OleF: an Inverse-Free Online Cipher

FSE 2017, Tokyo, Japan

Ritam Bhaumik and Mridul Nandi

Indian Statistical Institute, Kolkata

8 March 2017

Online Encryption

Online Encryption

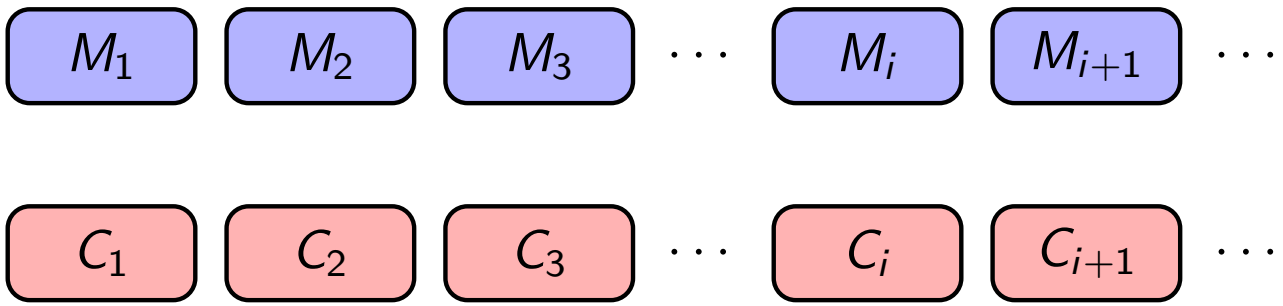
- **What does Online mean?**

Online Encryption

- **What does Online mean?**
  - i-th ciphertext block not affected by $(> i)$-th plaintext blocks

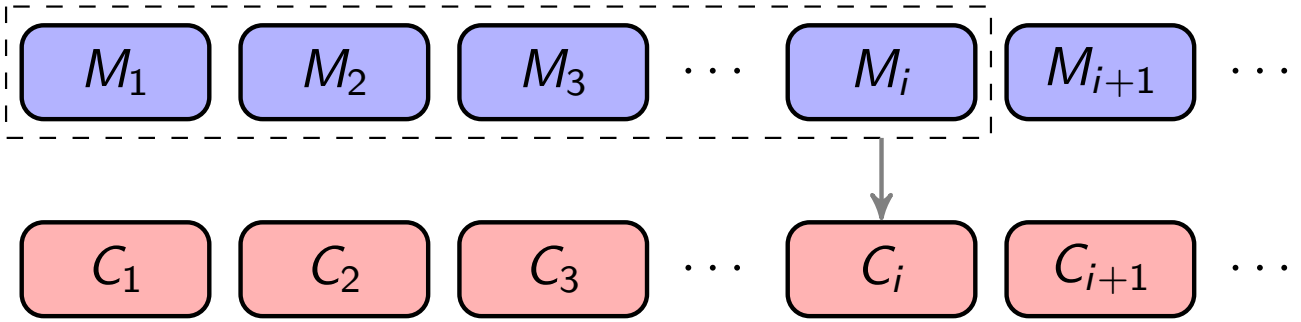
Online Encryption

- **What does Online mean?**
  - i-th ciphertext block not affected by $(> i)$-th plaintext blocks



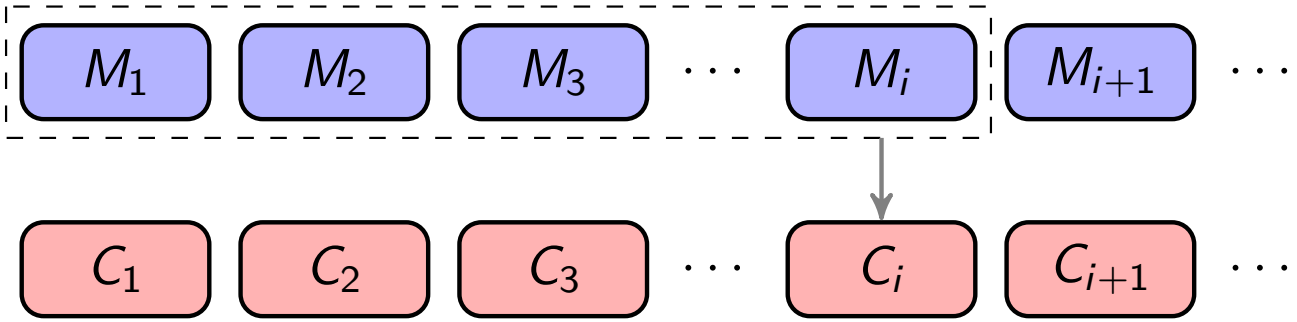
Online Encryption

- **What does Online mean?**
  - i-th ciphertext block not affected by $(> i)$-th plaintext blocks



Online Encryption

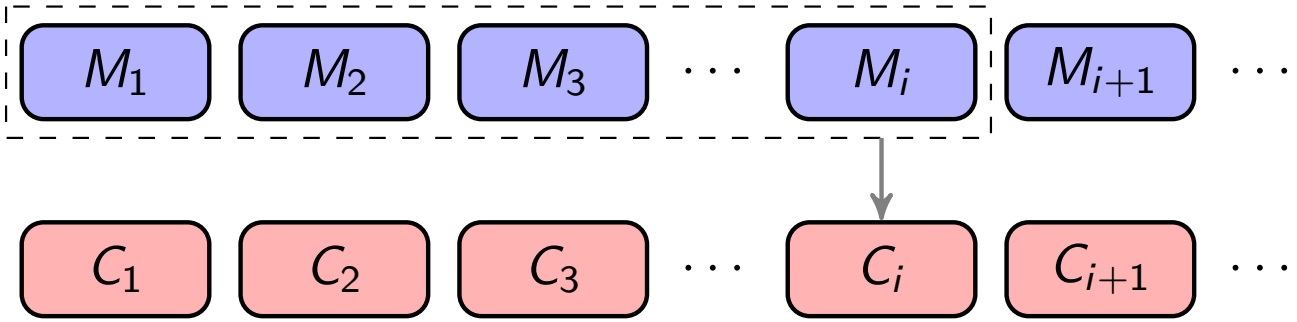
- **What does Online mean?**
  - i-th ciphertext block not affected by $(> i)$-th plaintext blocks



  - this is the classical definition; there can be variants

Online Encryption

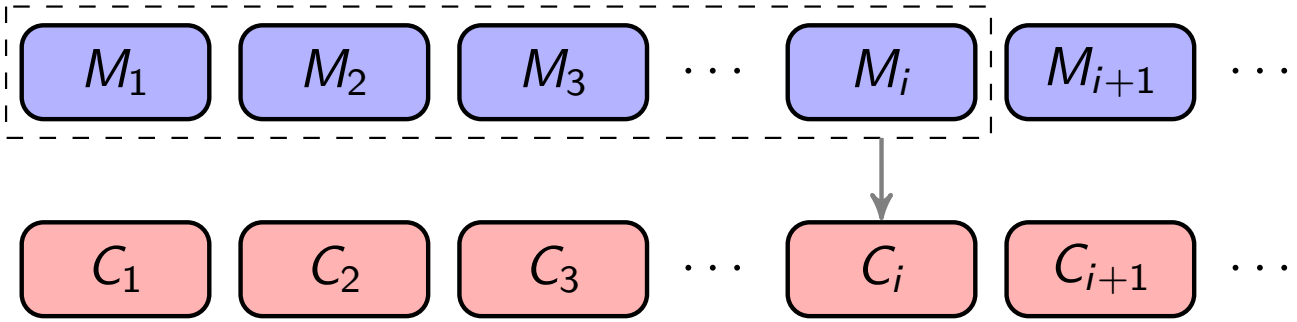
- **What does Online mean?**
  - i-th ciphertext block not affected by $(> i)$-th plaintext blocks



  - this is the classical definition; there can be variants
  - central idea: single-pass computation

Online Encryption

- **What does Online mean?**
  - i-th ciphertext block not affected by $(> i)$-th plaintext blocks



  - this is the classical definition; there can be variants
  - central idea: single-pass computation
  - frequently low-memory as well

Online Encryption: Security Implications

Online Encryption: Security Implications

- **Online vs. Full**

Online Encryption: Security Implications

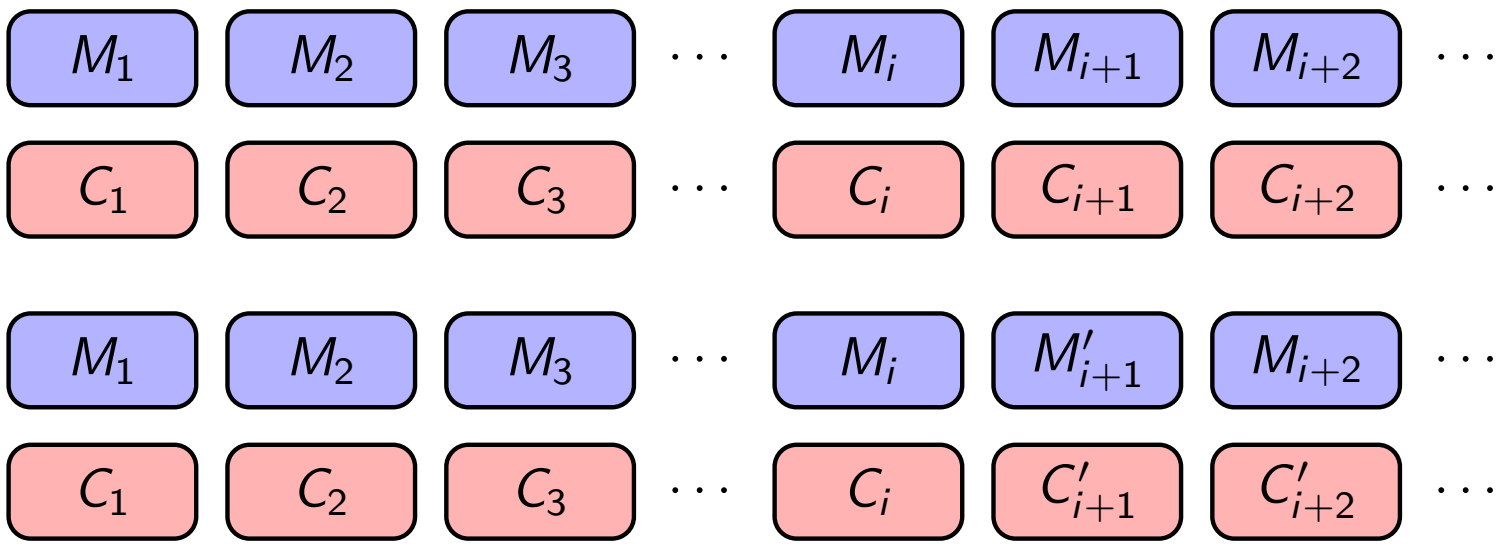
- **Online vs. Full**
  - full encryption only reveals whether two plaintexts are identical

Online Encryption: Security Implications

- **Online vs. Full**
  - full encryption only reveals whether two plaintexts are identical
  - online encryption leaks length of common prefix of plaintexts

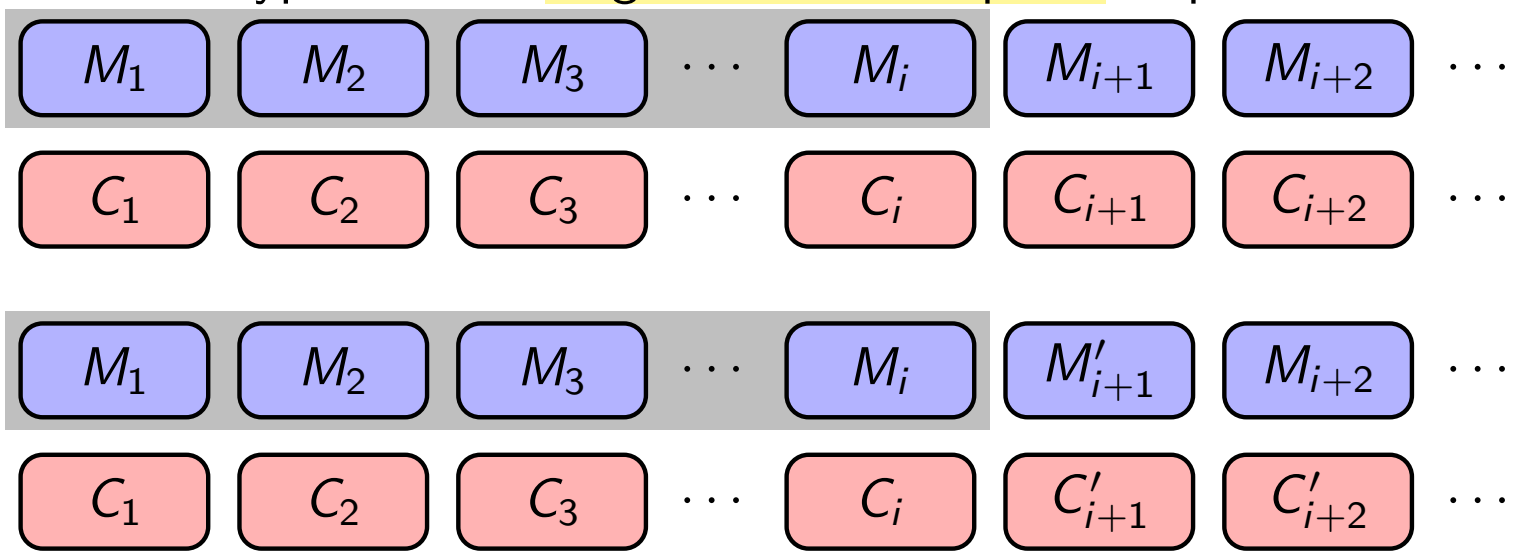
Online Encryption: Security Implications

- **Online vs. Full**
  - full encryption only reveals whether two plaintexts are identical
  - online encryption leaks length of common prefix of plaintexts



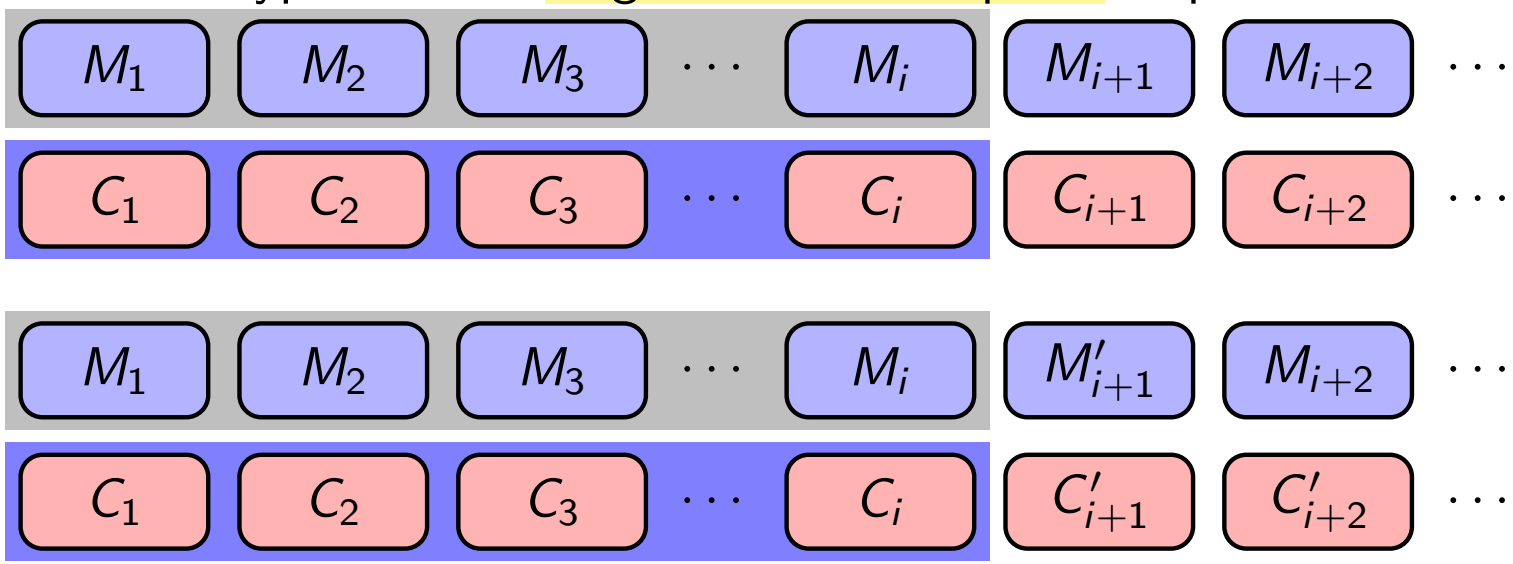
Online Encryption: Security Implications

- **Online vs. Full**
  - full encryption only reveals whether two plaintexts are identical
  - online encryption leaks length of common prefix of plaintexts



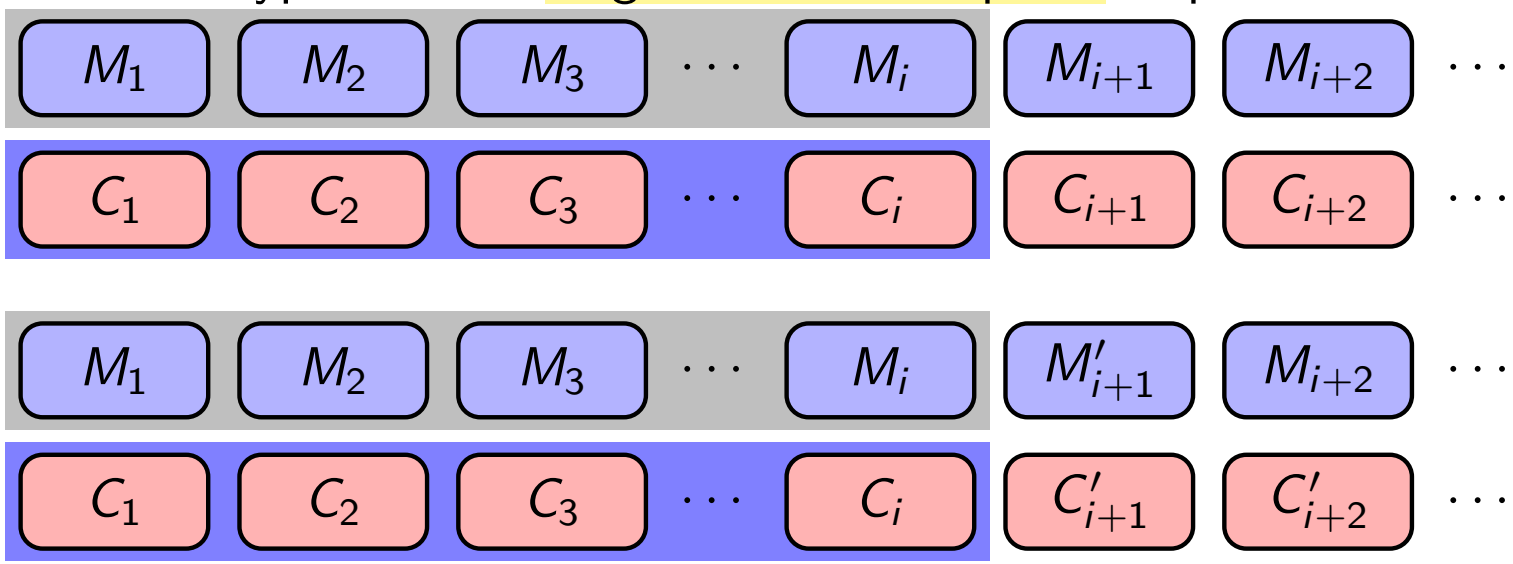
Online Encryption: Security Implications

- **Online vs. Full**
  - full encryption only reveals whether two plaintexts are identical
  - online encryption leaks length of common prefix of plaintexts



Online Encryption: Security Implications

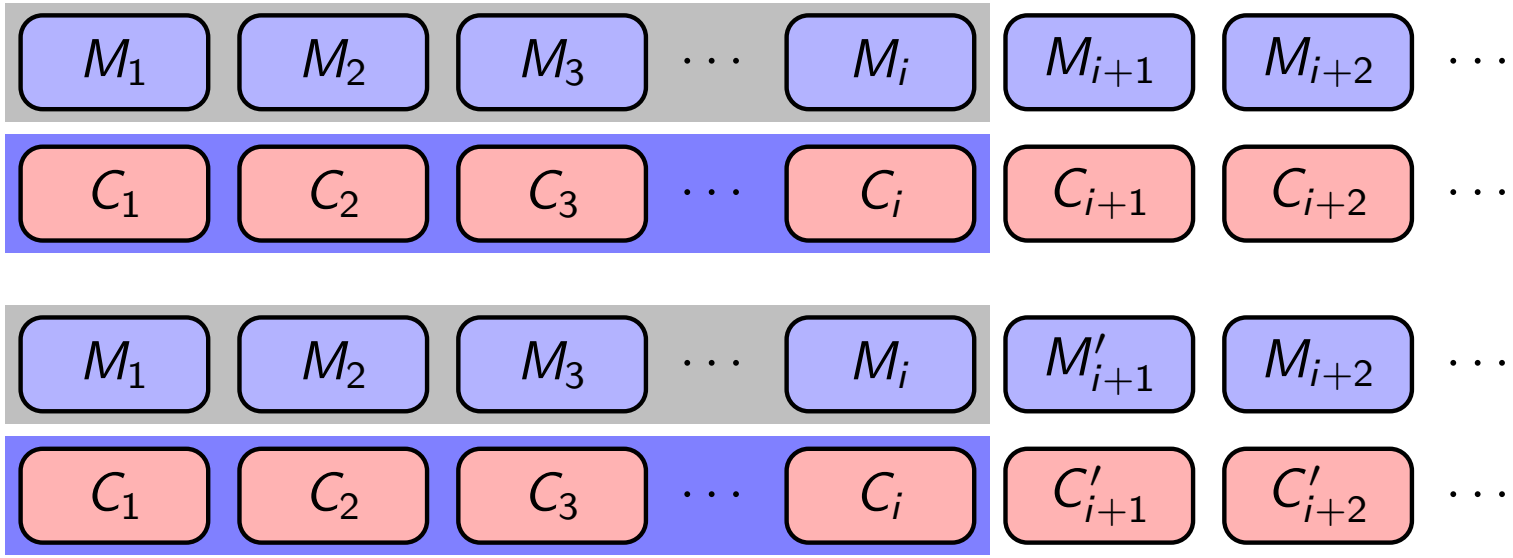
- **Online vs. Full**
  - full encryption only reveals whether two plaintexts are identical
  - online encryption leaks length of common prefix of plaintexts



  - *this is the only security degradation*

Online Encryption: Security Implications

- **Online vs. Full**
  - full encryption only reveals whether two plaintexts are identical
  - online encryption leaks length of common prefix of plaintexts



  - *this is the only security degradation*
  - performance often outweighs this degradation

Another Way to Look at Online Encryption

Another Way to Look at Online Encryption

- **Encrypt-and-Propagate Model**

Another Way to Look at Online Encryption

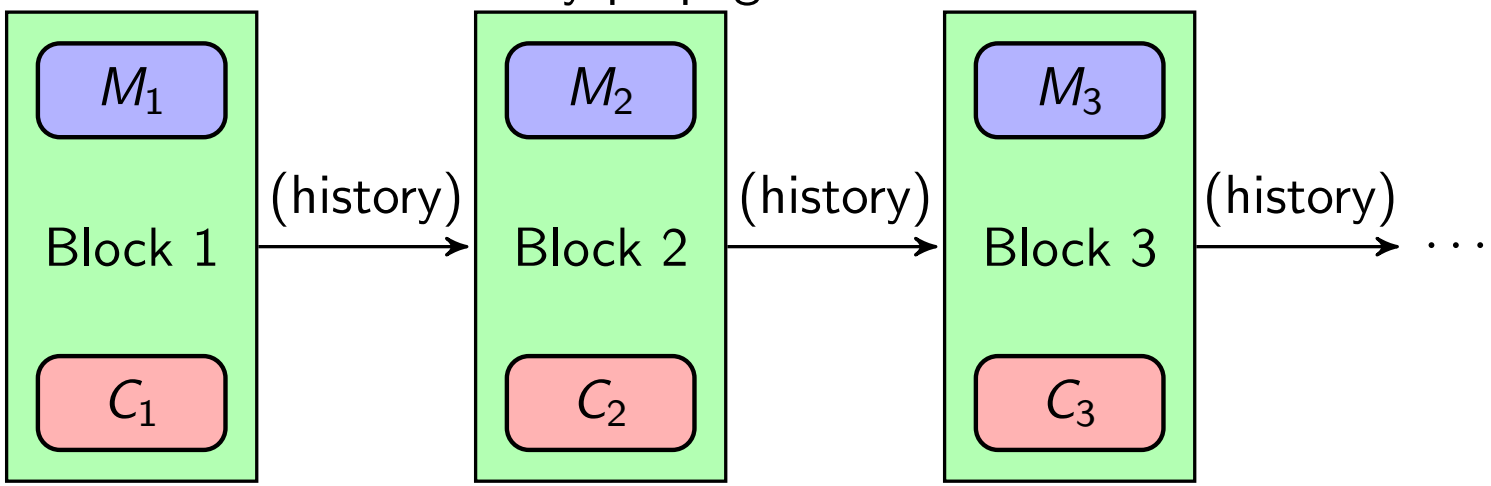
- **Encrypt-and-Propagate Model**
  - encryption proceeds block-by-block

Another Way to Look at Online Encryption

- **Encrypt-and-Propagate Model**
  - encryption proceeds block-by-block
  - information about history propagated down the line

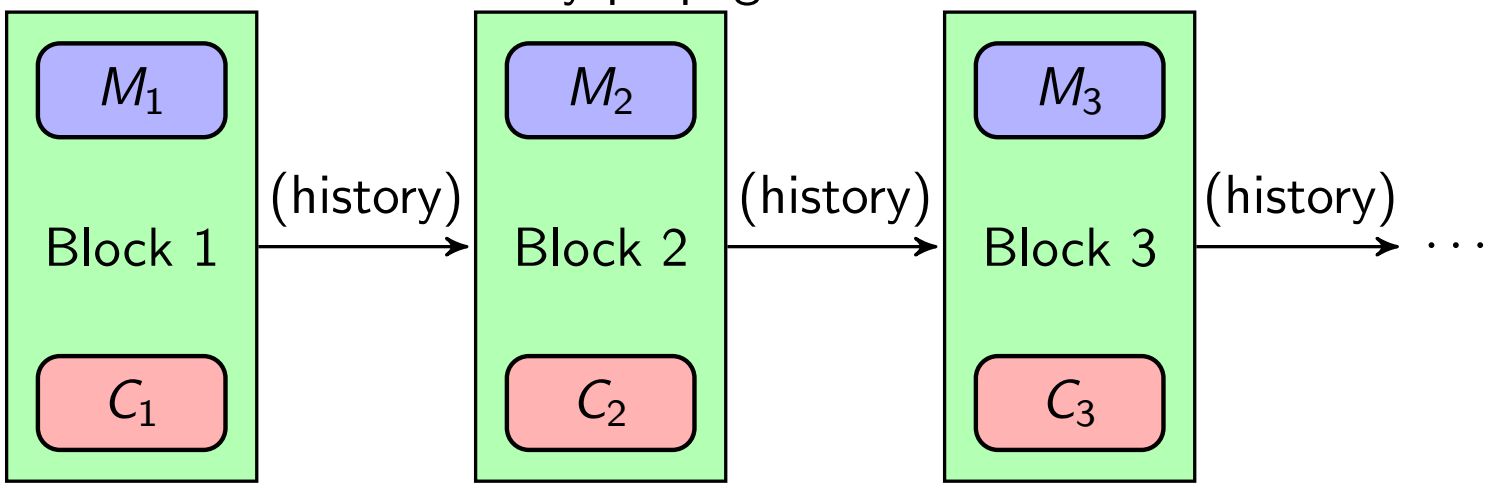
Another Way to Look at Online Encryption

- **Encrypt-and-Propagate Model**
  - encryption proceeds block-by-block
  - information about history propagated down the line



Another Way to Look at Online Encryption

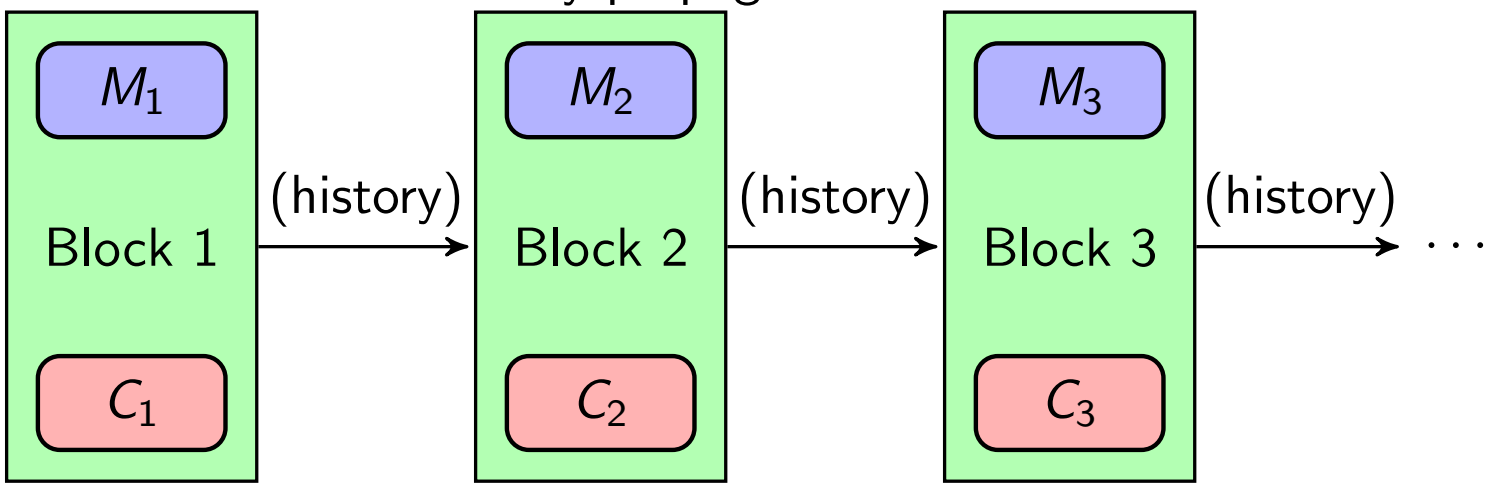
- **Encrypt-and-Propagate Model**
  - encryption proceeds block-by-block
  - information about history propagated down the line



  - We take this model as a paradigm

Another Way to Look at Online Encryption

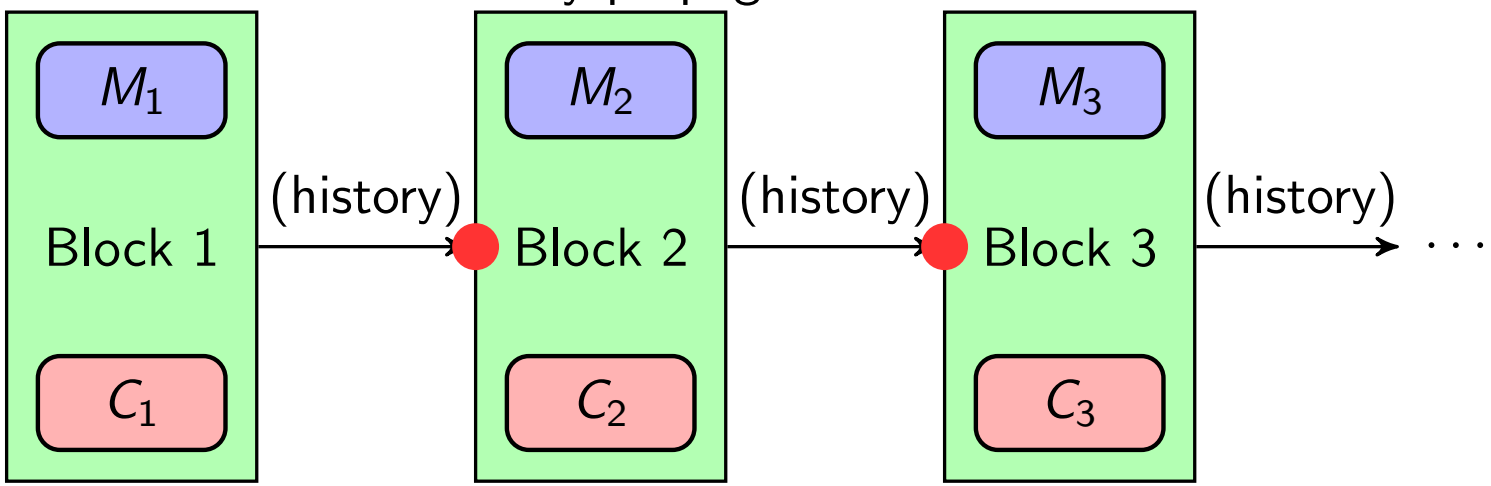
- **Encrypt-and-Propagate Model**
  - encryption proceeds block-by-block
  - information about history propagated down the line



  - We take this model as a paradigm
  - **Design Sub-Goals:**

Another Way to Look at Online Encryption

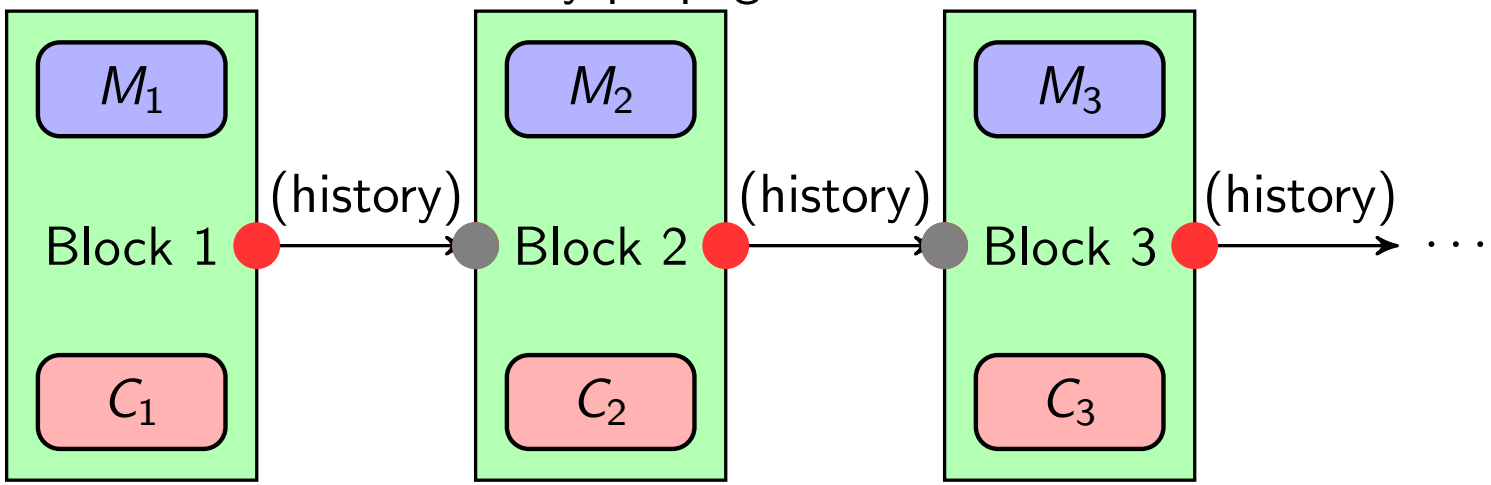
- **Encrypt-and-Propagate Model**
  - encryption proceeds block-by-block
  - information about history propagated down the line



  - We take this model as a paradigm
  - **Design Sub-Goals:**
    - To choose an appropriate way for injecting history

Another Way to Look at Online Encryption

- **Encrypt-and-Propagate Model**
  - encryption proceeds block-by-block
  - information about history propagated down the line



  - We take this model as a paradigm
  - **Design Sub-Goals:**
    - To choose an appropriate way for injecting history
    - To choose a suitable function for encoding history

Inverse-Free Encryption of Single Block

Inverse-Free Encryption of Single Block

- **What is an Inverse-Free Mode?**

Inverse-Free Encryption of Single Block

- **What is an Inverse-Free Mode?**
  - Only encryption calls to underlying blockcipher E_K

Inverse-Free Encryption of Single Block

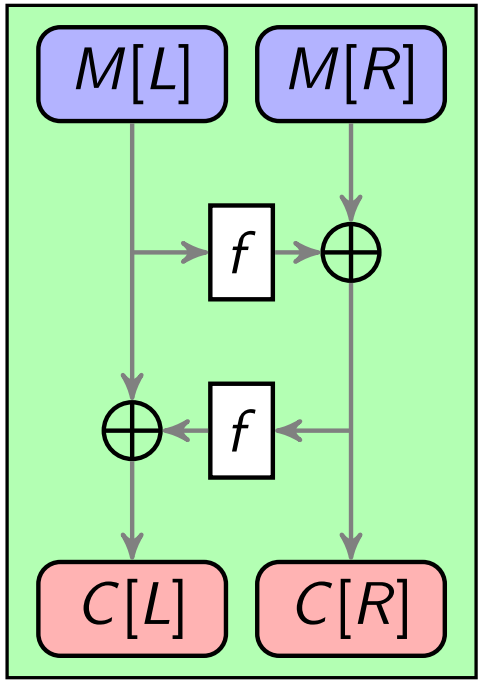
- **What is an Inverse-Free Mode?**
  - Only encryption calls to underlying blockcipher E_K
  - E_K^{-1} is never called, *not even during decryption*

Inverse-Free Encryption of Single Block

- **What is an Inverse-Free Mode?**
  - Only encryption calls to underlying blockcipher E_K
  - E_K^{-1} is never called, *not even during decryption*
  - In birthday-secure constructions, E_K can be replaced by a prf

Inverse-Free Encryption of Single Block

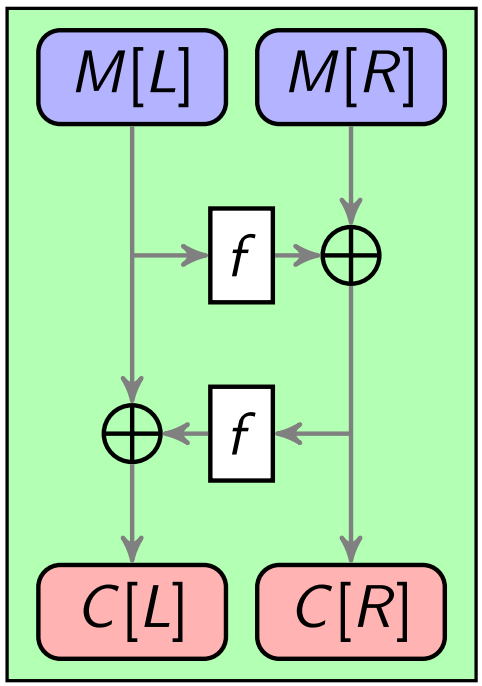
- **What is an Inverse-Free Mode?**
  - Only encryption calls to underlying blockcipher E_K
  - E_K^{-1} is never called, *not even during decryption*
  - In birthday-secure constructions, E_K can be replaced by a prf
  - Famous example: **Feistel Mode**



$$M = M[L]||M[R]$$
$$C = C[L]||C[R]$$

Inverse-Free Encryption of Single Block

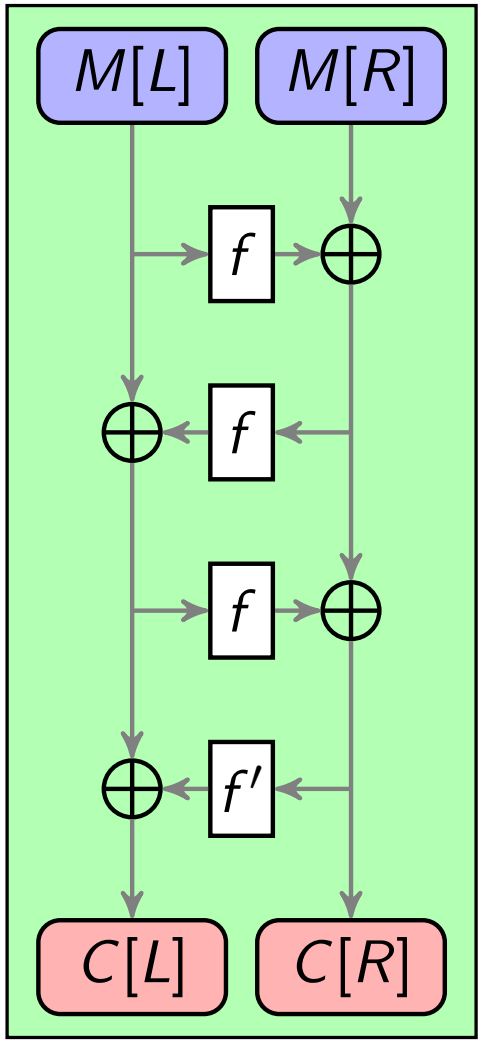
- **What is an Inverse-Free Mode?**
  - Only encryption calls to underlying blockcipher E_K
  - E_K^{-1} is never called, *not even during decryption*
  - In birthday-secure constructions, E_K can be replaced by a prf
  - Famous example: **Feistel Mode**
  - Unfortunately, one full Feistel round is insecure



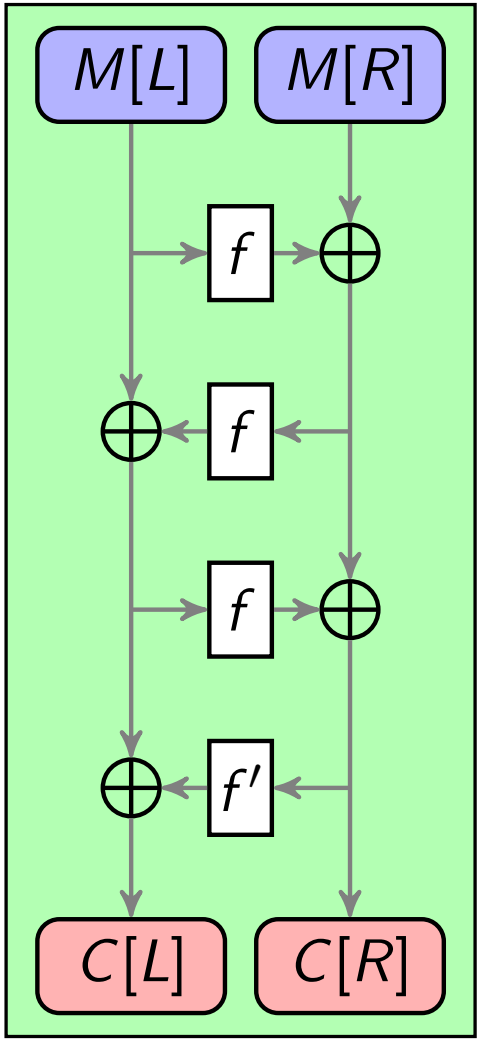
$$M = M[L]||M[R]$$
$$C = C[L]||C[R]$$

Inverse-Free Encryption of Single Block

- **What is an Inverse-Free Mode?**
  - Only encryption calls to underlying blockcipher E_K
  - E_K^{-1} is never called, *not even during decryption*
  - In birthday-secure constructions, E_K can be replaced by a prf
  - Famous example: **Feistel Mode**
  - Unfortunately, one full Feistel round is insecure
  - So we go instead for two full rounds of Feistel

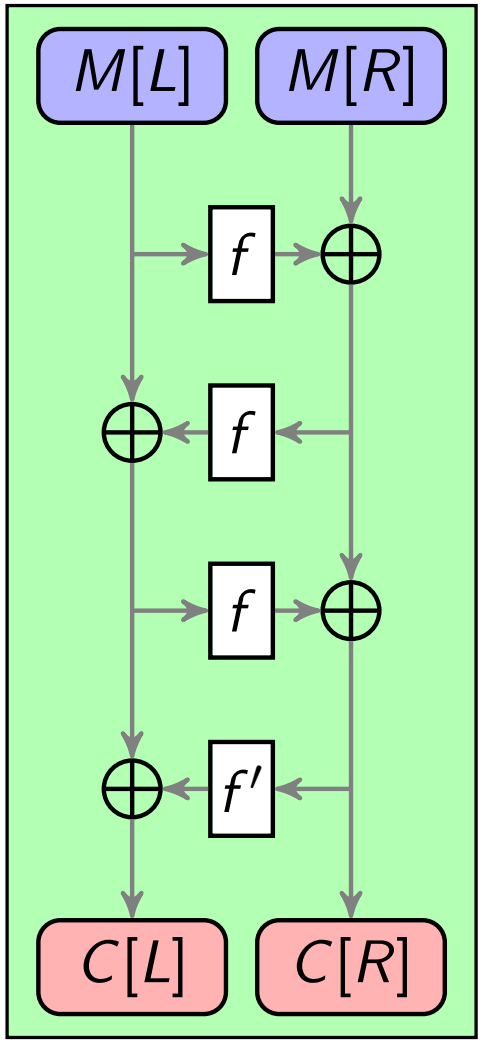


Design Sub-Goal 1: Injecting History



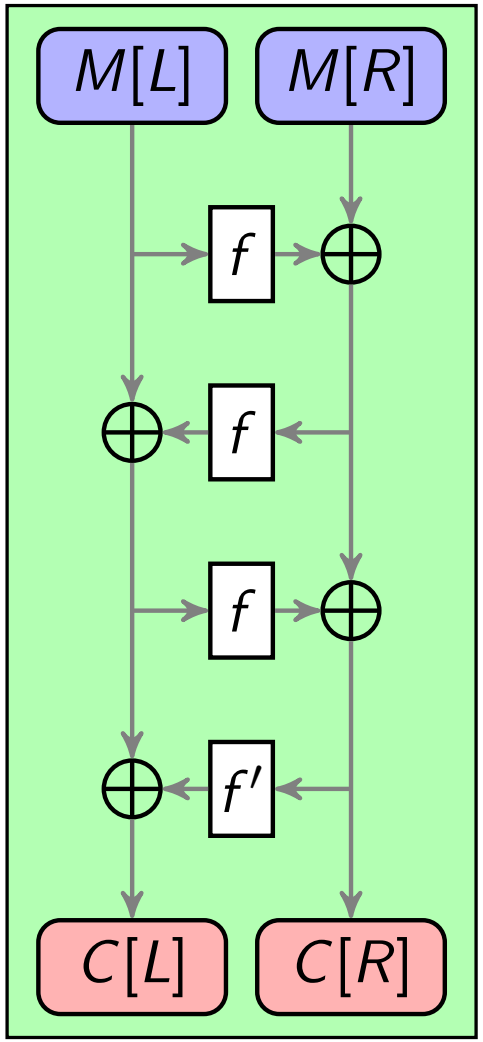
Design Sub-Goal 1: Injecting History

- **Adding the Tweak**



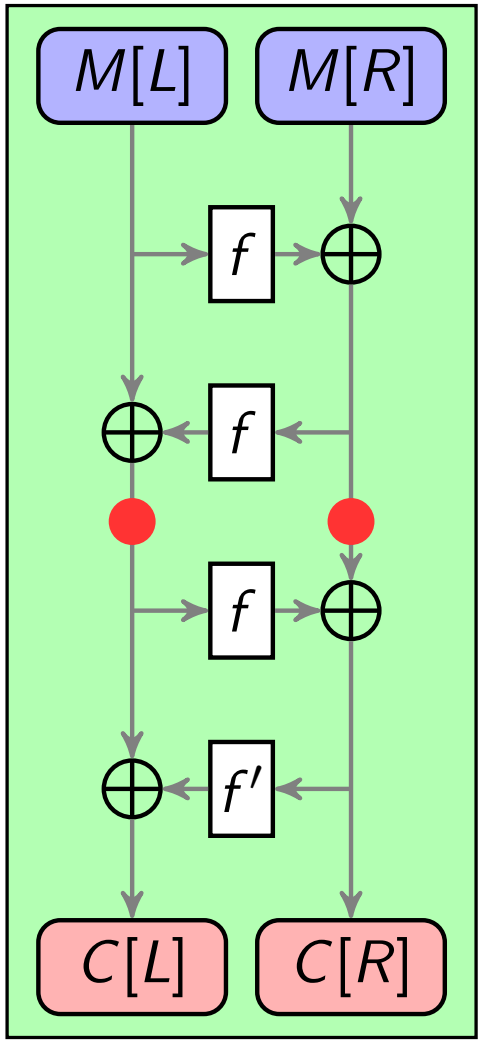
Design Sub-Goal 1: Injecting History

- **Adding the Tweak**
  - We henceforth refer to the encoded history as a tweak T



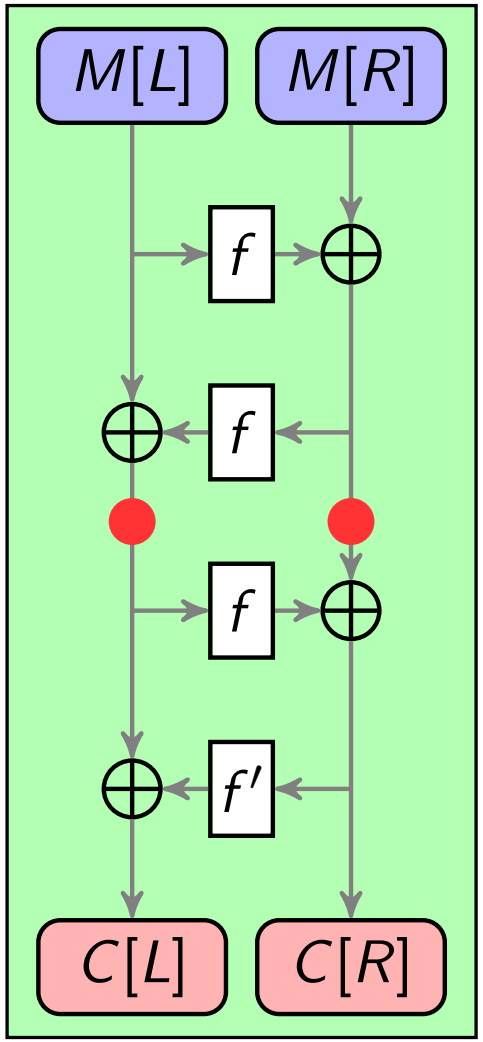
Design Sub-Goal 1: Injecting History

- **Adding the Tweak**
  - We henceforth refer to the encoded history as a tweak T
  - Straightforward choice: XOR-ing in the middle strands



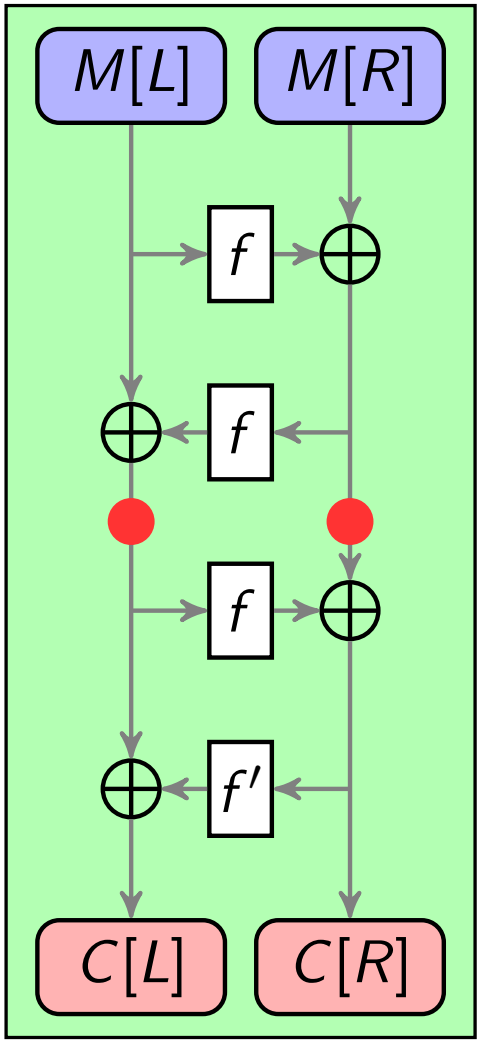
Design Sub-Goal 1: Injecting History

- **Adding the Tweak**
  - We henceforth refer to the encoded history as a tweak T
  - Straightforward choice: XOR-ing in the middle strands
  - But this becomes CBC-linear mix-CBC



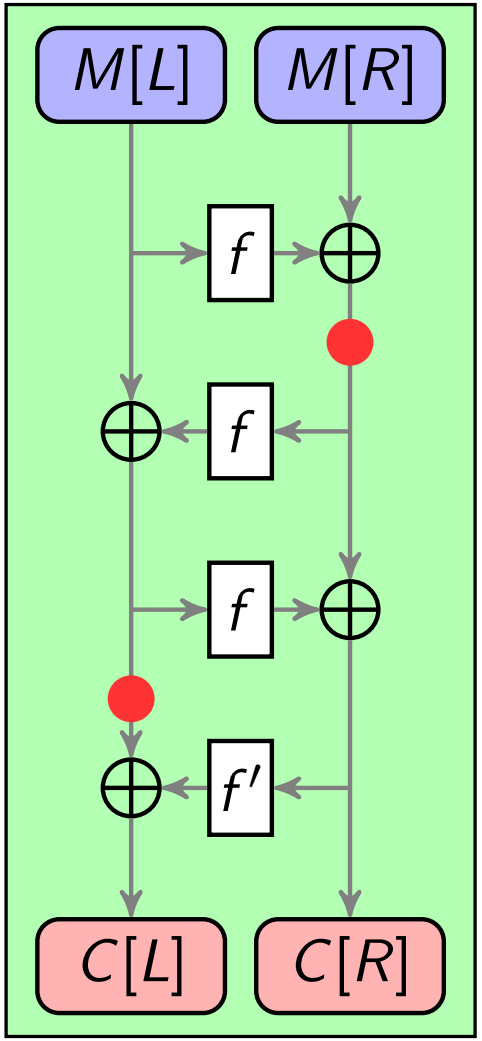
Design Sub-Goal 1: Injecting History

- **Adding the Tweak**
  - We henceforth refer to the encoded history as a tweak T
  - Straightforward choice: XOR-ing in the middle strands
  - But this becomes CBC-linear mix-CBC
  - not secure

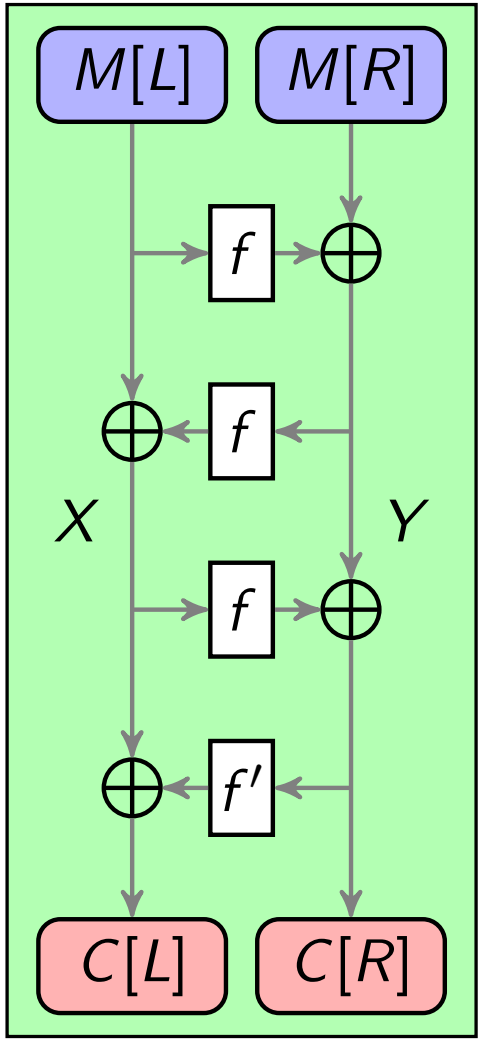


Design Sub-Goal 1: Injecting History

- **Adding the Tweak**
  - We henceforth refer to the encoded history as a tweak T
  - Straightforward choice: XOR-ing in the middle strands
  - But this becomes CBC-linear mix-CBC
  - not secure
  - So we inject after first and third f-call

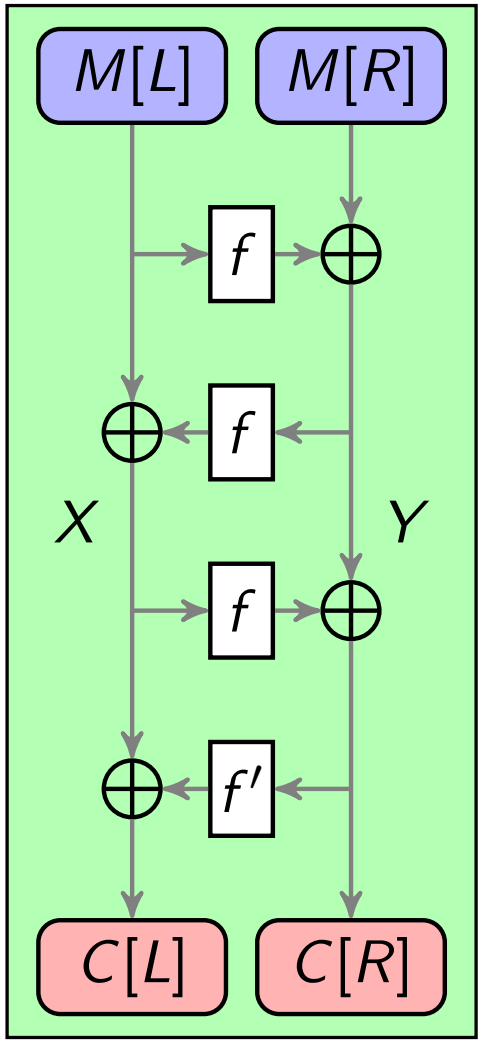


Design Sub-Goal 2: Encoding History



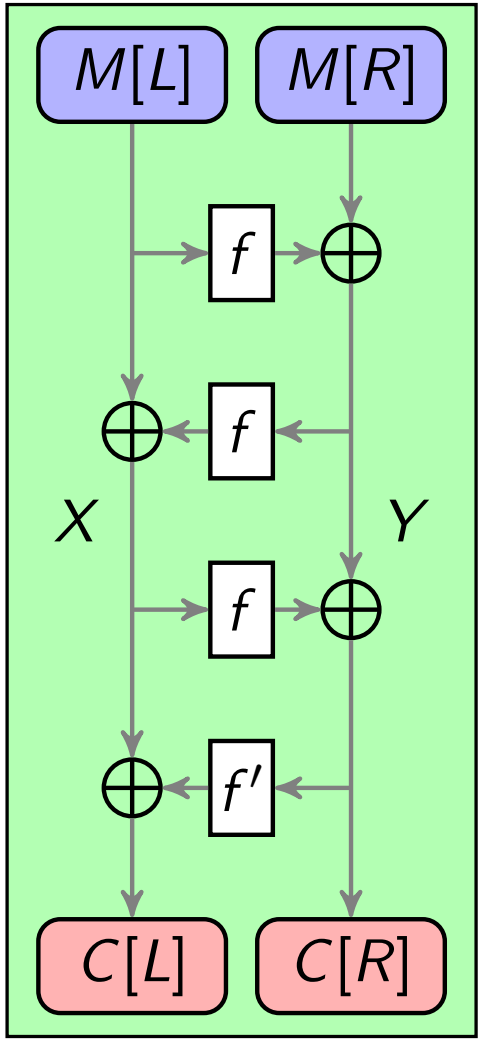
Design Sub-Goal 2: Encoding History

- **Generating the Tweak**



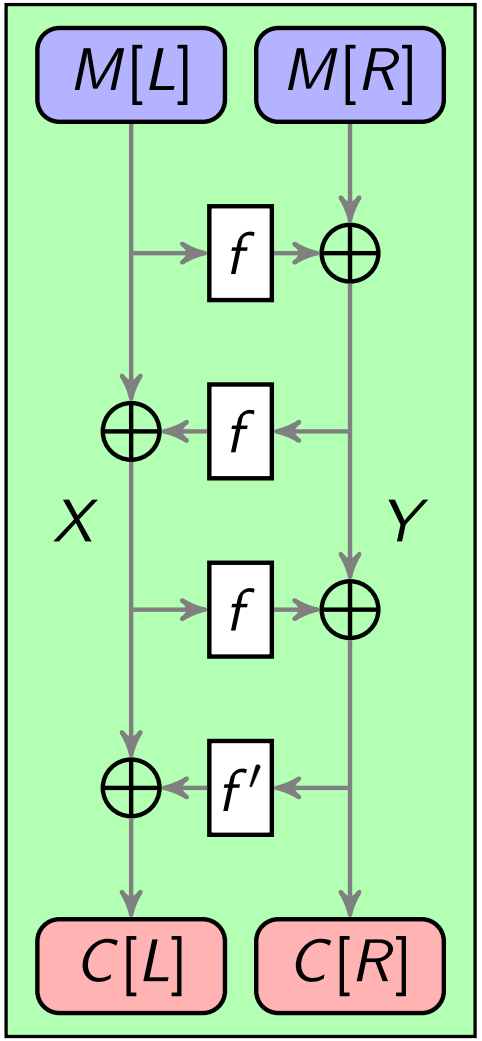
Design Sub-Goal 2: Encoding History

- **Generating the Tweak**
  - Naive choice:
    $M[L] \oplus M[R] \oplus C[L] \oplus C[R]$



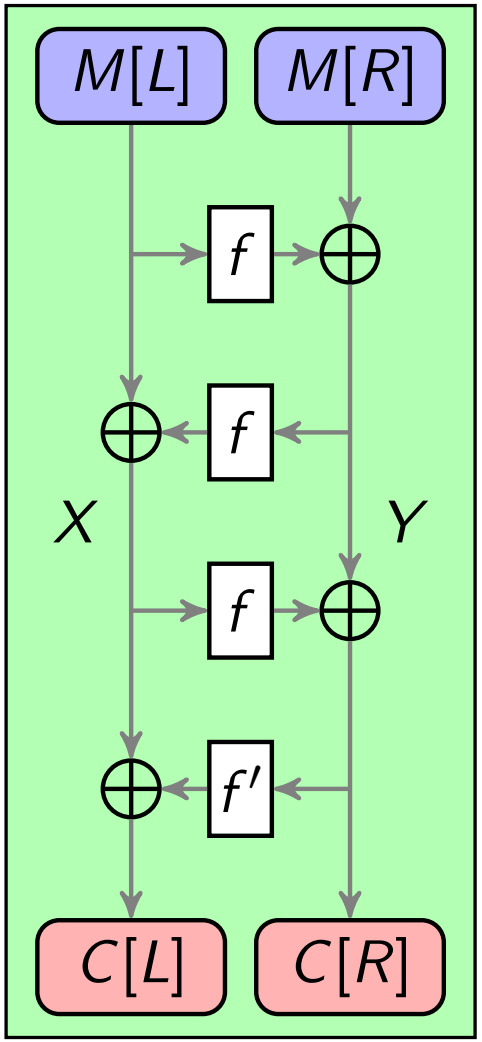
Design Sub-Goal 2: Encoding History

- **Generating the Tweak**
  - Naive choice:
    $M[L] \oplus M[R] \oplus C[L] \oplus C[R]$
  - But this gives adversary control over ΔT



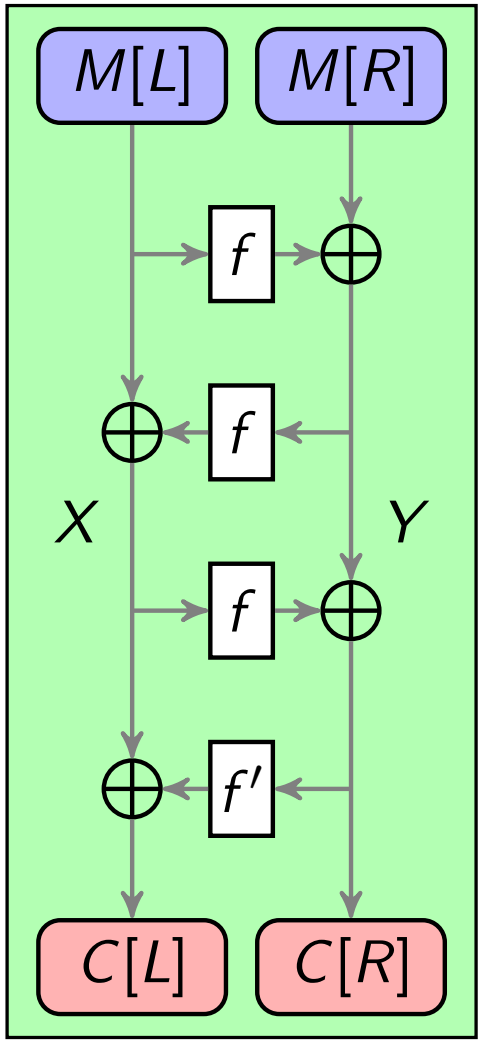
Design Sub-Goal 2: Encoding History

- **Generating the Tweak**
  - Naive choice:
    $M[L] \oplus M[R] \oplus C[L] \oplus C[R]$
  - But this gives adversary control over ΔT
  - not secure



Design Sub-Goal 2: Encoding History

- **Generating the Tweak**
  - Naive choice: $M[L] \oplus M[R] \oplus C[L] \oplus C[R]$
  - But this gives adversary control over ΔT
  - not secure
  - So we choose $X \oplus Y$



Balanced Linear Permutations

Balanced Linear Permutations

- In our design we use balanced linear permutations b_1, b_2, b_3, b_4

Balanced Linear Permutations

- In our design we use balanced linear permutations b_1, b_2, b_3, b_4
- $b_i(x) := \alpha^i \cdot x$

Balanced Linear Permutations

- In our design we use balanced linear permutations b_1, b_2, b_3, b_4
- $b_i(x) := \alpha^i \cdot x$
- Called balanced because b_i and $b_i + I$ are full-rank

Balanced Linear Permutations

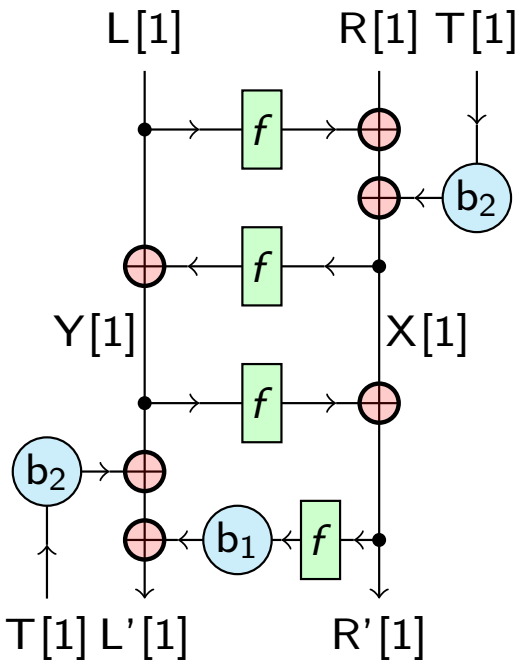
- In our design we use balanced linear permutations b_1, b_2, b_3, b_4
- $b_i(x) := \alpha^i \cdot x$
- Called balanced because b_i and $b_i + I$ are full-rank
- In addition, for this choice, $b_i + b_j$ is also full-rank

Balanced Linear Permutations

- In our design we use balanced linear permutations b_1, b_2, b_3, b_4
- $b_i(x) := \alpha^i \cdot x$
- Called balanced because b_i and $b_i + I$ are full-rank
- In addition, for this choice, $b_i + b_j$ is also full-rank
- For breaking symmetry, we take $f' := b_1 \circ f$

Final Construction

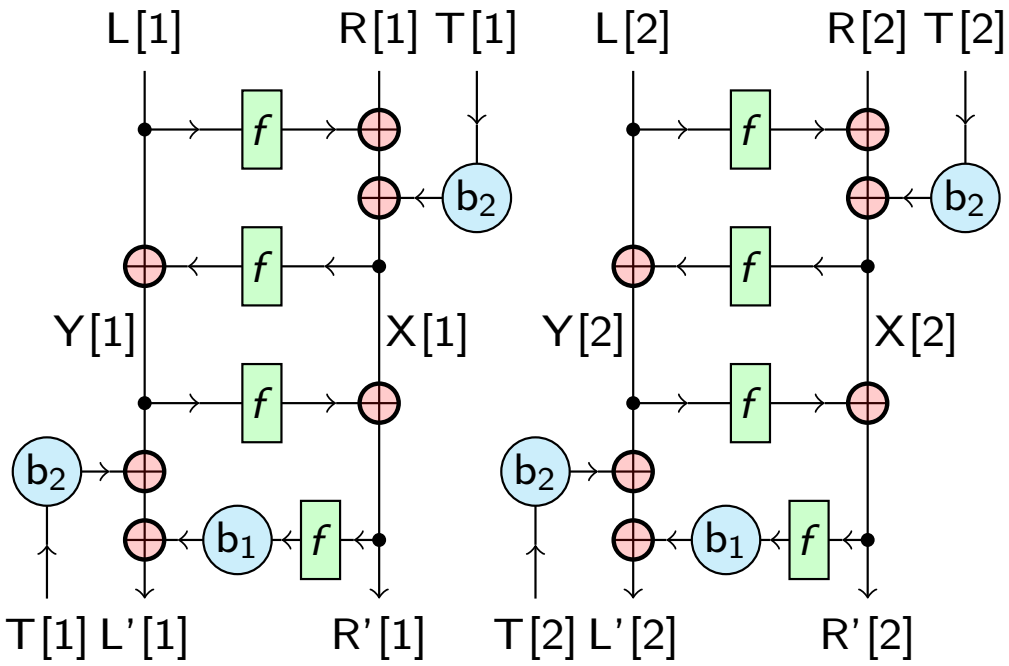
Final Construction



$T[1] = 0$

Figure: OleF for ℓ Complete Diblocks

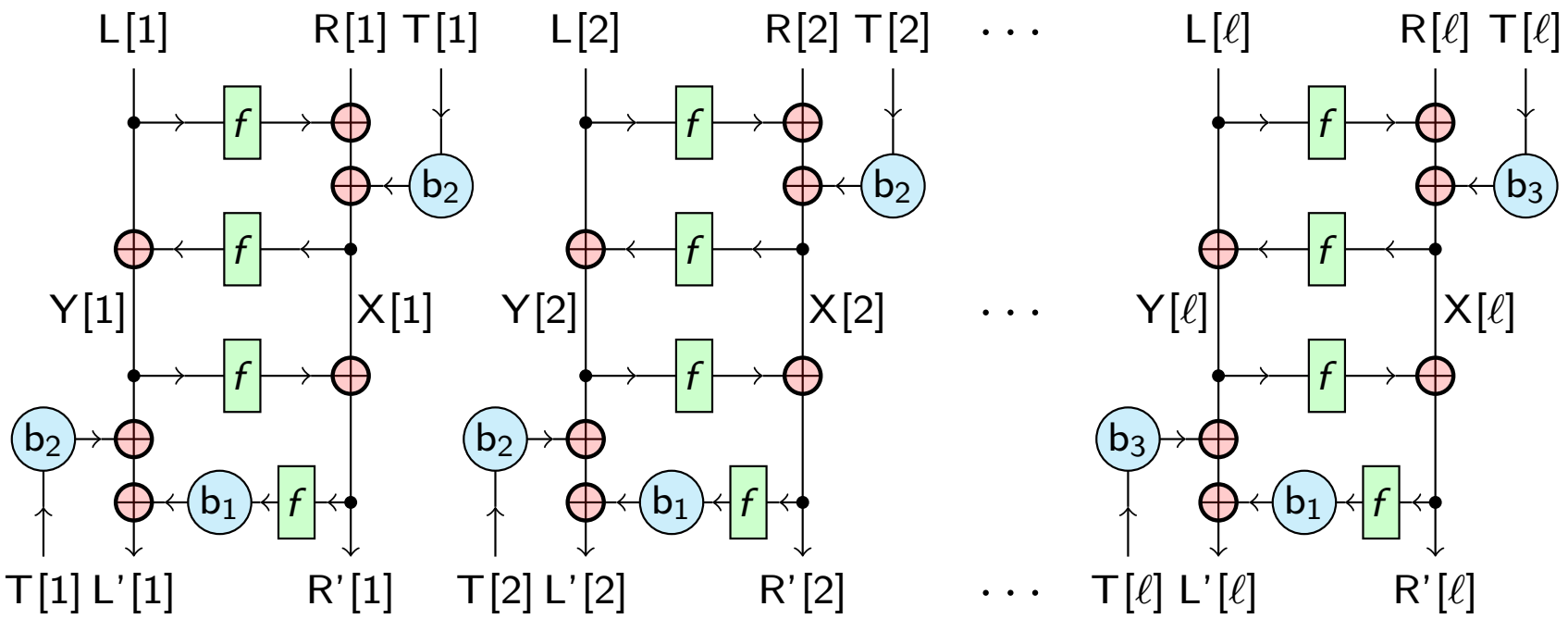
Final Construction



$T[1] = 0$ $\quad$ $T[2] = X[1] \oplus Y[1]$

Figure: OleF for ℓ Complete Diblocks

Final Construction



$T[1] = 0$ $\quad T[2] = X[1] \oplus Y[1]$ $\quad T[\ell] = X[\ell - 1] \oplus Y[\ell - 1]$

Figure: OleF for ℓ Complete Diblocks

Handling Partial Blocks

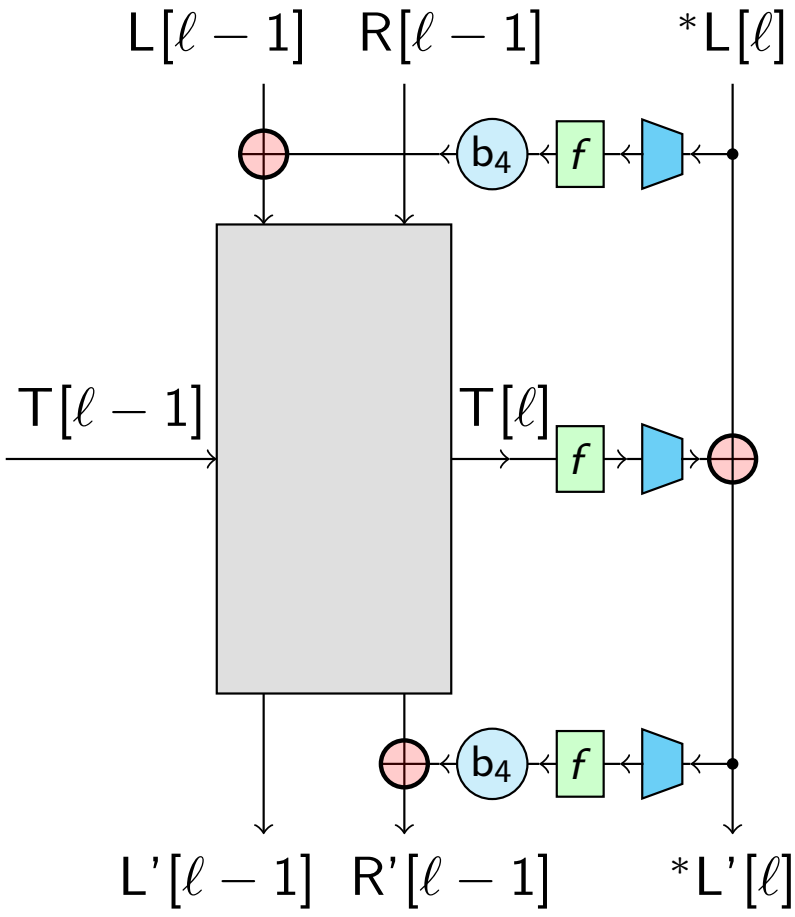


Figure: OleF for Partial Diblocks, where $*\mathrm{L}[\ell]$ has less than n bits

Handling Partial Blocks

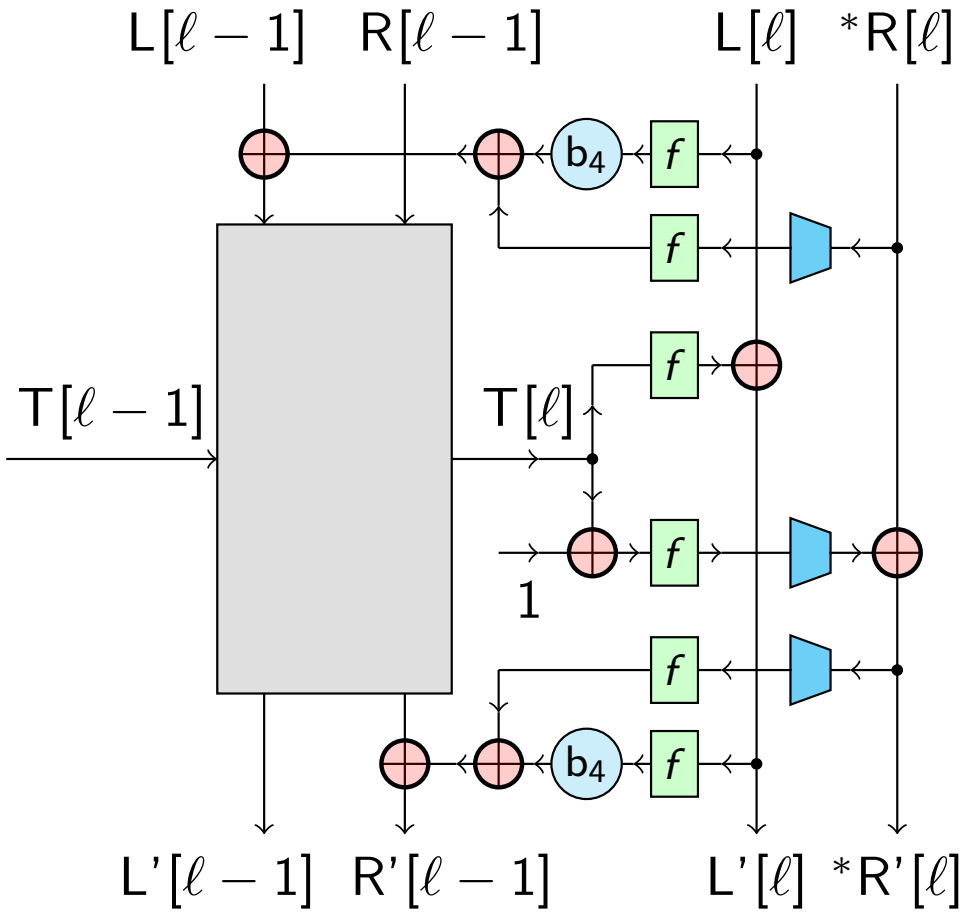


Figure: OleF for Partial Diblocks, where $*R[\ell]$ has less than n bits

Security Results

Security Results

- Online sprp advantage bounded by $\frac{7\sigma^2}{2^n}$ plus prf-advantage of f

Security Results

- Online sprp advantage bounded by $\frac{7\sigma^2}{2^n}$ plus prf-advantage of f
- Achieves the ideal security notion for online ciphers in birthday-bound

Security Results

- Online sprp advantage bounded by $\frac{7\sigma^2}{2^n}$ plus prf-advantage of f
- Achieves the ideal security notion for online ciphers in birthday-bound
- Proof uses Patarin's Technique

Advantages

Advantages

- Possibly optimal number of f-calls

Advantages

- Possibly optimal number of f-calls
- Inverse-free, hence low footprint

Advantages

- Possibly optimal number of f-calls
- Inverse-free, hence low footprint
- Online, low-memory

Advantages

- Possibly optimal number of f-calls
- Inverse-free, hence low footprint
- Online, low-memory
- At least one full block of randomness per query

Advantages

- Possibly optimal number of f-calls
- Inverse-free, hence low footprint
- Online, low-memory
- At least one full block of randomness per query
- Can be used to get online authenticated encryption

Thank you for your attention.

Judge a man by his questions rather than his answers. [Voltaire]