Chaffinch: Confidentiality in the Face of Legal Threats

Richard Clayton and George Danezis
University of Cambridge
Information Hiding, 2002

Presenter: Weikun Yang
December 9, 2015

Confidential Message Passing

Confidential Message Passing

Confidential Message Passing

Alice

Confidential Message Passing

Alice

Confidential Message Passing

Alice

Bob

Confidential Message Passing

Alice

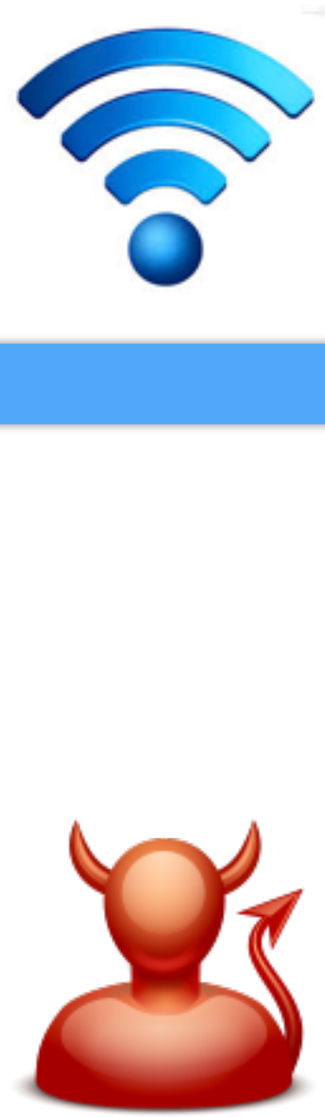
Bob

Confidential Message Passing

Alice

Bob

Confidential Message Passing



Alice

Bob

Confidential Message Passing

Alice

Eve

Bob

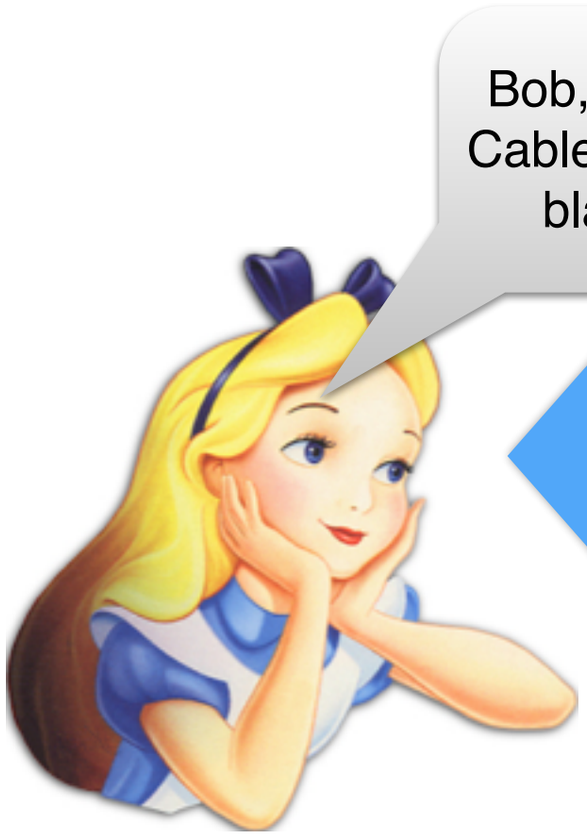
Confidential Message Passing

Alice

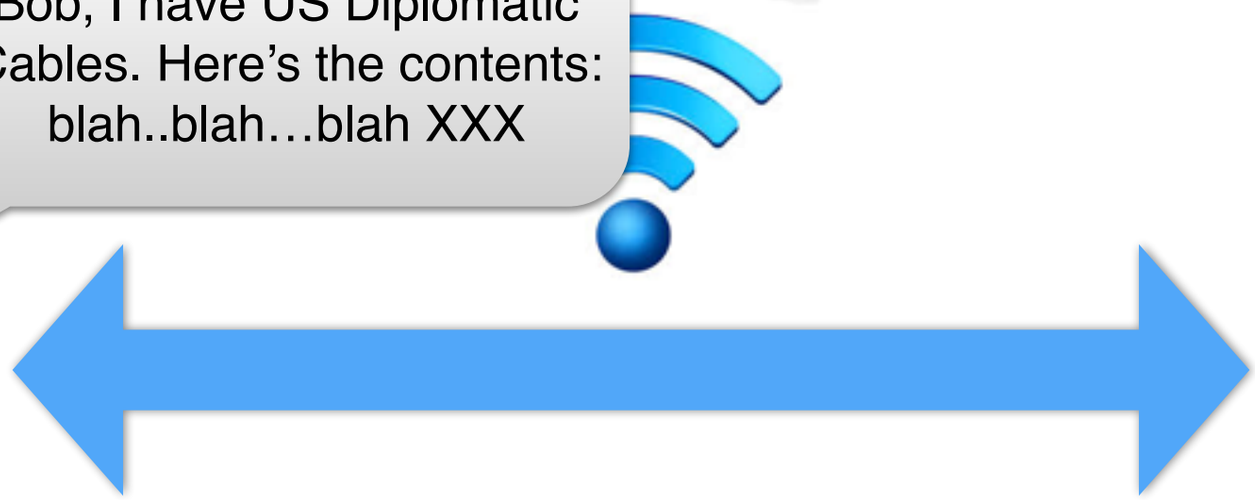
Eve

Bob

Confidential Message Passing



Bob, I have US Diplomatic Cables. Here's the contents: blah..blah…blah XXX

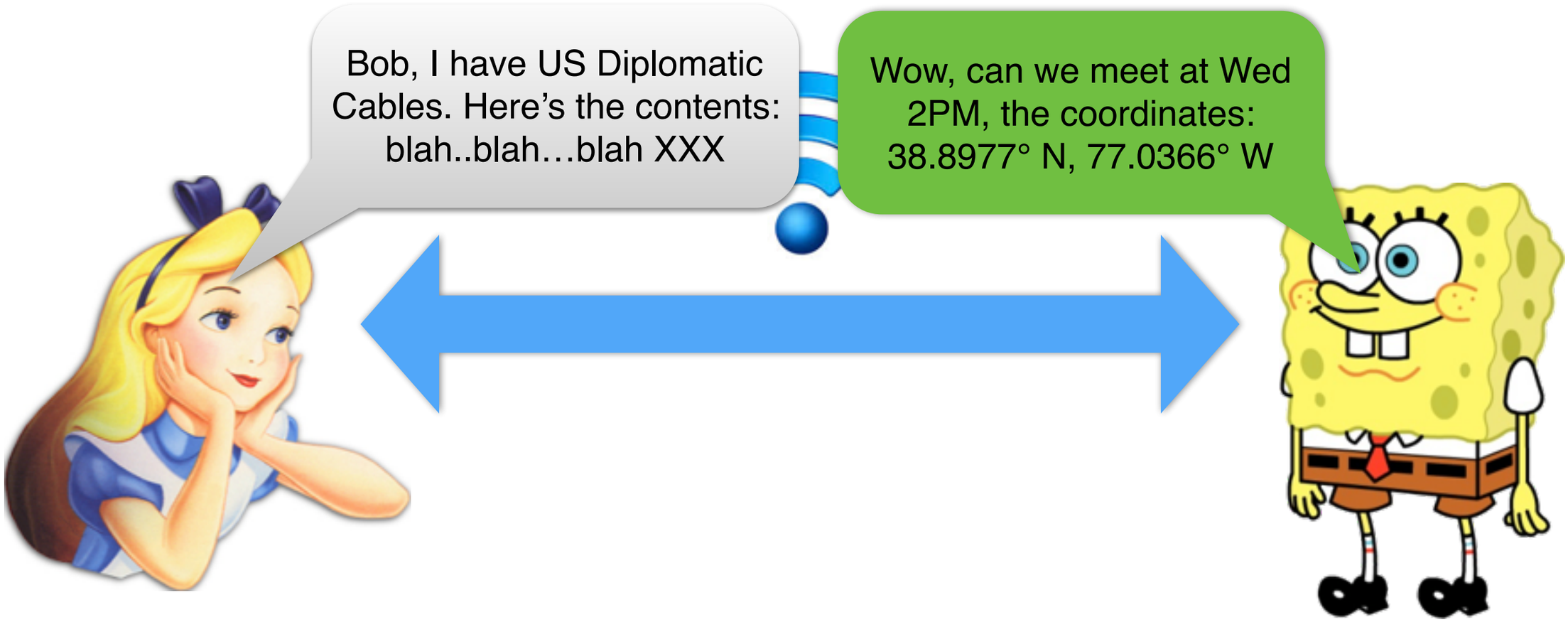


Alice

Eve

Bob

Confidential Message Passing

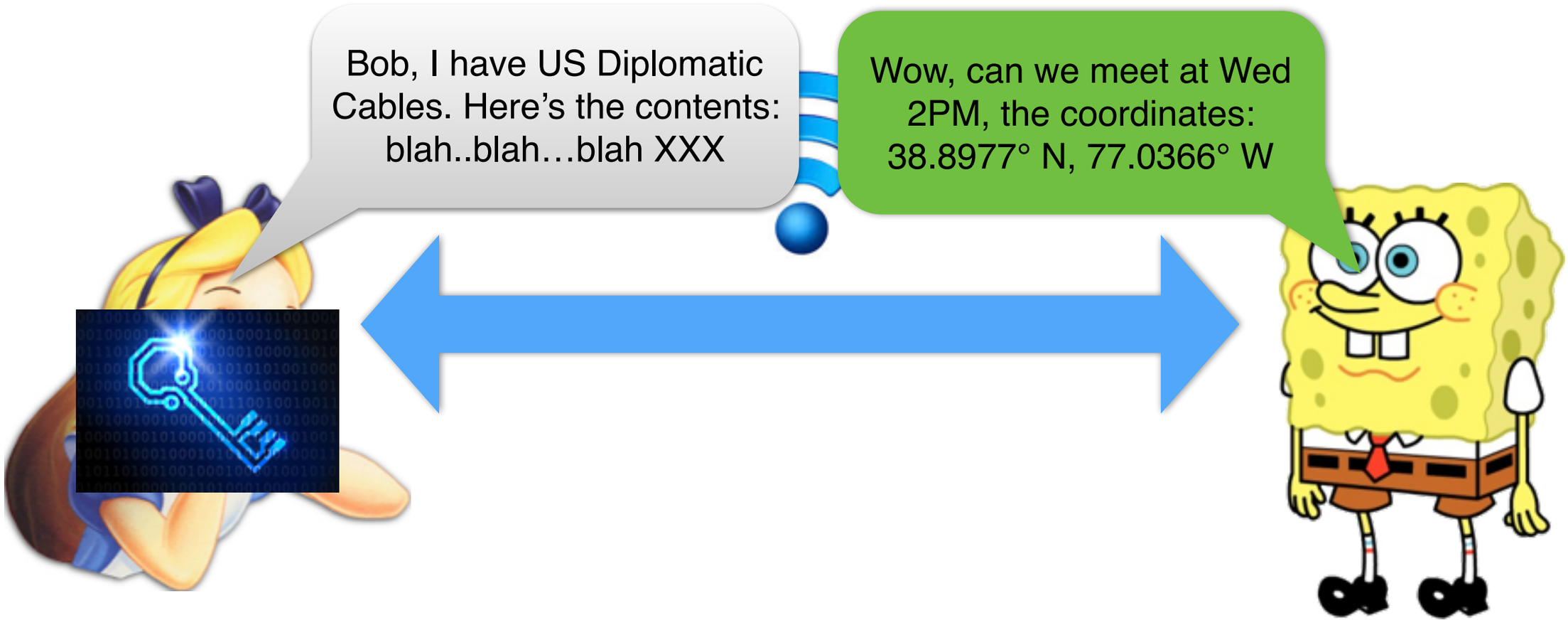


Alice

Eve

Bob

Confidential Message Passing

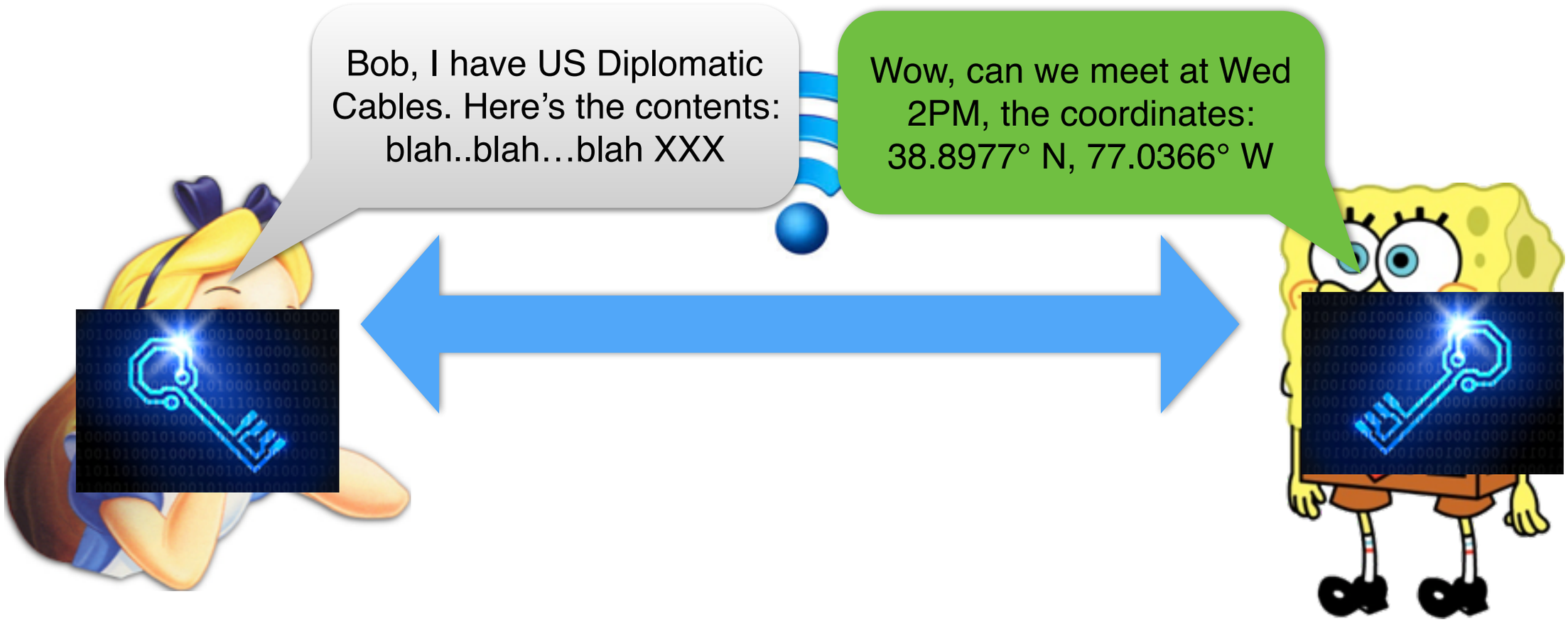


Alice

Eve

Bob

Confidential Message Passing

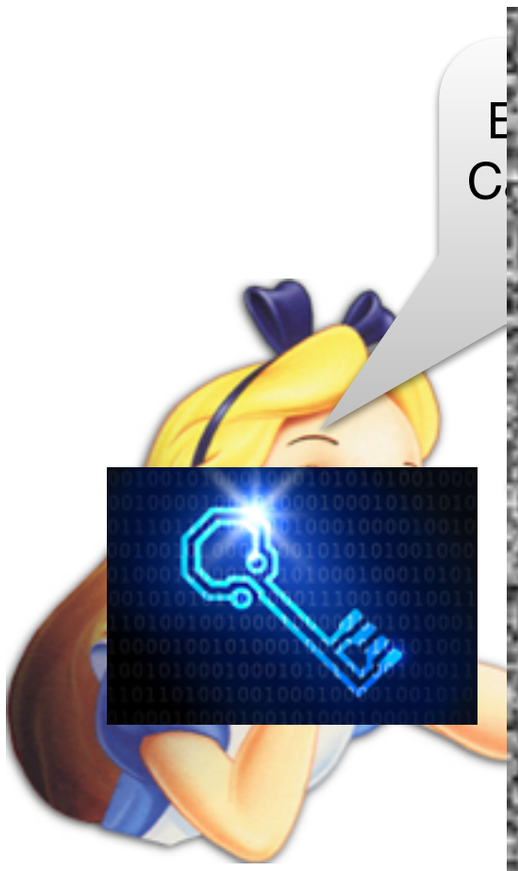


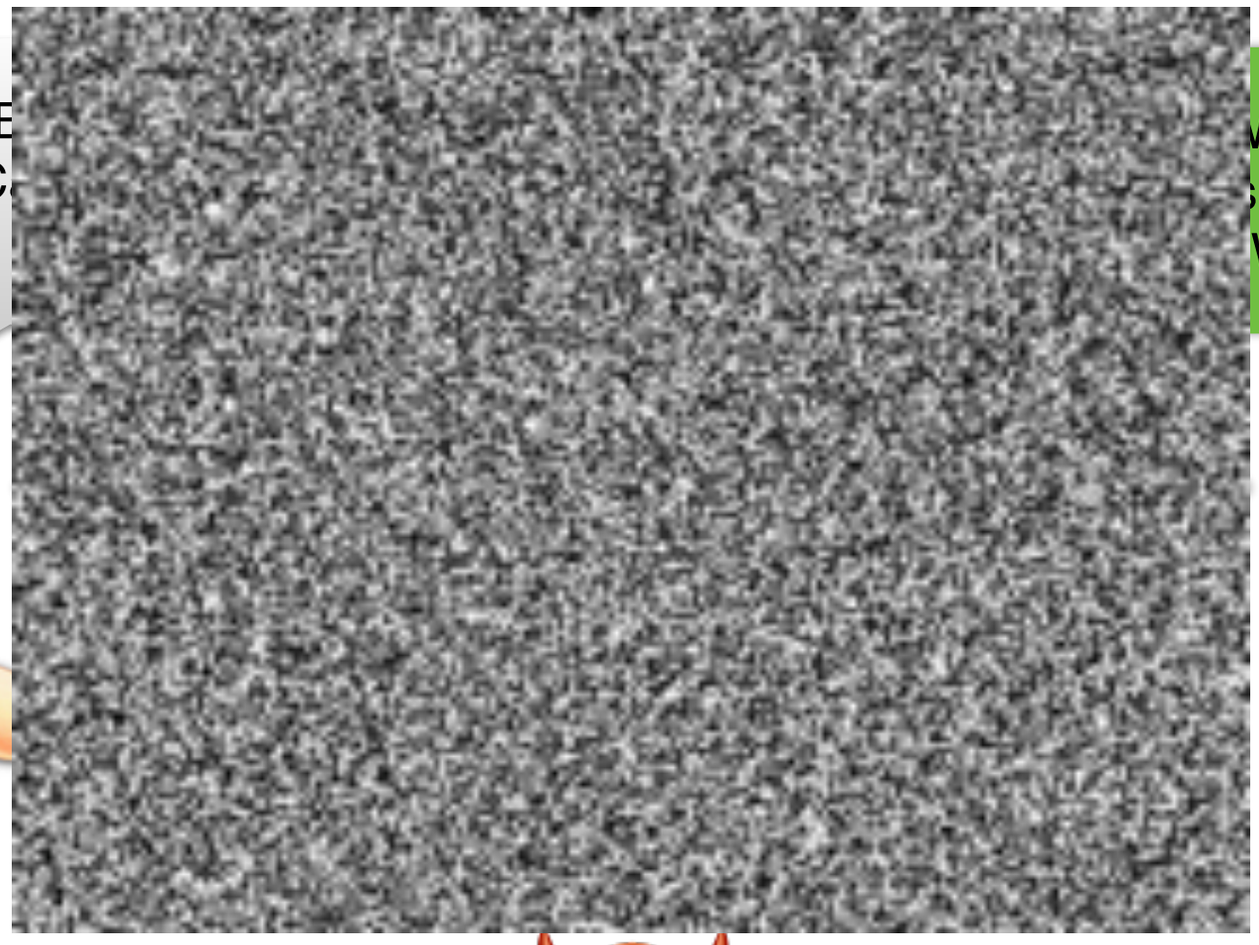
Alice

Eve

Bob

Confidential Message Passing



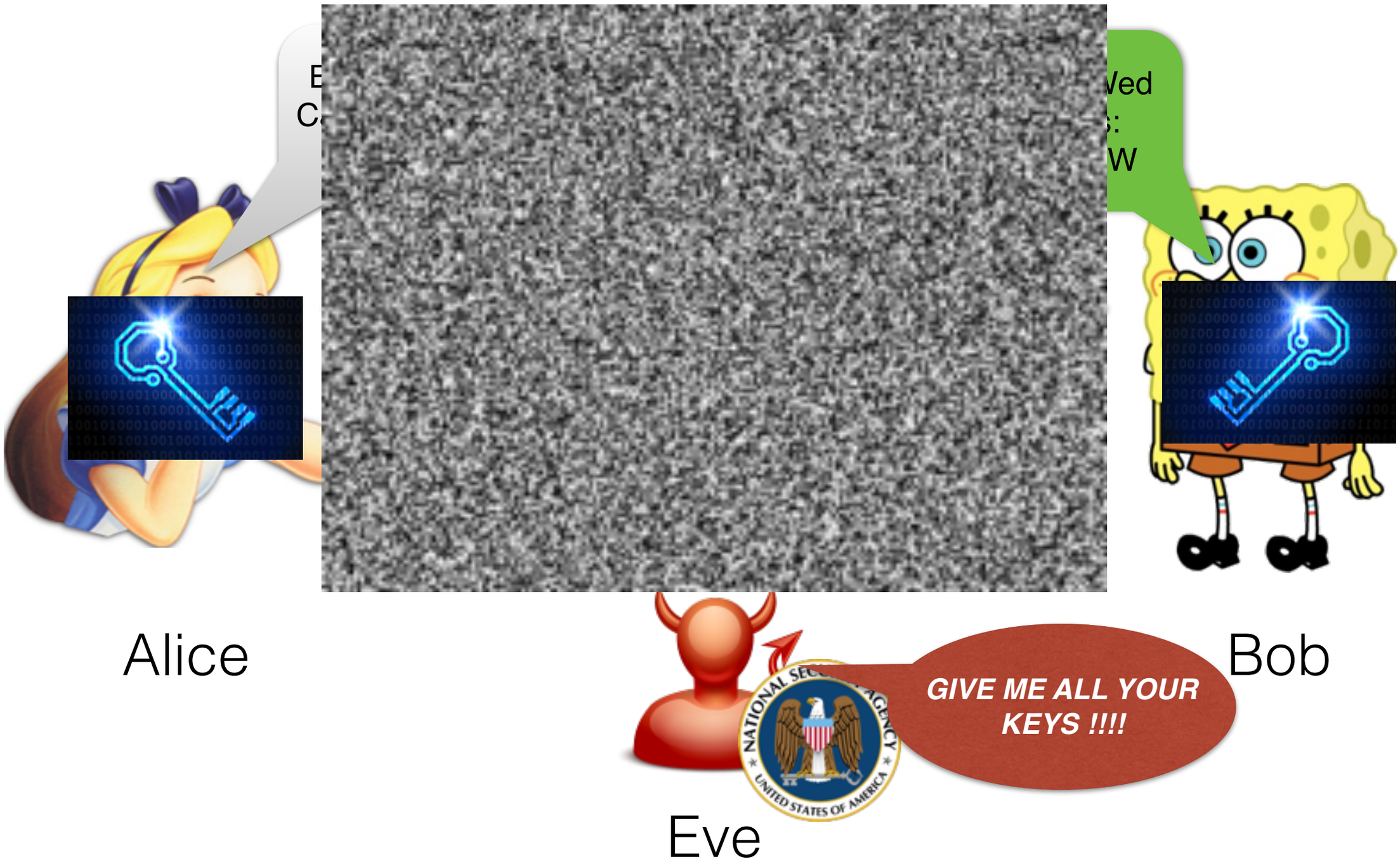


Alice

Eve

Bob

Confidential Message Passing

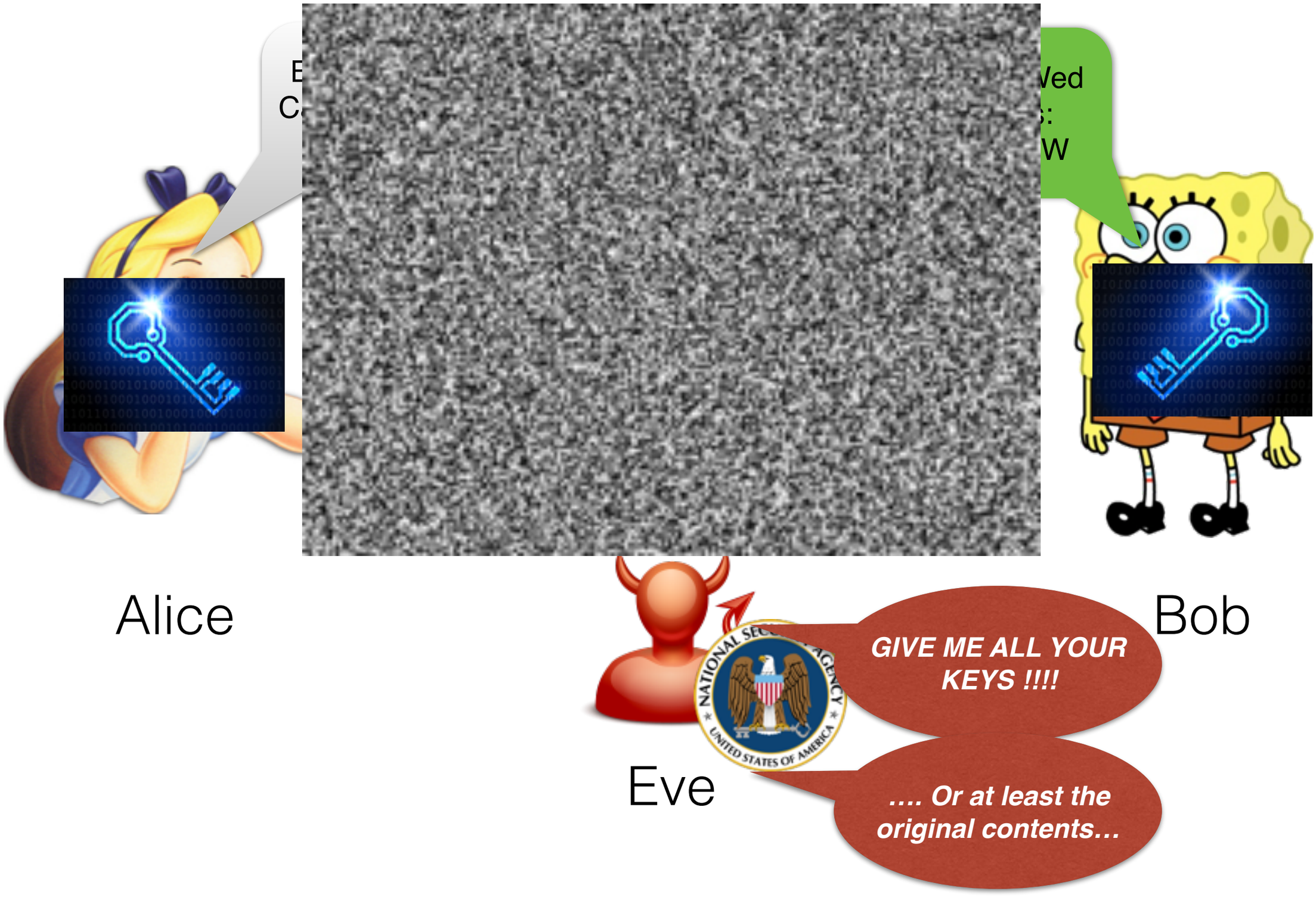


Alice

Bob

Eve

Confidential Message Passing



Goals (plausible deniability)

Goals (plausible deniability)

- **Confidentiality** (transform plaintext into random bits)

Goals (plausible deniability)

- **Confidentiality** (transform plaintext into random bits)

Goals (plausible deniability)

- **Confidentiality** (transform plaintext into random bits)
- **Deny the existence of plaintext** (surrender 2nd key)

Goals (plausible deniability)

- **Confidentiality** (transform plaintext into random bits)

- **Deny the existence of plaintext** (surrender 2nd key)

- **Deny act of encryption** (using authentication only)

Non-goals

Non-goals

- Hide or authenticate Identities

Non-goals

- Hide or authenticate Identities

Non-goals

- Hide or authenticate Identities
- Deny the existence of communication (DenaLi)

Original *Chaffing and Winnowing*

- Hi Bob, Meet me at 7PM Love-Alice

    - (1, Hi Larry, 532105)
    - (1, Hi Bob, 465231)

    - (2, Meet me at, 782290)
    - (2, I'll call you at, 793122)

    - (3, 6PM, 891231)
    - (3, 7PM, 344287)

    - (4, Yours-Susan, 553419)
    - (4, Love-Alice, 312265)

Original *Chaffing and Winnowing*

- Hi Bob, Meet me at 7PM Love-Alice

  - (1, Hi Larry, 532105)
  - (1, Hi Bob, 465231)
  - (2, Meet me at, 782290)
  - (2, I'll call you at, 793122)
  - (3, 6PM, 891231)
  - (3, 7PM, 344287)
  - (4, Yours-Susan, 553419)
  - (4, Love-Alice, 312265)

seq	*msg*	*auth*

Additions by Chaffinch

- **All-or-Nothing transformation** (more randomness, more effort for attacker)

- **Pass multiple messages.**

Chaffinch

Session	
Message 3 Section 1	Message 3 Authenticator
Message 1 Section 1	Message 1 Authenticator
Random	Random
Message 1 Section 2	Message 1 Authenticator
Message 2 Section 1	Message 2 Authenticator
Random	Random
Message 1 Section 3	Message 1 Authenticator

Fig. 1. Conceptual view of a Chaffinch block

Chaffinch

Session	
Message 3 Section 1	Message 3 Authenticator
Message 1 Section 1	Message 1 Authenticator
Random	Random
Message 1 Section 2	Message 1 Authenticator
Message 2 Section 1	Message 2 Authenticator
Random	Random
Message 1 Section 3	Message 1 Authenticator

Fig. 1. Conceptual view of a Chaffinch block

Chaffinch

Session	
Message 3 Section	Message 3 Authenticator
Message 1 Section	Message 1 Authenticator
Random	Random
Message 1 Section 2	Message 1 Authenticator
Message 2 Section 1	Message 2 Authenticator
Random	Random
Message 1 Section 3	Message 1 Authenticator

4byte

Fig. 1. Conceptual view of a Chaffinch block

Chaffinch

Session	
Message 3 Section	Message 3 Authenticator
Message 1 Section	Message 1 Authenticator
Random	Random
Message 1 Section 2	Message 1 Authenticator
Message 2 Section 1	Message 2 Authenticator
Random	Random
Message 1 Section 3	Message 1 Authenticator

4byte

Fig. 1. Conceptual view of a Chaffinch block

Chaffinch

Session	
Message 3 Section	Message 3 Authenti
Message 1 Section	Message 1 Authenti
Random	Random
Message 1 Section 2	Message 1 Authenticator
Message 2 Section 1	Message 2 Authenticator
Random	Random
Message 1 Section 3	Message 1 Authenticator

4byte

10bit

Fig. 1. Conceptual view of a Chaffinch block

Chaffinch

Session	
Message 3 Section	Message 3 Authenti
Message 1 Section	Message 1 Authenti
Random	Random
Message 1 Section 2	Message 1 Authenticator
Message 2 Section 1	Message 2 Authenticator
Random	Random
Message 1 Section 3	Message 1 Authenticator

4byte

10bit

Fig. 1. Conceptual view of a Chaffinch block

Chaffinch

Session	
Message 3 Section	Message 3 Authenti
Message 1 Section	Message 1 Authenti
Random	Random
Message 1 Section 2	Message 1 Authenticator
Message 2 Section 1	Message 2 Authenticator
Random	Random
Message 1 Section 3	Message 1 Authenticator

4byte

10bit

128+ sections

Fig. 1. Conceptual view of a Chaffinch block

Block Construction

- Encode the messages
- Compute the authenticators

Message Generation (BEAR)

Message Generation (BEAR)

- $L \,||\, R <= Nonce \,||\, M$

Message Generation (BEAR)

- $L \,||\, R <= Nonce \,||\, M$
- $L <= \; L \oplus Hash(R)$

Message Generation (BEAR)

- $L \,||\, R <= Nonce \,||\, M$
- $L <= L \oplus Hash(R)$
- $R <= R \oplus PRGen(L)$

Message Generation (BEAR)

- $L \,||\, R <= Nonce \,||\, M$
- $L <= \ L \oplus Hash(R)$
- $R <= R \oplus PRGen(L)$
- $L <= L \oplus Hash(R)$

Authenticator Generation (PRGen + BEAR)

Authenticator Generation (PRGen + BEAR)

- $L \,||\, R <= PRGen(AuthKey \oplus Session)$

Authenticator Generation (PRGen + BEAR)

- $L \,||\, R <= PRGen(AuthKey \oplus Session)$
- $L <= \ L \oplus Hash(R)$

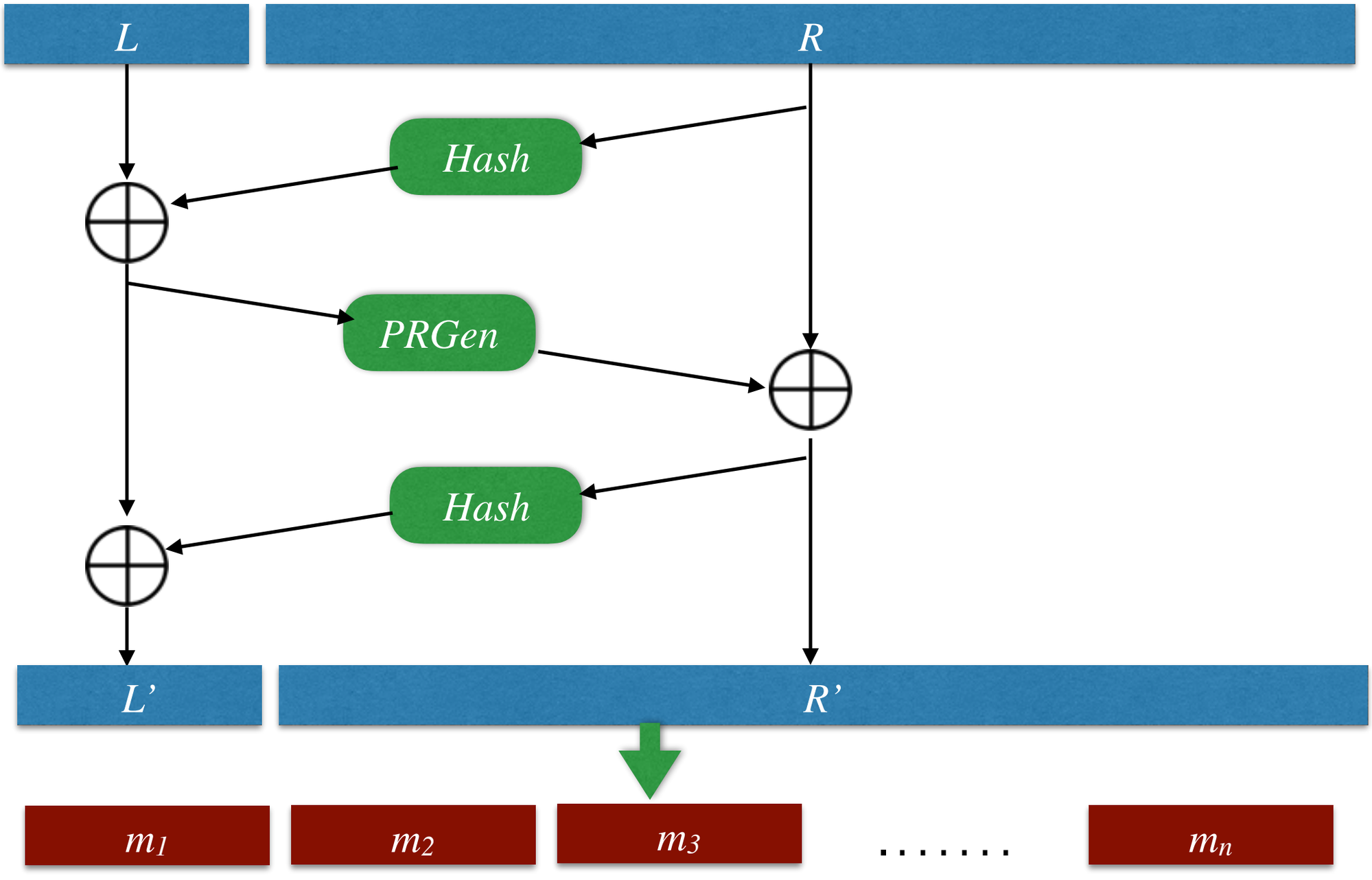
Authenticator Generation (PRGen + BEAR)

- $L \,||\, R <= PRGen(AuthKey \oplus Session)$
- $L <= \ L \oplus Hash(R)$
- $R <= R \oplus PRGen(L)$

Authenticator Generation (PRGen + BEAR)

- $L \,||\, R <= PRGen(AuthKey \oplus Session)$
- $L <= \ L \oplus Hash(R)$
- $R <= R \oplus PRGen(L)$
- $L <= L \oplus Hash(R)$

BEAR



Block Construction

- Choose random arrangement of `(msg, auth)` pairs
- Sections of the same message stay in order
- Prepend with metadata `nonce, session, length, hash(nonce || full msg)`

Message Reconstruction

Message Reconstruction

- 10 bit `auth` give collisions (95% under 128 attempt)

Message Reconstruction

- 10 bit `auth` give collisions (95% under 128 attempt)
- Depth-first search to select correct sections

Message Reconstruction

- 10 bit `auth` give collisions (95% under 128 attempt)
- Depth-first search to select correct sections
- Match received *auth* with actual *auth*

Message Reconstruction

- 10 bit `auth` give collisions (95% under 128 attempt)
- Depth-first search to select correct sections
- Match received *auth* with actual *auth*
- Choose right sequence

Message Reconstruction

- 10 bit `auth` give collisions (95% under 128 attempt)
- Depth-first search to select correct sections
- Match received *auth* with actual *auth*
- Choose right sequence

0, 1, 2, 3, 2, 0, 2, 1, 3, 2, 3, 4

Message Reconstruction

- 10 bit `auth` give collisions (95% under 128 attempt)
- Depth-first search to select correct sections
- Match received *auth* with actual *auth*
- Choose right sequence

0, 1, 2, 3, 2, 0, 2, 1, 3, 2, 3, 4

0, 1, 2, 3, 2, 0, 2, 1, 3, 2, 3, 4

Message Reconstruction

- 10 bit `auth` give collisions (95% under 128 attempt)
- Depth-first search to select correct sections
- Match received *auth* with actual *auth*
- Choose right sequence

0, 1, 2, 3, 2, 0, 2, 1, 3, 2, 3, 4

0, 1, 2, 3, 2, 0, 2, 1, 3, 2, 3, 4

0, 1, 2, 3, 2, 0, 2, 1, 3, 2, 3, 4

Message Reconstruction

- 10 bit `auth` give collisions (95% under 128 attempt)
- Depth-first search to select correct sections
- Match received *auth* with actual *auth*

- Choose right sequence

0, 1, 2, 3, 2, 0, 2, 1, 3, 2, 3, 4

0, 1, 2, 3, 2, 0, 2, 1, 3, 2, 3, 4

0, 1, 2, 3, 2, 0, 2, 1, 3, 2, 3, 4

0, 1, 2, 3, 2, 0, 2, 1, 3, 2, 3, 4

Technical Attacks

Technical Attacks

- `AuthKey` kept secret: Eve doesn't know which sections to look at.

Technical Attacks

- `AuthKey` kept secret: Eve doesn't know which sections to look at.
- `nonce` and `session`: `msg` and `auth` look random, and totally independent

Technical Attacks

- `AuthKey` kept secret: Eve doesn't know which sections to look at.

- `nonce` and `session`: `msg` and `auth` look random, and totally independent

- BEAR transformation: messages are reclaimed "*all-or-nothing*", and maximum effort for brute-force.

Legal Threats

Legal Threats

- When asked “intelligible form”: *deny any encryption*

Legal Threats

- When asked “intelligible form”: *deny any encryption*
- Asked further: *give cover message*

Legal Threats

- When asked “intelligible form”: *deny any encryption*
- Asked further: *give cover message*
- When asked for keys: *give cover keys*

Legal Threats

- When asked “intelligible form”: *deny any encryption*
- Asked further: *give cover message*
- When asked for keys: *give cover keys*
- Rubber-hose cryptanalysis: *give all keys. That’s it.*

Legal Threats

- When asked “intelligible form”: *deny any encryption*
- Asked further: *give cover message*
- When asked for keys: *give cover keys*
- Rubber-hose cryptanalysis: *give all keys. That’s it.*

Legal Threats

- When asked “intelligible form”: *deny any encryption*
- Asked further: *give cover message*
- When asked for keys: *give cover keys*
- Rubber-hose cryptanalysis: *give all keys. That’s it.*
- Consistent behaviors of *BOTH* parties!

Weaknesses

Weaknesses

- Non-goals (ID auth/hiding, key-exchange)

Weaknesses

- Non-goals (ID auth/hiding, key-exchange)
- No implementation, not a complete system

Weaknesses

- Non-goals (ID auth/hiding, key-exchange)
- No implementation, not a complete system
- Probabilistic message recovery: timing attack

Weaknesses

- Non-goals (ID auth/hiding, key-exchange)
- No implementation, not a complete system
- Probabilistic message recovery: timing attack
- Bandwidth and computation overhead

Chaffinch: Confidentiality in the Face of Legal Threats

Richard Clayton and George Danezis
University of Cambridge
Information Hiding, 2002

Presenter: Weikun Yang
December 9, 2015

(improved) “All-or-Nothing” Transformation

$$m'_i = m_i \oplus E(K', i) \quad \text{for } i = 1, \ldots, s$$

$$M = K' \oplus h_1 \oplus h_2 \oplus \ldots \oplus h_s$$

$$h_i = E(K_0, m'_i \oplus Z) \quad \text{for } i = 1, \ldots, s$$

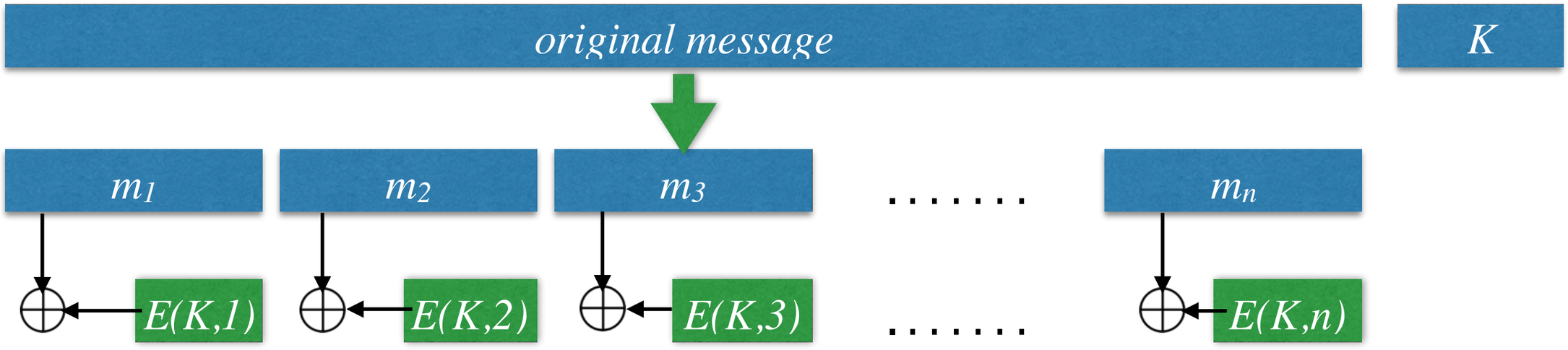
$$Z = \text{HASH}(m'_1, m'_2, \ldots, m'_s)$$

(improved) “All-or-Nothing” Transformation

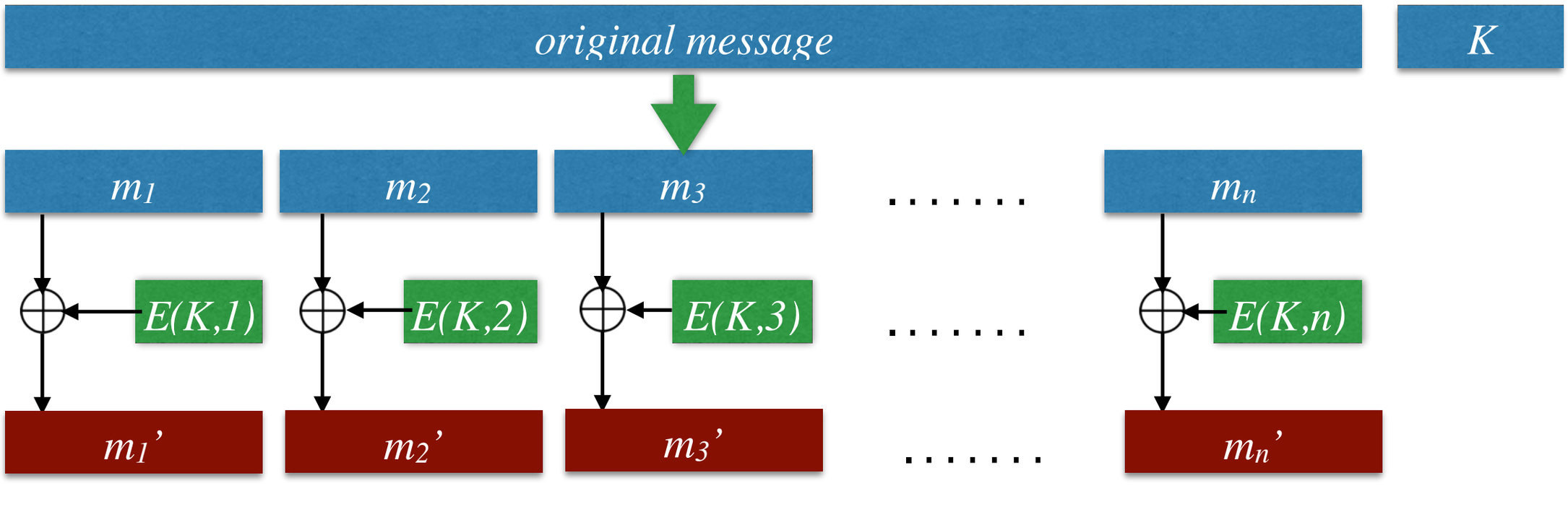
(improved) “All-or-Nothing” Transformation

(improved) “All-or-Nothing” Transformation

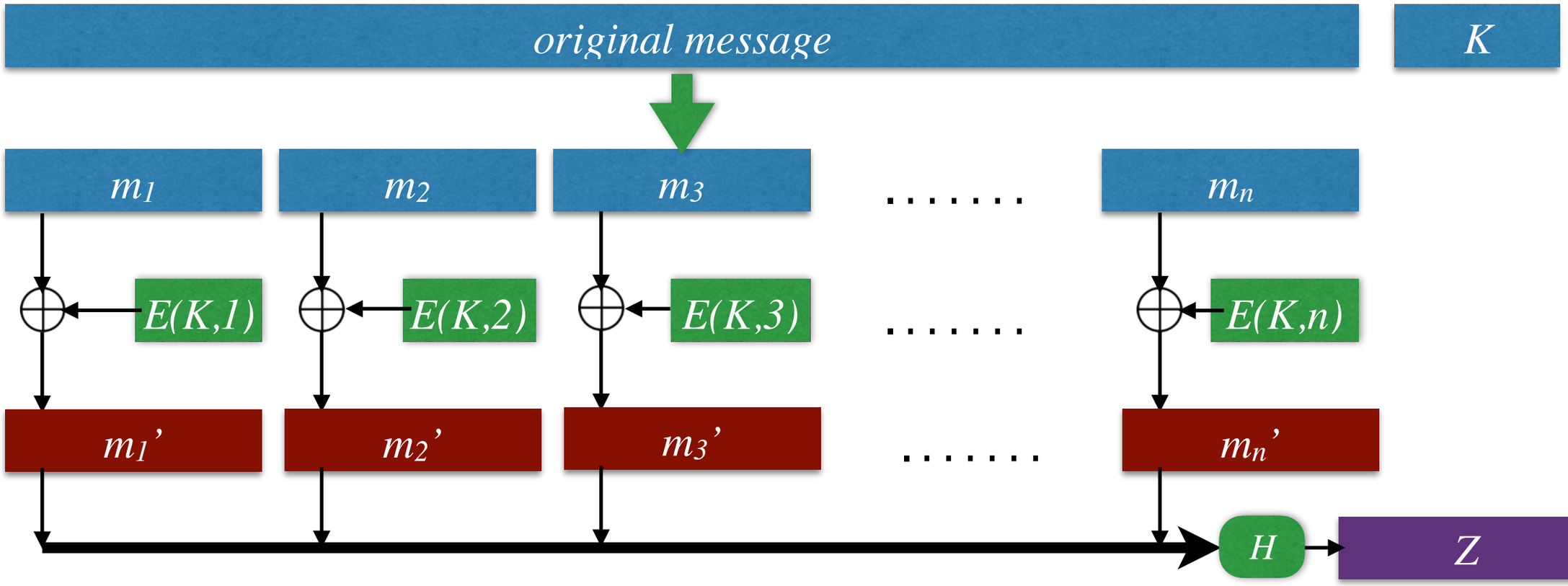
(improved) “All-or-Nothing” Transformation



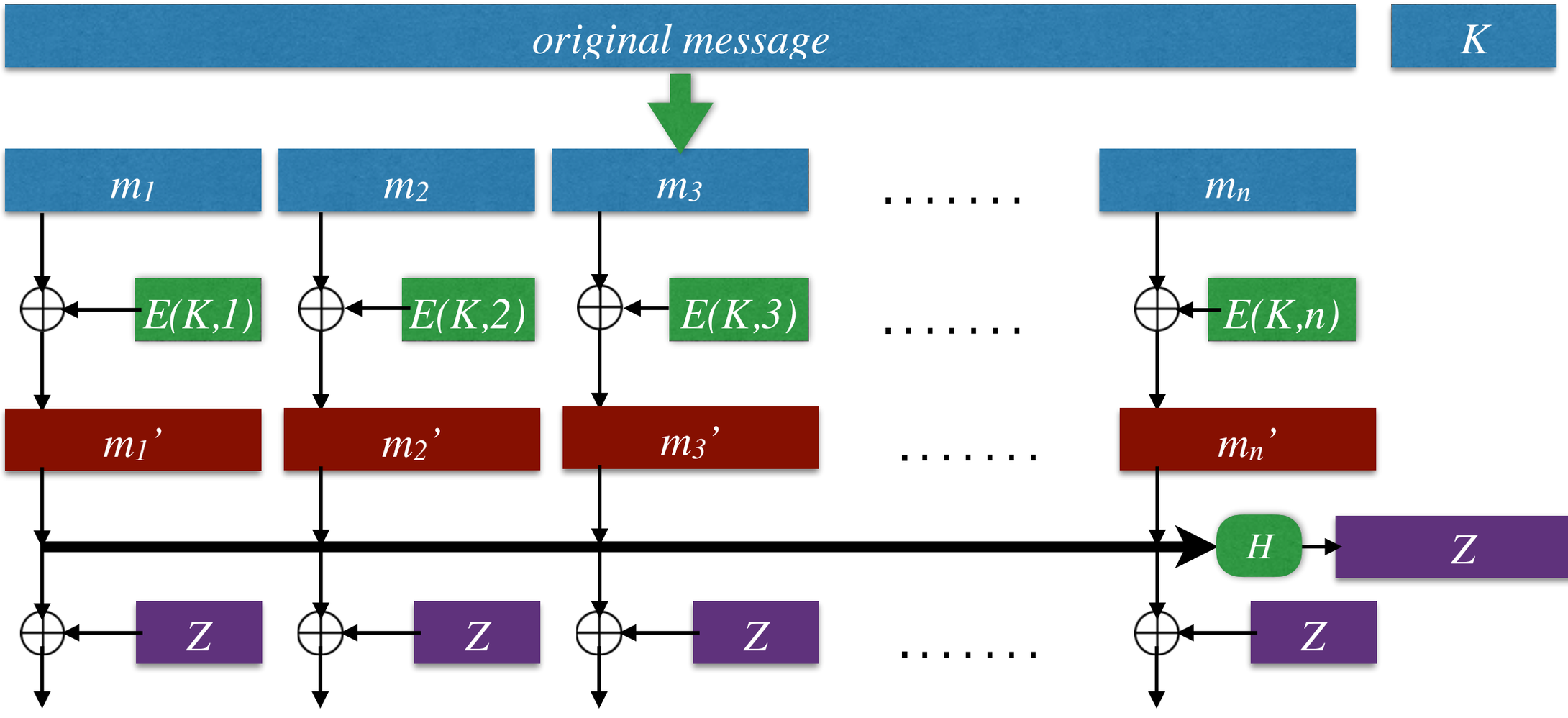
(improved) “All-or-Nothing” Transformation



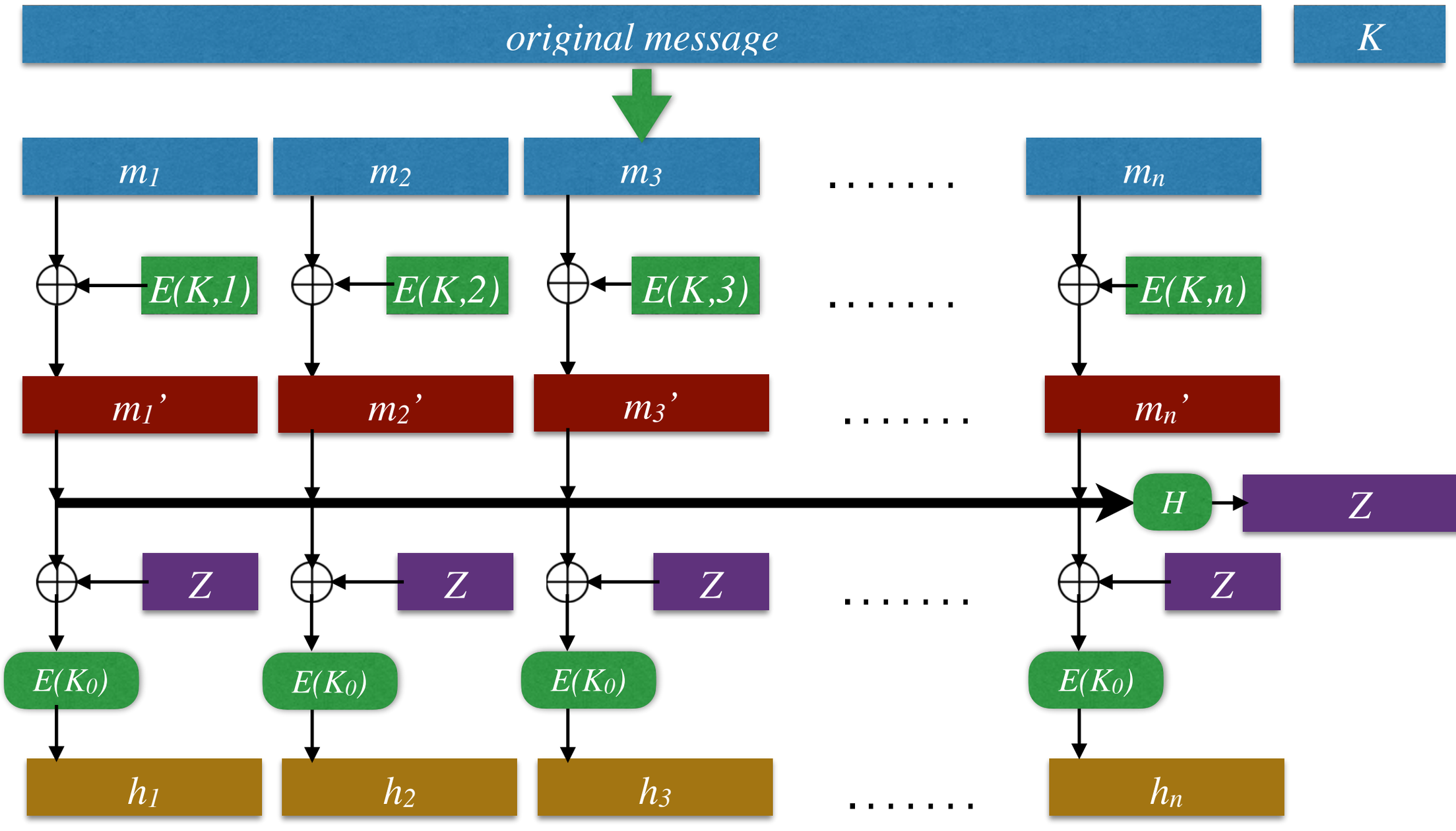
(improved) “All-or-Nothing” Transformation



(improved) “All-or-Nothing” Transformation



(improved) “All-or-Nothing” Transformation



(improved) “All-or-Nothing” Transformation