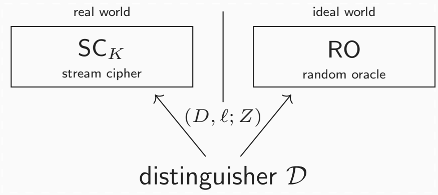
$$\begin{aligned}\mathbf{Adv}(\mathcal{D}) &= 2 \cdot \mathbf{Pr}\,(\text{success}) - 1 \\ &= \mathbf{Pr}\left(\mathcal{D}^{\mathsf{SC}_K} \text{ returns head}\right) - \mathbf{Pr}\left(\mathcal{D}^{\mathsf{RO}} \text{ returns head}\right)\end{aligned}$$

- $\mathcal{D}$ is limited by certain constraints
  - Data (or online) complexity q: total cost of queries $\mathcal{D}$ can make
  - Computation (or time) complexity t: everything that $\mathcal{D}$ can do "on its own"

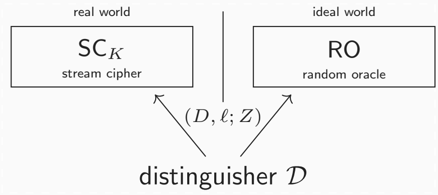
Stream Cipher Security, Formal



- Two oracles: SC_K (for secret key K) and RO (secret)

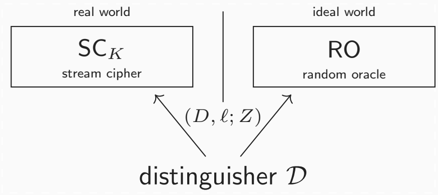


- Two oracles: SC_K (for secret key K) and RO (secret)
- Distinguisher $\mathcal{D}$ has query access to one of these



- Two oracles: SC_K (for secret key K) and RO (secret)
- Distinguisher $\mathcal{D}$ has query access to one of these
- $\mathcal{D}$ tries to determine which oracle it communicates with

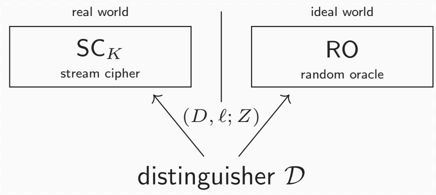
Stream Cipher Security, Formal



- Two oracles: SC_K (for secret key K) and RO (secret)
- Distinguisher $\mathcal{D}$ has query access to one of these
- $\mathcal{D}$ tries to determine which oracle it communicates with
- Its advantage is defined as:

$$\mathbf{Adv}_{\mathsf{SC}}^{\mathrm{prf}}(\mathcal{D}) = \Delta_{\mathcal{D}}\left(\mathsf{SC}_K \; ; \; \mathsf{RO}\right) = \left|\mathbf{Pr}\left(\mathcal{D}^{\mathsf{SC}_K} = 1\right) - \mathbf{Pr}\left(\mathcal{D}^{\mathsf{RO}} = 1\right)\right|$$

Stream Cipher Security, Formal



- Two oracles: SC_K (for secret key K) and RO (secret)
- Distinguisher $\mathcal{D}$ has query access to one of these
- $\mathcal{D}$ tries to determine which oracle it communicates with
- Its advantage is defined as:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{SC}}(\mathcal{D}) = \Delta_{\mathcal{D}}\left(\mathsf{SC}_K \; ; \; \mathsf{RO}\right) = \left|\mathbf{Pr}\left(\mathcal{D}^{\mathsf{SC}_K} = 1\right) - \mathbf{Pr}\left(\mathcal{D}^{\mathsf{RO}} = 1\right)\right|$$

- $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{SC}}(q, t)$: maximum advantage over any distinguisher with complexity q, t

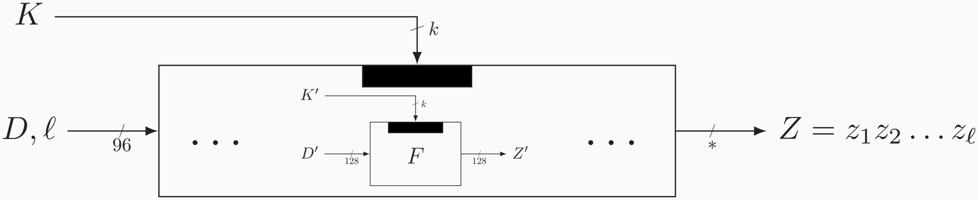
Generic Stream Cipher Design

Generic Stream Cipher Design (1/2)

- Classical approach: LFSRs strengthened with non-linear component
- Modern approach: building construction from smaller cryptographic primitive

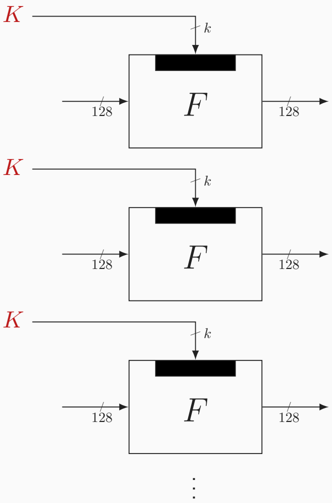
- Classical approach: LFSRs strengthened with non-linear component
- Modern approach: building construction from smaller cryptographic primitive

- Suppose (for the sake of argument):
  - we **know** how to build a strong stream cipher F with fixed-length output
  - we **want** to build a stream cipher with variable-length output



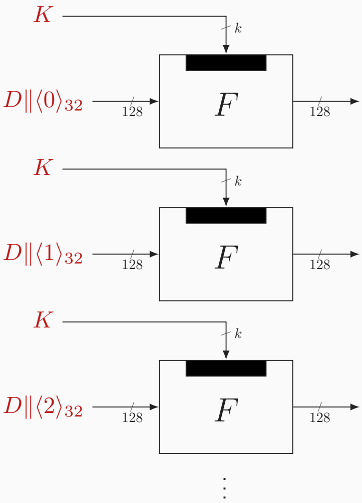
Design

- Feed K to primitive



Design

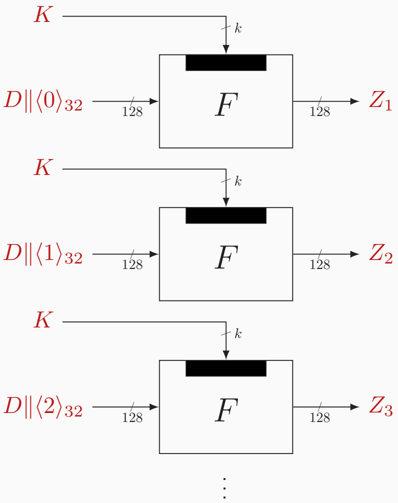
- Feed K to primitive
- Evaluate primitive as often as needed, with D concatenated with counter



Design

- Feed K to primitive
- Evaluate primitive as often as needed, with D concatenated with counter
- Concatenate outputs:

$$Z = Z_1 \parallel Z_2 \parallel Z_3 \parallel \cdots$$



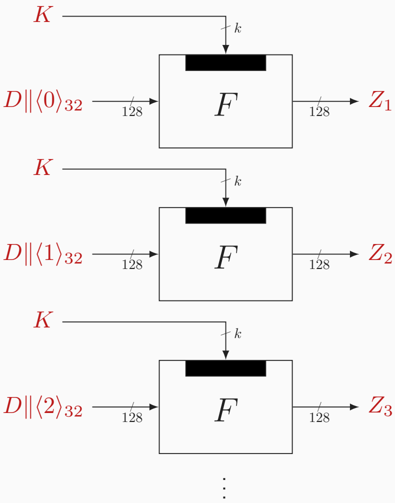
Design

- Feed K to primitive
- Evaluate primitive as often as needed, with D concatenated with counter
- Concatenate outputs:

$$Z = Z_1 \parallel Z_2 \parallel Z_3 \parallel \cdots$$

Security

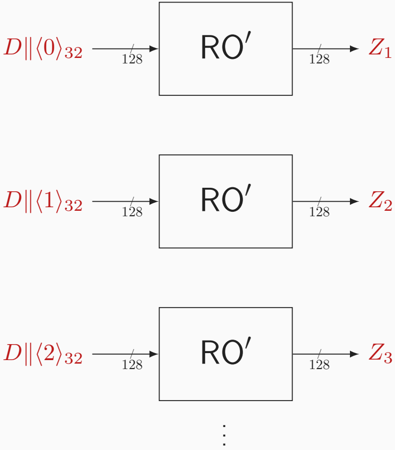
- If F_K is hard to distinguish from a RO′



Design

- Feed K to primitive
- Evaluate primitive as often as needed, with D concatenated with counter
- Concatenate outputs:

$$Z = Z_1 \parallel Z_2 \parallel Z_3 \parallel \cdots$$



Security

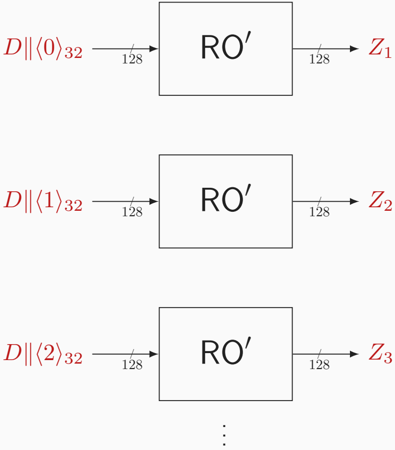
- If F_K is hard to distinguish from a RO′

Generic Stream Cipher Design (2/2)

Design

- Feed K to primitive
- Evaluate primitive as often as needed, with D concatenated with counter
- Concatenate outputs:

$$Z = Z_1 \parallel Z_2 \parallel Z_3 \parallel \cdots$$



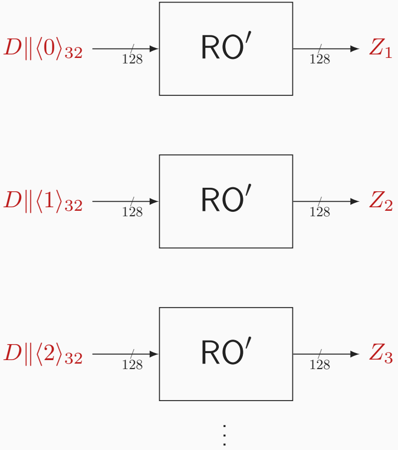
Security

- If F_K is hard to distinguish from a RO′
- Then construction is hard to distinguish from RO

Design

- Feed K to primitive
- Evaluate primitive as often as needed, with D concatenated with counter
- Concatenate outputs:

$$Z = Z_1 \parallel Z_2 \parallel Z_3 \parallel \cdots$$



Security

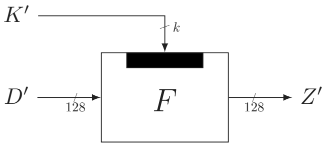
- If F_K is hard to distinguish from a RO′
- Then construction is hard to distinguish from RO
- For the purists: $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{SC}[F]}(q,t) \leq \mathbf{Adv}^{\mathrm{prf}}_{F}(q,t')$

Design

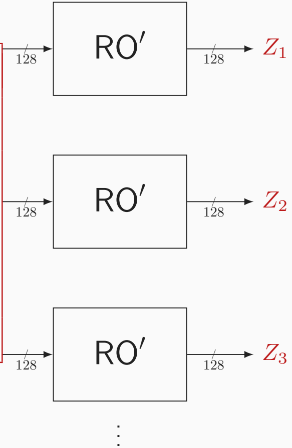
- Feed K to primitive
- Evaluate primitive as concatenated with co
- Concatenate outputs:

$Z = Z_1$

Unfortunately, we do not know how to easily construct a function



that behaves like a RO′

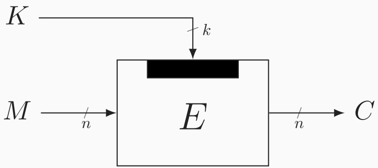


Security

- If F_K is hard to disti
- Then construction is hard to distinguish from RO
- For the purists: $\mathbf{Adv}^{\text{prf}}_{\mathsf{SC}[F]}(q,t) \leq \mathbf{Adv}^{\text{prf}}_{F}(q,t')$

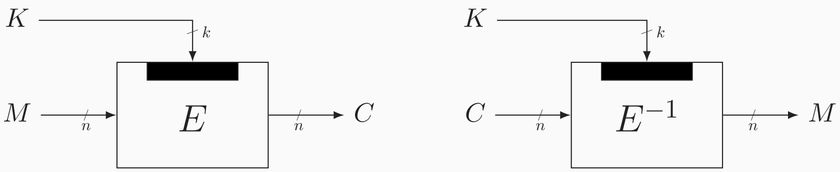
Block Ciphers

Block Ciphers

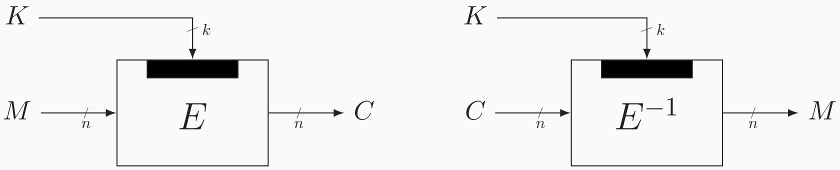


- Using key K, message M is bijectively transformed to ciphertext C
- Key, plaintext, and ciphertext are typically of fixed size

Block Ciphers



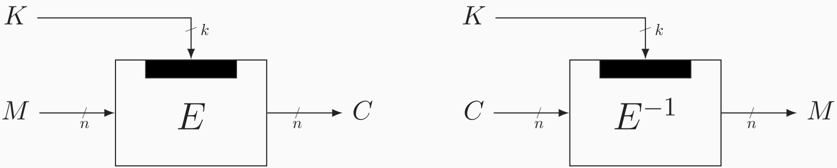
- Using key K, message M is bijectively transformed to ciphertext C
- Key, plaintext, and ciphertext are typically of fixed size
- For fixed key, E_K is invertible and the inverse is denoted as E_K^{-1}



- Using key K, message M is bijectively transformed to ciphertext C
- Key, plaintext, and ciphertext are typically of fixed size
- For fixed key, E_K is invertible and the inverse is denoted as E_K^{-1}
- Example [DR02]:

$$\text{AES-128}\colon \{0,1\}^{128} \times \{0,1\}^{128} \to \{0,1\}^{128}$$
$$(K, M) \mapsto C$$

Block Ciphers

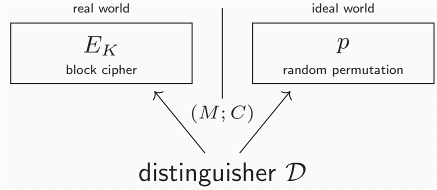


- Using key K, message M is bijectively transformed to ciphertext C
- Key, plaintext, and ciphertext are typically of fixed size
- For fixed key, E_K is invertible and the inverse is denoted as E_K^{-1}
- Example [DR02]:

$$\text{AES-128}\colon \{0,1\}^{128} \times \{0,1\}^{128} \to \{0,1\}^{128}$$
$$(K, M) \mapsto C$$

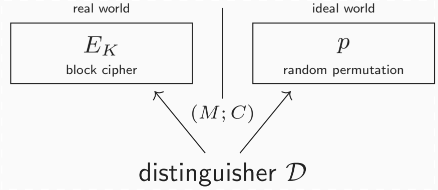
- A good block cipher should behave like a random permutation

Block Cipher Security



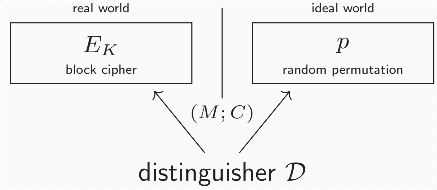
- Two oracles: E_K (for secret key K) and p (secret)

Block Cipher Security



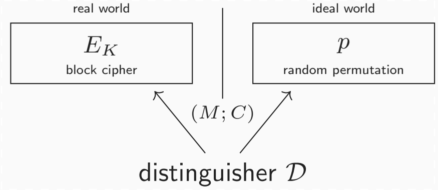
- Two oracles: E_K (for secret key K) and p (secret)
- Distinguisher $\mathcal{D}$ has query access to one of these

Block Cipher Security



- Two oracles: E_K (for secret key K) and p (secret)
- Distinguisher $\mathcal{D}$ has query access to one of these
- $\mathcal{D}$ tries to determine which oracle it communicates with

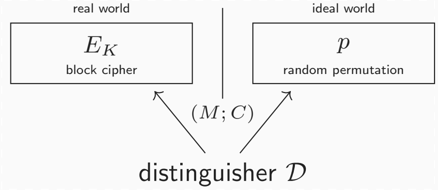
Block Cipher Security



- Two oracles: E_K (for secret key K) and p (secret)
- Distinguisher $\mathcal{D}$ has query access to one of these
- $\mathcal{D}$ tries to determine which oracle it communicates with
- Its advantage is defined as:

$$\mathbf{Adv}_E^{\mathrm{prp}}(\mathcal{D}) = \Delta_{\mathcal{D}}\left(E_K \ ; \ p\right) = \left|\mathbf{Pr}\left(\mathcal{D}^{E_K} = 1\right) - \mathbf{Pr}\left(\mathcal{D}^{p} = 1\right)\right|$$

Block Cipher Security

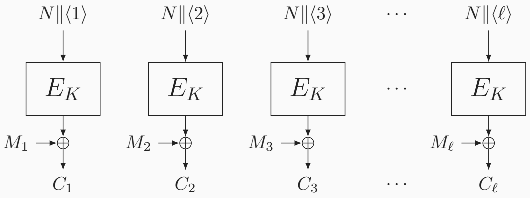


- Two oracles: E_K (for secret key K) and p (secret)
- Distinguisher $\mathcal{D}$ has query access to one of these
- $\mathcal{D}$ tries to determine which oracle it communicates with
- Its advantage is defined as:

$$\mathbf{Adv}_E^{\mathrm{prp}}(\mathcal{D}) = \Delta_{\mathcal{D}}\left(E_K\ ;\ p\right) = \left|\mathbf{Pr}\left(\mathcal{D}^{E_K} = 1\right) - \mathbf{Pr}\left(\mathcal{D}^{p} = 1\right)\right|$$

- $\mathbf{Adv}_E^{\mathrm{prp}}(q, t)$: maximum advantage over any $\mathcal{D}$ with query/time complexity q/t

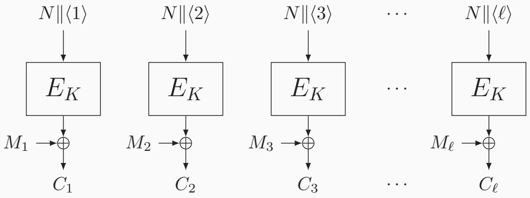
Counter Mode Encryption



Features

- Stream-based encryption mode
- Fully parallelizable (encryption and decryption) and extremely simple
- Decryption needs no E_K^{-1}

Counter (CTR) Mode



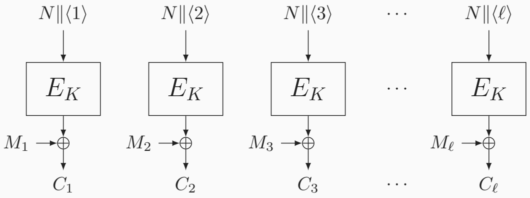
Features

- Stream-based encryption mode
- Fully parallelizable (encryption and decryption) and extremely simple
- Decryption needs no E_K^{-1}

Security

- "Hopefully" secure as long as N is never repeated and E_K is a secure PRP

Counter (CTR) Mode



Features

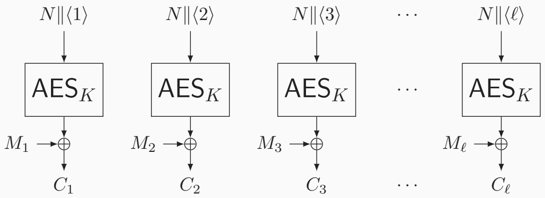
- Stream-based encryption mode
- Fully parallelizable (encryption and decryption) and extremely simple
- Decryption needs no E_K^{-1}

Security

- "Hopefully" secure as long as N is never repeated and E_K is a secure PRP
- Let us investigate that!

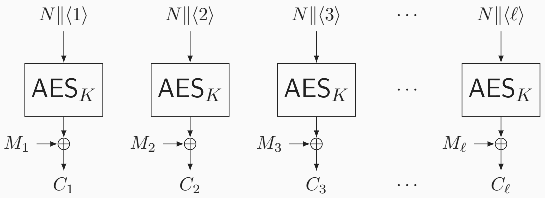
Security of Counter Mode Based on AES

- Let us consider counter mode based on AES: CTR[AES_K]



Security of Counter Mode Based on AES

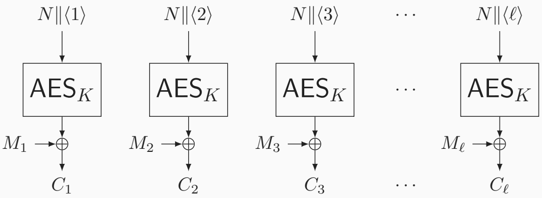
- Let us consider counter mode based on AES: CTR[AES_K]



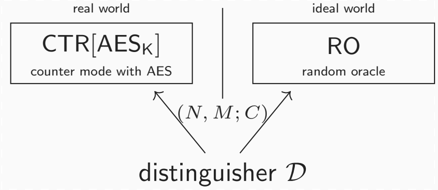
- We focus on the keystream generation portion

Security of Counter Mode Based on AES

- Let us consider counter mode based on AES: CTR[AES$_K$]

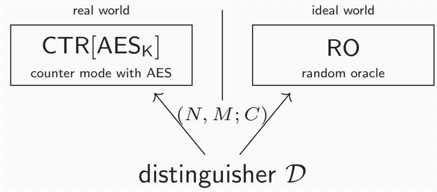


- We focus on the keystream generation portion
- Assumptions
  - Distinguisher never repeats nonce N
  - AES itself is sufficiently secure: $\mathbf{Adv}^{\mathrm{prp}}_{\mathsf{AES}}(q,t)$ is small



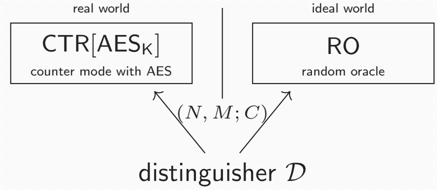
- Two oracles: CTR[AES$_K$] (for secret key K) and RO (secret)

Security of Counter Mode Based on AES: Model

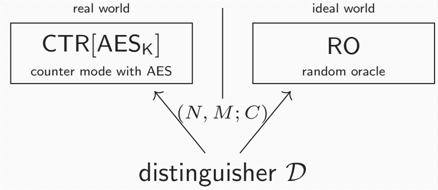


- Two oracles: CTR[AES$_K$] (for secret key K) and RO (secret)
- Distinguisher $\mathcal{D}$ has query access to one of these

Security of Counter Mode Based on AES: Model



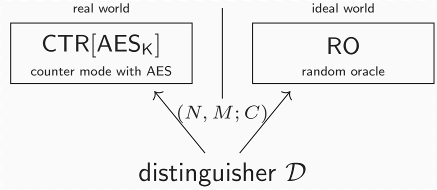
- Two oracles: $\text{CTR}[\text{AES}_K]$ (for secret key K) and RO (secret)
- Distinguisher $\mathcal{D}$ has query access to one of these
- $\mathcal{D}$ tries to determine which oracle it communicates with



- Two oracles: CTR[AES$_K$] (for secret key K) and RO (secret)
- Distinguisher $\mathcal{D}$ has query access to one of these
- $\mathcal{D}$ tries to determine which oracle it communicates with
- Its advantage is defined as:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[AES]}}(\mathcal{D}) = \Delta_{\mathcal{D}}\left(\mathsf{CTR[AES}_K\mathsf{]}\ ;\ \mathsf{RO}\right) = \left|\mathbf{Pr}\left(\mathcal{D}^{\mathsf{CTR[AES}_K\mathsf{]}} = 1\right) - \mathbf{Pr}\left(\mathcal{D}^{\mathsf{RO}} = 1\right)\right|$$

Security of Counter Mode Based on AES: Model

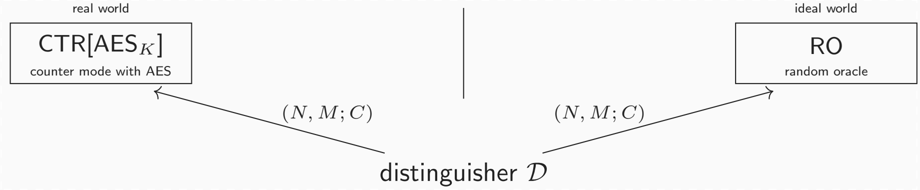


- Two oracles: CTR[AES$_K$] (for secret key K) and RO (secret)
- Distinguisher $\mathcal{D}$ has query access to one of these
- $\mathcal{D}$ tries to determine which oracle it communicates with
- Its advantage is defined as:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[AES]}}(\mathcal{D}) = \Delta_{\mathcal{D}}\left(\mathsf{CTR}[\mathsf{AES}_K]\;;\;\mathsf{RO}\right) = \left|\mathbf{Pr}\left(\mathcal{D}^{\mathsf{CTR}[\mathsf{AES}_K]} = 1\right) - \mathbf{Pr}\left(\mathcal{D}^{\mathsf{RO}} = 1\right)\right|$$

- $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[AES]}}(q, t)$: maximum advantage over any $\mathcal{D}$ with q/t blocks/time

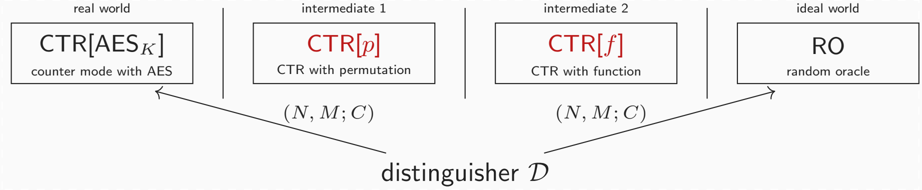
Proof: Overview



- For any (fixed) distinguisher $\mathcal{D}$ (later, we supremize over all), we have to bound:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[AES]}}(\mathcal{D}) = \Delta_{\mathcal{D}}\left(\mathsf{CTR[AES}_K\mathsf{]}\ ;\ \mathsf{RO}\right) = \left|\mathbf{Pr}\left(\mathcal{D}^{\mathsf{CTR[AES}_K\mathsf{]}} = 1\right) - \mathbf{Pr}\left(\mathcal{D}^{\mathsf{RO}} = 1\right)\right|$$

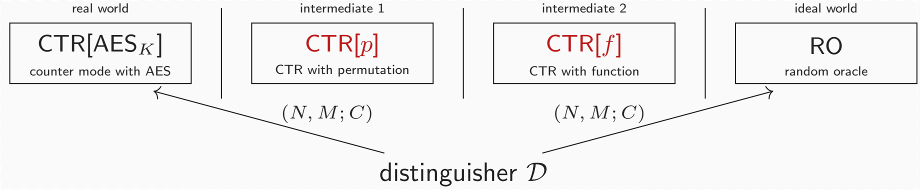
Proof: Overview



- For any (fixed) distinguisher $\mathcal{D}$ (later, we supremize over all), we have to bound:

$$\mathbf{Adv}^{\text{prf}}_{\text{CTR[AES]}}(\mathcal{D}) = \Delta_{\mathcal{D}}\left(\text{CTR[AES}_K] \; ; \; \text{RO}\right) = \left|\mathbf{Pr}\left(\mathcal{D}^{\text{CTR[AES}_K]} = 1\right) - \mathbf{Pr}\left(\mathcal{D}^{\text{RO}} = 1\right)\right|$$

- We add intermediate worlds CTR[p] and CTR[f] for random p and f



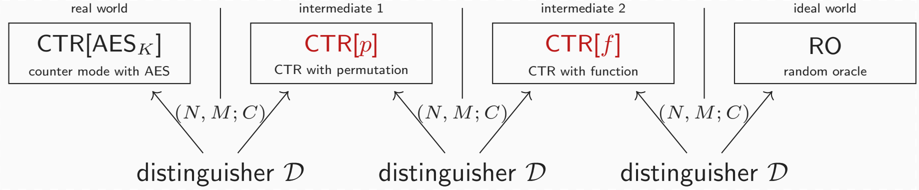
- For any (fixed) distinguisher $\mathcal{D}$ (later, we supremize over all), we have to bound:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[AES]}}(\mathcal{D}) = \Delta_{\mathcal{D}}\left(\mathsf{CTR}[\mathsf{AES}_K]\,;\ \mathsf{RO}\right) = \left|\mathbf{Pr}\left(\mathcal{D}^{\mathsf{CTR}[\mathsf{AES}_K]} = 1\right) - \mathbf{Pr}\left(\mathcal{D}^{\mathsf{RO}} = 1\right)\right|$$

- We add intermediate worlds $\mathsf{CTR}[p]$ and $\mathsf{CTR}[f]$ for random p and f
- By the triangle inequality:

$$\Delta_{\mathcal{D}}\left(\mathsf{CTR}[\mathsf{AES}_K]\,;\ \mathsf{RO}\right) \leq \Delta_{\mathcal{D}}\left(\mathsf{CTR}[\mathsf{AES}_K]\,;\ \mathsf{CTR}[p]\right) + \Delta_{\mathcal{D}}\left(\mathsf{CTR}[p]\,;\ \mathsf{CTR}[f]\right) + \Delta_{\mathcal{D}}\left(\mathsf{CTR}[f]\,;\ \mathsf{RO}\right)$$

Proof: Overview



- For any (fixed) distinguisher $\mathcal{D}$ (later, we supremize over all), we have to bound:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[AES]}}(\mathcal{D}) = \Delta_{\mathcal{D}}\left(\mathsf{CTR[AES}_K\mathsf{]}\ ;\ \mathsf{RO}\right) = \left|\mathbf{Pr}\left(\mathcal{D}^{\mathsf{CTR[AES}_K\mathsf{]}} = 1\right) - \mathbf{Pr}\left(\mathcal{D}^{\mathsf{RO}} = 1\right)\right|$$

- We add intermediate worlds CTR[p] and CTR[f] for random p and f
- By the triangle inequality:

$$\Delta_{\mathcal{D}}\left(\mathsf{CTR[AES}_K\mathsf{]}\ ;\ \mathsf{RO}\right) \leq \Delta_{\mathcal{D}}\left(\mathsf{CTR[AES}_K\mathsf{]}\ ;\ \mathsf{CTR}[p]\right) + \Delta_{\mathcal{D}}\left(\mathsf{CTR}[p]\ ;\ \mathsf{CTR}[f]\right) + \Delta_{\mathcal{D}}\left(\mathsf{CTR}[f]\ ;\ \mathsf{RO}\right)$$

Proof: From CTR[AES_K] to CTR[p]

- $\mathcal{D}$'s goal: distinguish CTR[AES_K] from CTR[p]

Proof: From $\mathrm{CTR}[\mathrm{AES}_K]$ to $\mathrm{CTR}[p]$

- $\mathcal{D}$'s goal: distinguish $\mathrm{CTR}[\mathrm{AES}_K]$ from $\mathrm{CTR}[p]$
- We replace $\mathcal{D}$ by a distinguisher $\mathcal{D}'$ that has more power

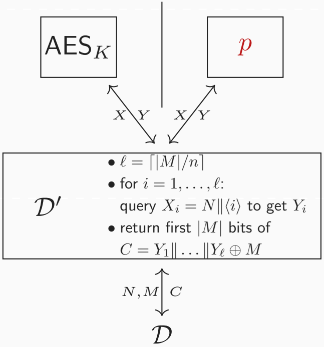
Proof: From CTR[AES$_K$] to CTR[p]

- $\mathcal{D}$'s goal: distinguish CTR[AES$_K$] from CTR[p]
- We replace $\mathcal{D}$ by a distinguisher $\mathcal{D}'$ that has more power
- $\mathcal{D}'$'s goal: distinguish AES$_K$ from p

Proof: From CTR[AES_K] to CTR[p]

- $\mathcal{D}$'s goal: distinguish CTR[AES_K] from CTR[p]
- We replace $\mathcal{D}$ by a distinguisher $\mathcal{D}'$ that has more power
- $\mathcal{D}'$'s goal: distinguish AES_K from p

- $\mathcal{D}'$ simulates the oracles of $\mathcal{D}$:
- Once $\mathcal{D}$ makes its final guess, $\mathcal{D}'$ makes the same guess



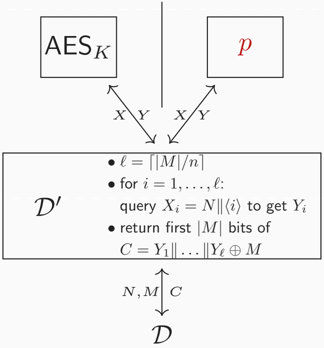
Proof: From CTR[AES_K] to CTR[p]

- $\mathcal{D}$'s goal: distinguish CTR[AES_K] from CTR[p]
- We replace $\mathcal{D}$ by a distinguisher $\mathcal{D}'$ that has more power
- $\mathcal{D}'$'s goal: distinguish AES_K from p

- $\mathcal{D}'$ simulates the oracles of $\mathcal{D}$:
- Once $\mathcal{D}$ makes its final guess, $\mathcal{D}'$ makes the same guess

- $\mathcal{D}'$ success probability turns out to be at least that of $\mathcal{D}$:

$$\Delta_{\mathcal{D}}\left(\text{CTR}[\text{AES}_K]\ ;\ \text{CTR}[p]\right) \leq \Delta_{\mathcal{D}'}\left(\text{AES}_K\ ;\ p\right)$$



Proof: From CTR[AES_K] to CTR[p]

- $\mathcal{D}$'s goal: distinguish CTR[AES_K] from CTR[p]
- We replace $\mathcal{D}$ by a distinguisher $\mathcal{D}'$ that has more power
- $\mathcal{D}'$'s goal: distinguish AES_K from p

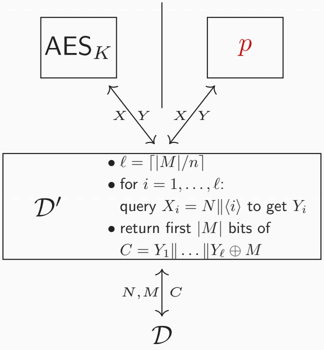
- $\mathcal{D}'$ simulates the oracles of $\mathcal{D}$:
- Once $\mathcal{D}$ makes its final guess, $\mathcal{D}'$ makes the same guess

- $\mathcal{D}'$ success probability turns out to be at least that of $\mathcal{D}$:

  $\Delta_{\mathcal{D}}\left(\text{CTR}[\text{AES}_K]\ ;\ \text{CTR}[p]\right) \leq \Delta_{\mathcal{D}'}\left(\text{AES}_K\ ;\ p\right)$
- But we have seen this distance before:

  $\Delta_{\mathcal{D}'}\left(\text{AES}_K\ ;\ p\right) = \mathbf{Adv}^{\text{prp}}_{\text{AES}}(\mathcal{D}') \leq \mathbf{Adv}^{\text{prp}}_{\text{AES}}(q, t')$

  (t' slightly larger than t)



Proof: From CTR$[p]$ to CTR$[f]$ (1/2)

- $\mathcal{D}$'s goal: distinguish CTR$[p]$ from CTR$[f]$

Proof: From $\mathsf{CTR}[p]$ to $\mathsf{CTR}[f]$ (1/2)

- $\mathcal{D}$'s goal: distinguish $\mathsf{CTR}[p]$ from $\mathsf{CTR}[f]$
- We replace $\mathcal{D}$ by a distinguisher $\mathcal{D}'$ that has more power

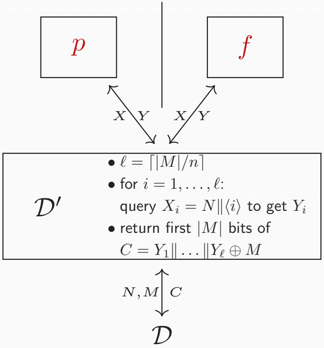
Proof: From CTR$[p]$ to CTR$[f]$ (1/2)

- $\mathcal{D}$'s goal: distinguish CTR$[p]$ from CTR$[f]$
- We replace $\mathcal{D}$ by a distinguisher $\mathcal{D}'$ that has more power
- $\mathcal{D}'$'s goal: distinguish p from f

Proof: From CTR[p] to CTR[f] (1/2)

- $\mathcal{D}$'s goal: distinguish CTR[p] from CTR[f]
- We replace $\mathcal{D}$ by a distinguisher $\mathcal{D}'$ that has more power
- $\mathcal{D}'$'s goal: distinguish p from f

- $\mathcal{D}'$ simulates the oracles of $\mathcal{D}$:
- Once $\mathcal{D}$ makes its final guess, $\mathcal{D}'$ makes the same guess



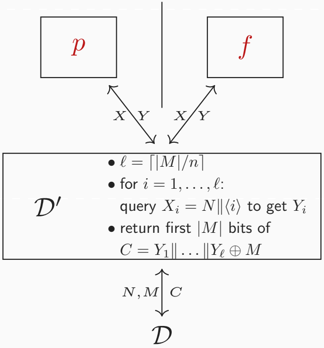
Proof: From CTR[p] to CTR[f] (1/2)

- $\mathcal{D}$'s goal: distinguish CTR[p] from CTR[f]
- We replace $\mathcal{D}$ by a distinguisher $\mathcal{D}'$ that has more power
- $\mathcal{D}'$'s goal: distinguish p from f

- $\mathcal{D}'$ simulates the oracles of $\mathcal{D}$:
- Once $\mathcal{D}$ makes its final guess, $\mathcal{D}'$ makes the same guess

- $\mathcal{D}'$ success probability turns out to be at least that of $\mathcal{D}$:

$$\Delta_{\mathcal{D}}(\text{CTR}[p]\ ;\ \text{CTR}[f]) \leq \Delta_{\mathcal{D}'}(p\ ;\ f)$$

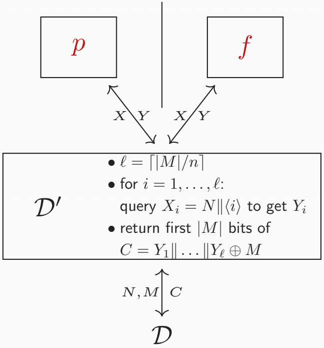


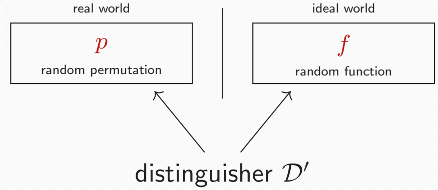
Proof: From CTR$[p]$ to CTR$[f]$ (1/2)

- $\mathcal{D}$'s goal: distinguish CTR$[p]$ from CTR$[f]$
- We replace $\mathcal{D}$ by a distinguisher $\mathcal{D}'$ that has more power
- $\mathcal{D}'$'s goal: distinguish p from f

- $\mathcal{D}'$ simulates the oracles of $\mathcal{D}$:
- Once $\mathcal{D}$ makes its final guess, $\mathcal{D}'$ makes the same guess

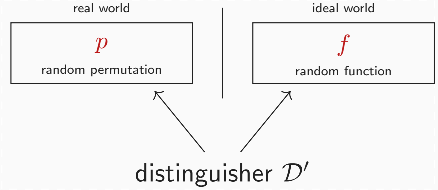
- $\mathcal{D}'$ success probability turns out to be at least that of $\mathcal{D}$:
  $\Delta_{\mathcal{D}}(\text{CTR}[p]\ ;\ \text{CTR}[f]) \leq \Delta_{\mathcal{D}'}(p\ ;\ f)$
- This is a well-known distance, called the RP-RF switch



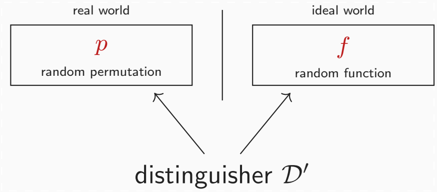


- Distinguisher $\mathcal{D}'$ gets q random n-bit samples:
  - real world: without replacement
  - ideal world: with replacement

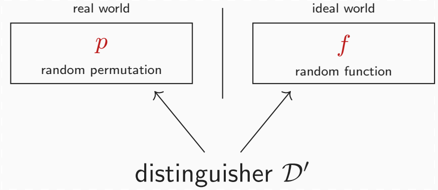
Proof: From CTR$[p]$ to CTR$[f]$ (2/2)



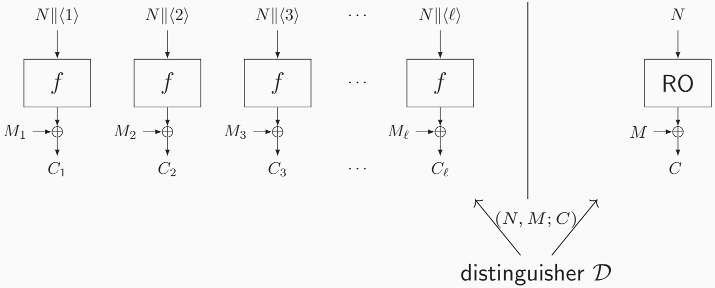
- Distinguisher $\mathcal{D}'$ gets q random n-bit samples:
  - real world: without replacement
  - ideal world: with replacement
- The two worlds can only be distinguished if f ever outputs colliding samples



- Distinguisher $\mathcal{D}'$ gets q random n-bit samples:
  - real world: without replacement
  - ideal world: with replacement
- The two worlds can only be distinguished if f ever outputs colliding samples
- This happens with probability at most $\binom{q}{2}/2^n$

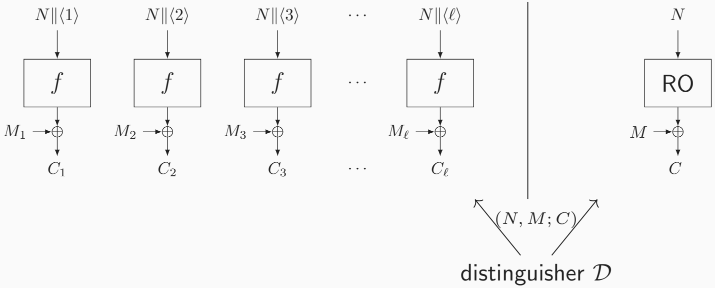


- Distinguisher $\mathcal{D}'$ gets q random n-bit samples:
    - real world: without replacement
    - ideal world: with replacement
- The two worlds can only be distinguished if f ever outputs colliding samples
- This happens with probability at most $\binom{q}{2}/2^n$
- Hence: $\Delta_{\mathcal{D}'}(p\ ;\ f) \leq \binom{q}{2}/2^n$



- In real world: f is a random function that is never evaluated for repeated $N\|\langle i\rangle$
- In ideal world: RO is a random oracle that is never evaluated for repeated N

Proof: From CTR$[f]$ to RO



- In real world: f is a random function that is never evaluated for repeated $N\|\langle i\rangle$
- In ideal world: RO is a random oracle that is never evaluated for repeated N
- Hence: $\Delta_{\mathcal{D}}\left(\text{CTR}[f]\ ;\ \text{RO}\right) = 0$

Proof: Conclusion

- Recall goal: bounding $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[AES]}}(\mathcal{D})$ for any $\mathcal{D}$ querying q blocks in t time

Proof: Conclusion

- Recall goal: bounding $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[AES]}}(\mathcal{D})$ for any $\mathcal{D}$ querying q blocks in t time
- From the triangle inequality and bounds on the three individual terms:

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[AES]}}(\mathcal{D}) &= \Delta_{\mathcal{D}}\left(\mathsf{CTR[AES}_K\mathsf{]}\ ;\ \mathsf{RO}\right) \\
&\leq \Delta_{\mathcal{D}}\left(\mathsf{CTR[AES}_K\mathsf{]}\ ;\ \mathsf{CTR}[p]\right) + \Delta_{\mathcal{D}}\left(\mathsf{CTR}[p]\ ;\ \mathsf{CTR}[f]\right) + \Delta_{\mathcal{D}}\left(\mathsf{CTR}[f]\ ;\ \mathsf{RO}\right) \\
&\leq \mathbf{Adv}^{\mathrm{prp}}_{\mathsf{AES}}(q, t') + \binom{q}{2}/2^n + 0
\end{aligned}$$

- Recall goal: bounding $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[AES]}}(\mathcal{D})$ for any $\mathcal{D}$ querying q blocks in t time
- From the triangle inequality and bounds on the three individual terms:

$$
\begin{aligned}
\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[AES]}}(\mathcal{D}) &= \Delta_{\mathcal{D}}\left(\mathsf{CTR}[\mathsf{AES}_K]\,;\,\mathsf{RO}\right)\\
&\le \Delta_{\mathcal{D}}\left(\mathsf{CTR}[\mathsf{AES}_K]\,;\,\mathsf{CTR}[p]\right) + \Delta_{\mathcal{D}}\left(\mathsf{CTR}[p]\,;\,\mathsf{CTR}[f]\right) + \Delta_{\mathcal{D}}\left(\mathsf{CTR}[f]\,;\,\mathsf{RO}\right)\\
&\le \mathbf{Adv}^{\mathrm{prp}}_{\mathsf{AES}}(q,t') + \binom{q}{2}/2^n + 0
\end{aligned}
$$

- As this reasoning holds for all distinguishers $\mathcal{D}$ querying q blocks in t time, we obtain:

$$
\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[AES]}}(q,t) \le \mathbf{Adv}^{\mathrm{prp}}_{\mathsf{AES}}(q,t') + \binom{q}{2}/2^n
$$

Beyond Birthday Bound Security

Birthday Paradox

For a random selection of 23 people, with a probability at least 50% two of them share the same birthday

For a random selection of 23 people, with a probability at least 50% two of them share the same birthday

General Birthday Paradox

- Consider space $\mathcal{S} = \{0,1\}^n$
- Randomly draw q elements from $\mathcal{S}$
- Expected number of collisions:

$$\mathbf{Ex}\left[\text{collisions}\right] = \binom{q}{2}/2^n$$

For a random selection of 23 people, with a probability at least 50% two of them share the same birthday

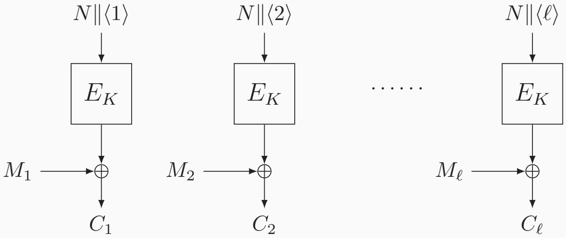
General Birthday Paradox

- Consider space $\mathcal{S} = \{0,1\}^n$
- Randomly draw q elements from $\mathcal{S}$
- Expected number of collisions:

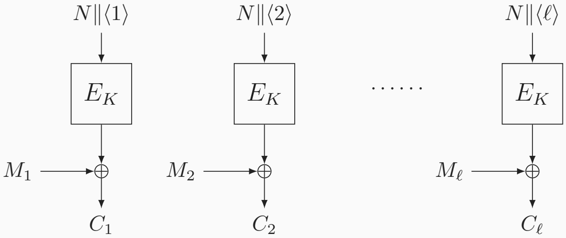
$$\mathbf{Ex}\left[\text{collisions}\right] = \binom{q}{2}/2^n$$

- Important phenomenon in cryptography

Counter Mode Based on Pseudorandom Permutation



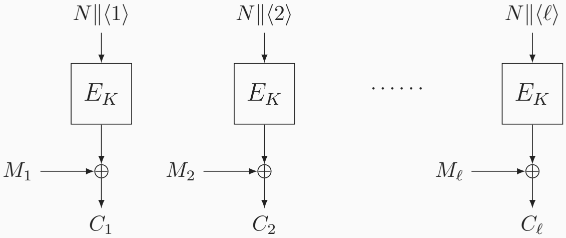
Counter Mode Based on Pseudorandom Permutation



- Security bound:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR}[E]}(q,t) \le \mathbf{Adv}^{\mathrm{prp}}_{E}(q,t') + \binom{q}{2}/2^n$$

Counter Mode Based on Pseudorandom Permutation

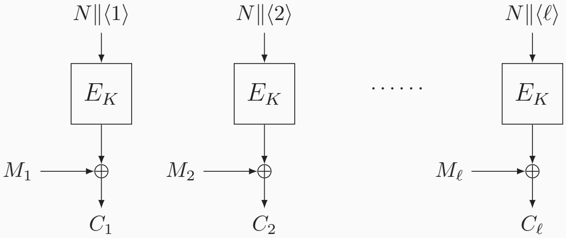


- Security bound:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR}[E]}(q,t) \le \mathbf{Adv}^{\mathrm{prp}}_{E}(q,t') + \binom{q}{2}/2^n$$

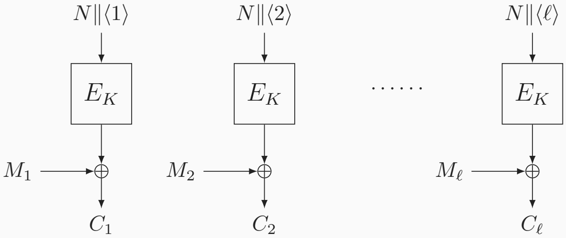
- $\mathsf{CTR}[E]$ is secure as long as:
  - E_K is a secure PRP
  - Number of encrypted blocks $q \ll 2^{n/2}$

Counter Mode Based on Pseudorandom Permutation



- $M_i \oplus C_i$ is distinct for all q blocks
- Unlikely to happen for random string

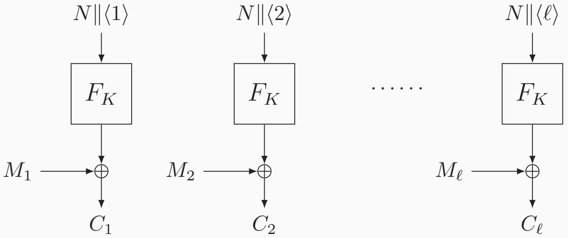
Counter Mode Based on Pseudorandom Permutation



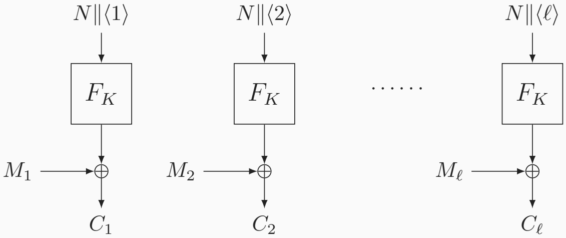
- $M_i \oplus C_i$ is distinct for all q blocks
- Unlikely to happen for random string
- Distinguishing attack in $q \approx 2^{n/2}$ blocks:

$$\binom{q}{2}/2^n \lesssim \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR}[E]}(q,t)$$

Counter Mode Based on Pseudorandom Function

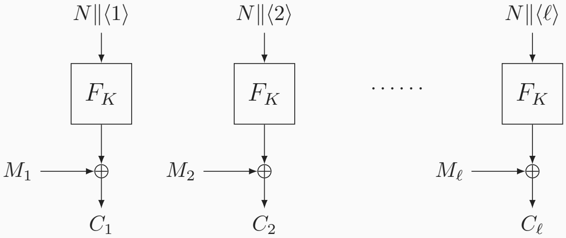


Counter Mode Based on Pseudorandom Function



- Security bound:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR}[F]}(q,t) \leq \mathbf{Adv}^{\mathrm{prf}}_{F}(q,t')$$

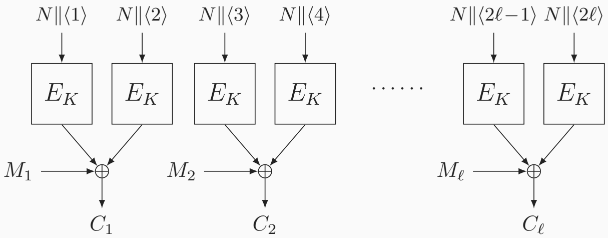


- Security bound:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR}[F]}(q,t) \le \mathbf{Adv}^{\mathrm{prf}}_{F}(q,t')$$

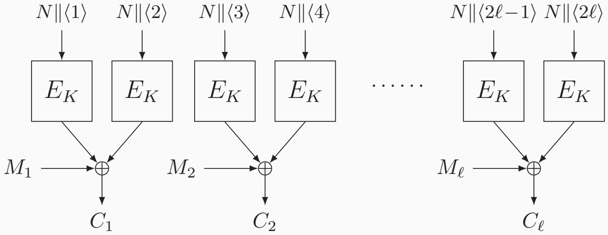
- CTR$[F]$ is secure as long as F_K is a secure PRF
- Birthday bound security loss disappeared

Counter Mode Based on XoP



- Security bound [Pat08a, DHT17]:

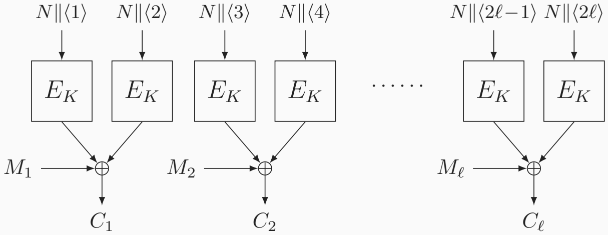
$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[XoP]}}(q,t) \leq \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{XoP}}(q,t')$$



- Security bound [Pat08a, DHT17]:

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[XoP]}}(q,t) &\leq \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{XoP}}(q,t') \\ &\leq \mathbf{Adv}^{\mathrm{prp}}_{E}(2q,t'') + q/2^n\end{aligned}$$

Counter Mode Based on XoP

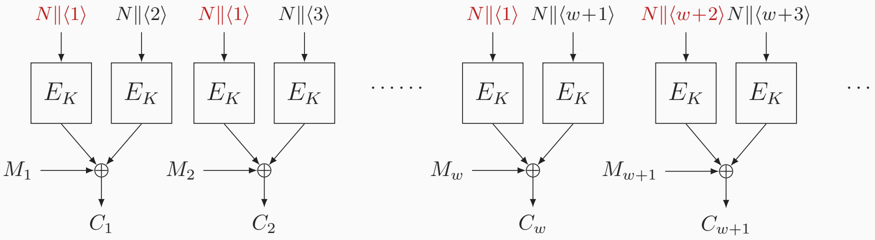


- Security bound [Pat08a, DHT17]:

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR[XoP]}}(q,t) &\leq \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{XoP}}(q,t') \\ &\leq \mathbf{Adv}^{\mathrm{prp}}_{E}(2q,t'') + q/2^n\end{aligned}$$

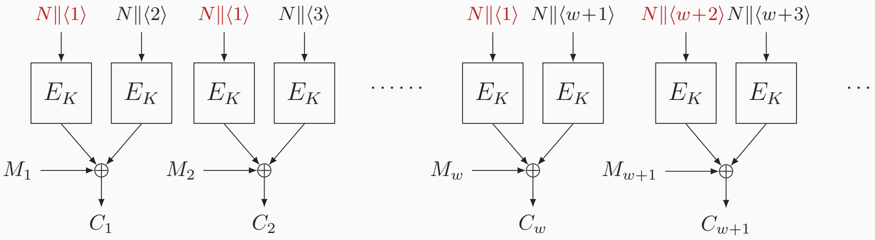
- Beyond birthday bound but 2x as expensive as CTR$[E]$

CENC by Iwata [Iwa06]



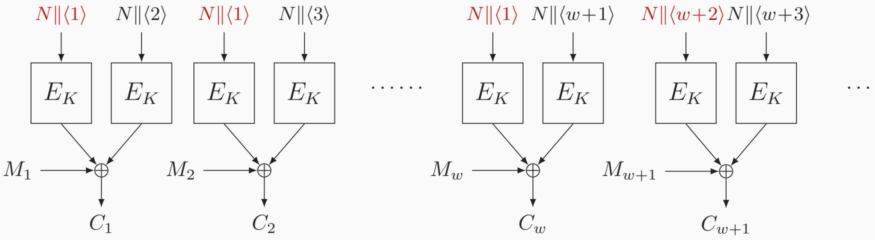
- One subkey used for $w \geq 1$ encryptions

CENC by Iwata [Iwa06]



- One subkey used for $w \geq 1$ encryptions
- Almost as expensive as CTR$[E]$

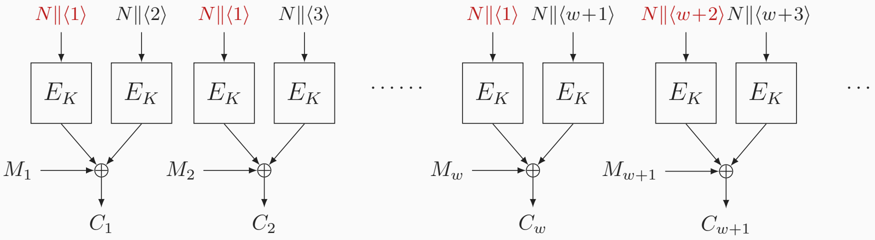
CENC by Iwata [Iwa06]



- One subkey used for $w \geq 1$ encryptions
- Almost as expensive as CTR$[E]$
- Security bound [IMV16]:

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR}[\mathsf{XoP}[w]]}(q,t) &\leq \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{XoP}[w]}(q,t') \\ &\leq \mathbf{Adv}^{\mathrm{prp}}_{E}((w+1)q,t'') + wq/2^n\end{aligned}$$

CENC by Iwata [Iwa06]



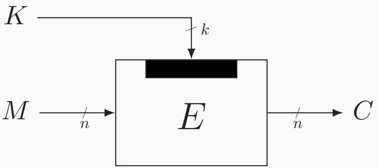
- One subkey used for $w \geq 1$ encryptions
- Almost as expensive as CTR$[E]$
- Security bound [IMV16]:

$$\begin{aligned}\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CTR}[\mathsf{XoP}[w]]}(q,t) &\leq \mathbf{Adv}^{\mathrm{prf}}_{\mathsf{XoP}[w]}(q,t') \\ &\leq \mathbf{Adv}^{\mathrm{prp}}_{E}((w+1)q,t'') + wq/2^n\end{aligned}$$

- Security of XoP and XoP$[w]$ can be proven using mirror theory [Pat03]

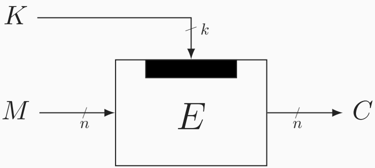
Accordion Modes

Block Ciphers



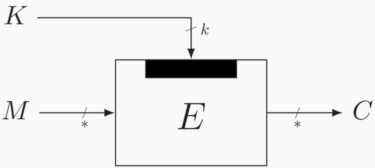
- Message M encrypted to ciphertext C with secret key K
- Fixed block size

Block Ciphers

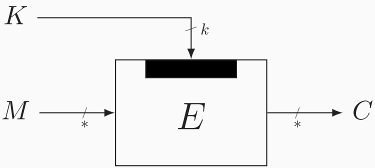


- Message M encrypted to ciphertext C with secret key K
- Fixed block size
- In order to encrypt variable sized messages, we need a mode of operation
  - These modes require a nonce

Wide Block Ciphers

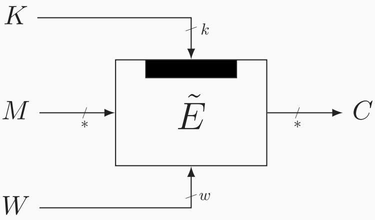


- Alternatively, we can design a wide block cipher
- A wide block cipher is a block cipher with a variable block size



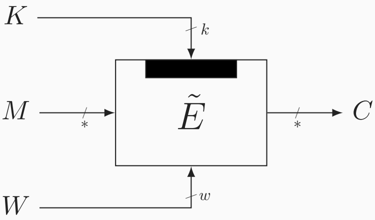
- Alternatively, we can design a wide block cipher
- A wide block cipher is a block cipher with a variable block size
- Every part of the output (ideally) depends on every part of the input

Tweakable Wide Block Ciphers



- A tweakable wide block cipher additionally has a tweak
- Tweak W public, ciphertext completely changes with a different tweak

Tweakable Wide Block Ciphers



- A tweakable wide block cipher additionally has a tweak
- Tweak W public, ciphertext completely changes with a different tweak
- Useful for e.g. disk encryption, where every sector gets its own tweak

NIST's Incentive to Develop Accordion Mode

- March 2024: NIST announced quest for tweakable wide block ciphers
- There is a workshop right now aimed to discuss ideas on requirements, designs, security goals, targets, ...

NIST's Incentive to Develop Accordion Mode

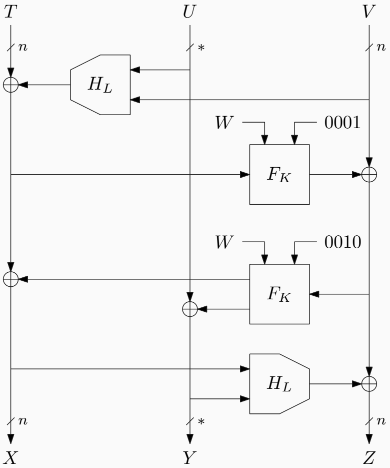
- March 2024: NIST announced quest for tweakable wide block ciphers
- There is a workshop right now aimed to discuss ideas on requirements, designs, security goals, targets, . . .
- Quote from the website:
  NIST plans to develop a new mode of the AES that is a tweakable, variable-input-length-strong pseudorandom permutation (VIL-SPRP) with a reduction proof to the security of the underlying block cipher.

NIST's Incentive to Develop Accordion Mode

- March 2024: NIST announced quest for tweakable wide block ciphers
- There is a workshop right now aimed to discuss ideas on requirements, designs, security goals, targets, . . .
- Quote from the website:
  NIST plans to develop a new mode of the AES that is a tweakable, variable-input-length-strong pseudorandom permutation (VIL-SPRP) with a reduction proof to the security of the underlying block cipher.

Now: high-level idea of our recent proposals

Docked Double Decker [GDM19]



Building Blocks

- F_K: stream cipher
- H_L: universal hash

Construction

- Feistel-like structure
- Outer lanes of fixed size
- Inner lane of variable size

Goals and Hurdles

Goals

- Instantiation using components as used in NIST standardized schemes:
  - AES [DR02, DR20]
  - Operations in binary extension fields, e.g., as in GHASH [MV04]

Goals and Hurdles

Goals

- Instantiation using components as used in NIST standardized schemes:
    - AES [DR02, DR20]
    - Operations in binary extension fields, e.g., as in GHASH [MV04]
- Present birthday bound secure $ddd\text{-}AES$ and beyond birthday bound secure $bbb\text{-}ddd\text{-}AES$ that seamlessly fit NIST's accordion idea

Goals and Hurdles

Goals

- Instantiation using components as used in NIST standardized schemes:
  - AES [DR02, DR20]
  - Operations in binary extension fields, e.g., as in GHASH [MV04]
- Present birthday bound secure $ddd\text{-}AES$ and beyond birthday bound secure $bbb\text{-}ddd\text{-}AES$ that seamlessly fit NIST's accordion idea

Hurdles

- AES is not a tweakable blockcipher
- AES is rather small (circular reasoning?)
- AES in typical stream cipher modes only gives birthday bound security

Efficient Instantiations of Docked Double Decker [DMMT24]

ddd-AES

- H_L instantiated using Polyval (as in GCM-SIV)
- F_K instantiated as variant of CTR: tweak used to randomize inputs to AES_K

Efficient Instantiations of Docked Double Decker [DMMT24]

ddd-AES

- H_L instantiated using Polyval (as in GCM-SIV)
- F_K instantiated as variant of CTR: tweak used to randomize inputs to AES_K

bbb-ddd-AES

- H_L instantiated using Polyval (as in GCM-SIV)
- F_K instantiated as variant of CENC: tweak used to randomize inputs to AES_K

Efficient Instantiations of Docked Double Decker [DMMT24]

ddd-AES

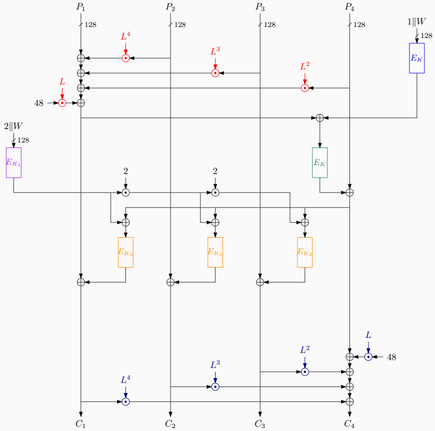
- H_L instantiated using Polyval (as in GCM-SIV)
- F_K instantiated as variant of CTR: tweak used to randomize inputs to AES_K

bbb-ddd-AES

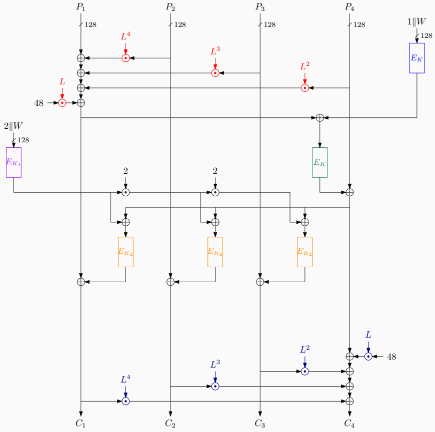
- H_L instantiated using Polyval (as in GCM-SIV)
- F_K instantiated as variant of CENC: tweak used to randomize inputs to AES_K

Instantiations turn out to be very competitive and well parallelizable

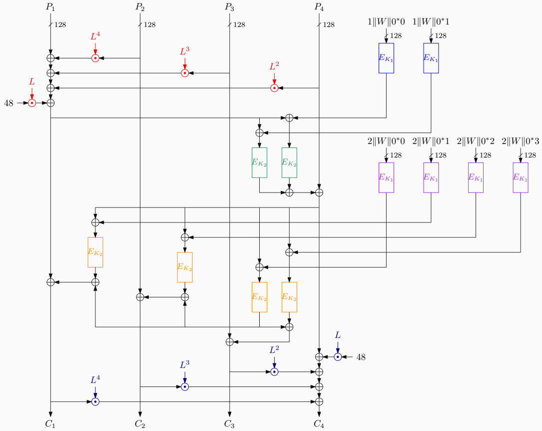
Implementation Design of *ddd-AES* (512-Bit Message)



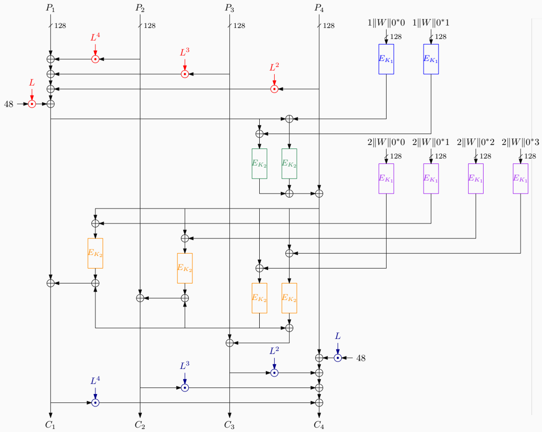
Implementation Design of *ddd-AES* (512-Bit Message)



Implementation Design of $bbb\text{-}ddd\text{-}AES$ (512-Bit Message)



Implementation Design of $bbb\text{-}ddd\text{-}AES$ (512-Bit Message)



Conclusion

Conclusion

Provable Security in Symmetric Cryptography

- Basic modes proved secure using quite simple ideas
- More sophisticated modes require nice tricks in graph theory
- Often this boils down to trying to upper or lower bound solutions

Conclusion

Provable Security in Symmetric Cryptography

- Basic modes proved secure using quite simple ideas
- More sophisticated modes require nice tricks in graph theory
- Often this boils down to trying to upper or lower bound solutions

Current Directions in Provable Security

- Difficulties in beyond birthday bound security
- Accordion modes
- Arithmetization-oriented modes

Conclusion

Provable Security in Symmetric Cryptography

- Basic modes proved secure using quite simple ideas
- More sophisticated modes require nice tricks in graph theory
- Often this boils down to trying to upper or lower bound solutions

Current Directions in Provable Security

- Difficulties in beyond birthday bound security
- Accordion modes
- Arithmetization-oriented modes

Thank you for your attention!

Benoît Cogliati, Avijit Dutta, Mridul Nandi, Jacques Patarin, and Abishanka Saha.
Proof of Mirror Theory for a Wide Range of $\xi_{\max}$.
In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 470–501. Springer, 2023.

Shan Chen and John P. Steinberger.
Tight Security Bounds for Key-Alternating Ciphers.
In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 327–350. Springer, 2014.

Wei Dai, Viet Tung Hoang, and Stefano Tessaro.
Information-Theoretic Indistinguishability via the Chi-Squared Method.
In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 497–523. Springer, 2017.

Christoph Dobraunig, Krystian Matusiewicz, Bart Mennink, and Alexander Tereschenko.
Efficient Instances of Docked Double Decker With AES, and Application to Authenticated Encryption.
Cryptology ePrint Archive, Report 2024/084, 2024.
https://eprint.iacr.org/2024/084.

Joan Daemen and Vincent Rijmen.
The Design of Rijndael: AES - The Advanced Encryption Standard.
Information Security and Cryptography. Springer, 2002.

References iii

Joan Daemen and Vincent Rijmen.
The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition.
Information Security and Cryptography. Springer, 2020.

Aldo Gunsing, Joan Daemen, and Bart Mennink.
Deck-Based Wide Block Cipher Modes and an Exposition of the Blinded Keyed Hashing Model.
IACR Trans. Symmetric Cryptol., 2019(4):1–22, 2019.

Shay Gueron, Adam Langley, and Yehuda Lindell.
AES-GCM-SIV: Specification and Analysis.
Cryptology ePrint Archive, Report 2017/168, 2017.
http://eprint.iacr.org/2017/168.

Tetsu Iwata, Bart Mennink, and Damian Vizár.
CENC is Optimally Secure.
Cryptology ePrint Archive, Report 2016/1087, 2016.
`http://eprint.iacr.org/2016/1087`.

Tetsu Iwata.
New Blockcipher Modes of Operation with Beyond the Birthday Bound Security.
In Matthew J. B. Robshaw, editor, *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers*, volume 4047 of *Lecture Notes in Computer Science*, pages 310–327. Springer, 2006.

Bart Mennink and Samuel Neves.
Encrypted Davies-Meyer and Its Dual: Towards Optimal Security Using Mirror Theory.
In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 556–583. Springer, 2017.

David A. McGrew and John Viega.
The Security and Performance of the Galois/Counter Mode (GCM) of Operation.
In Anne Canteaut and Kapalee Viswanathan, editors, *Progress in Cryptology - INDOCRYPT 2004, 5th International Conference on Cryptology in India, Chennai, India, December 20-22, 2004, Proceedings*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355. Springer, 2004.

Jacques Patarin.
Luby-Rackoff: 7 Rounds Are Enough for $2^{n(1-\epsilon)}$ Security.
In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 513–529. Springer, 2003.

Jacques Patarin.
A Proof of Security in $O(2^n)$ for the Xor of Two Random Permutations.
In Reihaneh Safavi-Naini, editor, *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings*, volume 5155 of *Lecture Notes in Computer Science*, pages 232–248. Springer, 2008.

Jacques Patarin.
The "Coefficients H" Technique.
In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, volume 5381 of *Lecture Notes in Computer Science*, pages 328–345. Springer, 2008.

Phillip Rogaway.
Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC.
In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, volume 3329 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2004.

Mirror Theory (Intuition)

System of Equations

- Consider r distinct unknowns $\mathcal{P} = \{P_1, \ldots, P_r\}$
- Consider a system of q equations of the form:

$$
\begin{aligned}
P_{a_1} \oplus P_{b_1} &= \lambda_1 \\
P_{a_2} \oplus P_{b_2} &= \lambda_2 \\
&\vdots \\
P_{a_q} \oplus P_{b_q} &= \lambda_q
\end{aligned}
$$

for some surjection $\varphi : \{a_1, b_1, \ldots, a_q, b_q\} \to \{1, \ldots, r\}$

System of Equations

- Consider r distinct unknowns $\mathcal{P} = \{P_1, \ldots, P_r\}$
- Consider a system of q equations of the form:

$$\begin{aligned} P_{a_1} \oplus P_{b_1} &= \lambda_1 \\ P_{a_2} \oplus P_{b_2} &= \lambda_2 \\ &\vdots \\ P_{a_q} \oplus P_{b_q} &= \lambda_q \end{aligned}$$

for some surjection $\varphi : \{a_1, b_1, \ldots, a_q, b_q\} \to \{1, \ldots, r\}$

Goal

- Lower bound on the number of solutions to $\mathcal{P}$

Mirror Theory

- Extremely powerful lower bound

Mirror Theory

- Extremely powerful lower bound
- First introduced by Patarin in 2003 [Pat03]

Mirror Theory

- Extremely powerful lower bound
- First introduced by Patarin in 2003 [Pat03]
- Has remained rather unknown since introduction until 2017 [MN17]

Mirror Theory

- Extremely powerful lower bound
- First introduced by Patarin in 2003 [Pat03]
- Has remained rather unknown since introduction until 2017 [MN17]
- Has been debated since

Mirror Theory

- Extremely powerful lower bound
- First introduced by Patarin in 2003 [Pat03]
- Has remained rather unknown since introduction until 2017 [MN17]
- Has been debated since
- Conclusive proof given in 2023 [CDN+23]

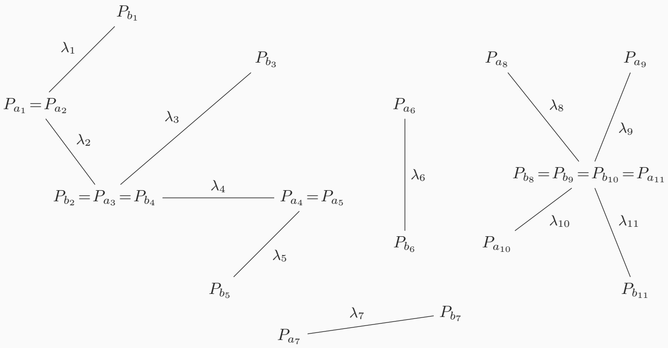
- Extremely powerful lower bound
- First introduced by Patarin in 2003 [Pat03]
- Has remained rather unknown since introduction until 2017 [MN17]
- Has been debated since
- Conclusive proof given in 2023 [CDN+23]

Now: graph-based intuition behind mirror theory

System of Equations

- r distinct unknowns $\mathcal{P} = \{P_1, \ldots, P_r\}$
- System of equations $P_{a_i} \oplus P_{b_i} = \lambda_i$
- Surjection $\varphi : \{a_1, b_1, \ldots, a_q, b_q\} \to \{1, \ldots, r\}$

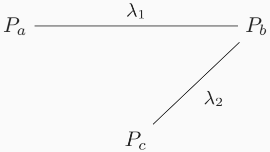
Graph Based View



Mirror Theory: Toy Example 1

- System of equations:

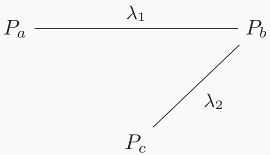
$$P_a \oplus P_b = \lambda_1$$
$$P_b \oplus P_c = \lambda_2$$



Mirror Theory: Toy Example 1

- System of equations:

$$P_a \oplus P_b = \lambda_1$$
$$P_b \oplus P_c = \lambda_2$$



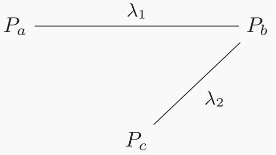
If $\lambda_1 = 0$ or $\lambda_2 = 0$ or $\lambda_1 = \lambda_2$

- Contradiction: $P_a = P_b$ or $P_b = P_c$ or $P_a = P_c$
- Scheme is degenerate

Mirror Theory: Toy Example 1

- System of equations:

$$P_a \oplus P_b = \lambda_1$$
$$P_b \oplus P_c = \lambda_2$$



If $\lambda_1 = 0$ or $\lambda_2 = 0$ or $\lambda_1 = \lambda_2$

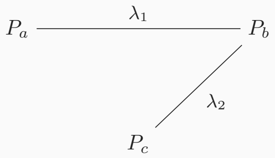
- Contradiction: $P_a = P_b$ or $P_b = P_c$ or $P_a = P_c$
- Scheme is degenerate

If $\lambda_1, \lambda_2 \neq 0$ and $\lambda_1 \neq \lambda_2$

- 2^n choices for P_a

- System of equations:

$$P_a \oplus P_b = \lambda_1$$
$$P_b \oplus P_c = \lambda_2$$



If $\lambda_1 = 0$ or $\lambda_2 = 0$ or $\lambda_1 = \lambda_2$

- Contradiction: $P_a = P_b$ or $P_b = P_c$ or $P_a = P_c$
- Scheme is degenerate

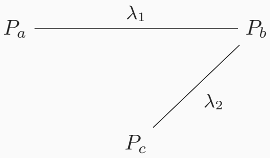
If $\lambda_1, \lambda_2 \neq 0$ and $\lambda_1 \neq \lambda_2$

- 2^n choices for P_a
- Fixes $P_b = \lambda_1 \oplus P_a$ (which is $\neq P_a$ as desired)

Mirror Theory: Toy Example 1

- System of equations:

$$P_a \oplus P_b = \lambda_1$$
$$P_b \oplus P_c = \lambda_2$$



If $\lambda_1 = 0$ or $\lambda_2 = 0$ or $\lambda_1 = \lambda_2$

- Contradiction: $P_a = P_b$ or $P_b = P_c$ or $P_a = P_c$
- Scheme is degenerate

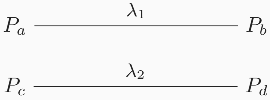
If $\lambda_1, \lambda_2 \neq 0$ and $\lambda_1 \neq \lambda_2$

- 2^n choices for P_a
- Fixes $P_b = \lambda_1 \oplus P_a$ (which is $\neq P_a$ as desired)
- Fixes $P_c = \lambda_2 \oplus P_b$ (which is $\neq P_a, P_b$ as desired)

Mirror Theory: Toy Example 2

- System of equations:

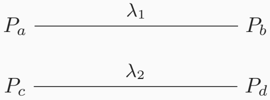
$$P_a \oplus P_b = \lambda_1$$
$$P_c \oplus P_d = \lambda_2$$



Mirror Theory: Toy Example 2

- System of equations:

$$P_a \oplus P_b = \lambda_1$$
$$P_c \oplus P_d = \lambda_2$$

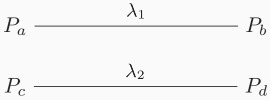


If $\lambda_1 = 0$ or $\lambda_2 = 0$

- Contradiction: $P_a = P_b$ or $P_b = P_c$
- Scheme is degenerate

- System of equations:

$$P_a \oplus P_b = \lambda_1$$
$$P_c \oplus P_d = \lambda_2$$



If $\lambda_1 = 0$ or $\lambda_2 = 0$

- Contradiction: $P_a = P_b$ or $P_b = P_c$
- Scheme is degenerate

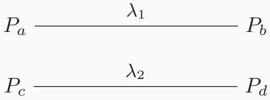
If $\lambda_1, \lambda_2 \neq 0$

- 2^n choices for P_a (which fixes P_b)

Mirror Theory: Toy Example 2

- System of equations:

$$P_a \oplus P_b = \lambda_1$$
$$P_c \oplus P_d = \lambda_2$$



If $\lambda_1 = 0$ or $\lambda_2 = 0$

- Contradiction: $P_a = P_b$ or $P_b = P_c$
- Scheme is degenerate

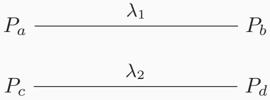
If $\lambda_1, \lambda_2 \neq 0$

- 2^n choices for P_a (which fixes P_b)
- For P_c and P_d we require
  - $P_c \neq P_a, P_b$
  - $P_d = \lambda_2 \oplus P_c \neq P_a, P_b$

Mirror Theory: Toy Example 2

- System of equations:

$$P_a \oplus P_b = \lambda_1$$
$$P_c \oplus P_d = \lambda_2$$



If $\lambda_1 = 0$ or $\lambda_2 = 0$

- Contradiction: $P_a = P_b$ or $P_b = P_c$
- Scheme is degenerate

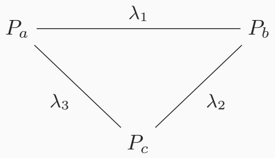
If $\lambda_1, \lambda_2 \neq 0$

- 2^n choices for P_a (which fixes P_b)
- For P_c and P_d we require
  - $P_c \neq P_a, P_b$
  - $P_d = \lambda_2 \oplus P_c \neq P_a, P_b$
- At least $2^n - 4$ choices for P_c (which fixes P_d)

- System of equations:

$$P_a \oplus P_b = \lambda_1$$
$$P_b \oplus P_c = \lambda_2$$
$$P_c \oplus P_a = \lambda_3$$

- Assume $\lambda_i \neq 0$ and $\lambda_i \neq \lambda_j$

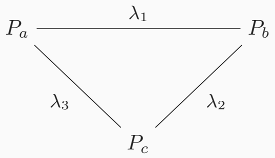


Mirror Theory: Toy Example 3

- System of equations:

$$P_a \oplus P_b = \lambda_1$$
$$P_b \oplus P_c = \lambda_2$$
$$P_c \oplus P_a = \lambda_3$$

- Assume $\lambda_i \neq 0$ and $\lambda_i \neq \lambda_j$



If $\boldsymbol{\lambda_1 \oplus \lambda_2 \oplus \lambda_3 \neq 0}$

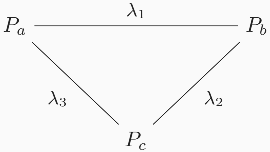
- Contradiction: equations sum to $0 = \lambda_1 \oplus \lambda_2 \oplus \lambda_3$
- Scheme contains a circle

Mirror Theory: Toy Example 3

- System of equations:

$$P_a \oplus P_b = \lambda_1$$
$$P_b \oplus P_c = \lambda_2$$
$$P_c \oplus P_a = \lambda_3$$

- Assume $\lambda_i \neq 0$ and $\lambda_i \neq \lambda_j$



If $\boldsymbol{\lambda_1 \oplus \lambda_2 \oplus \lambda_3 \neq 0}$

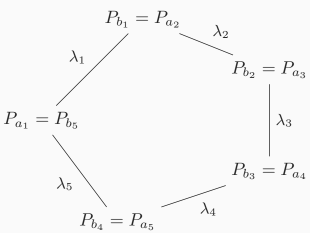
- Contradiction: equations sum to $0 = \lambda_1 \oplus \lambda_2 \oplus \lambda_3$
- Scheme contains a circle

If $\boldsymbol{\lambda_1 \oplus \lambda_2 \oplus \lambda_3 = 0}$

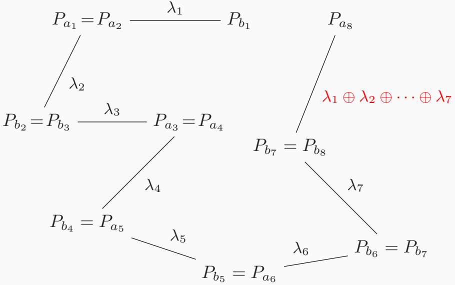
- One redundant equation, no contradiction
- Removing this equation brings us back at toy example 1

Mirror Theory: Two Problematic Cases

Circle



Degeneracy



System of Equations

- r distinct unknowns $\mathcal{P} = \{P_1, \ldots, P_r\}$
- System of equations $P_{a_i} \oplus P_{b_i} = \lambda_i$
- Surjection $\varphi : \{a_1, b_1, \ldots, a_q, b_q\} \to \{1, \ldots, r\}$

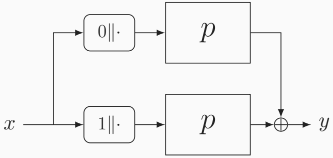
Main Result [CDN+23]

If the system of equations is circle-free and non-degenerate, the number of solutions to $\mathcal{P}$ is at least

$$\frac{(2^n)_r}{2^{nq}}$$

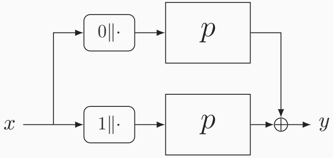
provided the maximum tree size ξ satisfies $\xi^2 \lesssim \min\{2^n/(12r), 2^{n/2}/n\}$

Mirror Theory Applied to XoP



General Setting

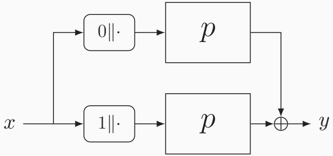
- Distinguisher gets transcript $\tau = \{(x_1, y_1), \ldots, (x_q, y_q)\}$



General Setting

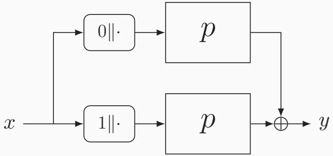
- Distinguisher gets transcript $\tau = \{(x_1, y_1), \ldots, (x_q, y_q)\}$
- Each tuple relates to $0\|x_i \mapsto p(0\|x_i) =: P_{a_i}$ and $1\|x_i \mapsto p(1\|x_i) =: P_{b_i}$

Mirror Theory Applied to XoP



General Setting

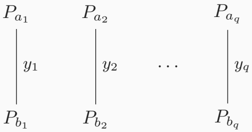
- Distinguisher gets transcript $\tau = \{(x_1, y_1), \ldots, (x_q, y_q)\}$
- Each tuple relates to $0\|x_i \mapsto p(0\|x_i) =: P_{a_i}$ and $1\|x_i \mapsto p(1\|x_i) =: P_{b_i}$
- System of q equations $P_{a_i} \oplus P_{b_i} = y_i$



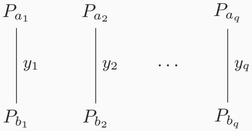
General Setting

- Distinguisher gets transcript $\tau = \{(x_1, y_1), \ldots, (x_q, y_q)\}$
- Each tuple relates to $0\|x_i \mapsto p(0\|x_i) =: P_{a_i}$ and $1\|x_i \mapsto p(1\|x_i) =: P_{b_i}$
- System of q equations $P_{a_i} \oplus P_{b_i} = y_i$
- Inputs to p are all distinct: $2q$ unknowns

Mirror Theory Applied to XoP



Mirror Theory Applied to XoP

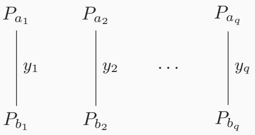


Applying Mirror Theory

- Circle-free: no collisions in inputs to p
- Non-degenerate: provided that $y_i \neq 0$ $(\forall i)$

  $\longrightarrow$ Call this a bad transcript
- Maximum tree size 2

Mirror Theory Applied to XoP



Applying Mirror Theory

- Circle-free: no collisions in inputs to p
- Non-degenerate: provided that $y_i \neq 0$ $(\forall i)$

  $\longrightarrow$ Call this a bad transcript
- Maximum tree size 2
- If $q \leq 2^n/96$: at least $\frac{(2^n)_{2q}}{2^{nq}}$ solutions to unknowns

Mirror Theory Applied to XoP

H-Coefficient Technique [Pat08b, CS14]

Let $\varepsilon \geq 0$ be such that for all good transcripts τ:

$$\frac{\mathbf{Pr}\left(\text{XoP gives } \tau\right)}{\mathbf{Pr}\left(f \text{ gives } \tau\right)} \geq 1 - \varepsilon$$

Then, $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{XoP}}(q) \leq \varepsilon + \mathbf{Pr}\left(\text{bad transcript for } f\right)$

H-Coefficient Technique [Pat08b, CS14]

Let $\varepsilon \geq 0$ be such that for all good transcripts τ:

$$\frac{\mathbf{Pr}\left(\text{XoP gives } \tau\right)}{\mathbf{Pr}\left(f \text{ gives } \tau\right)} \geq 1 - \varepsilon$$

Then, $\mathbf{Adv}^{\mathrm{prf}}_{\text{XoP}}(q) \leq \varepsilon + \mathbf{Pr}\left(\text{bad transcript for } f\right)$

- Bad transcript: if $y_i = 0$ for some i
  - $\mathbf{Pr}\left(\text{bad transcript for } f\right) = q/2^n$

H-Coefficient Technique [Pat08b, CS14]

Let $\varepsilon \geq 0$ be such that for all good transcripts τ:

$$\frac{\mathbf{Pr}\left(\text{XoP gives } \tau\right)}{\mathbf{Pr}\left(f \text{ gives } \tau\right)} \geq 1 - \varepsilon$$

Then, $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{XoP}}(q) \leq \varepsilon + \mathbf{Pr}\left(\text{bad transcript for } f\right)$

- Bad transcript: if $y_i = 0$ for some i
  - $\mathbf{Pr}\left(\text{bad transcript for } f\right) = q/2^n$
- For any good transcript:
  - $\mathbf{Pr}\left(\text{XoP gives } \tau\right) \geq \frac{(2^n)_{2q}}{2^{nq}} \cdot \frac{1}{(2^n)_{2q}}$

H-Coefficient Technique [Pat08b, CS14]

Let $\varepsilon \geq 0$ be such that for all good transcripts τ:

$$\frac{\mathbf{Pr}\left(\text{XoP gives } \tau\right)}{\mathbf{Pr}\left(f \text{ gives } \tau\right)} \geq 1 - \varepsilon$$

Then, $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{XoP}}(q) \leq \varepsilon + \mathbf{Pr}\left(\text{bad transcript for } f\right)$

- Bad transcript: if $y_i = 0$ for some i
  - $\mathbf{Pr}\left(\text{bad transcript for } f\right) = q/2^n$
- For any good transcript:
  - $\mathbf{Pr}\left(\text{XoP gives } \tau\right) \geq \frac{(2^n)_{2q}}{2^{nq}} \cdot \frac{1}{(2^n)_{2q}}$
  - $\mathbf{Pr}\left(f \text{ gives } \tau\right) = \frac{1}{2^{nq}}$

H-Coefficient Technique [Pat08b, CS14]

Let $\varepsilon \geq 0$ be such that for all good transcripts τ:

$$\frac{\mathbf{Pr}\left(\text{XoP gives } \tau\right)}{\mathbf{Pr}\left(f \text{ gives } \tau\right)} \geq 1 - \varepsilon$$

Then, $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{XoP}}(q) \leq \varepsilon + \mathbf{Pr}\left(\text{bad transcript for } f\right)$

- Bad transcript: if $y_i = 0$ for some i
  - $\mathbf{Pr}\left(\text{bad transcript for } f\right) = q/2^n$
- For any good transcript:
  - $\mathbf{Pr}\left(\text{XoP gives } \tau\right) \geq \frac{(2^n)_{2q}}{2^{nq}} \cdot \frac{1}{(2^n)_{2q}}$
  - $\mathbf{Pr}\left(f \text{ gives } \tau\right) = \frac{1}{2^{nq}}$

  $\Big\}\ \varepsilon = 0$

H-Coefficient Technique [Pat08b, CS14]

Let $\varepsilon \geq 0$ be such that for all good transcripts τ:

$$\frac{\mathbf{Pr}\left(\text{XoP gives } \tau\right)}{\mathbf{Pr}\left(f \text{ gives } \tau\right)} \geq 1 - \varepsilon$$

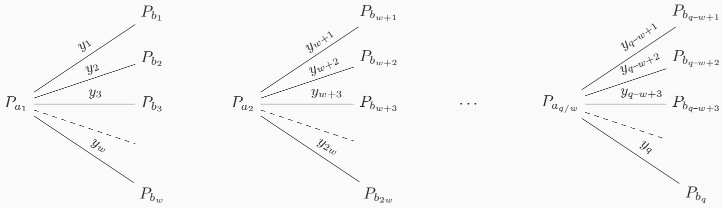
Then, $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{XoP}}(q) \leq \varepsilon + \mathbf{Pr}\left(\text{bad transcript for } f\right)$

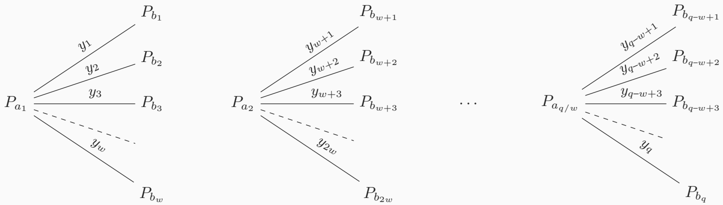
- Bad transcript: if $y_i = 0$ for some i
  - $\mathbf{Pr}\left(\text{bad transcript for } f\right) = q/2^n$
- For any good transcript:
  - $\mathbf{Pr}\left(\text{XoP gives } \tau\right) \geq \frac{(2^n)_{2q}}{2^{nq}} \cdot \frac{1}{(2^n)_{2q}}$
  - $\mathbf{Pr}\left(f \text{ gives } \tau\right) = \frac{1}{2^{nq}}$

  $\Big\}\ \varepsilon = 0$

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{XoP}}(q) \leq q/2^n$$

Mirror Theory Applied to CENC

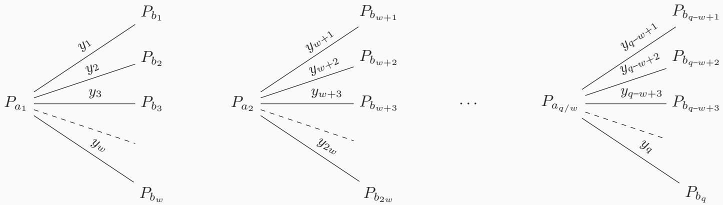




Applying Mirror Theory

- Circle-free: no collisions in inputs to p
- Non-degenerate: provided that $y_i \neq 0$ and $y_i \neq y_j$ $(\forall i, j)$ within all w-blocks

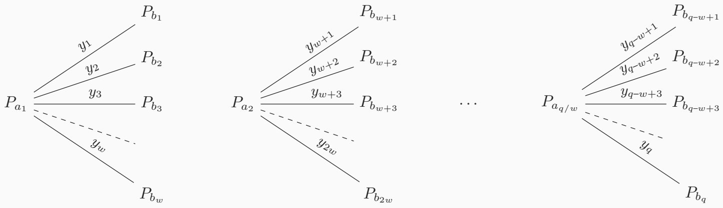
  $\longrightarrow$ Call this a bad transcript
- Maximum tree size $w + 1$



Applying Mirror Theory

- Circle-free: no collisions in inputs to p
- Non-degenerate: provided that $y_i \neq 0$ and $y_i \neq y_j$ $(\forall i, j)$ within all w-blocks

  $\longrightarrow$ Call this a bad transcript
- Maximum tree size $w + 1$
- If $(w+1)^3 q \leq 2^n/12$: at least $\frac{(2^n)_r}{2^{nq}}$ solutions to unknowns



Applying Mirror Theory

- Circle-free: no collisions in inputs to p
- Non-degenerate: provided that $y_i \neq 0$ and $y_i \neq y_j$ $(\forall i, j)$ within all w-blocks

  $\longrightarrow$ Call this a bad transcript
- Maximum tree size $w + 1$
- If $(w+1)^3 q \leq 2^n/12$: at least $\frac{(2^n)_r}{2^{nq}}$ solutions to unknowns
- H-coefficient technique: $\mathbf{Adv}^{\mathrm{prf}}_{\mathsf{CENC}}(q) \leq q/2^n + wq/2^{n+1}$

Accordion Modes (Instantiations)

Universal Hash Function Instantiation for Both $ddd\text{-}AES$ and $bbb\text{-}ddd\text{-}AES$

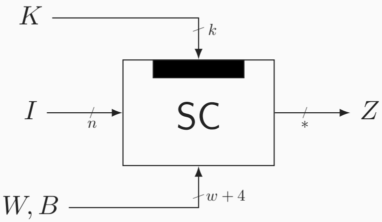
Polyval [GLL17]

- Operates on finite field $GF(2^{128})[x]/(x^{128} + x^{127} + x^{126} + x^{121} + 1)$
- Defined as follows, for a padded message $(I_1, I_2, \ldots, I_s)$:

$$\text{Polyval}_L(I_1, I_2, \ldots, I_s) = \sum_{i=1}^{s} \left(L^{s-i+1} \cdot I_i \cdot x^{-128\cdot(s-i+1)} \right)$$

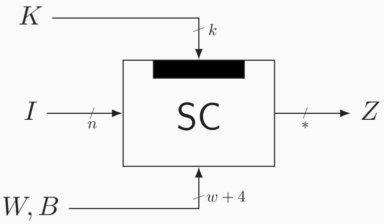
- We use zero-padding with length encoding

Recall Goal



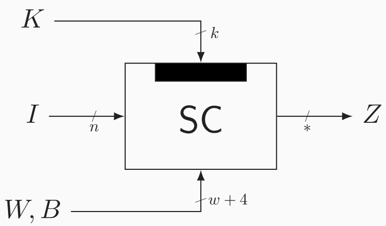
Stream Cipher Instantiation

Recall Goal



- Construction should be built on top of AES

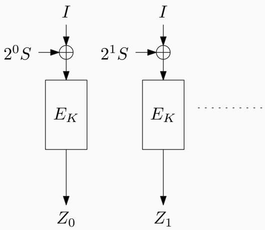
Recall Goal



- Construction should be built on top of AES
- We give one construction with birthday bound security
  one construction with beyond birthday bound security

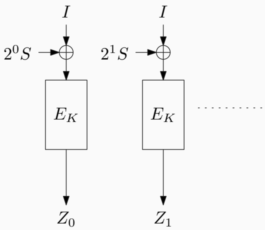
XE-style [Rog04] Tweakable Blockcipher in Counter Mode

- Let $S = E_K(B\|W)$



XE-style [Rog04] Tweakable Blockcipher in Counter Mode

- Let $S = E_K(B \| W)$



- Stream cipher (and thus ddd-AES) is $2^{n/2}$ PRF-secure

Bonus: Extension $ddd\text{-}AES^+$ to Accommodate Variable-Length Tweaks

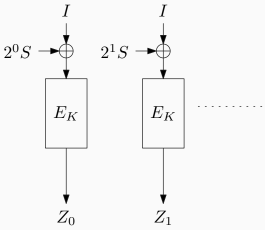
- ddd-AES almost seamlessly fits NIST's accordion idea
- Only thing missing: variable-length tweaks

Bonus: Extension $ddd\text{-}AES^+$ to Accommodate Variable-Length Tweaks

- $ddd\text{-}AES$ almost seamlessly fits NIST's accordion idea
- Only thing missing: variable-length tweaks

XE$^+$-style [Rog04] Tweakable Blockcipher in Counter Mode

- Pad B, W into $(W_0, W_1, \ldots, W_{l-1} \| B' \| 0^*)$ with $B' = B \oplus 1000$
- Let $S = E_K(W_0\|0) \oplus E_K(W_1\|1) \oplus \cdots \oplus E_K(W_{l-1}\|B'\|0^*\|(l-1))$

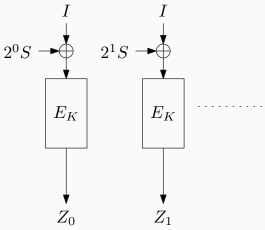


Bonus: Extension $ddd\text{-}AES^+$ to Accommodate Variable-Length Tweaks

- ddd-AES almost seamlessly fits NIST's accordion idea
- Only thing missing: variable-length tweaks

XE$^+$-style [Rog04] Tweakable Blockcipher in Counter Mode

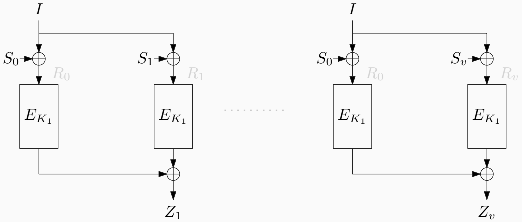
- Pad B, W into $(W_0, W_1, \ldots, W_{l-1} \| B' \| 0^*)$ with $B' = B \oplus 1000$
- Let $S = E_K(W_0\|0) \oplus E_K(W_1\|1) \oplus \cdots \oplus E_K(W_{l-1}\|B'\|0^*\|(l-1))$



- Stream cipher (and thus ddd-AES^+) is $2^{n/2}$ PRF-secure

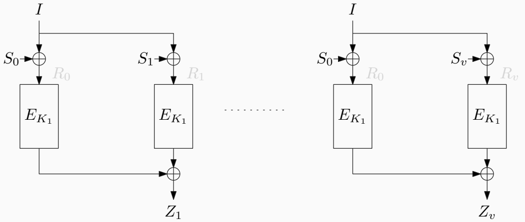
$\widetilde{\text{XoP}[w]}$ PRF in Counter Mode

- $\widetilde{\text{XoP}[w]}$: XoP$[w]$ as used in CENC [Iwa06], and extended to include tweak
  - Introduction is new and comes with separate security proof
  - Let $S_j = E_{K_2}(B\|W\|c\|j)$



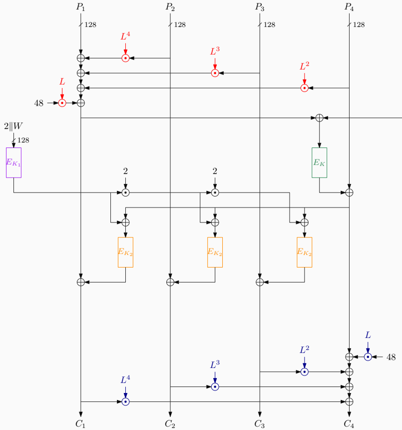
$\widetilde{\text{XoP}[w]}$ PRF in Counter Mode

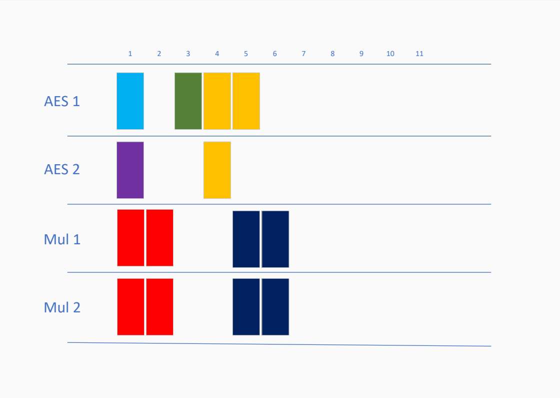
- $\widetilde{\text{XoP}[w]}$: $\text{XoP}[w]$ as used in CENC [Iwa06], and extended to include tweak
  - Introduction is new and comes with separate security proof
  - Let $S_j = E_{K_2}(B\|W\|c\|j)$



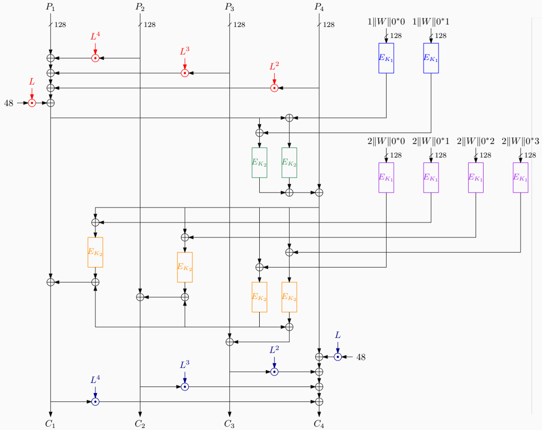
- Corresponding stream cipher runs $\widetilde{\text{XoP}[w]}$ in counter mode
- Stream cipher (and thus *bbb-ddd-AES*) is $2^{2n/3}$ PRF-secure when tweaks are not used too often

Implementation Design of $ddd\text{-}AES$ (512-Bit Message)

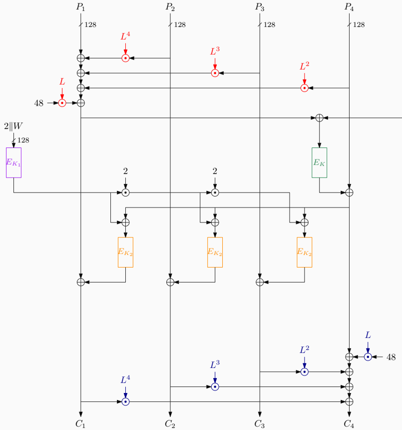


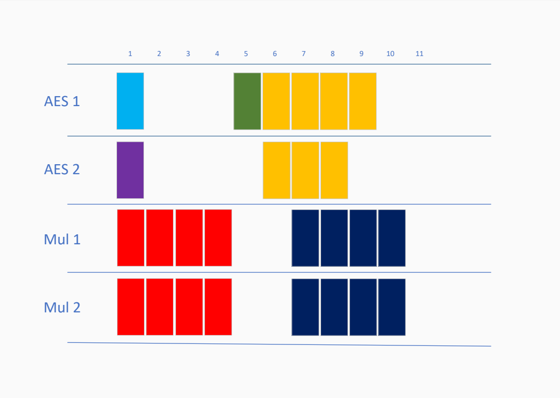


Implementation Design of *bbb-ddd-AES* (512-Bit Message)



Implementation Design of $ddd\text{-}AES$ (1024-Bit Message)





Implementation Design of $bbb\text{-}ddd\text{-}AES$ (1024-Bit Message)