Information-Theoretic Key Agreement from Close Secrets

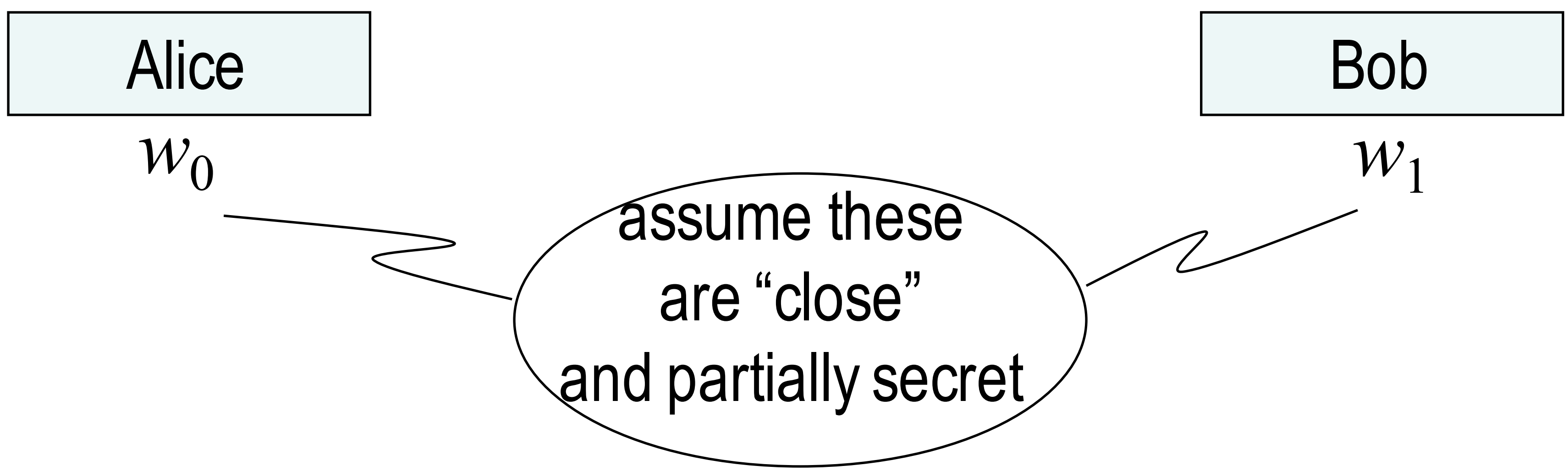
Leonid Reyzin

BOSTON UNIVERSITY

January 5, 2018

IISc

Close Secrets



Close Secrets

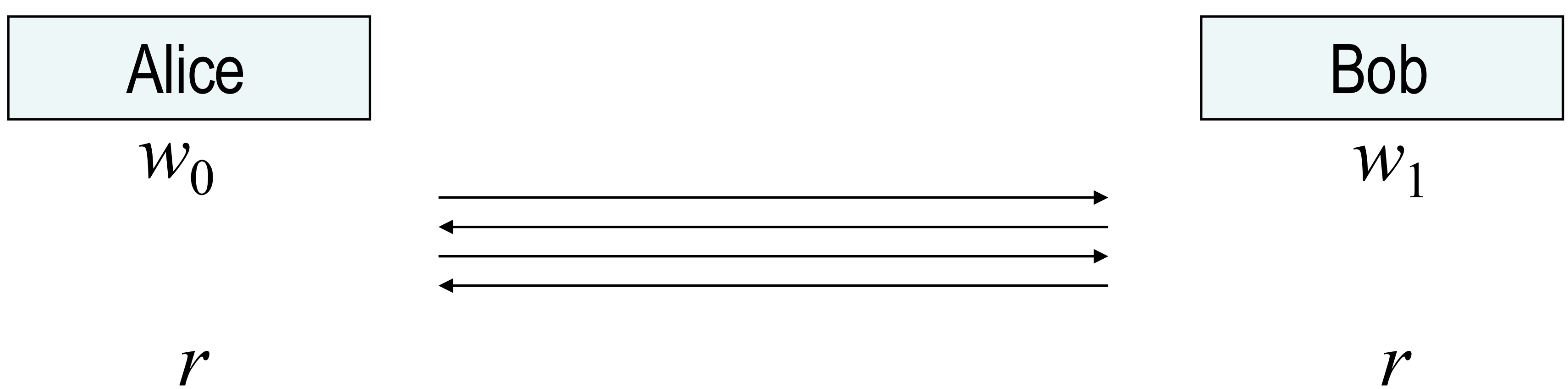
Alice

w_0

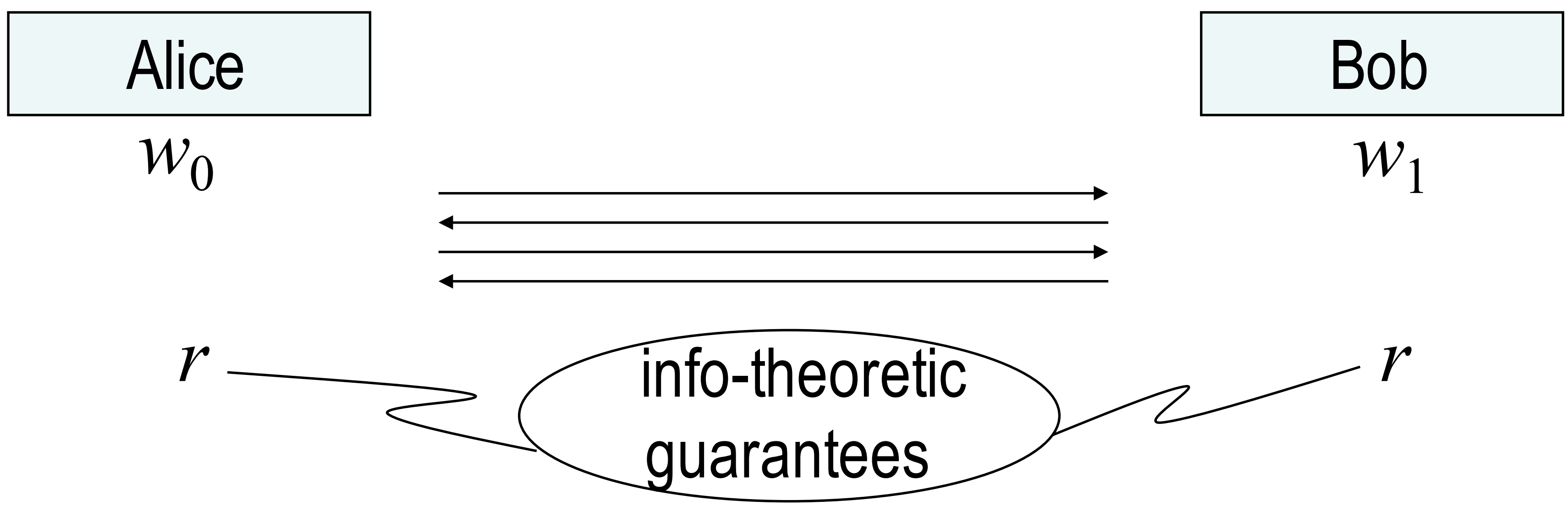
Bob

w_1

Key Agreement from Close Secrets



Information-Theoretic Key Agreement from Close Secrets

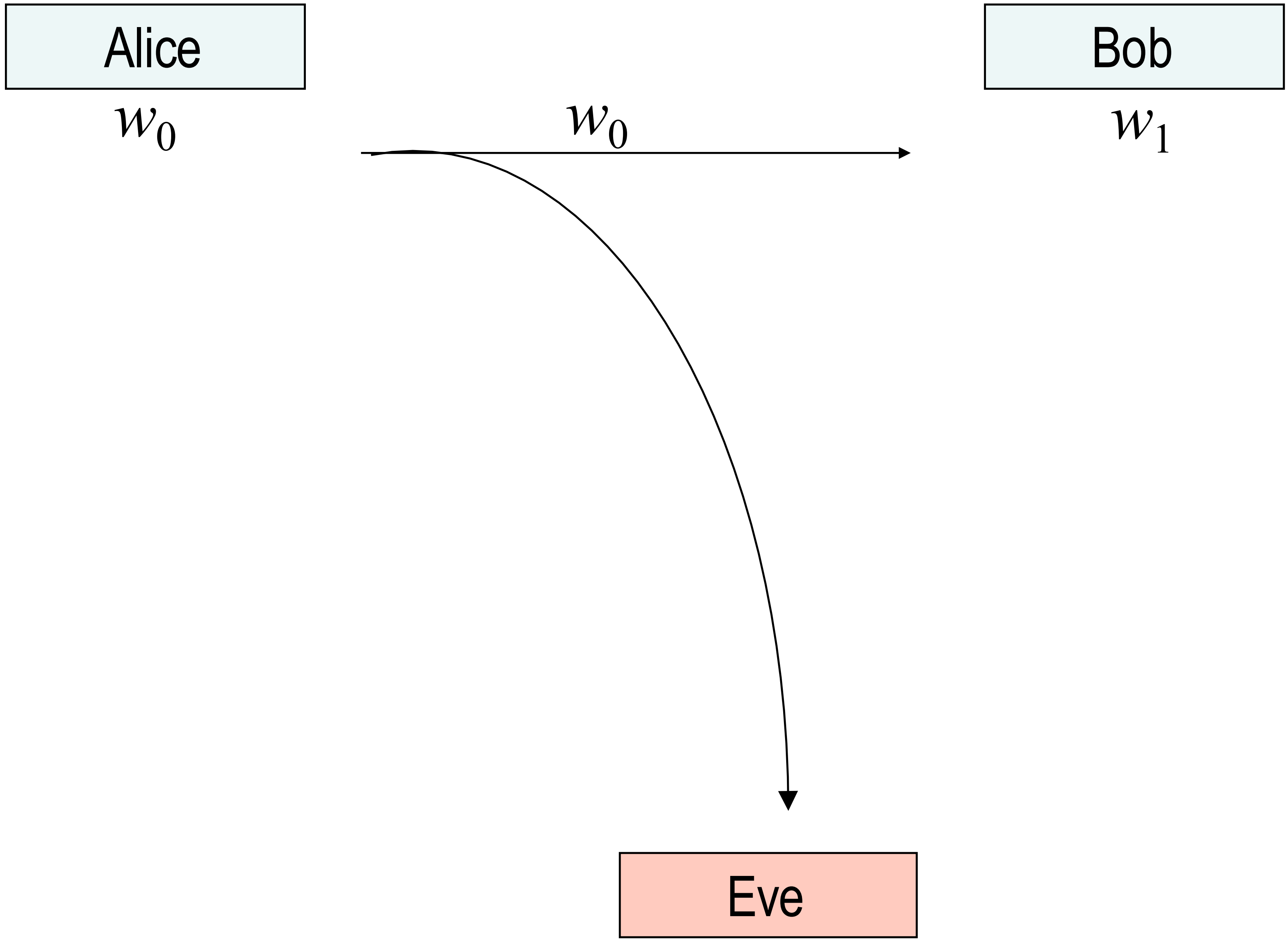


How do we get here?

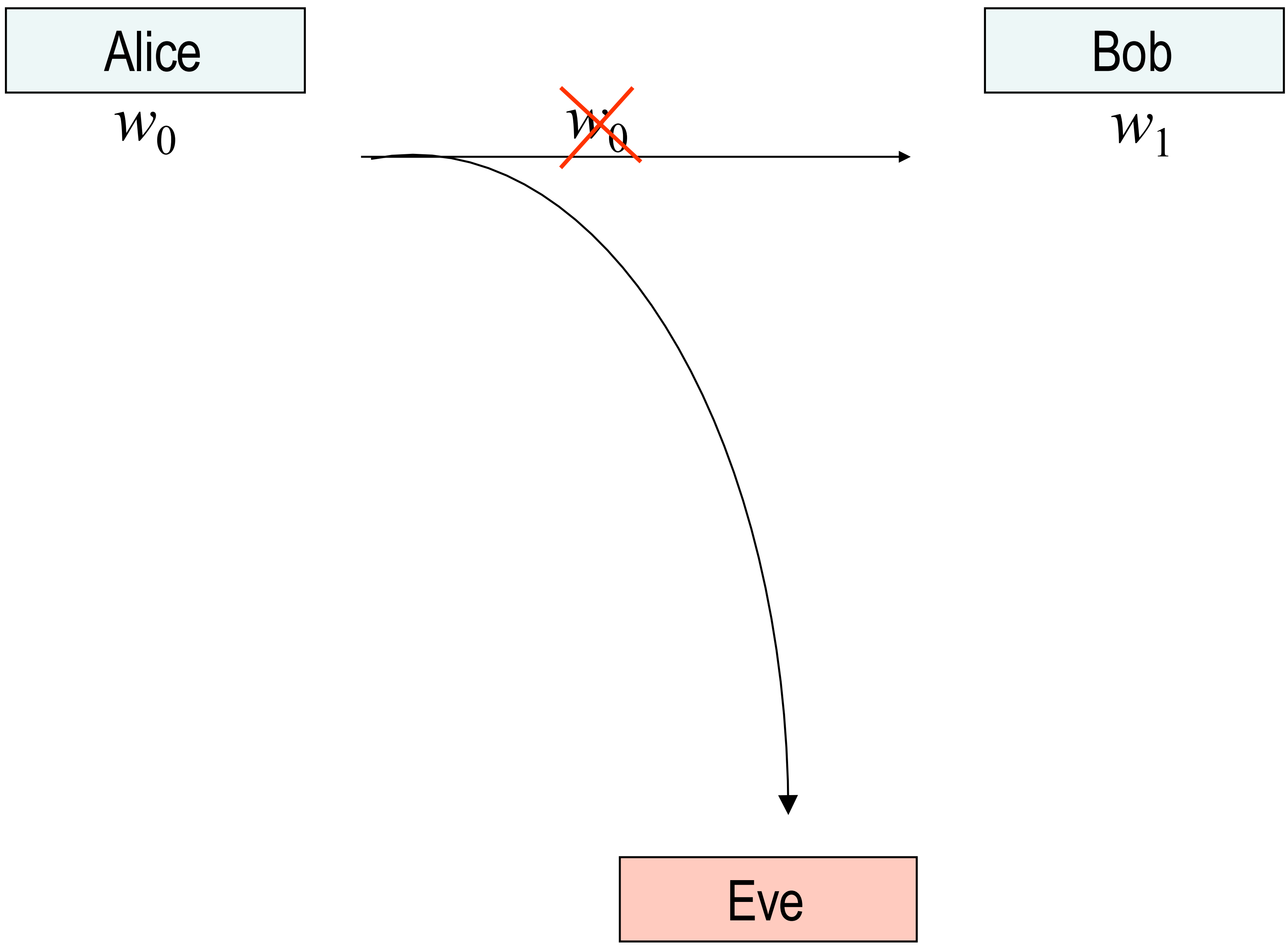
- Alice and Bob have a partially secret and partially noisy channel between them [Wyner 1975]
- Alice and Bob are running quantum key distribution
- Alice and Bob listen to a noisy beacon
- Alice and Bob are two cell phones shaken together
- Alice knows Bob's iris scan

Alice		Bob
w_0		w_1

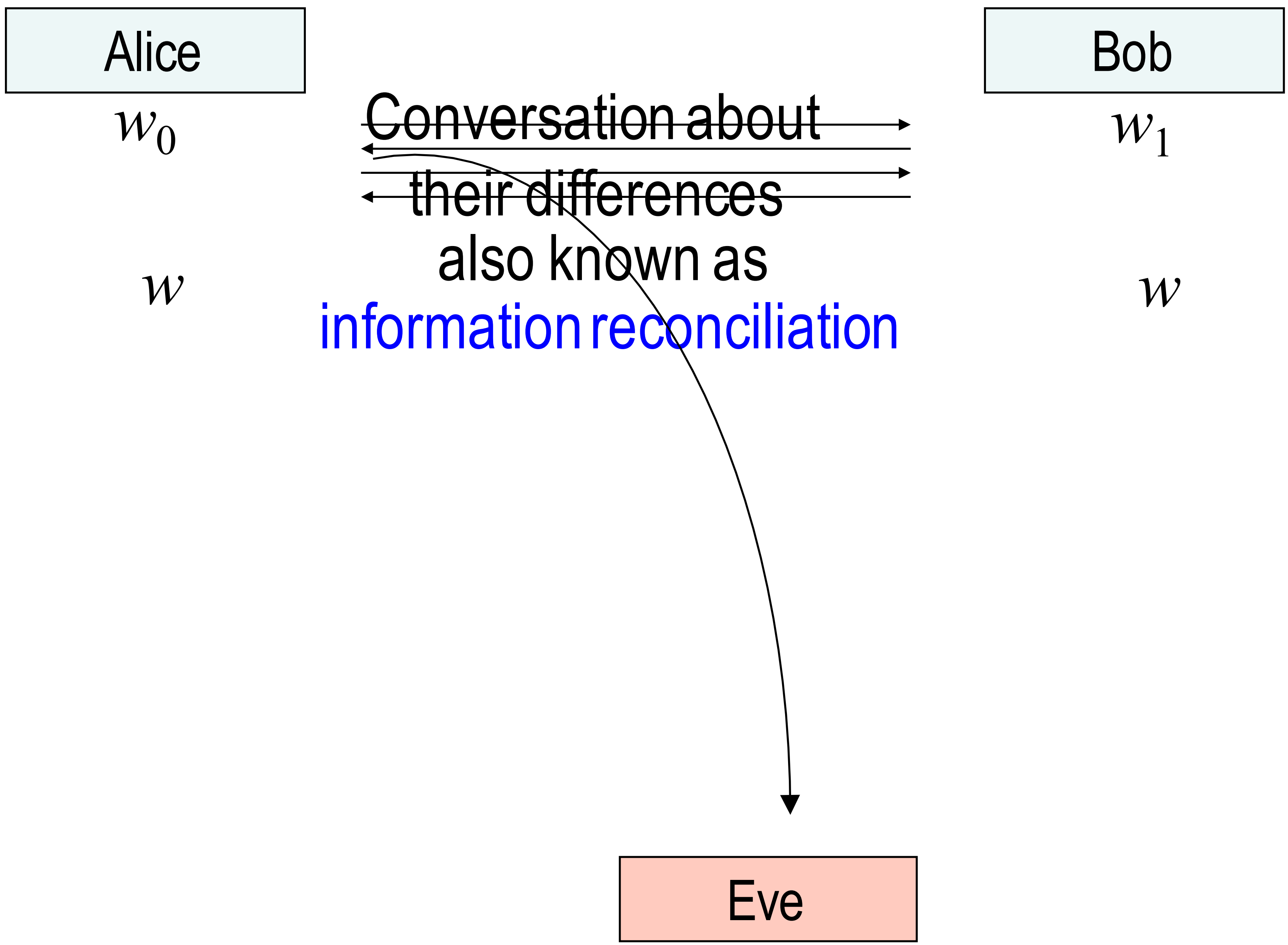
basic paradigm



basic paradigm



basic paradigm: passive adversary



basic paradigm: passive adversary

Alice

w_0

Bob

w_1

Conversation about their differences

also known as

information reconciliation

w w

some information E about w

Eve

basic paradigm: passive adversary

Alice — w_0

Bob — w_1

Conversation about their differences
also known as
information reconciliation

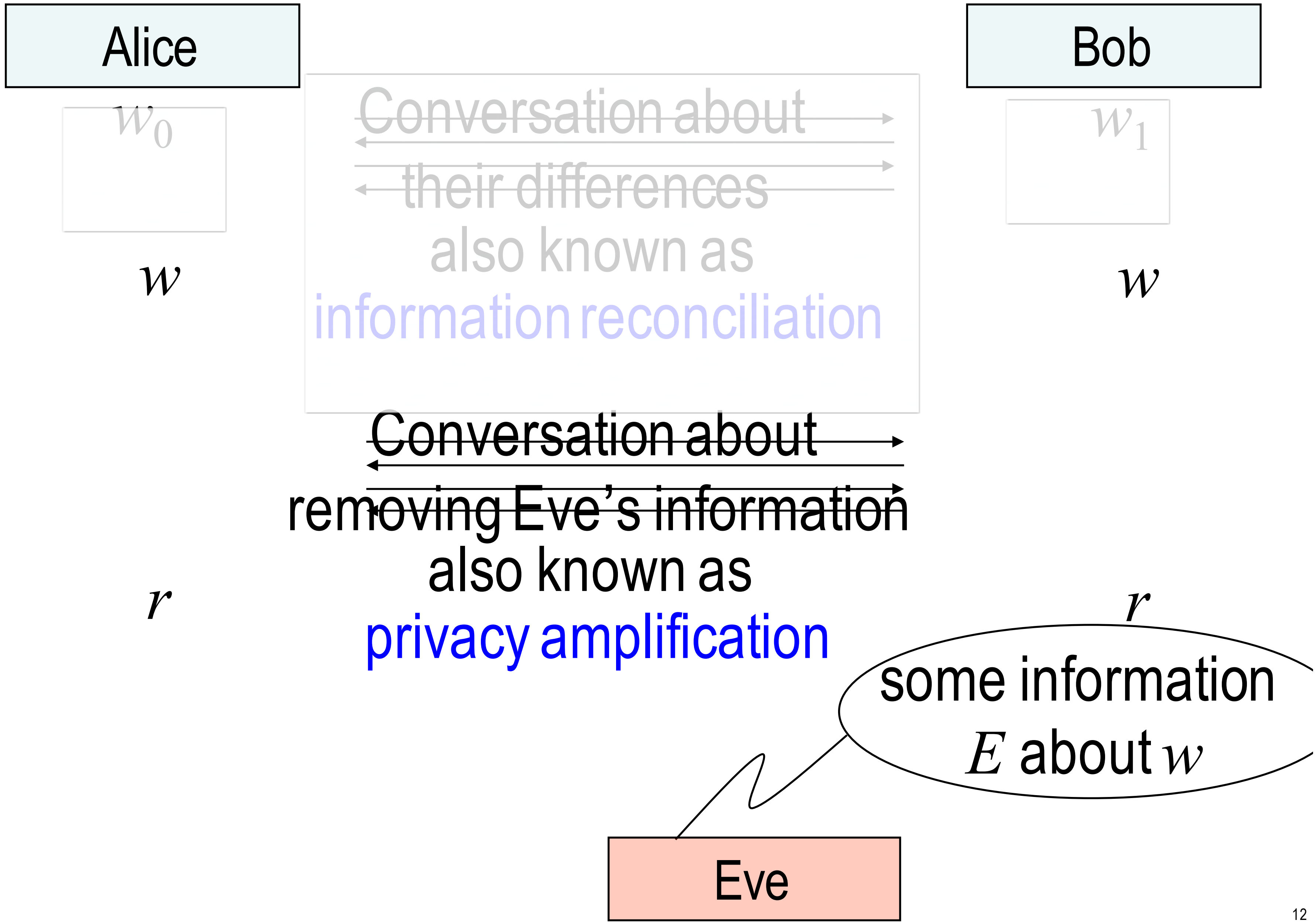
w — w

Conversation about removing Eve's information
also known as
privacy amplification

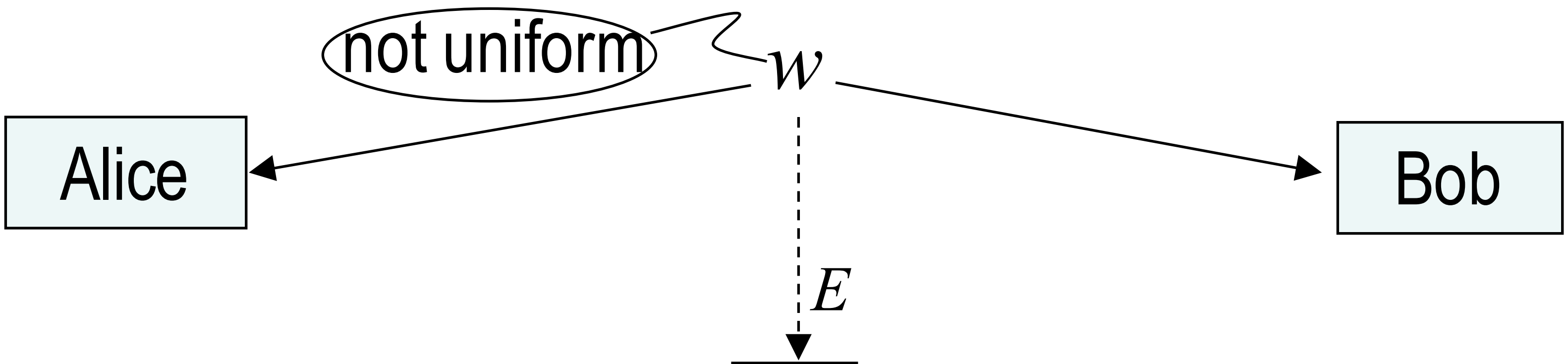
r — r

Eve: some information E about w

basic paradigm: passive adversary



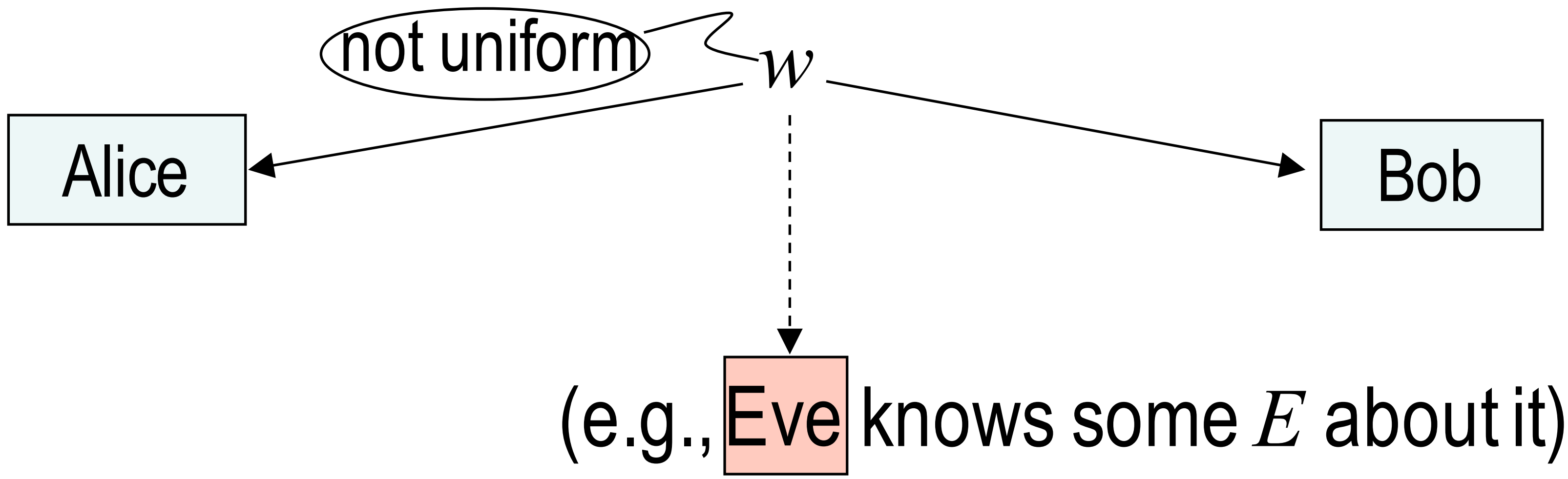
privacy amplification



(e.g., Eve knows some E about it)

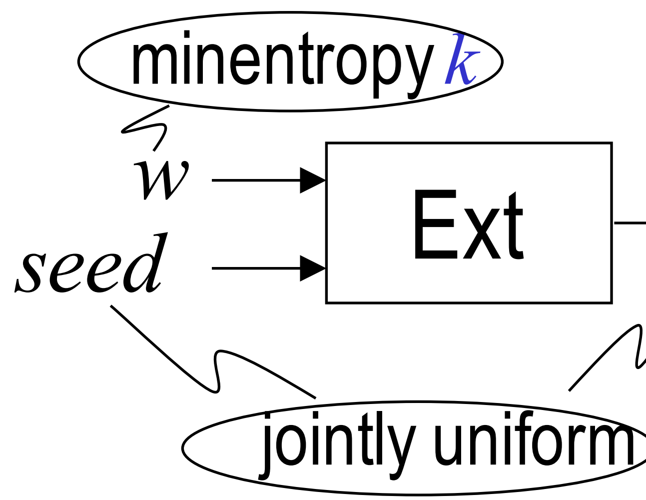
Goal: from a nonuniform secret w
agree on a uniform secret r

privacy amplification

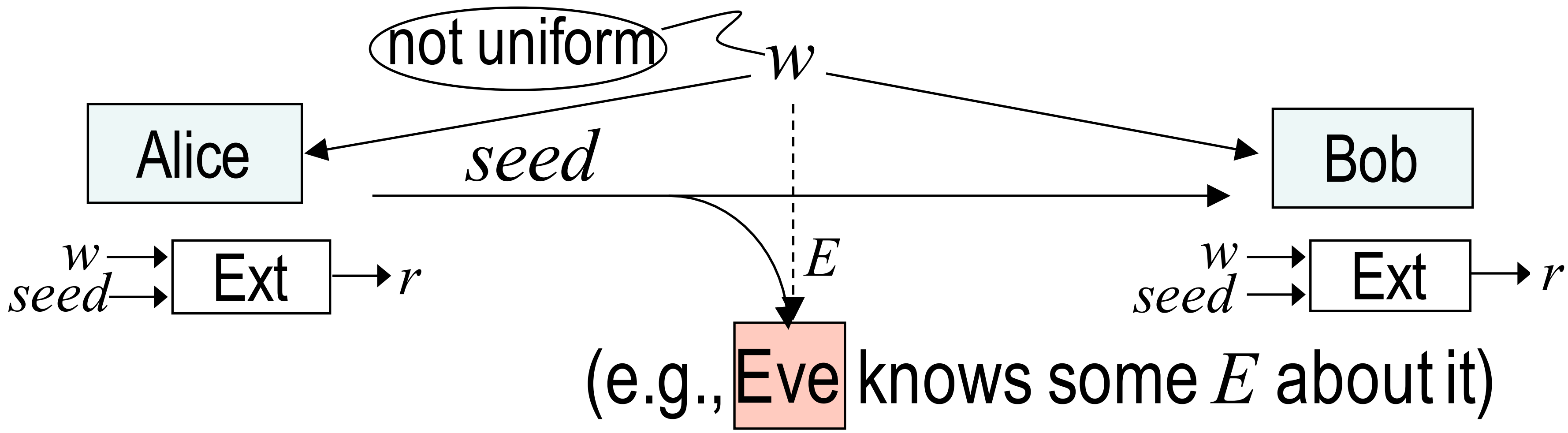


Goal: from a nonuniform secret w
agree on a uniform secret r

Solution: use a strong extractor

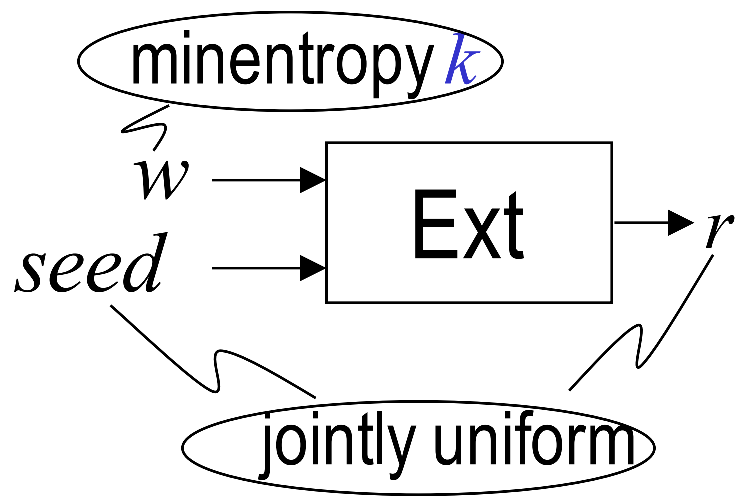


privacy amplification

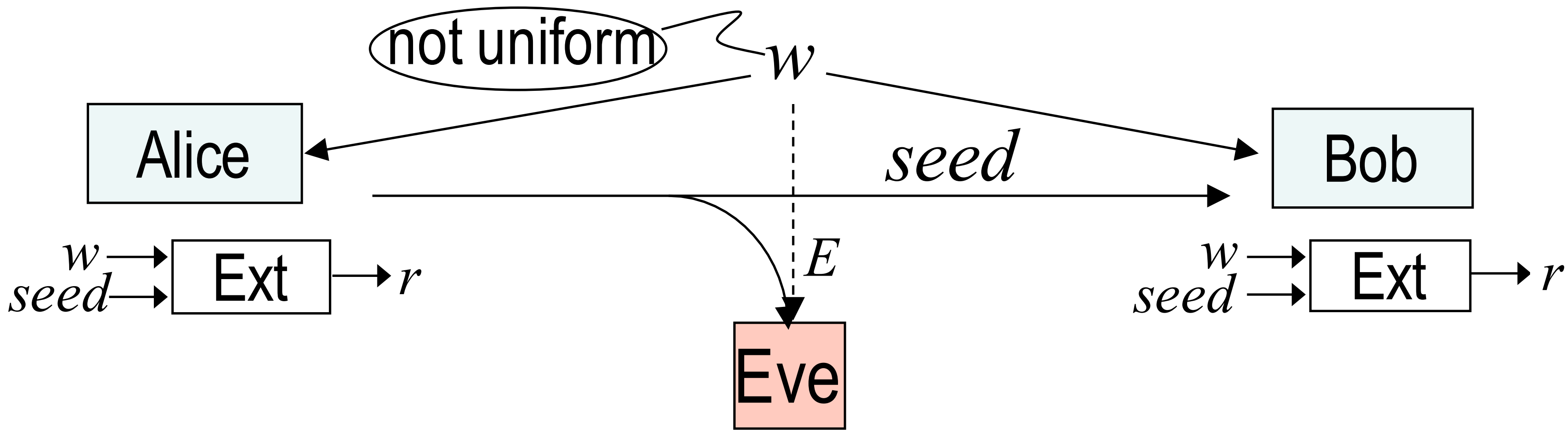


Goal: from a nonuniform secret w
agree on a uniform secret r

Solution: use a strong extractor



privacy amplification

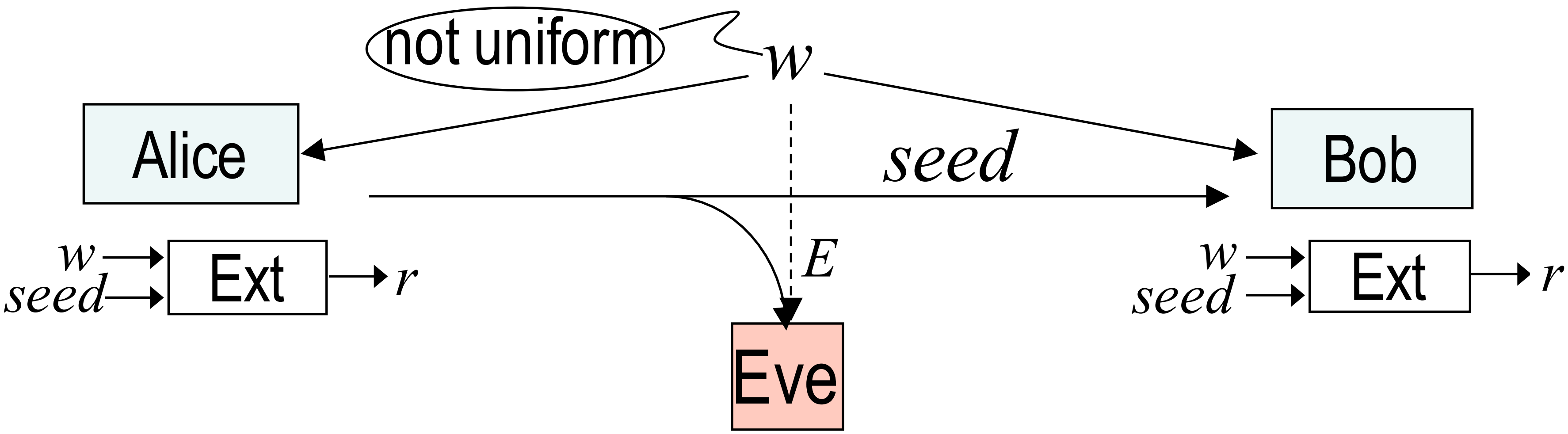


If average min-entropy $H_{\min}(W\,|E)$ is sufficiently high, and Ext is an average-case strong extractor, this works!

<u>Using universal hashing</u>:

If $H_{\min}(W|E) \geq k$, we get $(R, Seed, E) \approx_\varepsilon (U_m, Seed, E)$ for $m = k - 2\log(1/\varepsilon)$

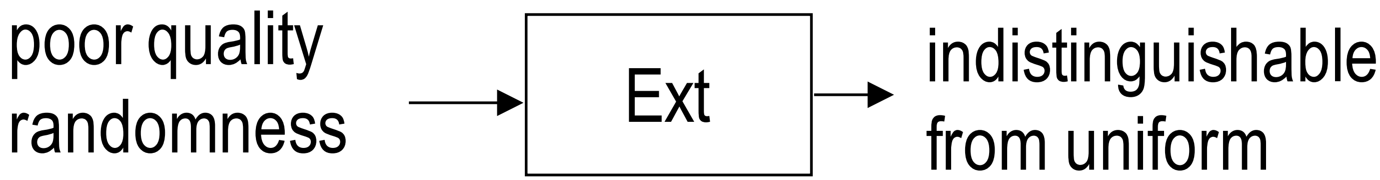
privacy amplification



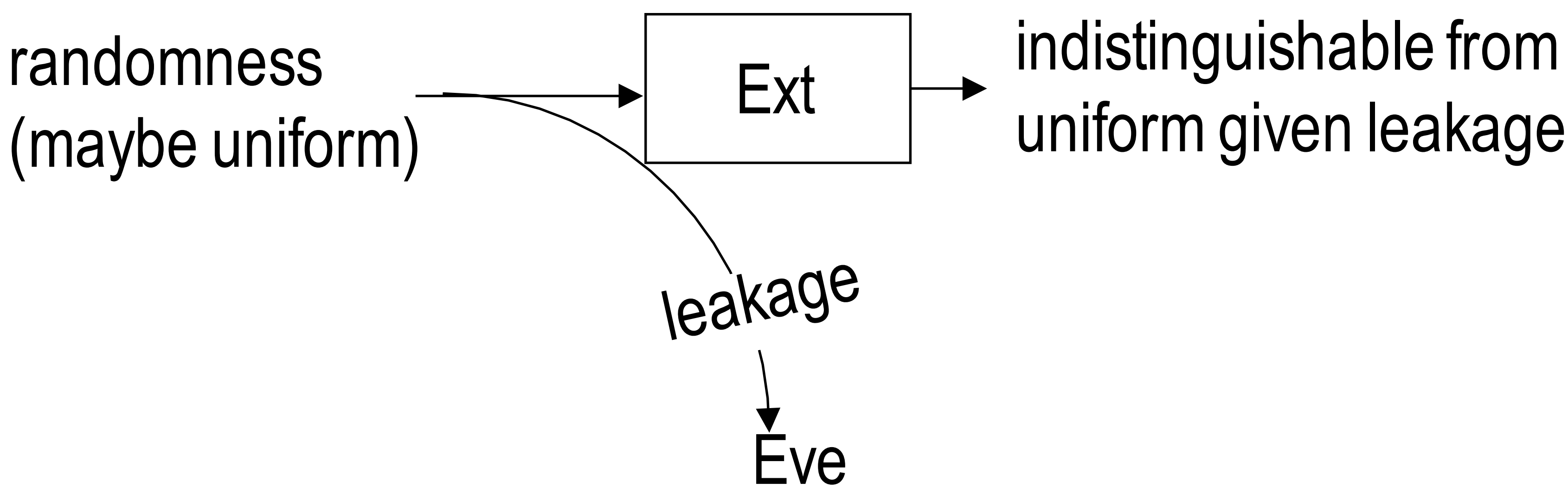
- Early work for specific distributions of w and classes of Eve's knowledge, motivated by quantum key agreement
- [Ozarow-Wyner 84]: nonconstructive solution
- [Bennett-Brassard-Robert 85]: universal hashing <u>for any</u> Eve's knowledge
- Early analysis used Shannon entropy for W as an input assumption and low mutual information between E and R as an output guarantee. Problem: Shannon entropy and mutual information are not great for security
- [Maurer 93, Bennett-Brassard-Crépeau-Maurer 95]: modern security notions

note the two views of extractors

[Santha-Vazirani]:



[Wyner]:



The equivalence of these two views wasn't obvious at first

basic paradigm: passive adversary

Alice

w_0

w

r

Bob

w_1

w

r

Conversation about

their differences

also known as

information reconciliation

Conversation about

removing Eve's information

also known as

privacy amplification

Eve

Outline

- Passive adversaries
  - Privacy amplification
  - Information reconciliation
- Active adversaries, w has a lot of entropy
  - Privacy amplification
  - Information reconciliation
- Active adversaries, w has little entropy
  - Privacy amplification
  - Information reconciliation

basic paradigm: passive adversary

Alice

w_0

Conversation about their differences

also known as

information reconciliation

Bob

w_1

w w

seed to a strong extractor

r r

Eve

basic paradigm: passive adversary

Alice

w_0

Conversation about their differences

also known as

information reconciliation

w

Bob

w_1

w

seed to a strong extractor

r

r

Eve

basic paradigm: passive adversary

Alice

w_0

w

~~Conversation about~~
~~their differences~~
also known as
information reconciliation

Bob

w_1

w

seed to a strong extractor

Goal: minimize amount of information leaked about w,
i.e., maximize $H_{\min}(W|\text{protocol messages})$

Eve

information reconciliation

Alice

w_0

focus today: single-message,
starting with Bennett-Brassard-Robert 85
(interactive protocols more rare
e.g., Brassard-Salvail 93)

Bob

w_1

w

w

Goal: minimize amount of information
leaked about w,
i.e., maximize $H_{\min}(W|\text{protocol messages})$

Eve

information reconciliation

Alice

w_0

focus today: single-message,
starting with Bennett-Brassard-Robert 85
(interactive protocols more rare
e.g., Brassard-Salvail 93)

Bob

w_1

Sketch(w_0)

w

w

Goal: minimize amount of information
leaked about w,
i.e., maximize $H_{\min}(W|\text{protocol messages})$

Eve

Aside: chain rule for $H_{\min}$

Def: $H_{\max}(E) = \log |\{e \mid \Pr[E = e]>0\}| = \log |\text{support}(E)|$

<u>Lemma</u>: $H_{\min}(X \mid E) \geq H_{\min}(X, E) - H_{\max}(E)$

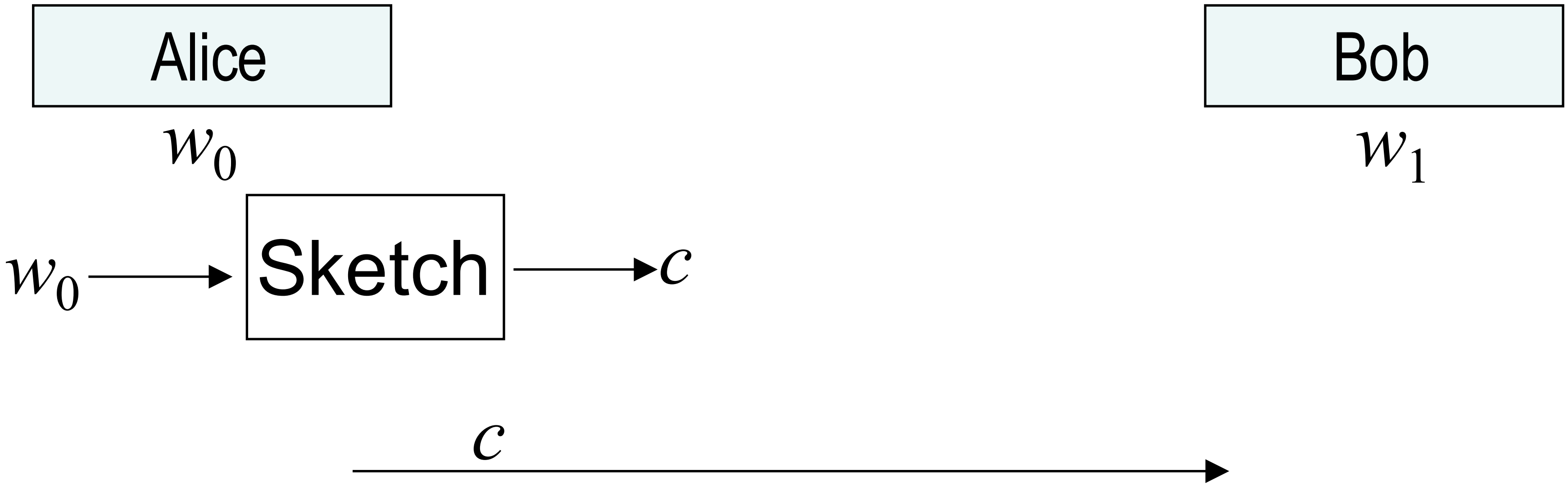
<u>Proof</u>: Reduction. Suppose $\Pr_{(x,e)}[A(e) \rightarrow x] = p$.

Let $B =$ pick a uniform g support(E); output $(A(g), g)$

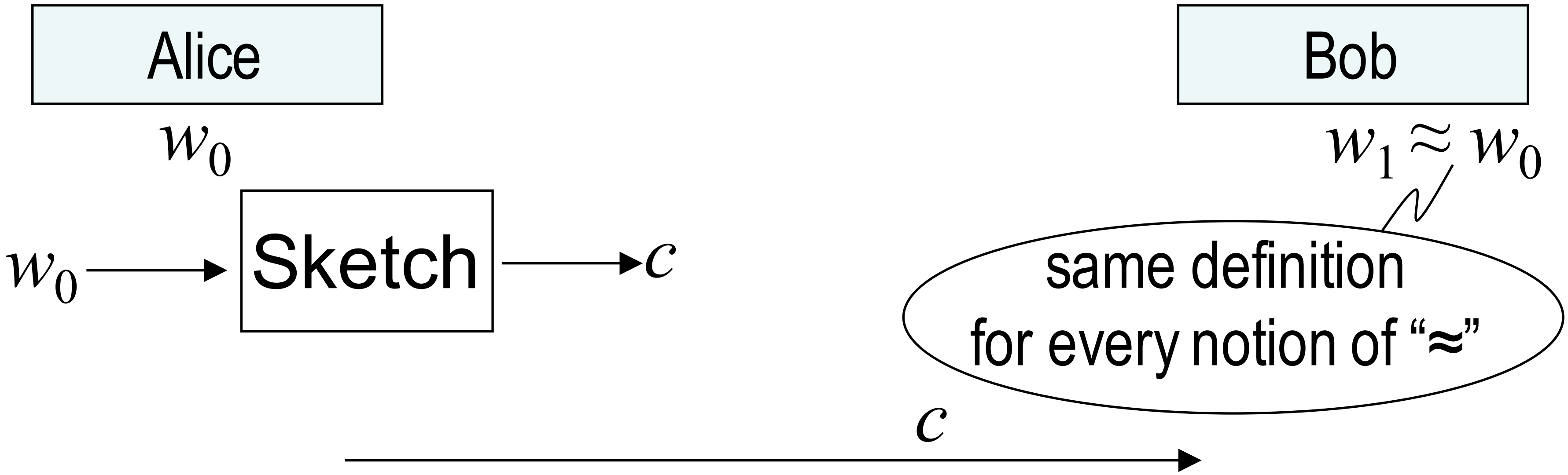
$$
\begin{aligned}
\Pr_{(x,e)}[B \rightarrow (x,e)] &\geq \Pr_{(x,e,g)}[e{=}g \text{ and } A(g) \rightarrow x] \\
&= \Pr_{(x,e,g)}[e{=}g \text{ and } A(e) \rightarrow x] \\
&= \Pr_{(x,e,g)}[e{=}g] \Pr_{(x,e,g)}[A(e) \rightarrow x] \\
&= p/|\text{support}(E)| \qquad \square
\end{aligned}
$$

<u>Lemma</u>: $H_{\min}(X \mid E_1, E_2) \geq H_{\min}(X, E_2 \mid E_1) - H_{\max}(E_2)$

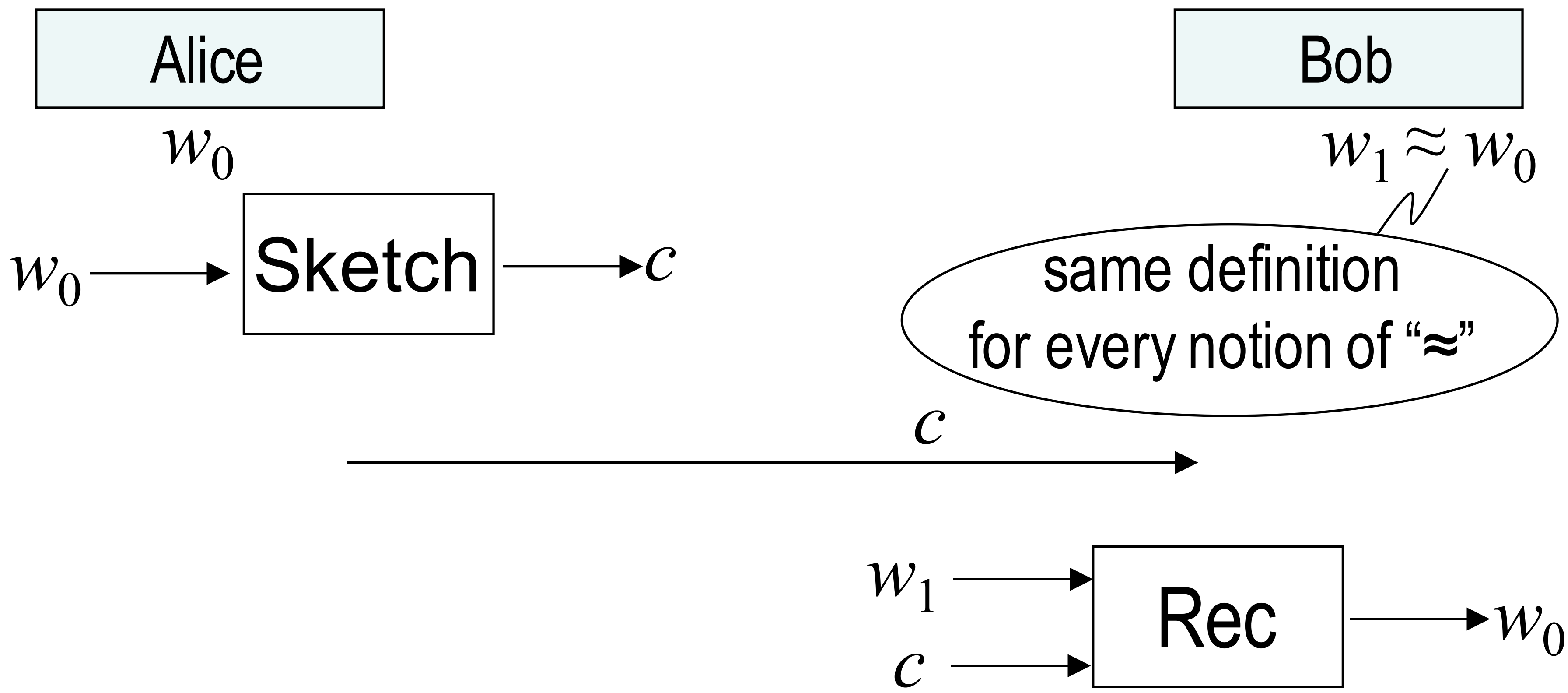
definition: secure sketch is a pair (Sketch, Rec)



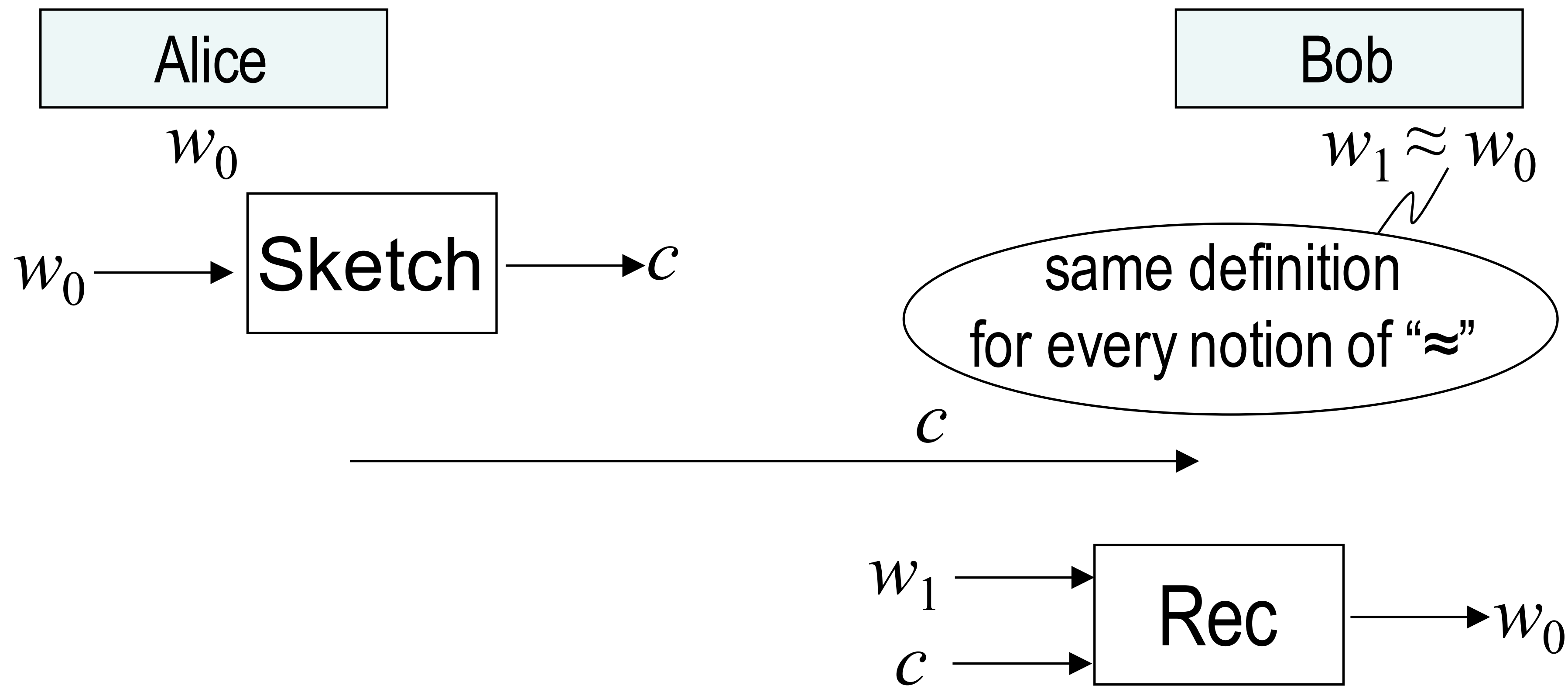
definition: secure sketch is a pair (Sketch, Rec)



definition: secure sketch is a pair (Sketch, Rec)



definition: secure sketch is a pair (Sketch, Rec)



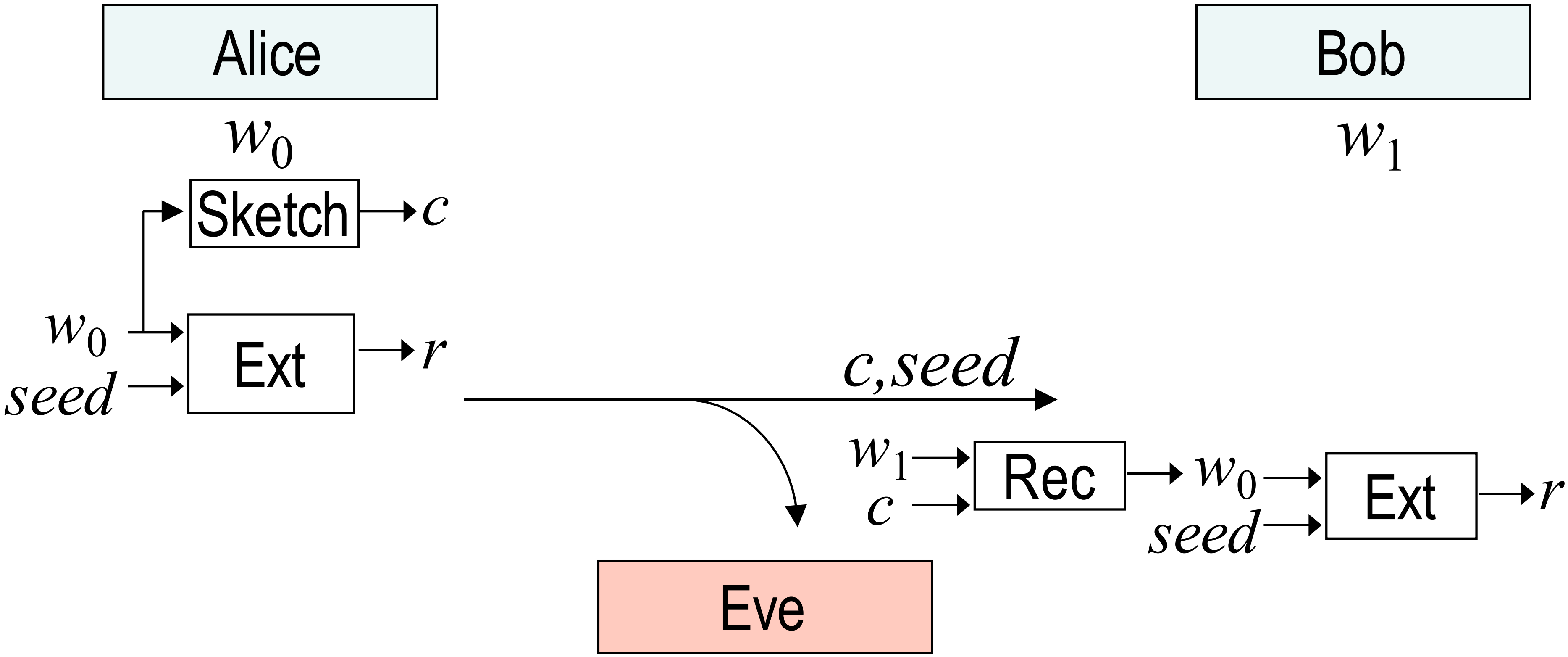
entropy loss l

Def [Dodis-Ostrovsky-R-Smith 04]:

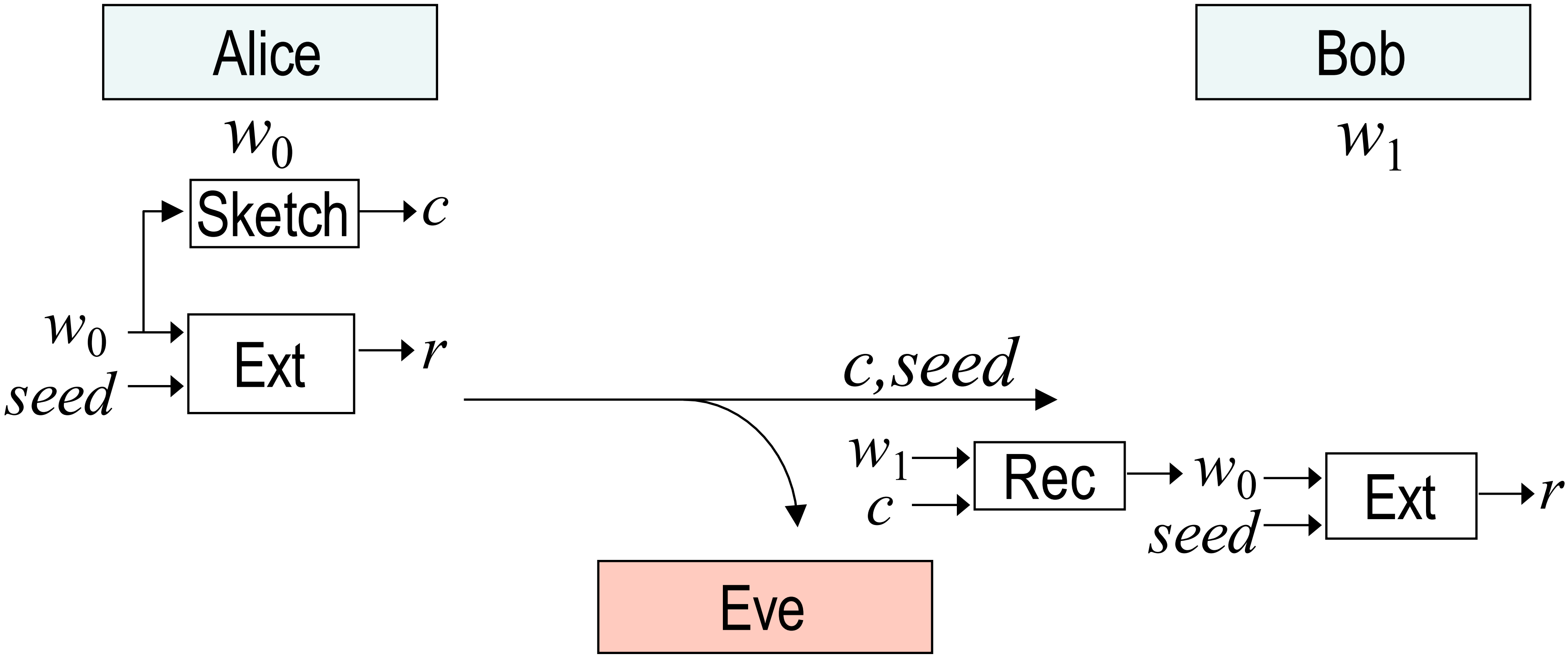
(Sketch, Rec) is a $(k, k-l)$-secure sketch if

$H_{\min}(W_0 \mid E) \geq k$ implies $H_{\min}(W_0 \mid E, \text{Sketch}(W_0)) \geq k - l$

information-reconciliation + privacy amplification



information-reconciliation + privacy amplification

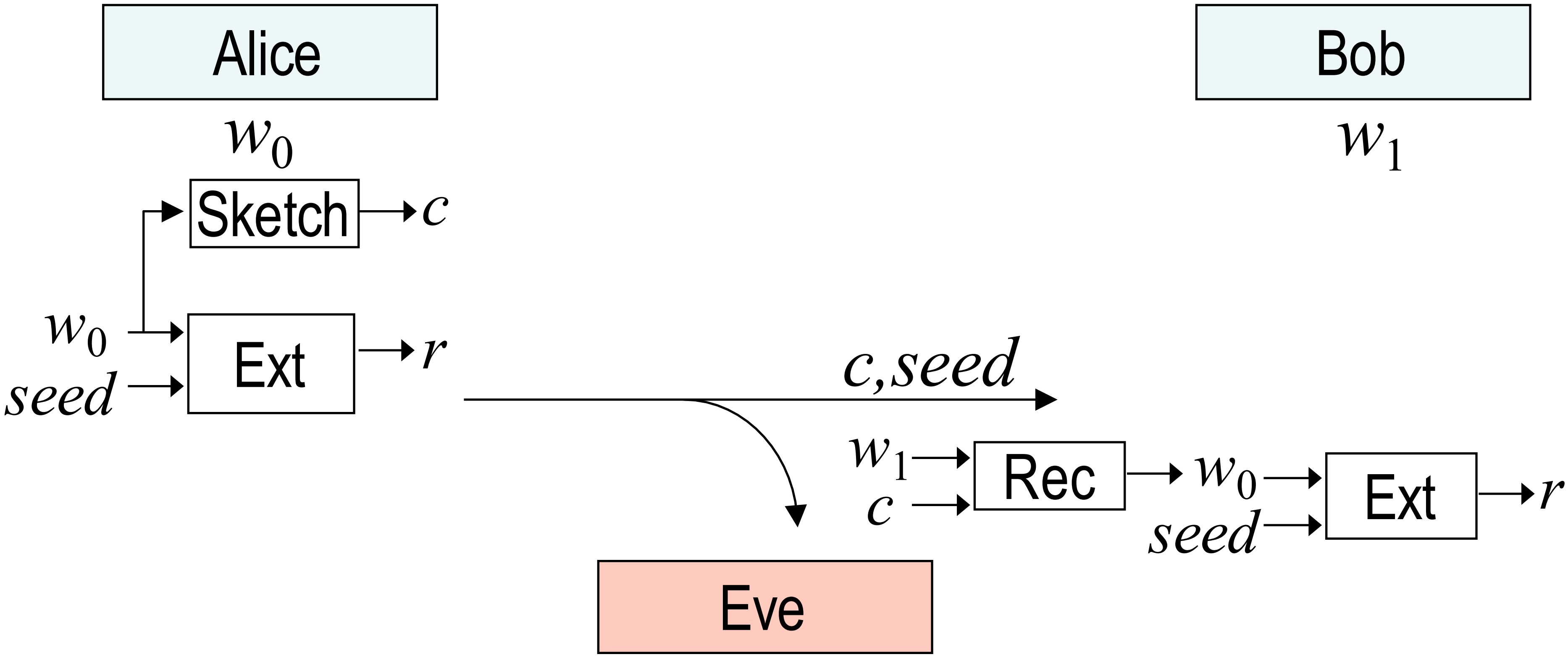


$$H_{\min}(W_0 \mid E) \geq k \;\Rightarrow\; H_{\min}(W_0 \mid E, \text{Sketch}(W_0)) \geq k - l$$

$(k - l,\ \varepsilon)$- Ext $\Rightarrow (R, C, Seed, E) \approx_{\varepsilon} (U_m, C, Seed, E)$

Thus can get $m = k - l - 2 \log (1/\varepsilon)$

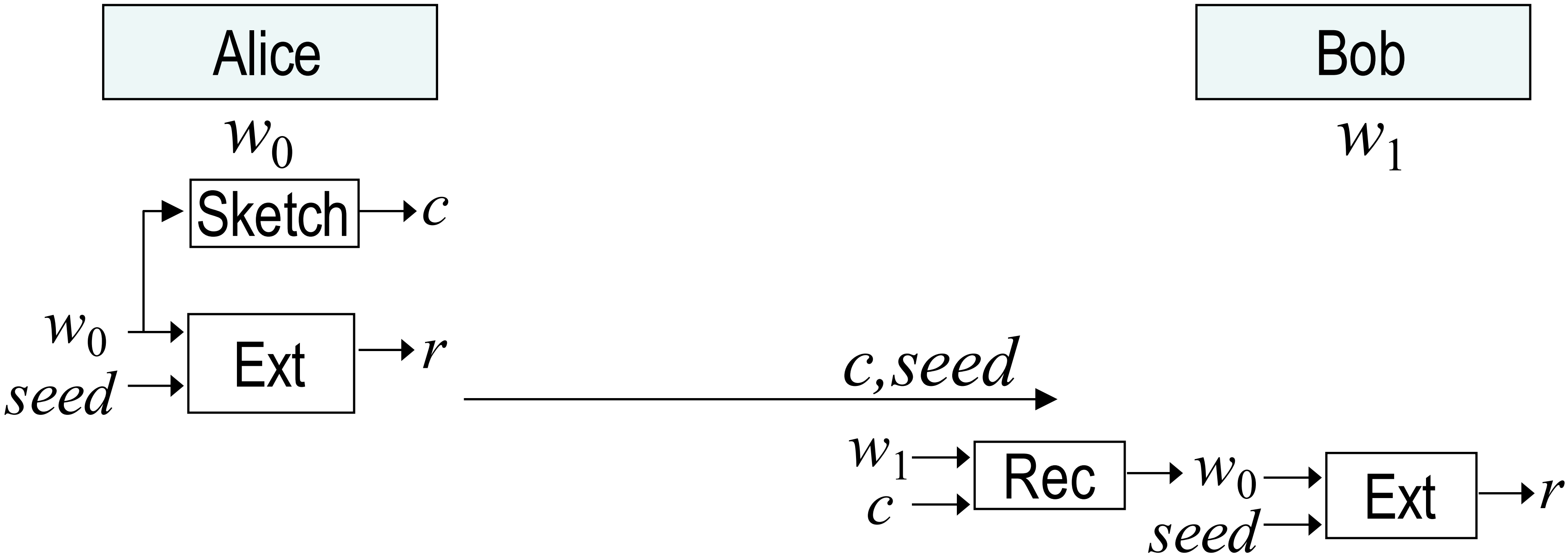
information-reconciliation + privacy amplification

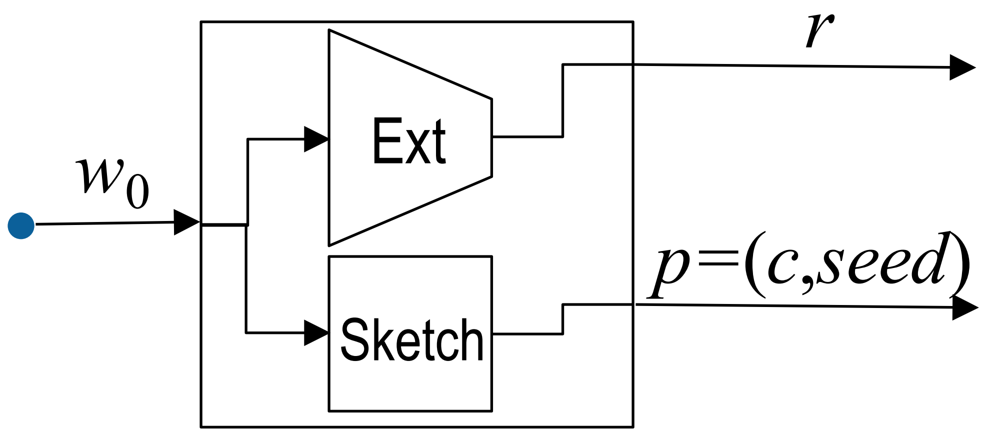


All in one message!

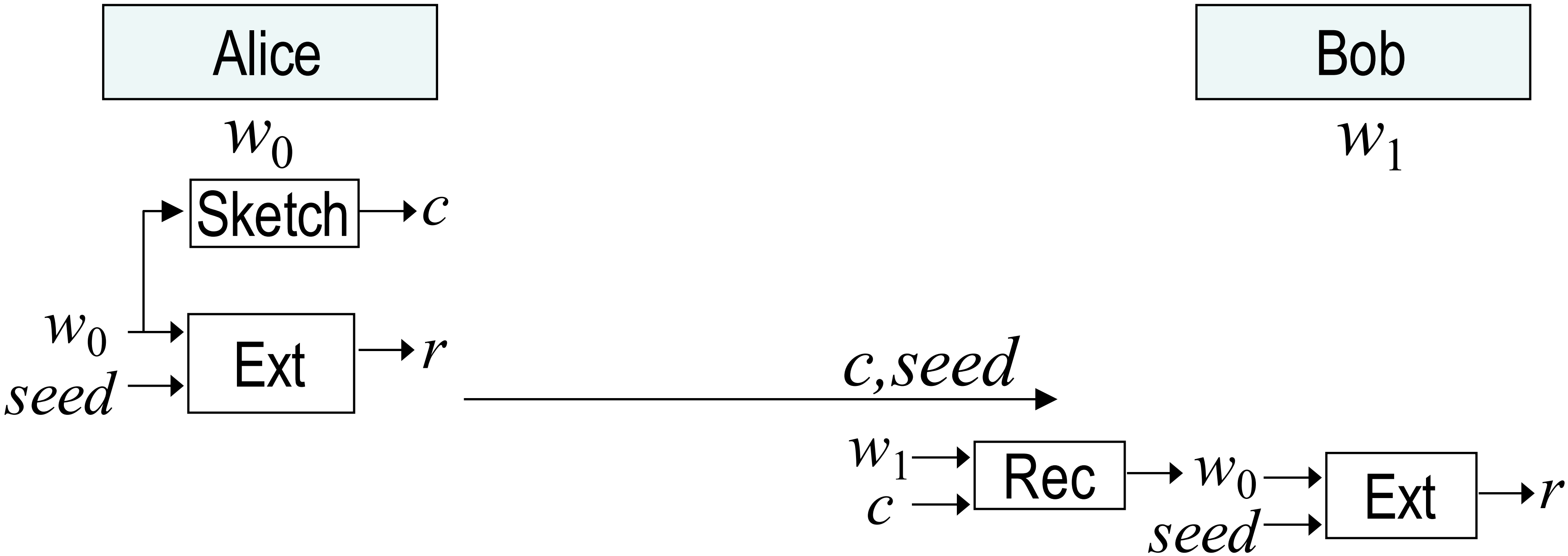
Let's take another view of what we've built…

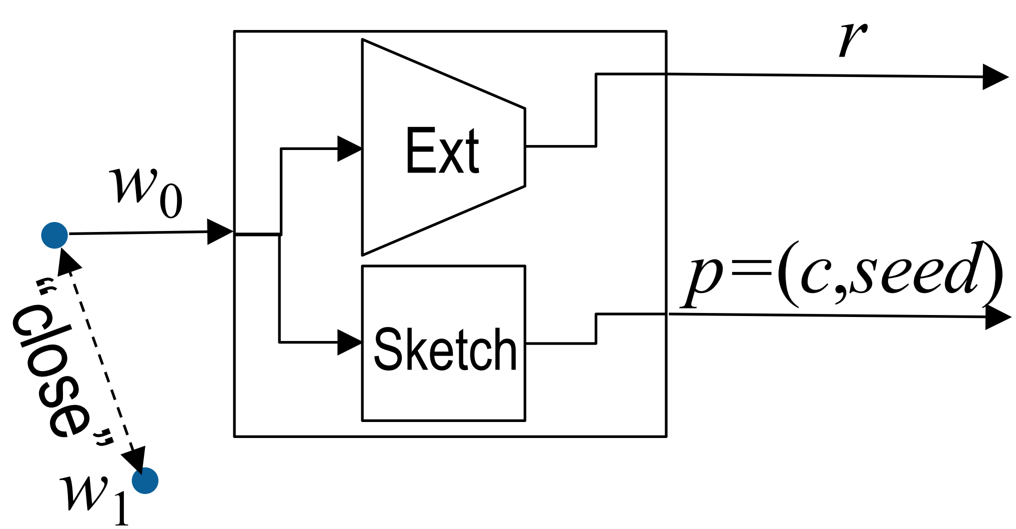
information-reconciliation + privacy amplification



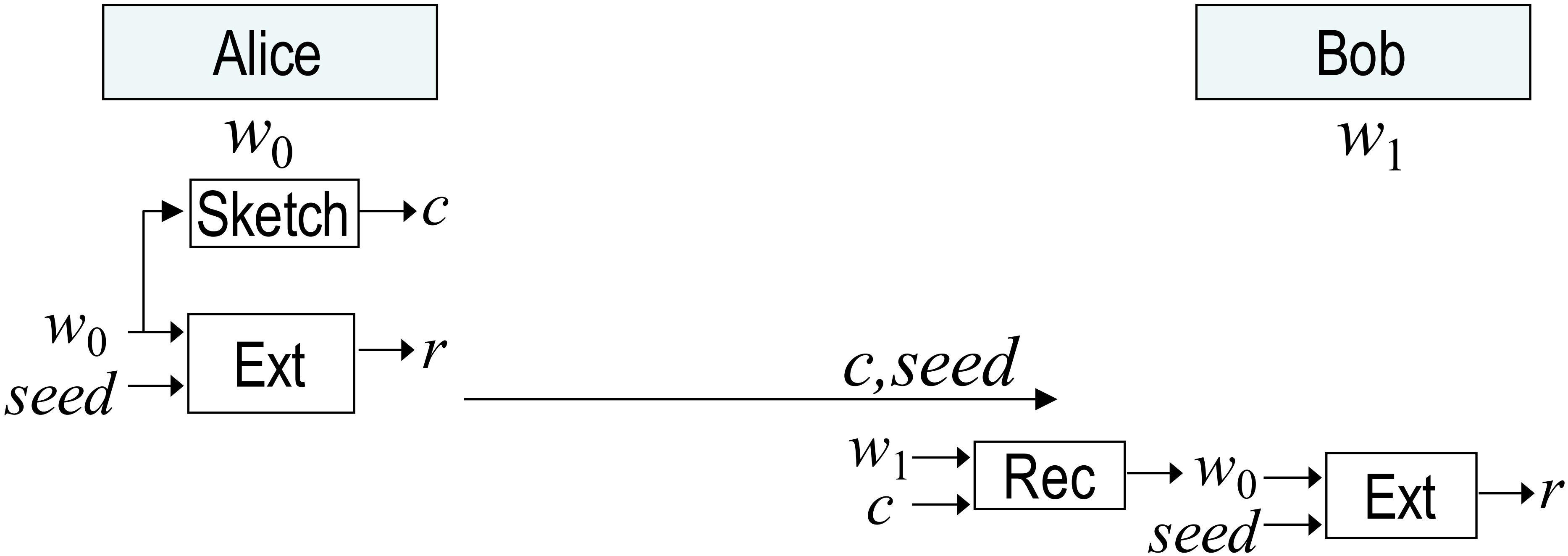


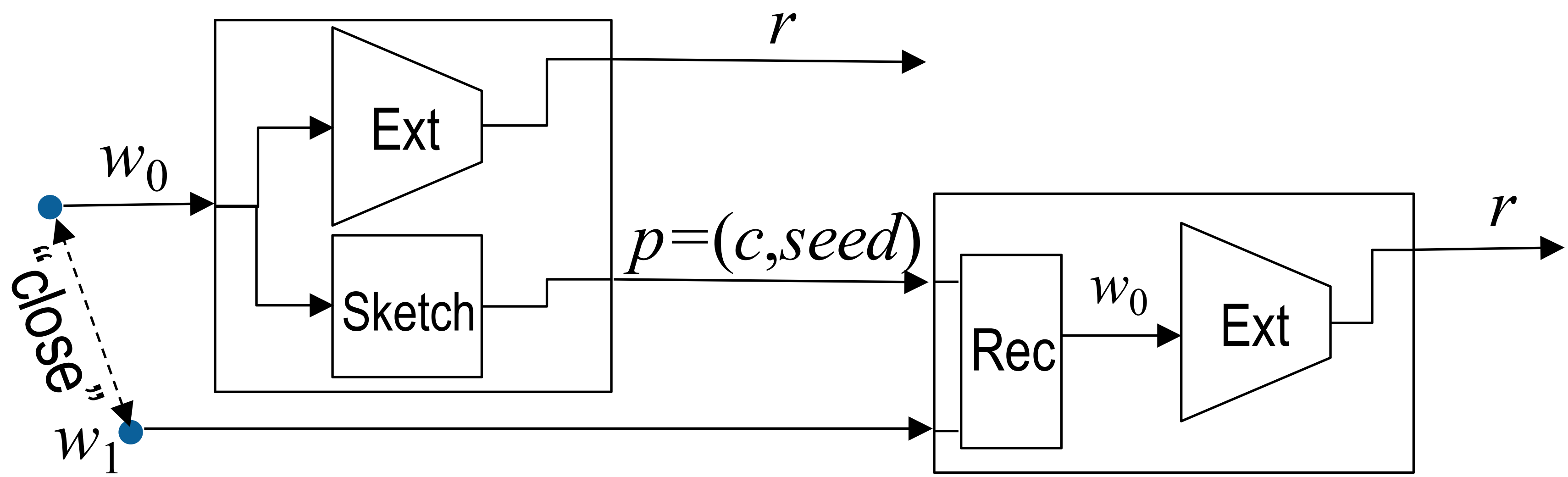
information-reconciliation + privacy amplification



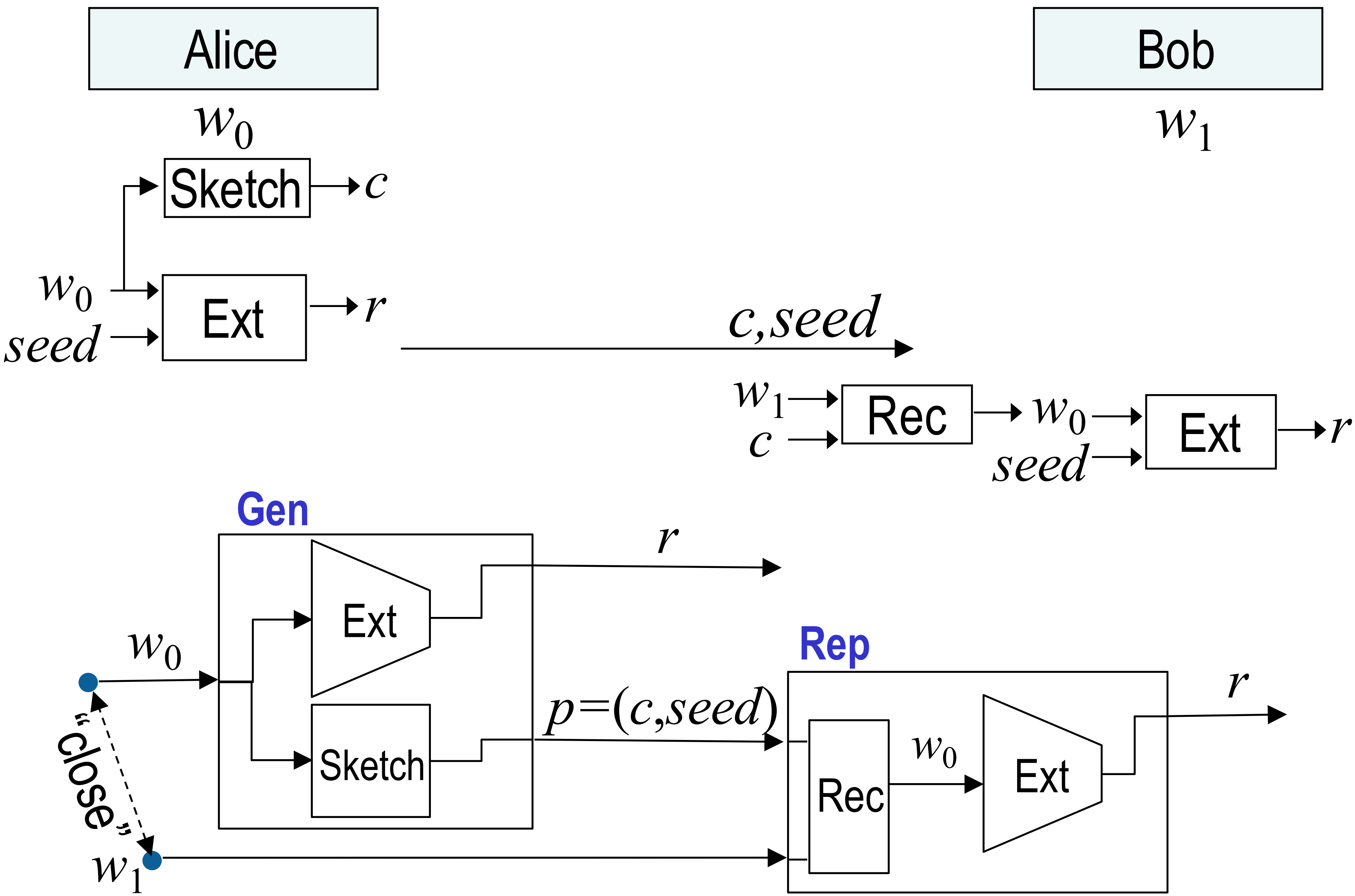


information-reconciliation + privacy amplification

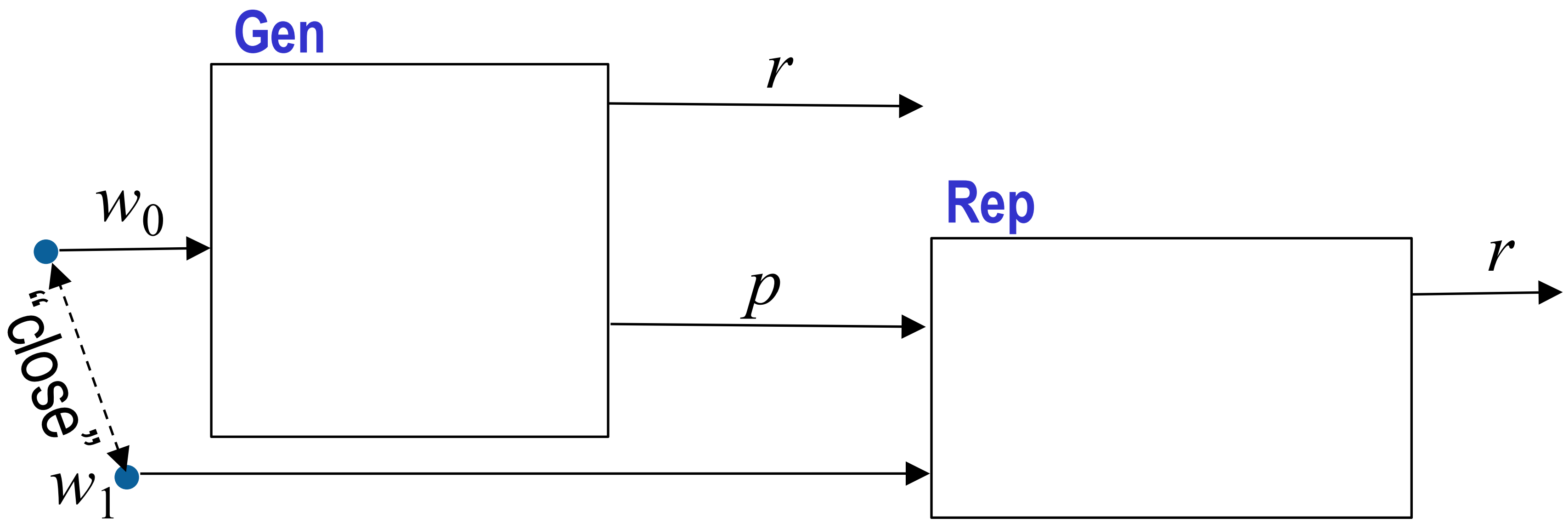




information-reconciliation + privacy amplification



information-reconciliation + privacy amplification



Fuzzy Extractors

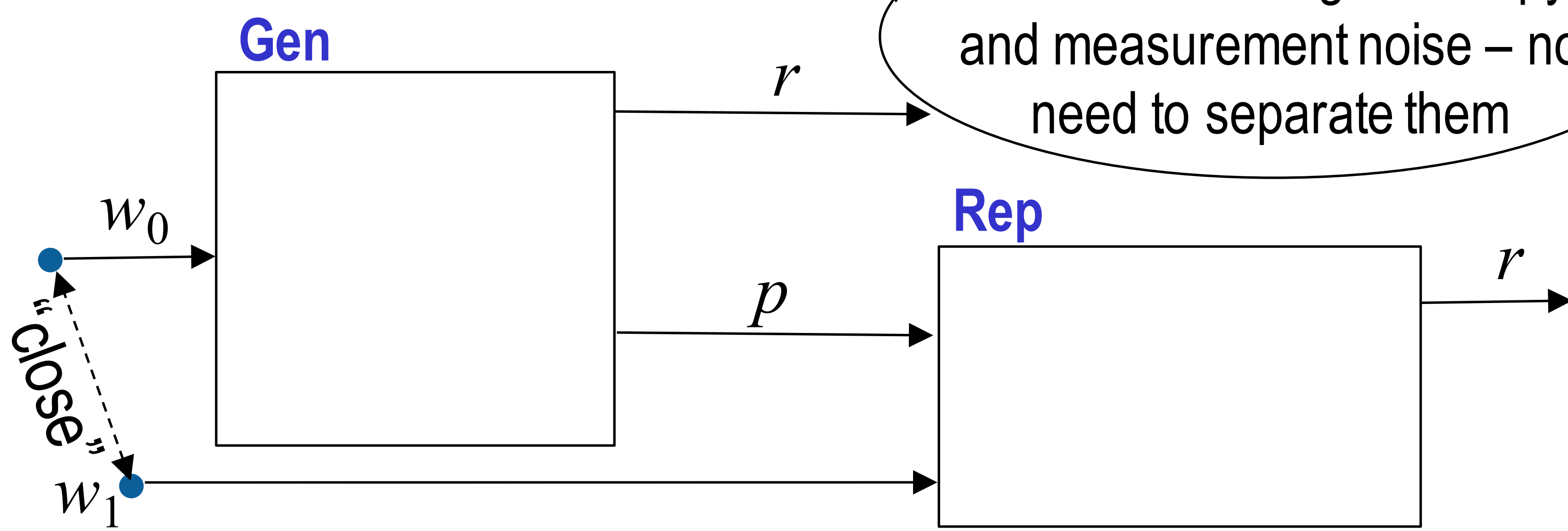
Single message information reconciliation + privacy amplification
= fuzzy extractor [Dodis-Ostrovsky-R-Smith 04]

<u>Definition of fuzzy extractors:</u>

Functionality requirement: if w_0 and w_1 are close, then Rep gets r

Security requirement: if $H_{\min}(W_0|E) \geq k$ then $(R,P,E) \approx_\varepsilon (U_m,P,E)$

includes "meaningful entropy" and measurement noise – no need to separate them



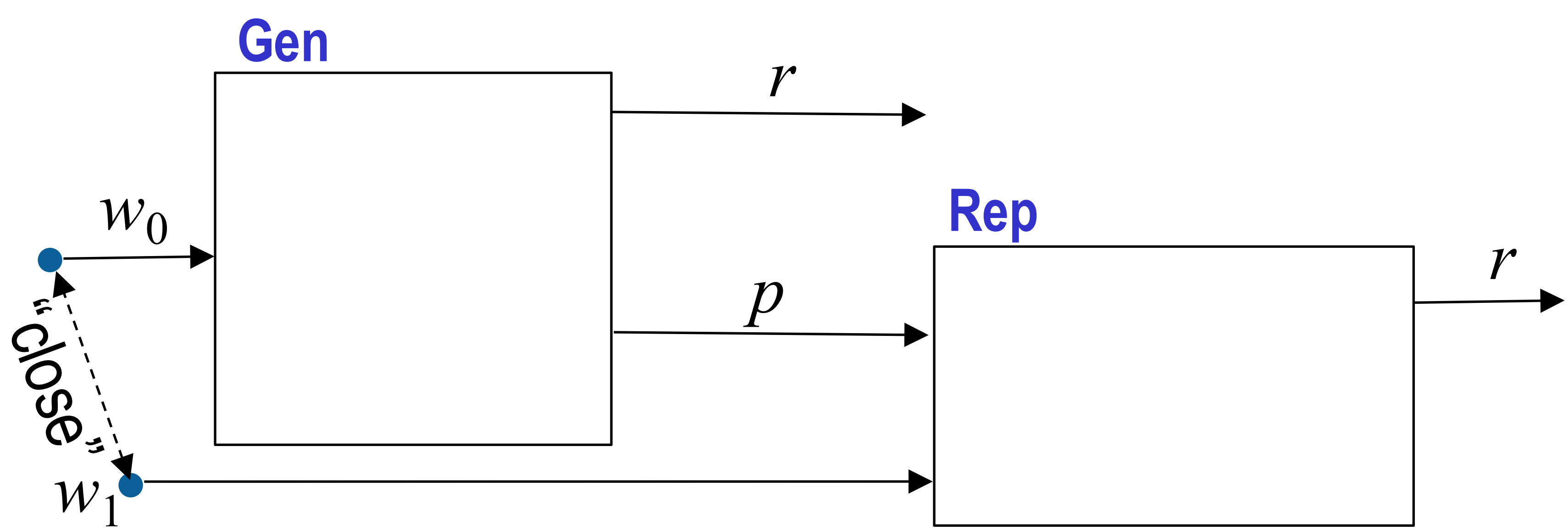
Fuzzy Extractors

Single message information reconciliation + privacy amplification
= fuzzy extractor [Dodis-Ostrovsky-R-Smith 04]

<u>Advantages of this view:</u>

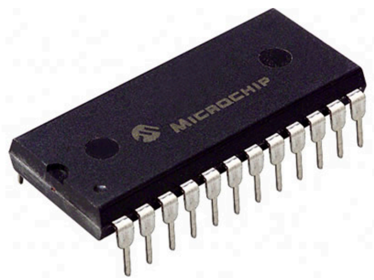
Can think of other constructions (not sketch+extract, computational)
[Canetti-Fuller-Paneth-R.-Smith Eurocrypt '15]

Single message p can be sent into the future!



Advantages of single-message protocols

Physically Unclonable Functions (PUFs)



Biometric Data

High-entropy sources are often noisy

– Initial reading $w_0 \neq$ later reading reading w_1, but is close

Fuzzy Extractor can derive a stable,
cryptographically strong output

– At initial enrollment of w_0, use Gen, store p

– All subsequent readings $w_{1,}$ w_2 … map to same output using Rep

Use r for any crypto scheme–e.g., a key to encrypt your sensitive data

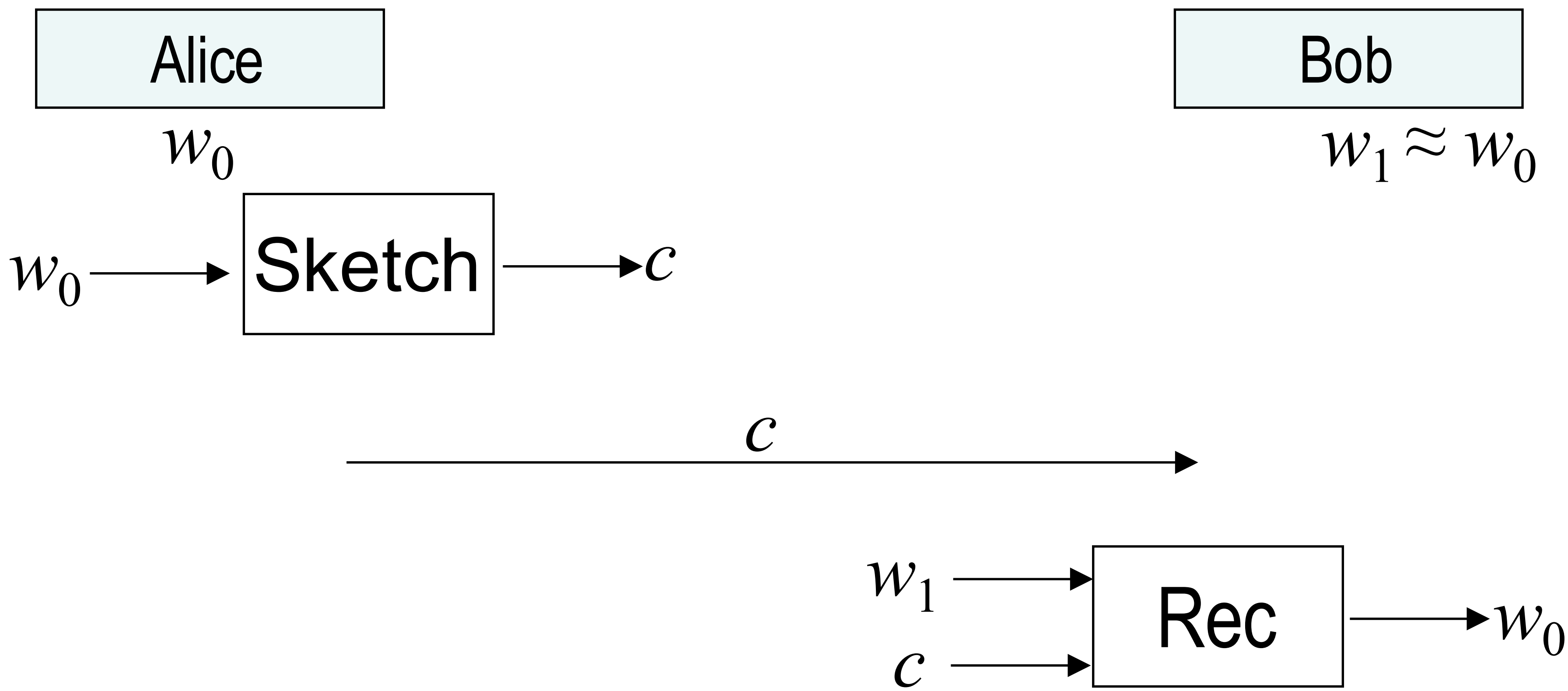
– E.g., self-enforcing, rather than server-enforced, authorization

Outline

- Passive adversaries
  - Privacy amplification
  - Fuzzy extractors
  - Information reconciliation
- Active adversaries, w has a lot of entropy
  - Privacy amplification
  - Information reconciliation
- Active adversaries, w has little entropy
  - Privacy amplification
  - Information reconciliation

How to build a secure sketch?

How to build a secure sketch?



Want:

$H_{\min}(W_0 \mid E) \geq k$ implies $H_{\min}(W_0 \mid E, \text{Sketch}(W_0)) \geq k - l$

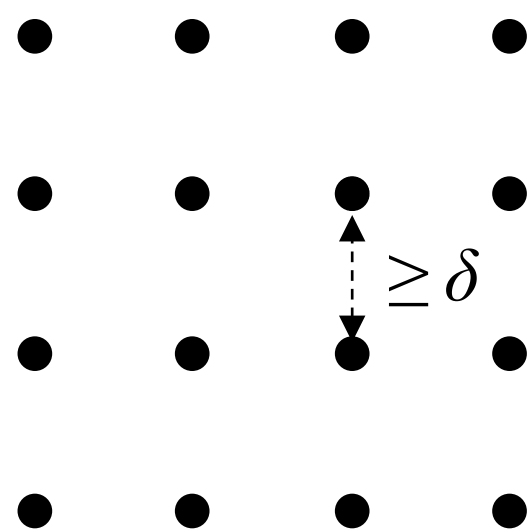
Focus for now: $\approx$ means Hamming distance

(w_0 and w_1 are strings over $\mathrm{GF}(q)$ that differ in $\leq t$ positions)

background: error-correcting codes

$(n, \mu, \delta)_q$ code $\mathrm{GF}(q)^\mu \longrightarrow \mathrm{GF}(q)^n$

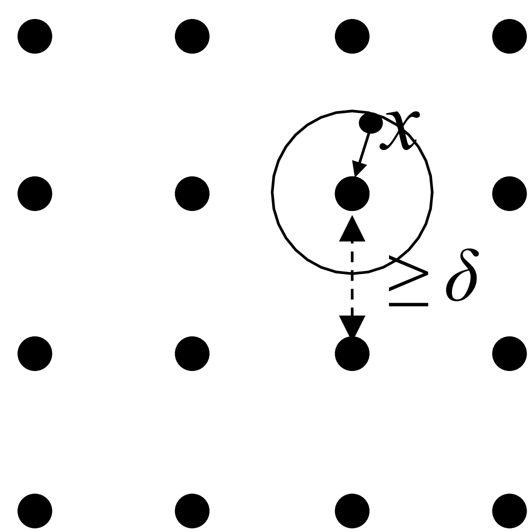
- encodes μ-symbol messages into n-symbol codewords
- any two codewords differ in at least δ locations
  - fewer than $\delta/2$ errors $\Rightarrow$ unique correct decoding



background: error-correcting codes

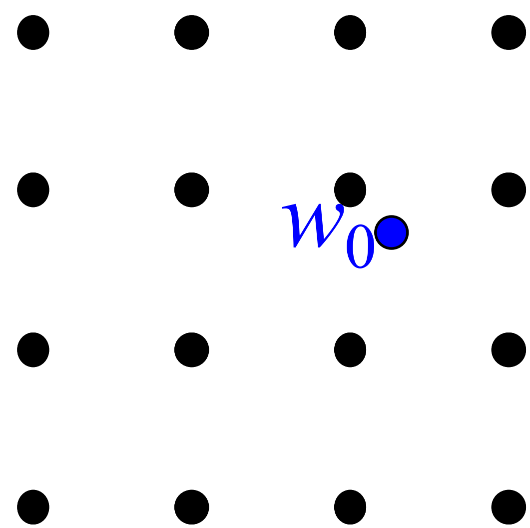
$(n, \mu, \delta)_q$ code $\mathrm{GF}(q)^\mu \rightarrow \mathrm{GF}(q)^n$

- encodes μ-symbol messages into n-symbol codewords
- any two codewords differ in at least δ locations
  - fewer than $\delta/2$ errors $\Rightarrow$ unique correct decoding
- Ignore the message space
- Think of decoding x as finding nearest codeword
- Efficiency of decoding and parameters n, μ, δ depend on the code



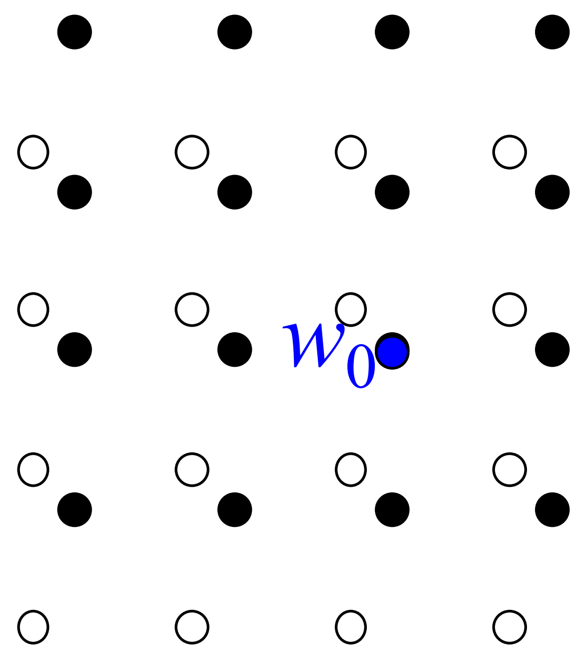
building secure sketches

- Idea: what if w_0 is a codeword in an ECC?
- Sketch = nothing; Rec = Decoding to find w_0 from w_1
- If w_0 not a codeword, simply shift the ECC



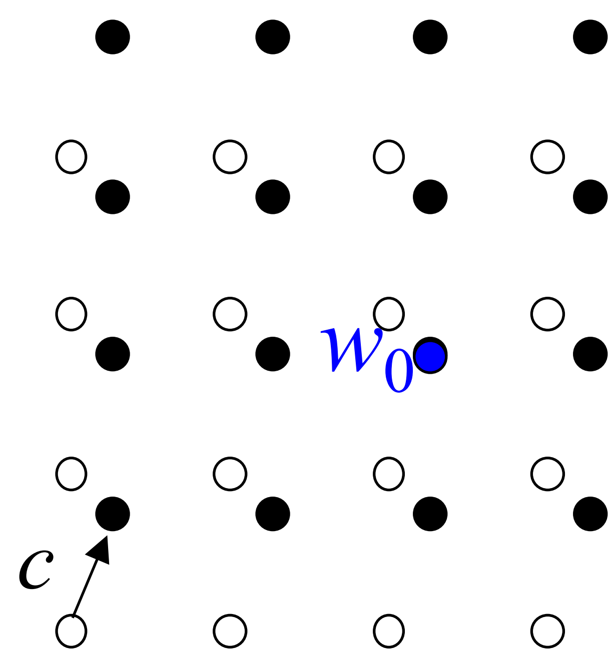
building secure sketches

- Idea: what if w_0 is a codeword in an ECC?
- Sketch = nothing; Rec = Decoding to find w_0 from w_1
- If w_0 not a codeword, simply shift the ECC



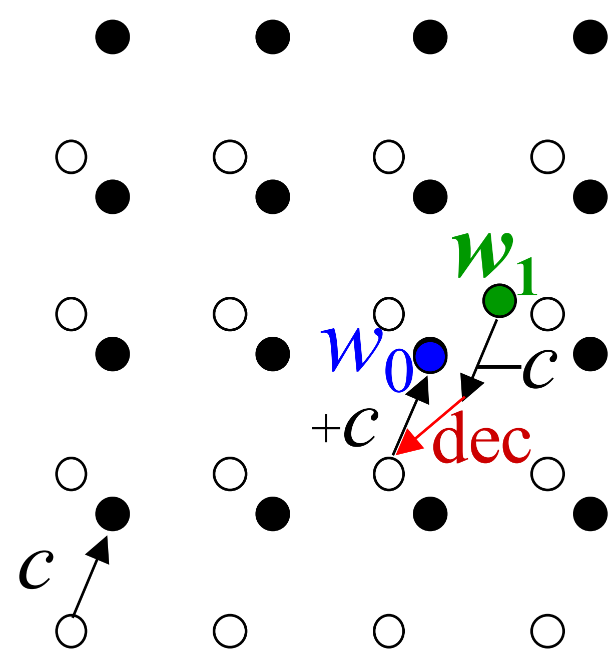
building secure sketches

- Idea: what if w_0 is a codeword in an ECC?
- Sketch = nothing; Rec = Decoding to find w_0 from w_1
- If w_0 not a codeword, simply shift the ECC
- Sketch (w_0) is the shift to random codeword:
  $c = w_0 -$ random codeword



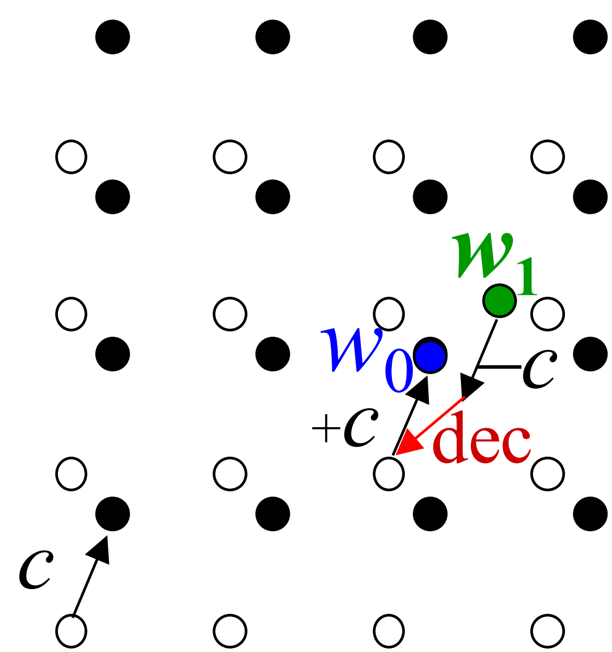
building secure sketches

- Idea: what if w_0 is a codeword in an ECC?
- Sketch = nothing; Rec = Decoding to find w_0 from w_1
- If w_0 not a codeword, simply shift the ECC
- Sketch (w_0) is the shift to random codeword: $c = w_0 -$ random codeword
- Rec: $\mathrm{dec}(w_1 - c) + c$



building secure sketches

- Idea: what if w_0 is a codeword in an ECC?
- Sketch = nothing; Rec = Decoding to find w_0 from w_1
- If w_0 not a codeword, simply shift the ECC
- Sketch (w_0) is the shift to random codeword: $c = w_0 -$ random codeword
- Rec: $\mathrm{dec}(w_1 - c) + c$



- Another view: w_0 is a one-time-pad for a message that's been encoded with the error-correcting code, so w_1 can decrypt

security analysis

$(n, \mu, \delta)_q$ code $\mathrm{GF}(q)^\mu \longrightarrow \mathrm{GF}(q)^n$

$c = w_0$ – random codeword

$$H_{\min}(W_0 \mid E, C) \geq H_{\min}(W_0, C \mid E) - H_{\max}(C) =$$

$$= H_{\min}(W_0, C \mid E) - n \log q$$

$$= H_{\min}(W_0 \mid E) + \mu \log q - n \log q$$

$$= H_{\min}(W_0 \mid E) - (n - \mu) \log q$$

entropy loss l

optimization for linear codes

$(n, \mu, \delta)_q$ code $\mathrm{GF}(q)^\mu \longrightarrow \mathrm{GF}(q)^n$

$c = w_0$ – random codeword

Suppose the codewords form a linear subspace of $\mathrm{GF}(q)^n$

Then there is a linear map (called "parity check matrix")

H: $\mathrm{GF}(q)^n \longrightarrow \mathrm{GF}(q)^{n-\mu}$ such that codewords = $\mathrm{Ker}\, H$

c = uniform choice from $\{w_0 - \mathrm{Ker}\, H\}$

Observe that $\{w_0 - \mathrm{Ker}\, H\} = \{x\colon Hx = Hw_0\}$

(l.h.s. $\subseteq$ r.h.s. by multiplication by H)

(l.h.s. $\supseteq$ r.h.s. because $x = w_0 - (w_0 - x)$)

optimization for linear codes

$(n, \mu, \delta)_q$ code $\mathrm{GF}(q)^\mu \longrightarrow \mathrm{GF}(q)^n$

$c = w_0$ – random codeword

Suppose the codewords form a linear subspace of $\mathrm{GF}(q)^n$

Then there is a linear map (called “parity check matrix”)
H: $\mathrm{GF}(q)^n \longrightarrow \mathrm{GF}(q)^{n-\mu}$ such that codewords = $\mathrm{Ker}\, H$

c = uniform choice from $\{w_0 - \mathrm{Ker}\, H\}$

Observe that $\{w_0 - \mathrm{Ker}\, H\} = \{x\colon Hx = Hw_0\}$

Thus, Sketch(w_0) can send Hw_0 (called ”syndrome of w_0”) and Rec can sample x by solving linear equations

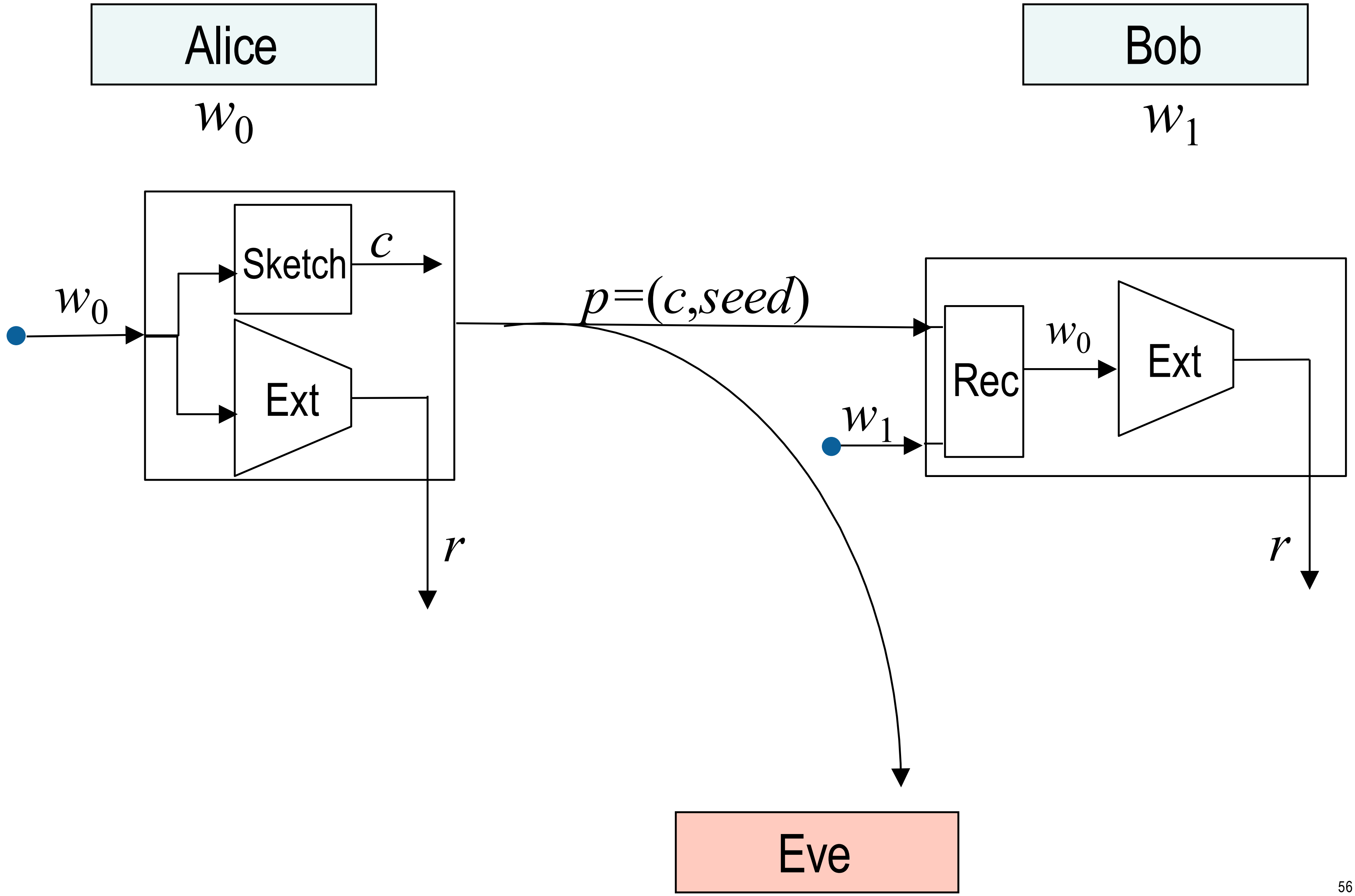
$$
\begin{aligned}
H_{\min}(W_0 \mid E, H\, W_0) &\geq H_{\min}(W_0 \mid E) - H_{\max}(H\, W_0) \\
&= H_{\min}(W_0 \mid E) - (n - \mu) \log q
\end{aligned}
$$

syndrome or code-offset construction

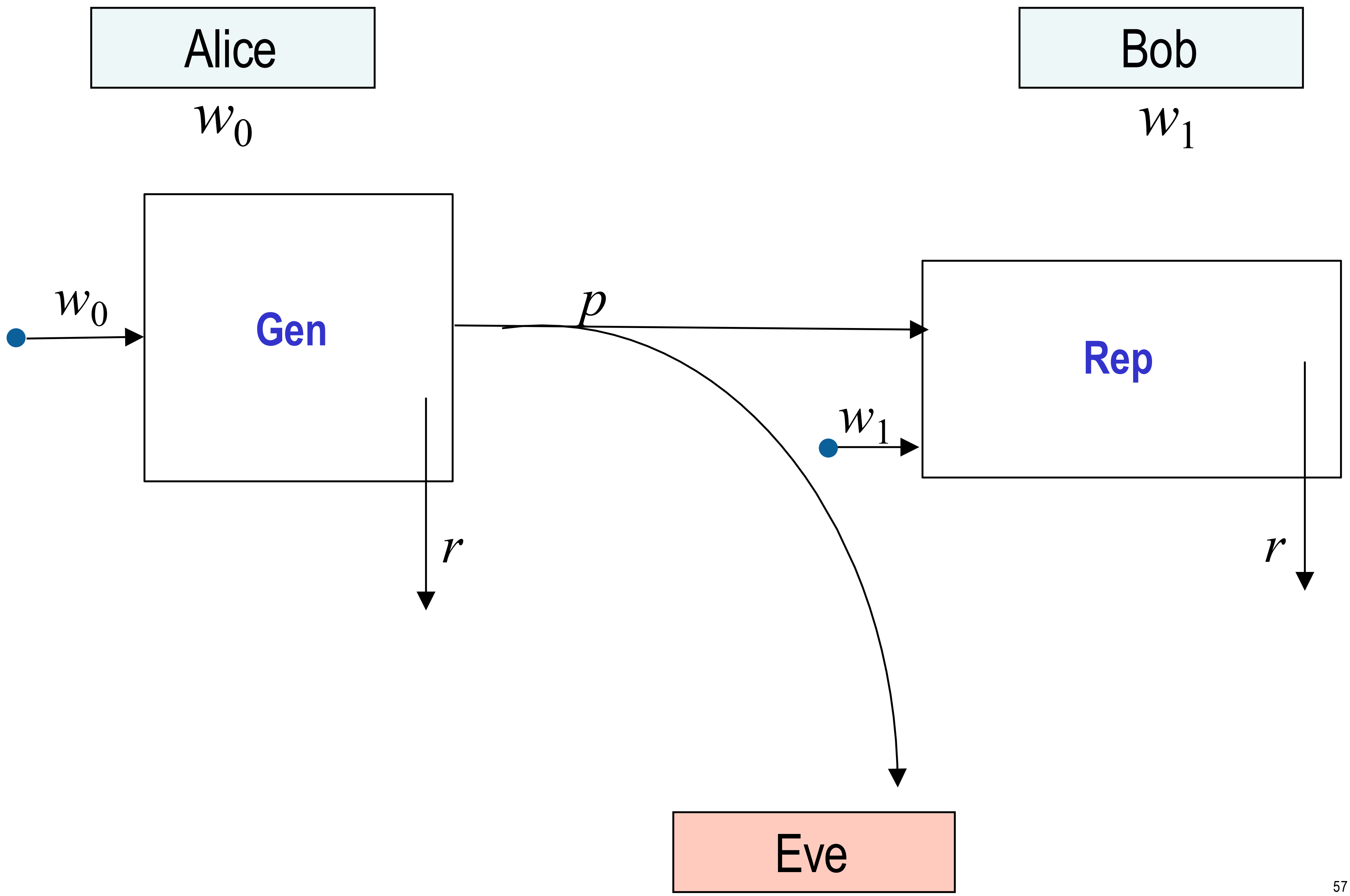
$$\text{Sketch}(w) = Hw \quad \text{OR} \quad \text{Sketch}(w) = w - \text{random codeword}$$

- If ECC μ symbols $\rightarrow n$ symbols and has distance δ:
  - Correct $\delta/2$ errors; entropy loss $l = n - \mu$ symbols
  - Higher error-tolerance means higher entropy loss (trade error-tolerance for security)
  - Can be viewed as redundant one-time pad
  - Hard to improve without losing generality (e.g., working only for some distributions of inputs, for example, [Yu et al. CHES 2011, Fuller el al. Asiacrypt 2016, Woodage et al. Crypto 2017])
- Construction is old but keeps being rediscovered
  - [Bennett-Brassard-Robert 1985] (from systematic codes), [Bennet-Brassard-Crépeau-Skubiszewska 1991] (syndrome), [Juels-Watenberg 2002] (code-offset)

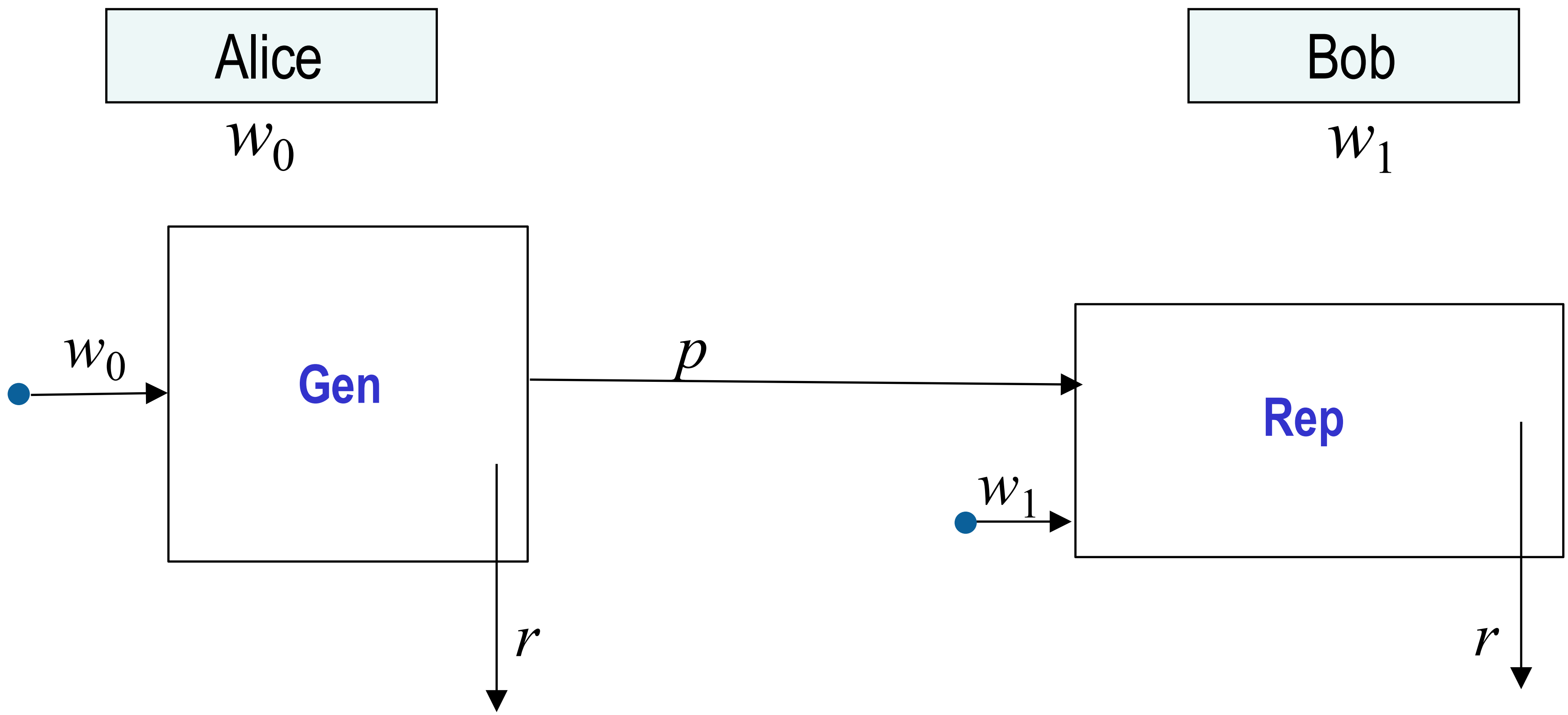
1-message key agreement for passive adversaries



1-message key agreement for passive adversaries



1-message key agreement for passive adversaries

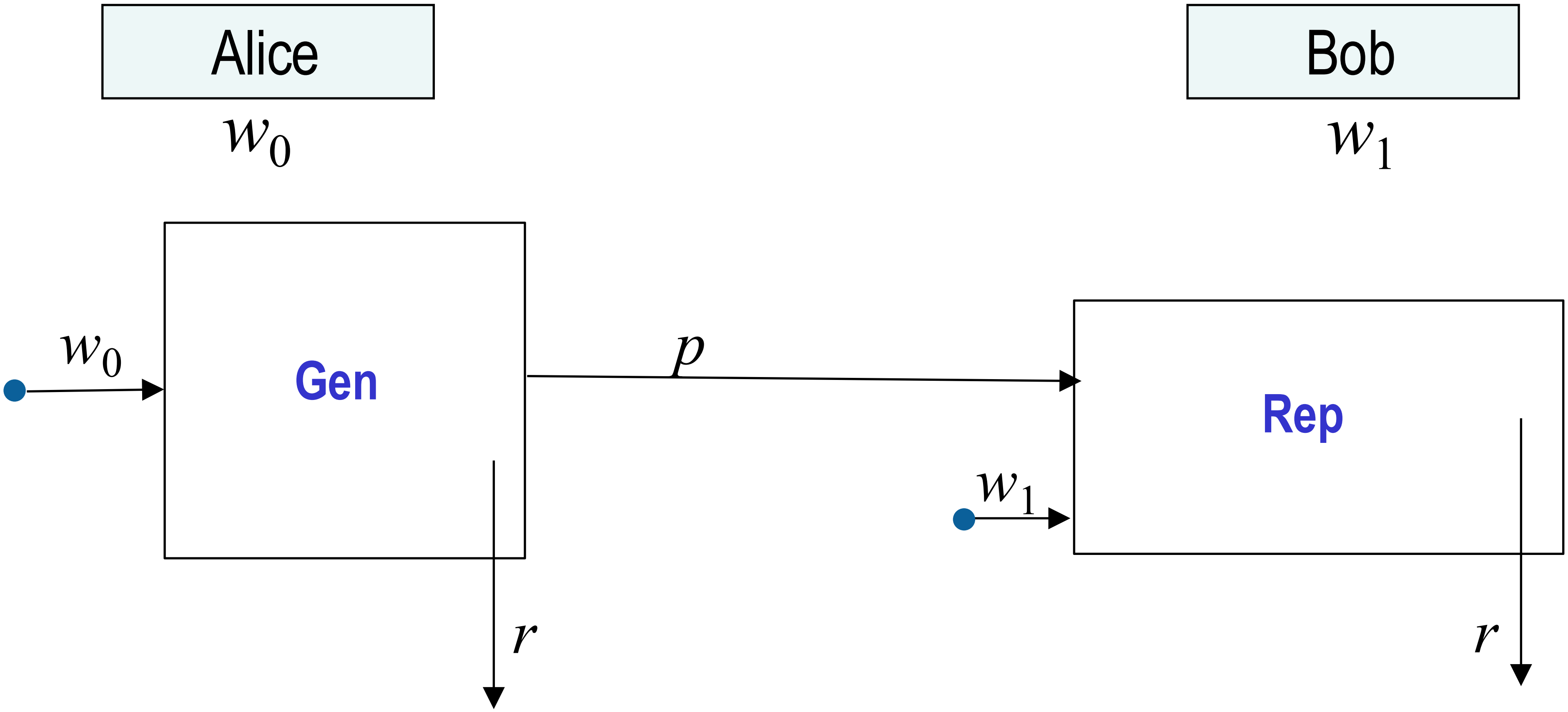


- Fuzzy extractors exist for other distances besides Hamming, including set difference, edit distance, point-set distance
- Some make specific assumptions on input distribution, some are computational rather than info-theoretic

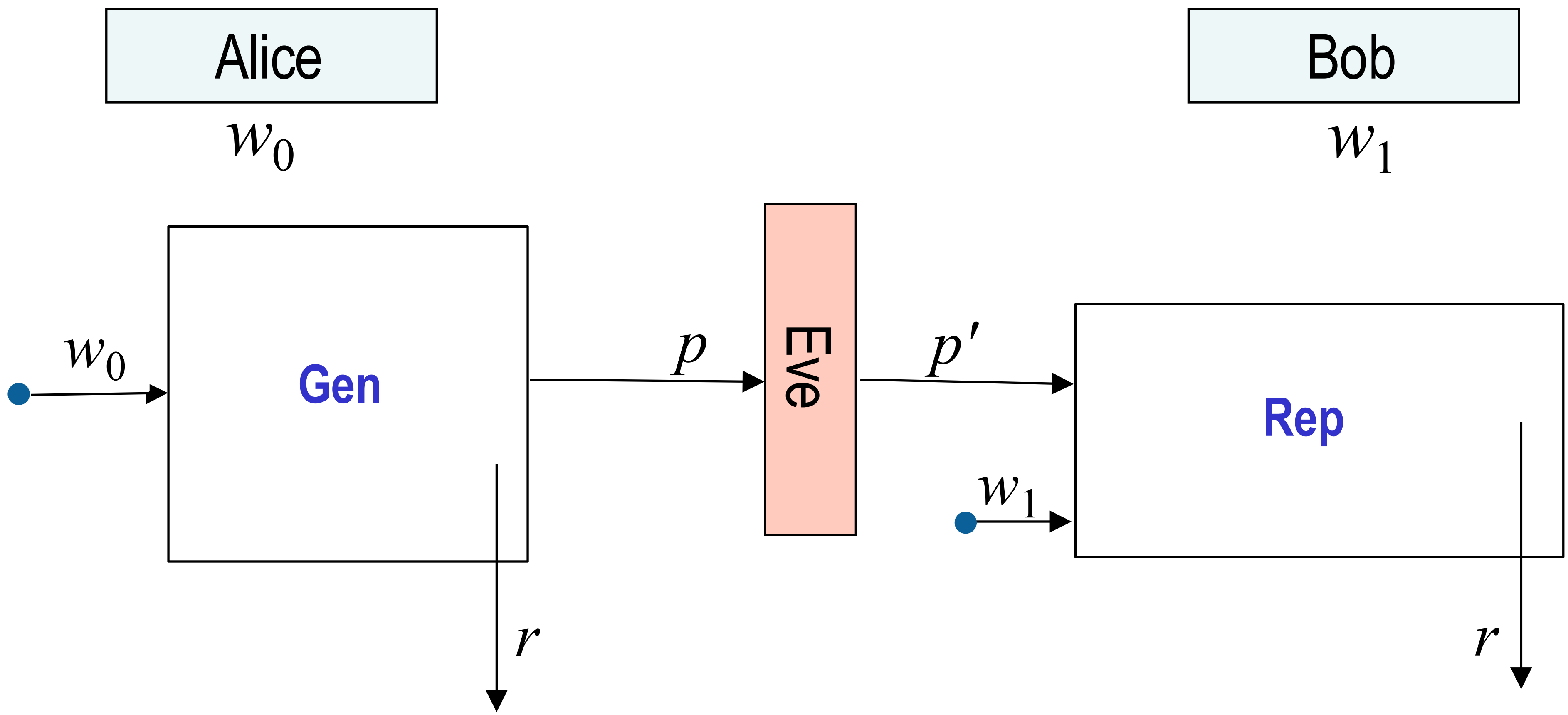
Outline

- Passive adversaries
  - Privacy amplification
  - Fuzzy extractors
  - Information reconciliation
- Active adversaries, w has a lot of entropy
  - Privacy amplification
  - Information reconciliation
- Active adversaries, w has little entropy
  - Privacy amplification
  - Information reconciliation

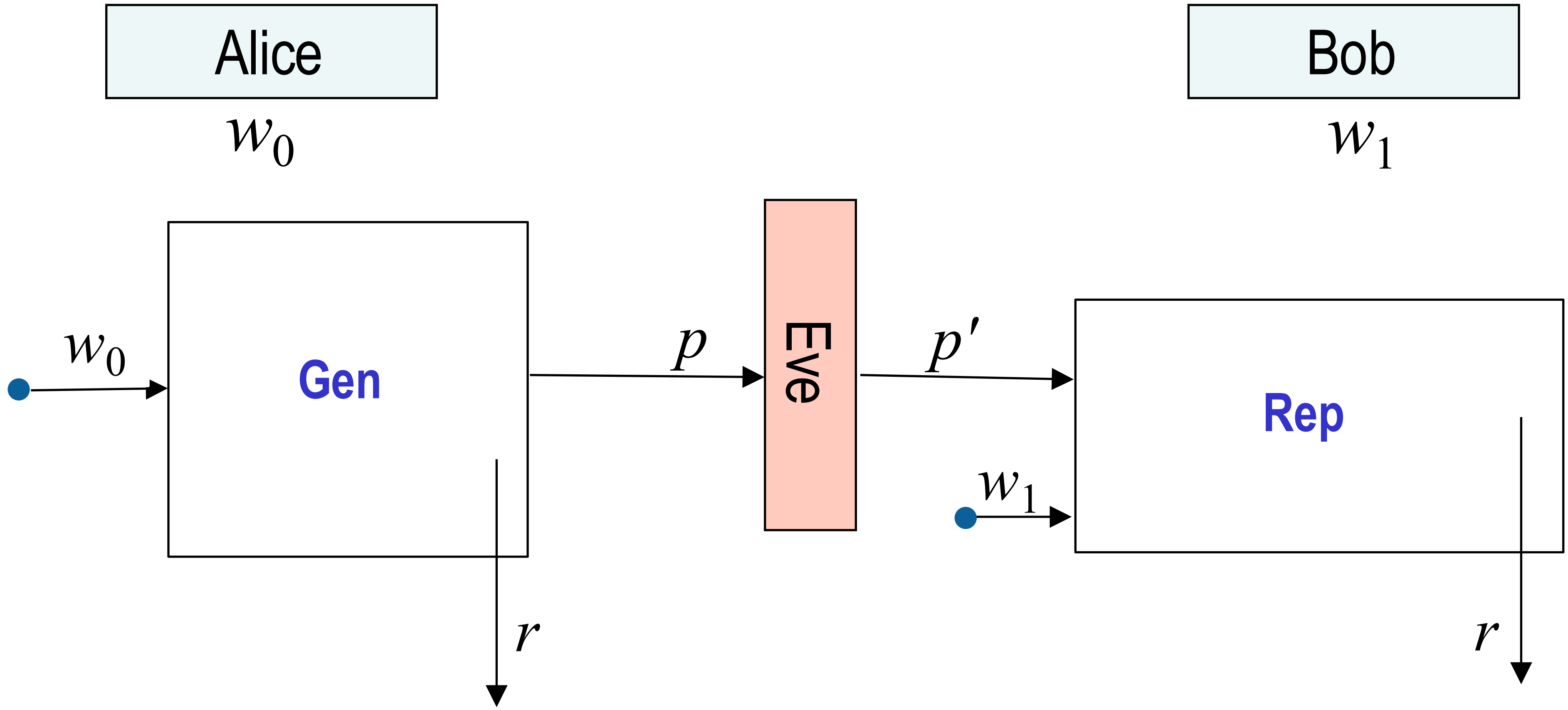
WHAT ABOUT ACTIVE ADVERSARIES?



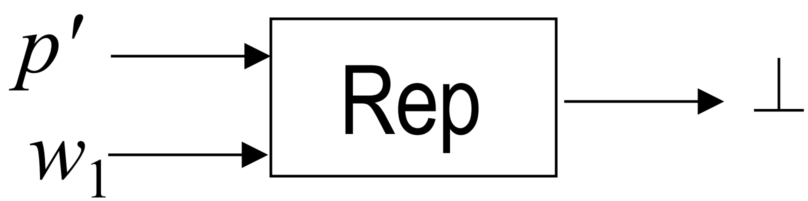
WHAT ABOUT ACTIVE ADVERSARIES?



WHAT ABOUT ACTIVE ADVERSARIES?



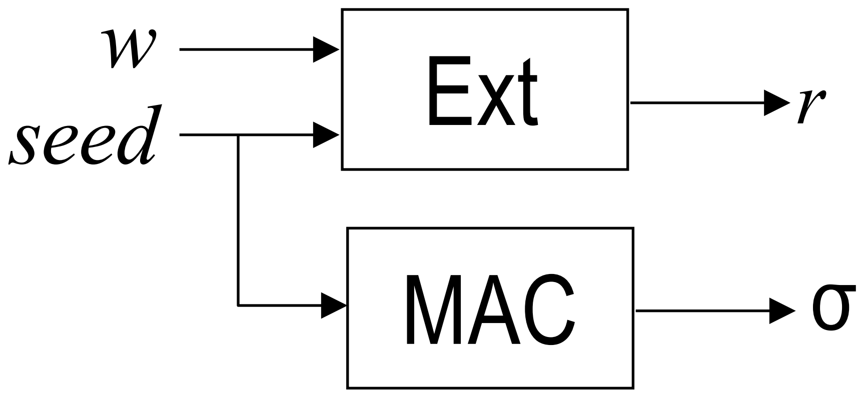
Robustness: as long as $w_0 \approx w_1$, if Eve(p) produces $p' \neq p$



(with 1 – negligible probability over w_0 & coins of Rep, Eve)

building $\overrightarrow{\text{robust}}$ extractors

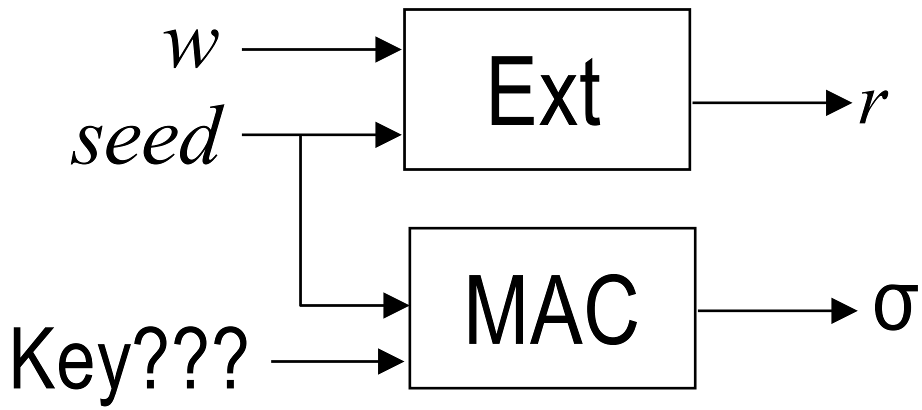
Idea 0:



$$p = (seed, \sigma)$$

building robust extractors

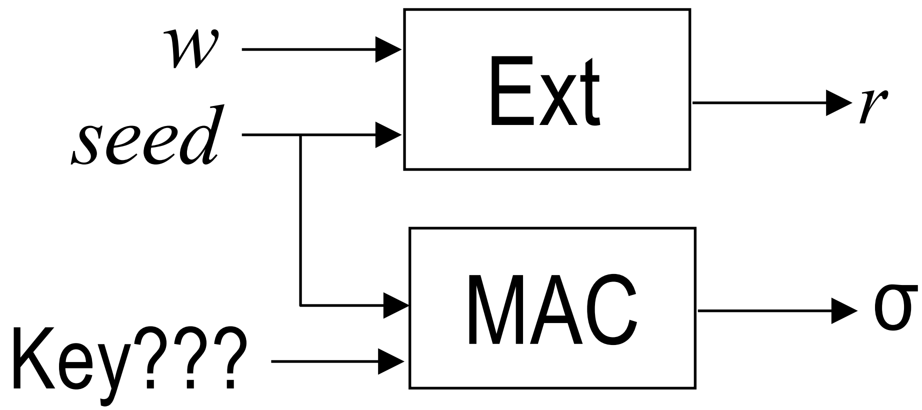
Idea 0:



$$p = (seed, \sigma)$$

building robust extractors

Idea 0:

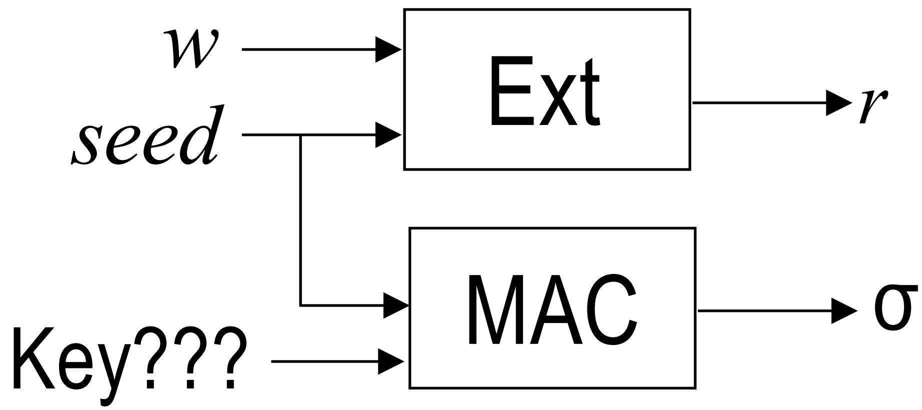


$p = (seed, \sigma)$

r? But if adversary changes $seed$, then r will change

building robust extractors

Idea 0:



$$p = (seed, \sigma)$$

r? But if adversary changes $seed$, then r will change

w?

Circularity!

$seed$ extracts from w

w authenticates $seed$

background: XOR-universal functions and MACs

- Define $f_a(\bullet)$ with v-bit outputs to be XOR-universal if

$$(\forall\, i \neq j, y)\ \Pr_a[f_a(i) \oplus f_a(j) = y] = 1/2^v$$

- Fact: $f_a(i) = ai$ is XOR-universal (b/c linear + uniform)
- Define $\text{MAC}_\kappa(\bullet)$ to be a δ-secure one-time message authentication code (MAC) if Pr[Eve wins] is at most δ:
  - Pick a random κ; ask Eve for i and give Eve $\sigma_i = \text{MAC}_\kappa(i)$
  - Eve wins by outputting $j \neq i$ and $\sigma_j = \text{MAC}_\kappa(j)$
- Claim: if $f_a(\bullet)$ is XOR-universal then

  $\text{MAC}_{a,b}(i) = f_a(i) \oplus b$ is a $1/2^v$ secure MAC

  - Proof: guessing $\sigma_j \Leftrightarrow$ guessing $f_a(i) \oplus f_a(j)$, but b hides a
- Thus $\text{MAC}_{a,b}(i) = ai + b$ is a $1/2^v$-secure MAC ($|a|=|b|=|i|=v$)

background: MACs with imperfect keys

- Pr[Eve wins] = $\mathrm{E}_{\kappa \text{ chosen uniformly}}$ Pr[Eve wins for key = κ] ≤ δ
- What if κ is not uniform but has min-entropy k?

$$\mathrm{E}_{\kappa \text{ chosen from some entropy-}k\text{ distribution}} f(\kappa) = \sum f(\kappa)\,\Pr[\kappa]$$

(because f is nonnegative) $\quad \leq \sum f(\kappa)\, 2^{-k}$

$$= 2^{|\kappa|-k} \sum f(\kappa)\, 2^{-|\kappa|}$$

$$= 2^{|\kappa|-k}\, \mathrm{E}_{\kappa \text{ chosen uniformly}} f(\kappa)$$

$$= 2^{|\kappa|-k}\, \delta$$

- Security gets reduced by entropy deficiency!
- Thus $\mathrm{MAC}_{a,b}(i) = ai+b$ is $(2^{2v-k}/2^{v} = 2^{v-k})$-secure whenever $H_{\min}(a,b) = k$

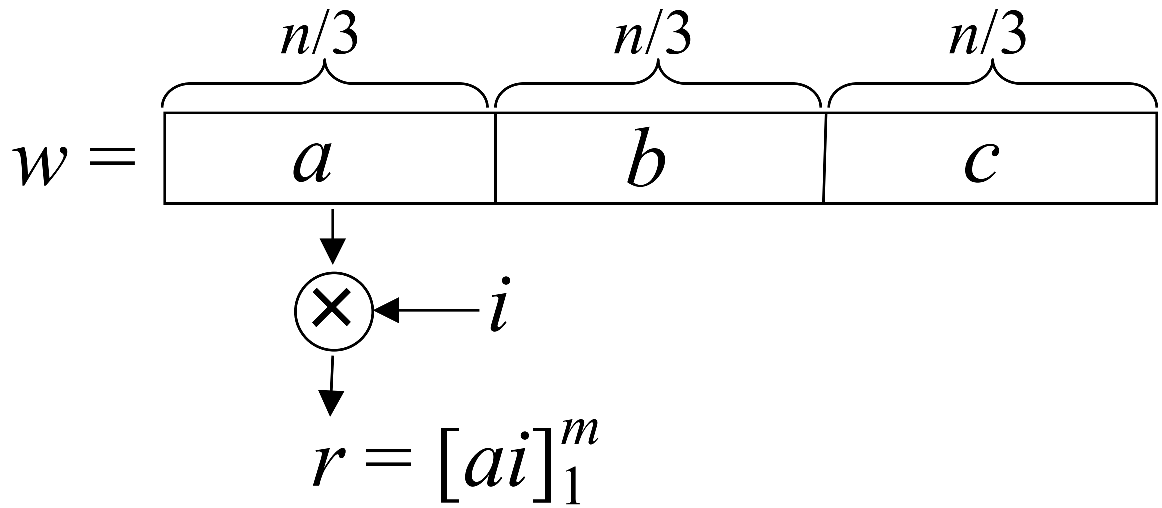
Outline

- Passive adversaries
  - Privacy amplification
  - Fuzzy extractors
  - Information reconciliation
- Active adversaries, w has a lot of entropy
  - Message authentication codes
  - Privacy amplification
  - Information reconciliation
- Active adversaries, w has little entropy
  - Privacy amplification
  - Information reconciliation

building robust extractors

Notation: $|w| = n$, $H_{\min}(w) = k$, "entropy deficiency" $n - k = g$

[Maurer-Wolf 97]



building robust extractors

<u>Notation</u>: $|w| = n$, $H_{\min}(w) = k$, "entropy deficiency" $n - k = g$

[Maurer-Wolf 97]

$$w = \underbrace{a}_{n/3}\ \underbrace{b}_{n/3}\ \underbrace{c}_{n/3}$$

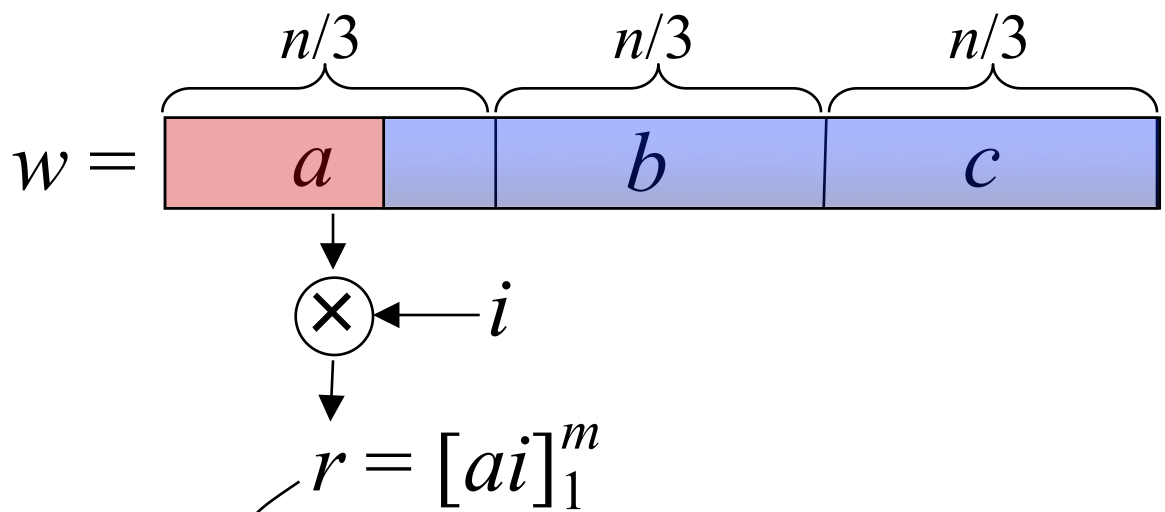
$$a \otimes i \rightarrow r = [ai]_1^m$$

ε-uniform
if $n/3 > m + g + 2\log\frac{1}{\varepsilon}$

building robust extractors

<u>Notation</u>: $|w| = n$, $H_{\min}(w) = k$, "entropy deficiency" $n - k = g$

[Maurer-Wolf 97]



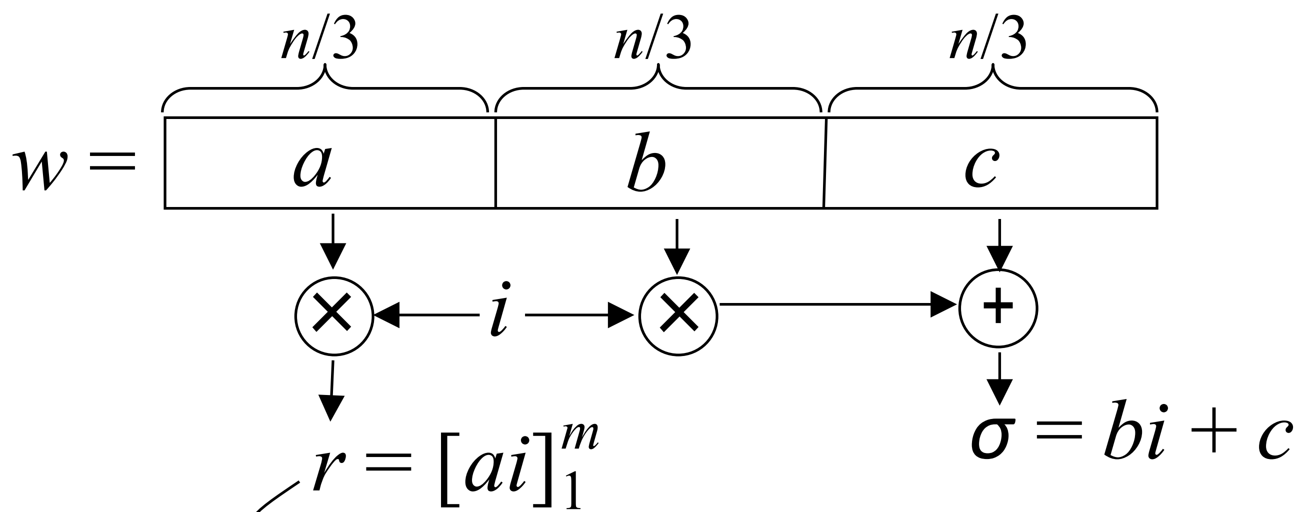
Extract if $k > 2n/3$

ε-uniform
if $n/3 > m + g + 2\log\frac{1}{\varepsilon}$

building robust extractors

Notation: $|w| = n$, $H_{\min}(w) = k$, "entropy deficiency" $n - k = g$

[Maurer-Wolf 97]



Extract if $k > 2n/3$

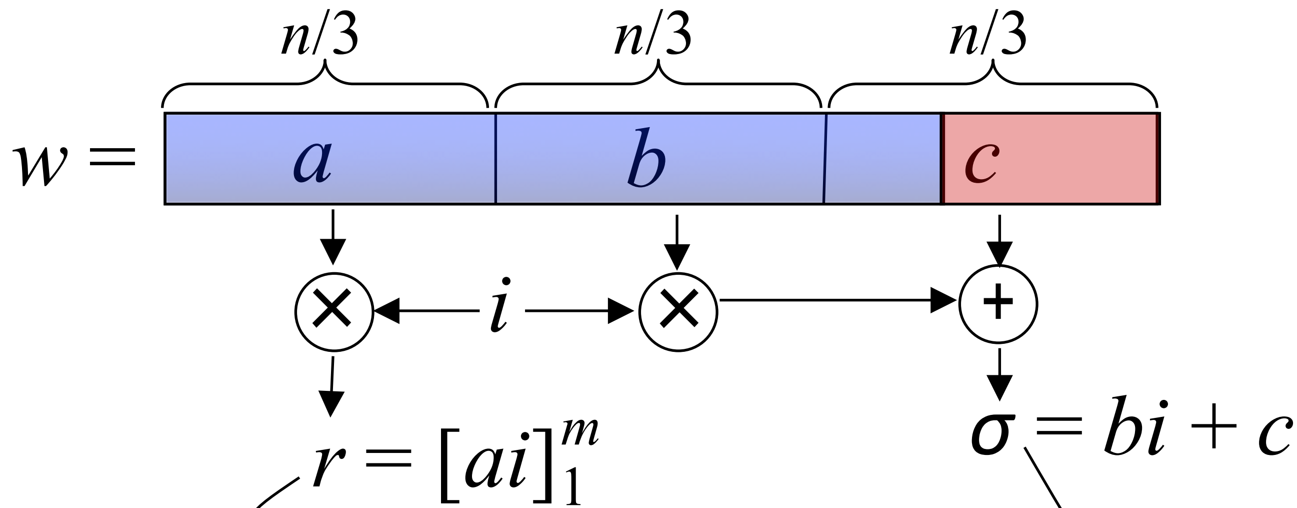
ε-uniform
if $n/3 > m + g + 2\log\frac{1}{\varepsilon}$

building robust extractors

Notation: $|w| = n$, $H_{\min}(w) = k$, "entropy deficiency" $n - k = g$

[Maurer-Wolf 97]

Extract if $k > 2n/3$



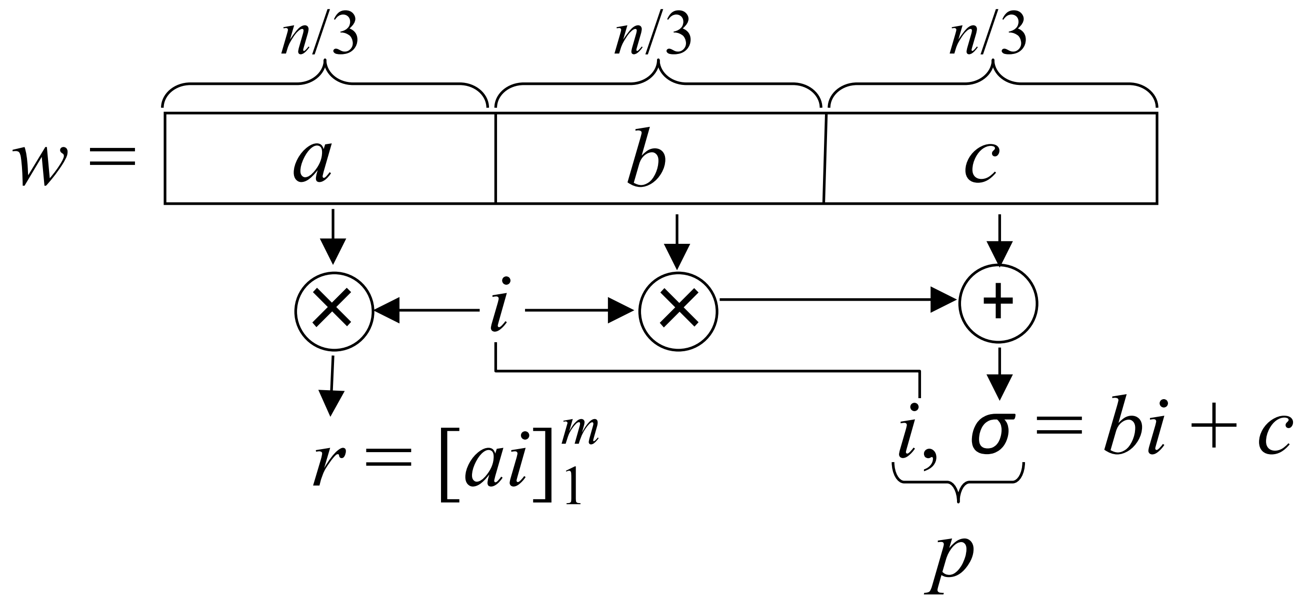
ε-uniform
if $n/3 > m + g + 2\log\frac{1}{\varepsilon}$

δ-robust
if $n/3 > g + \log\frac{1}{\delta}$

building robust extractors

<u>Notation</u>: $|w| = n$, $H_{\min}(w) = k$, "entropy deficiency" $n - k = g$

[Maurer-Wolf 97]



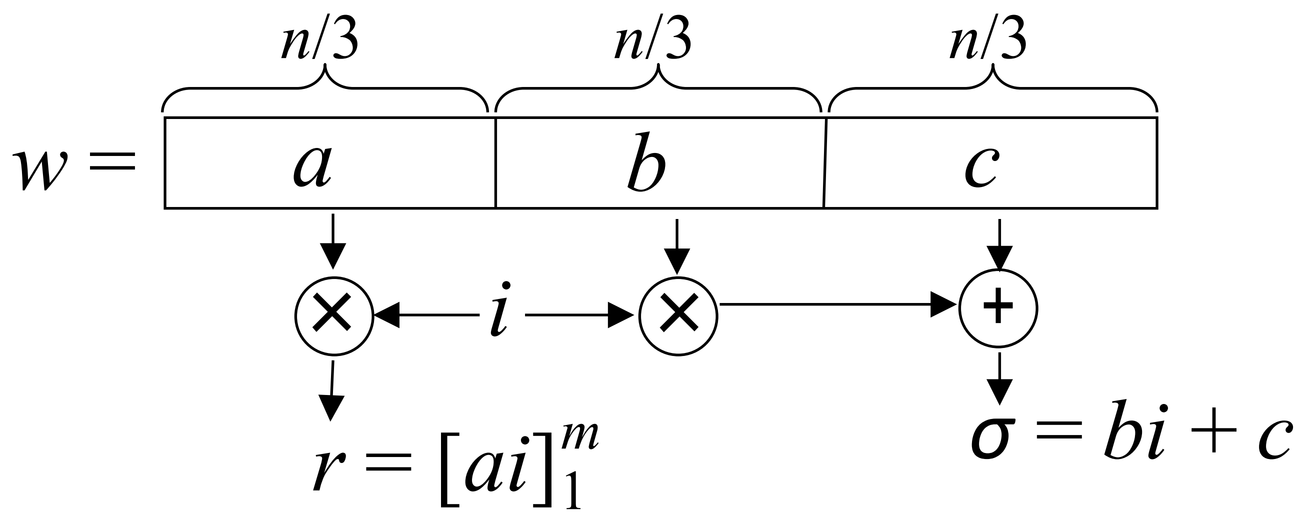
Extract if $k > 2n/3$

building robust extractors

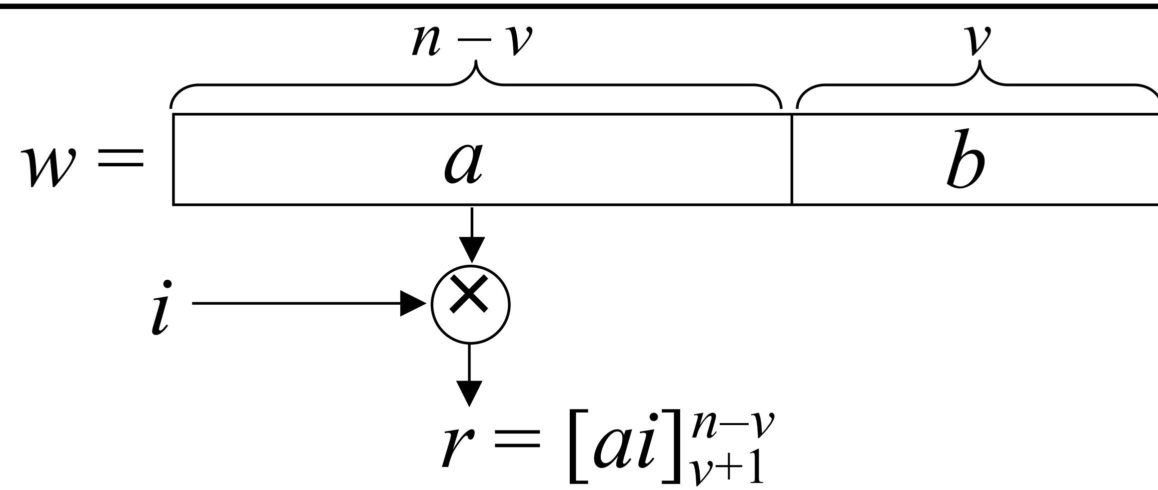
<u>Notation</u>: $|w| = n$, $H_{\min}(w) = k$, "entropy deficiency" $n - k = g$

[Maurer-Wolf 97]

Extract if $k > 2n/3$



[Dodis-Kanukurthi-Katz-Reyzin-Smith '12]

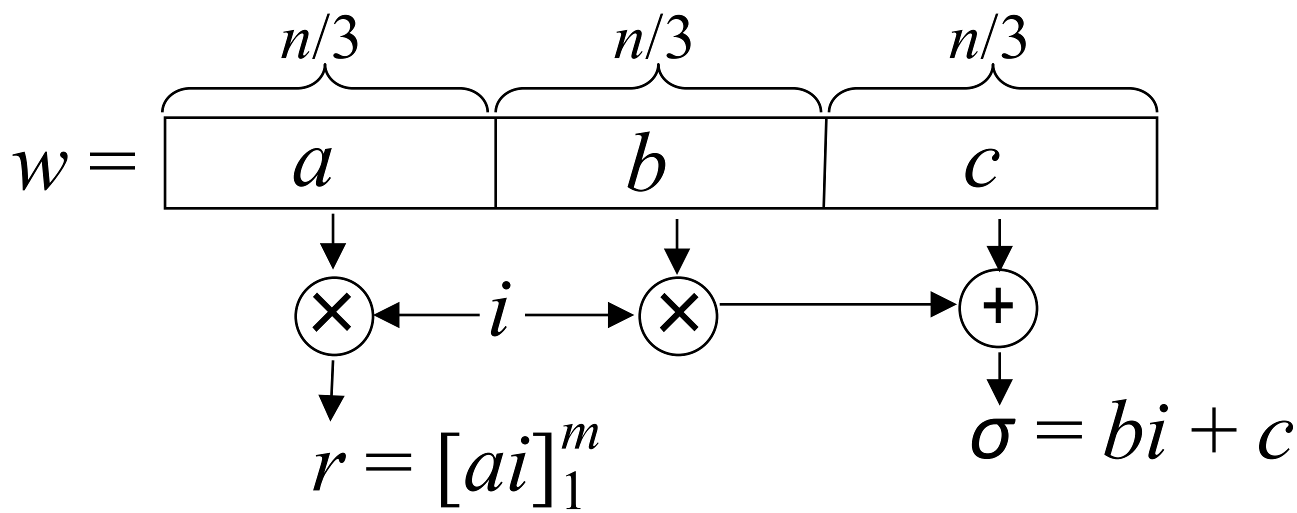


building robust extractors

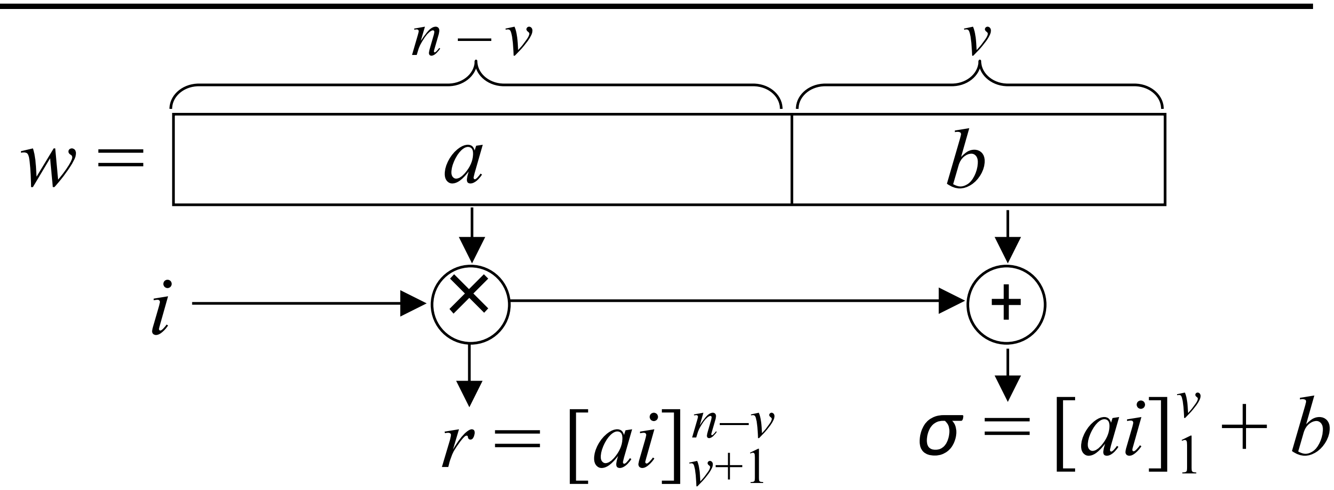
Notation: $|w| = n$, $H_{\min}(w) = k$, "entropy deficiency" $n - k = g$

[Maurer-Wolf 97]

Extract if $k > 2n/3$



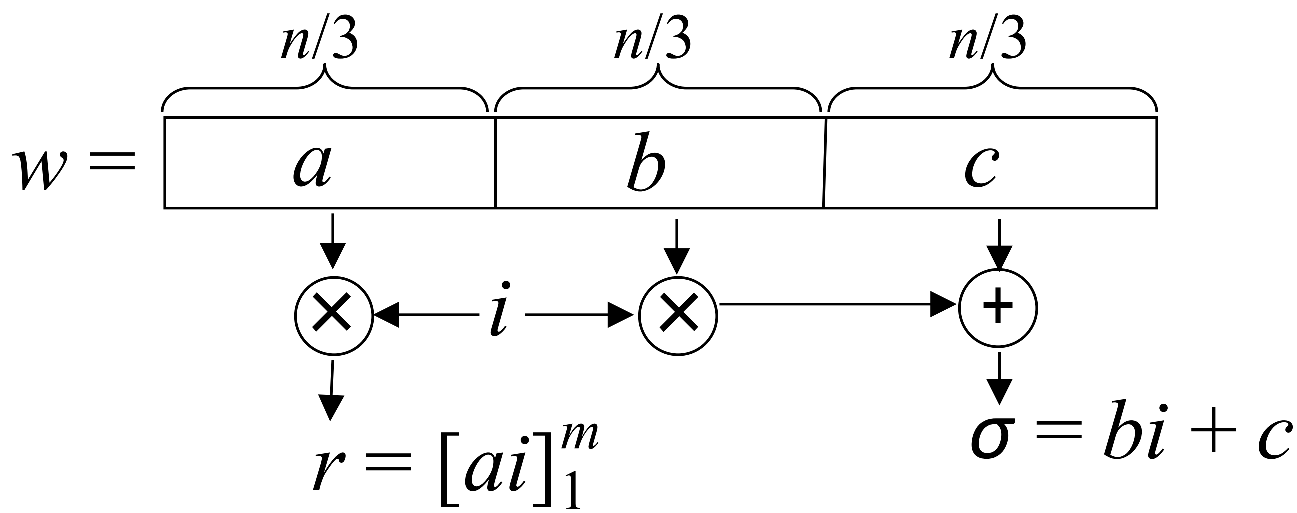
[Dodis-Kanukurthi-Katz-Reyzin-Smith '12]



building robust extractors

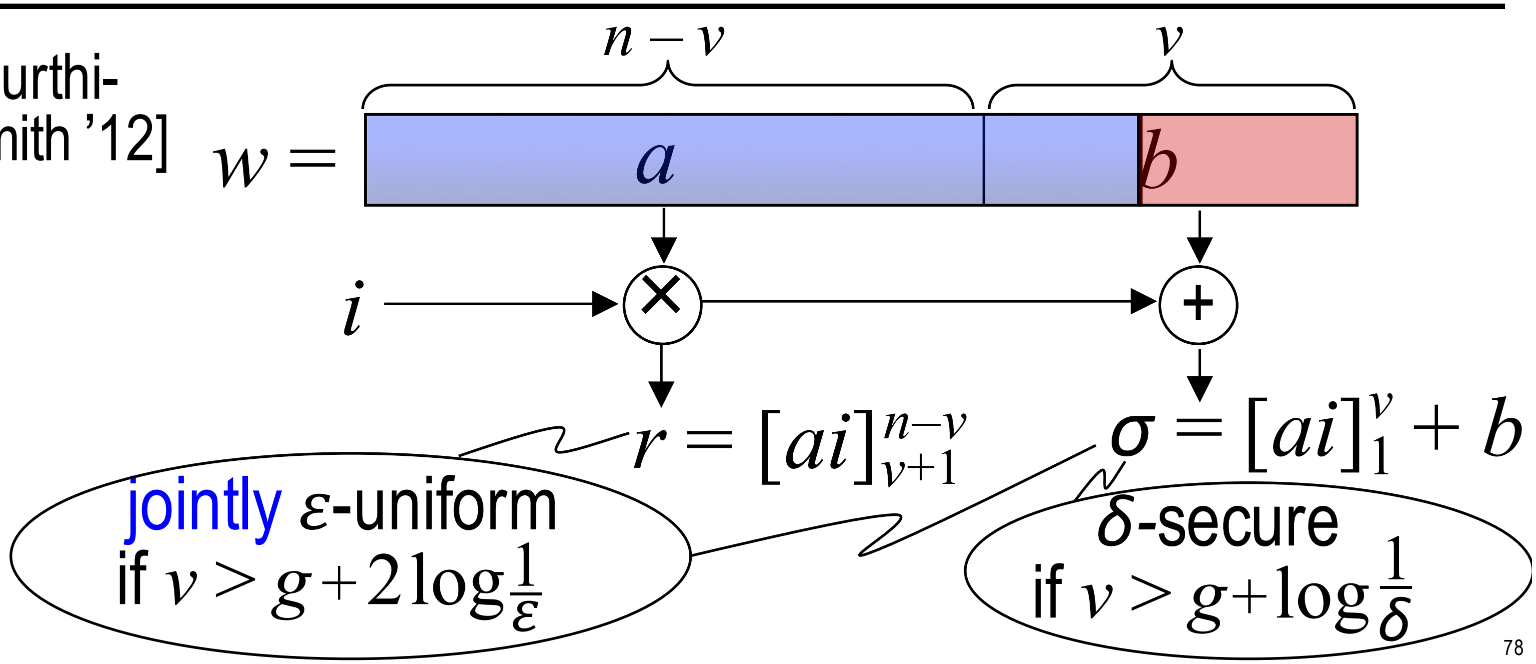
Notation: $|w| = n$, $H_{\min}(w) = k$, "entropy deficiency" $n - k = g$

[Maurer-Wolf 97]

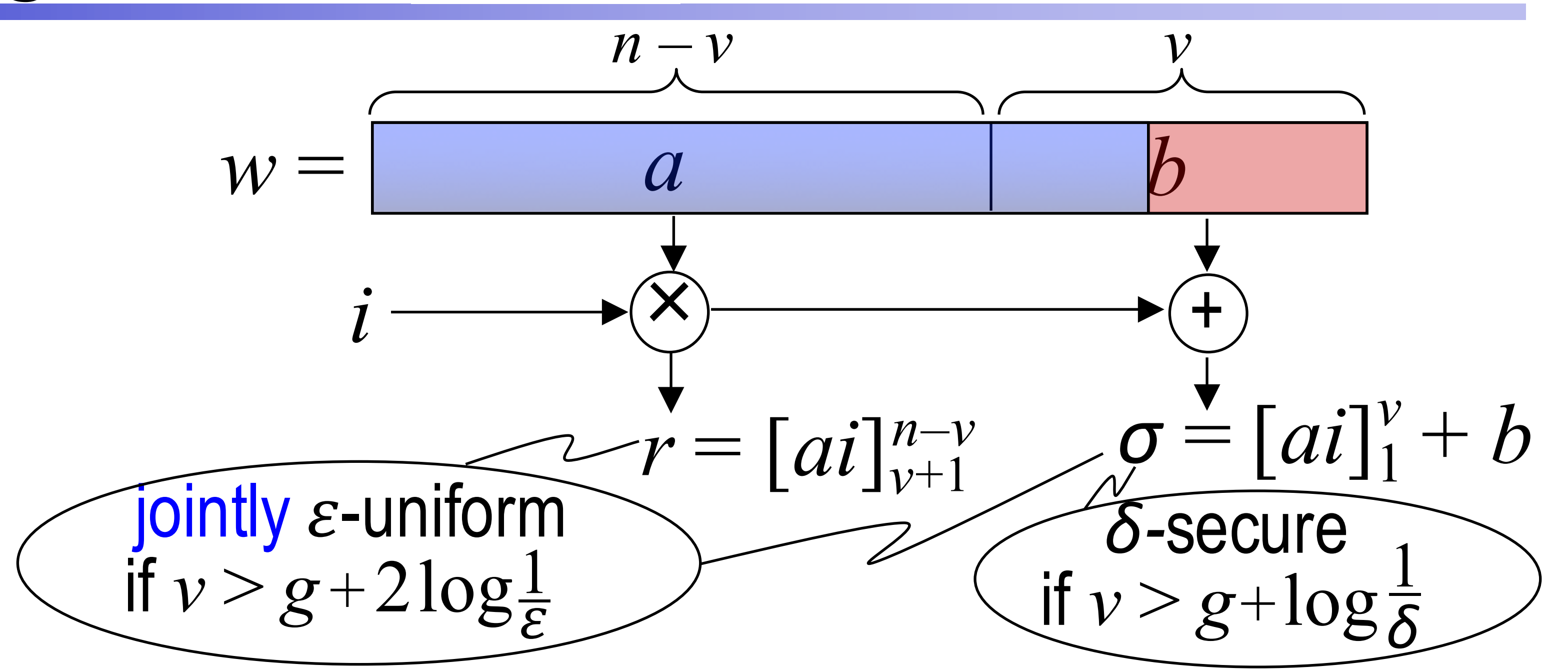


Extract if $k > 2n/3$

[Dodis-Kanukurthi-Katz-Reyzin-Smith '12]



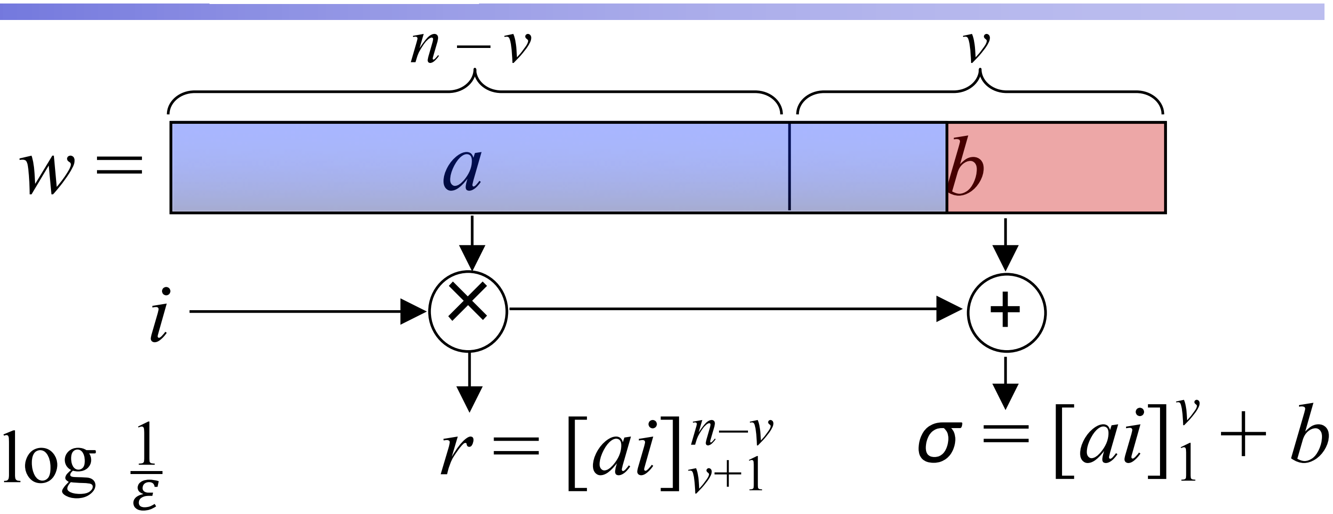
building robust extractors



Analysis:

- Extraction: $(R,\sigma)=ai+b$ is a universal hash family (few collisions) (i is the key, $w=(a,b)$ is the input) [ok by leftover hash lemma]
- Robustness: $\sigma=[ai]_1^v$ is XOR-universal ($w=(a,b)$ is the key, i is the input) [ok by Maurer-Wolf]

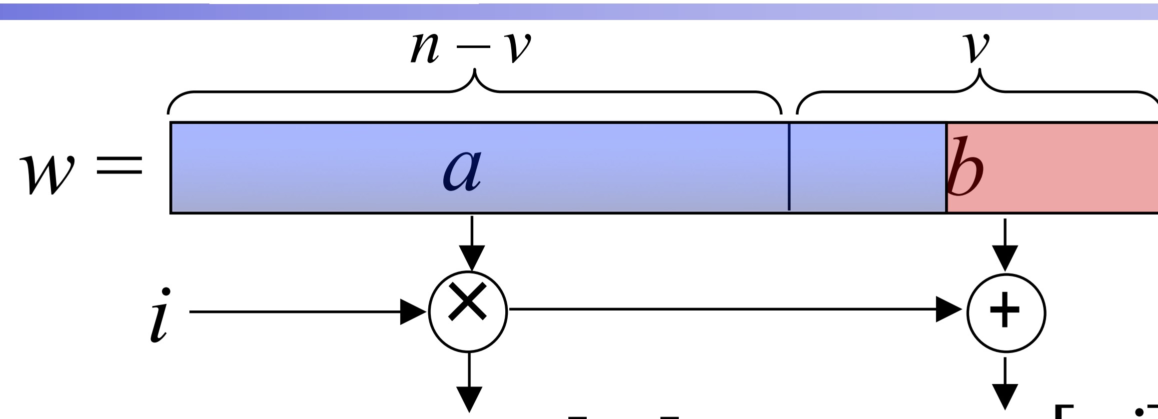
building robust extractors ?



Extract $k-g-2\log\frac{1}{\varepsilon}$

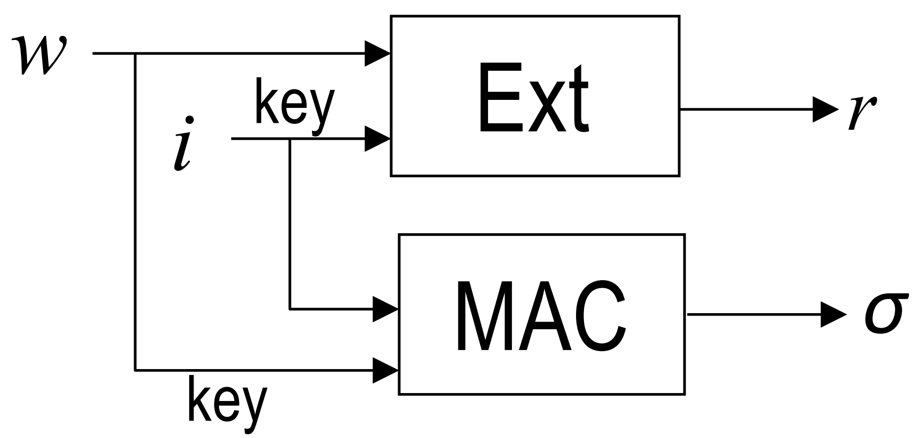
$k>n/2$ is necessary [Dodis-Wichs09]

building robust extractors ?



Extract $k-g-2\log\frac{1}{\varepsilon}$ $\qquad r=[ai]_{v+1}^{n-v}$ $\qquad \sigma=[ai]_1^v+b$

$k>n/2$ is necessary [Dodis-Wichs09]

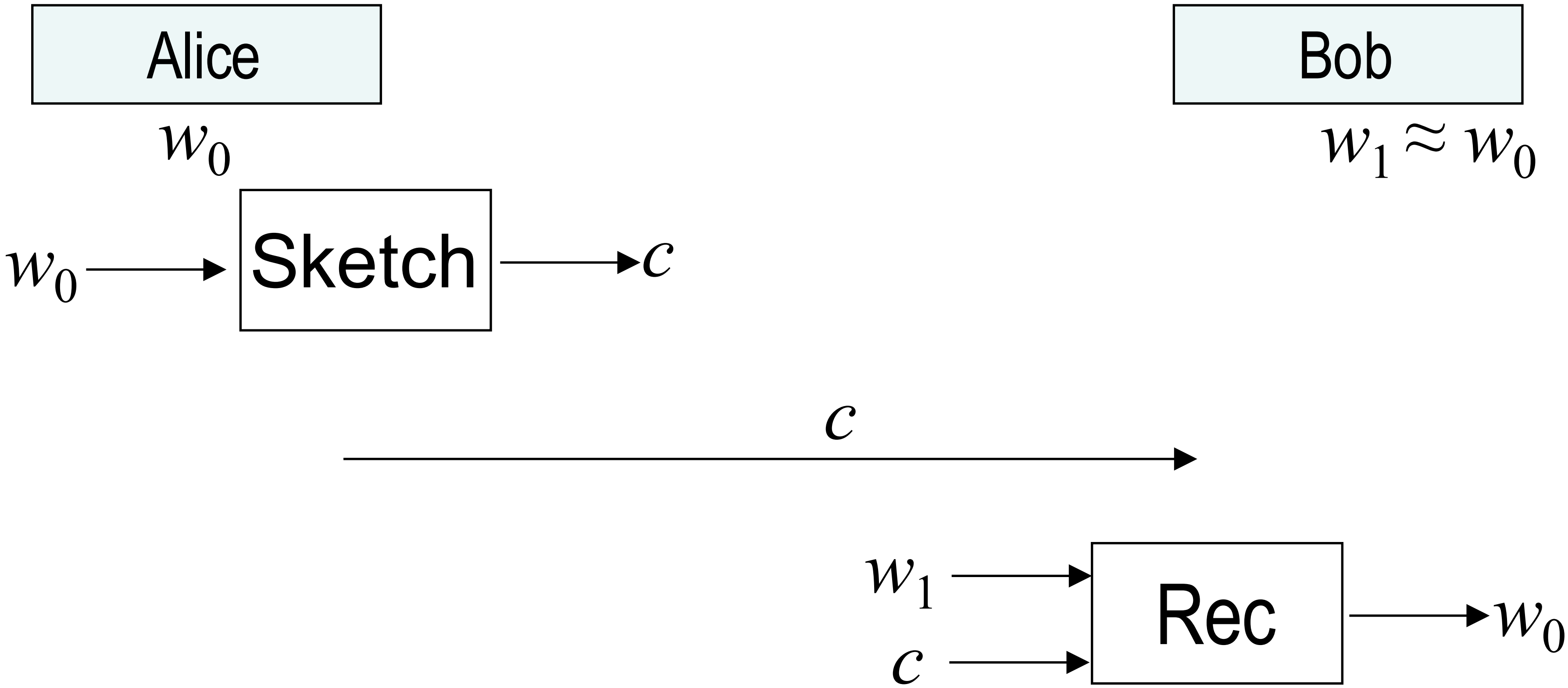


$p=(i,\sigma)$

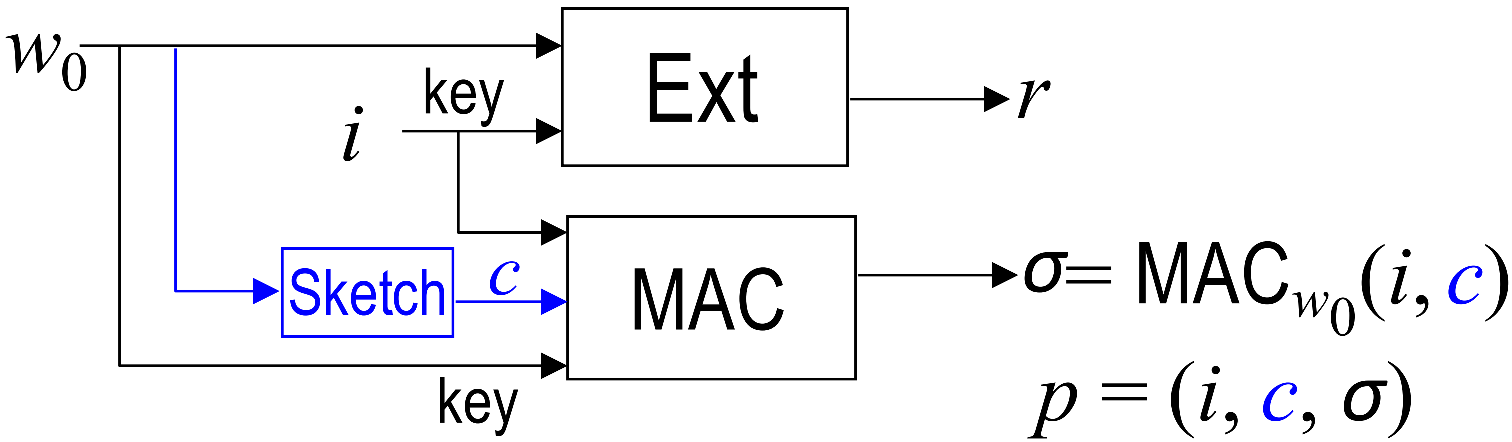
Outline

- Passive adversaries
  - Privacy amplification
  - Fuzzy extractors
  - Information reconciliation
- Active adversaries, w has a lot of entropy
  - Message authentication codes
  - Privacy amplification
  - Information reconciliation
- Active adversaries, w has little entropy
  - Privacy amplification
  - Information reconciliation

recall: secure sketch



building robust fuzzy extractors

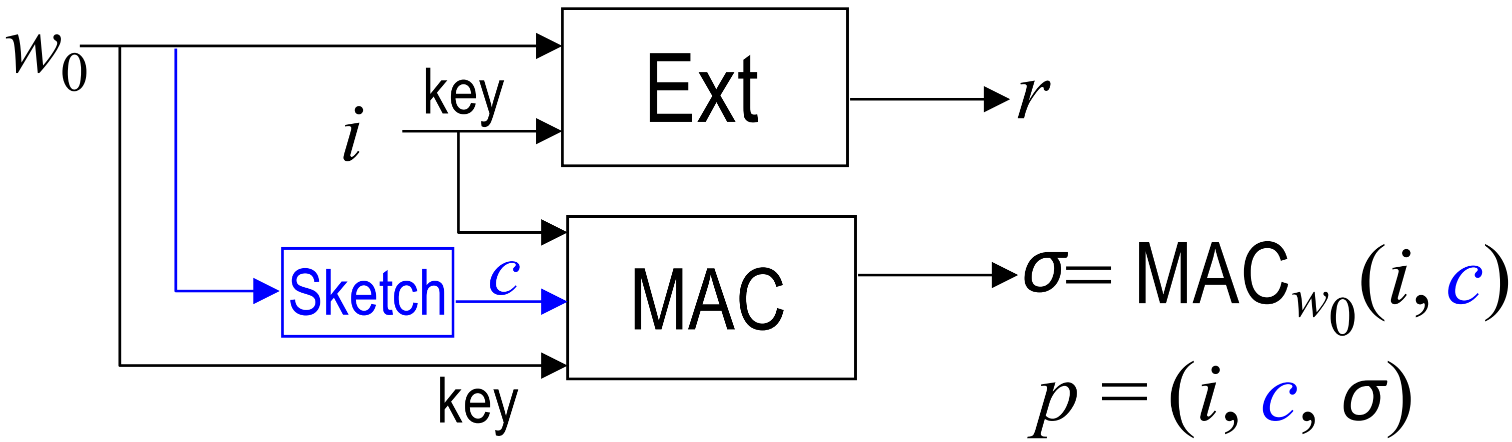


$p = (i, c, \sigma)$

How to MAC long messages? $\sigma = [a^2c + ai]_1^v + b$

(recall $w = a|b$)

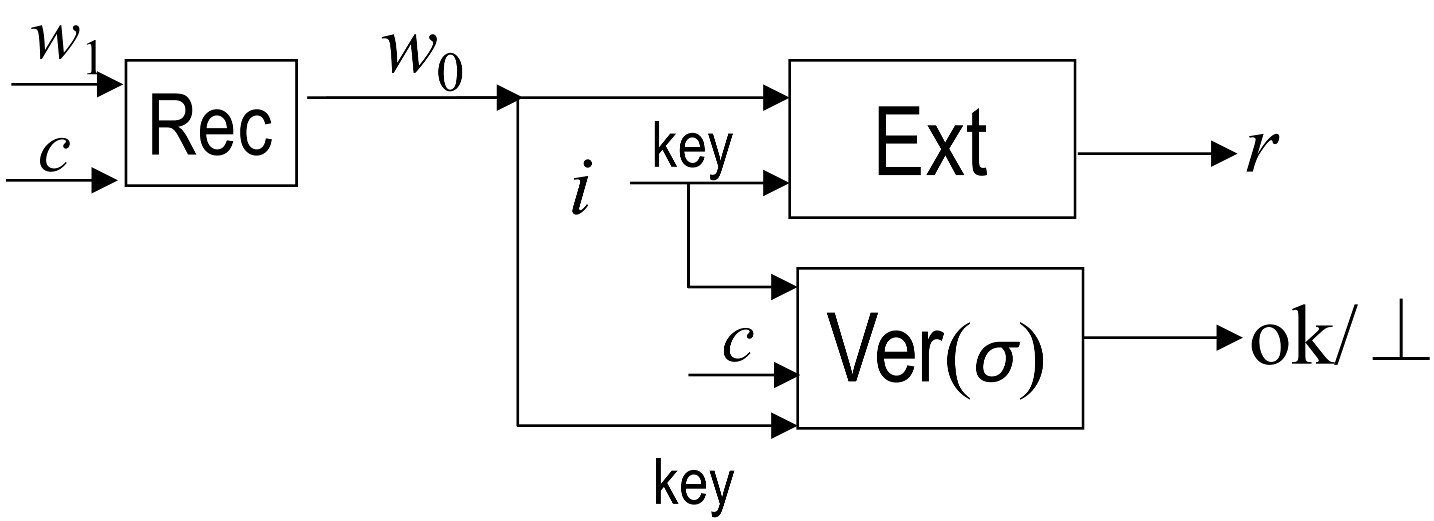
building robust fuzzy extractors



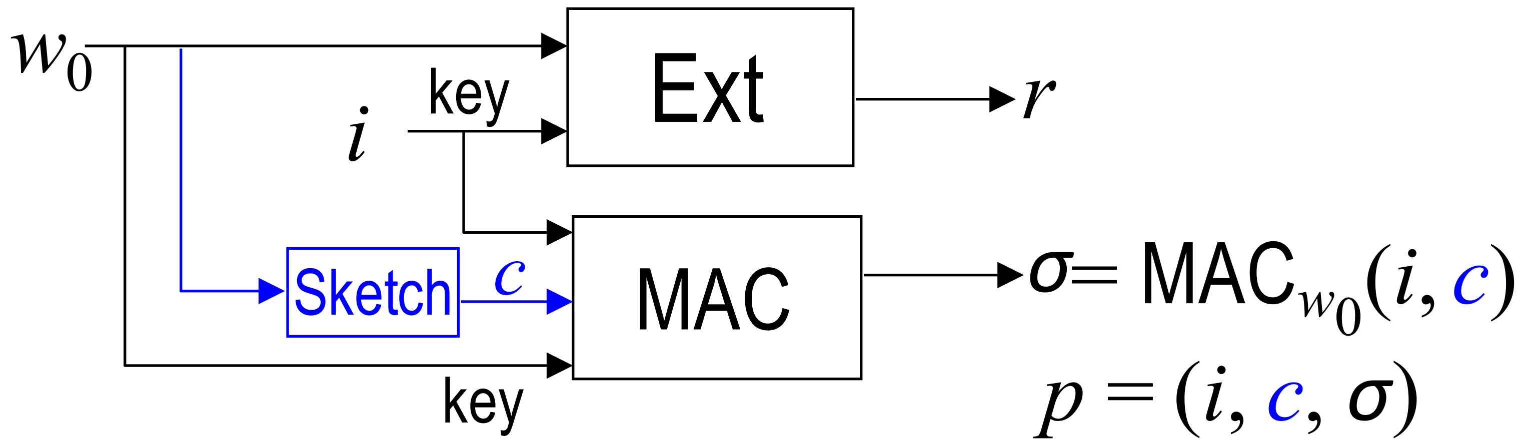
How to MAC long messages? $\sigma = [a^2 c + ai]_1^v + b$

(recall $w = a|b$)

How to Rep



building robust fuzzy extractors

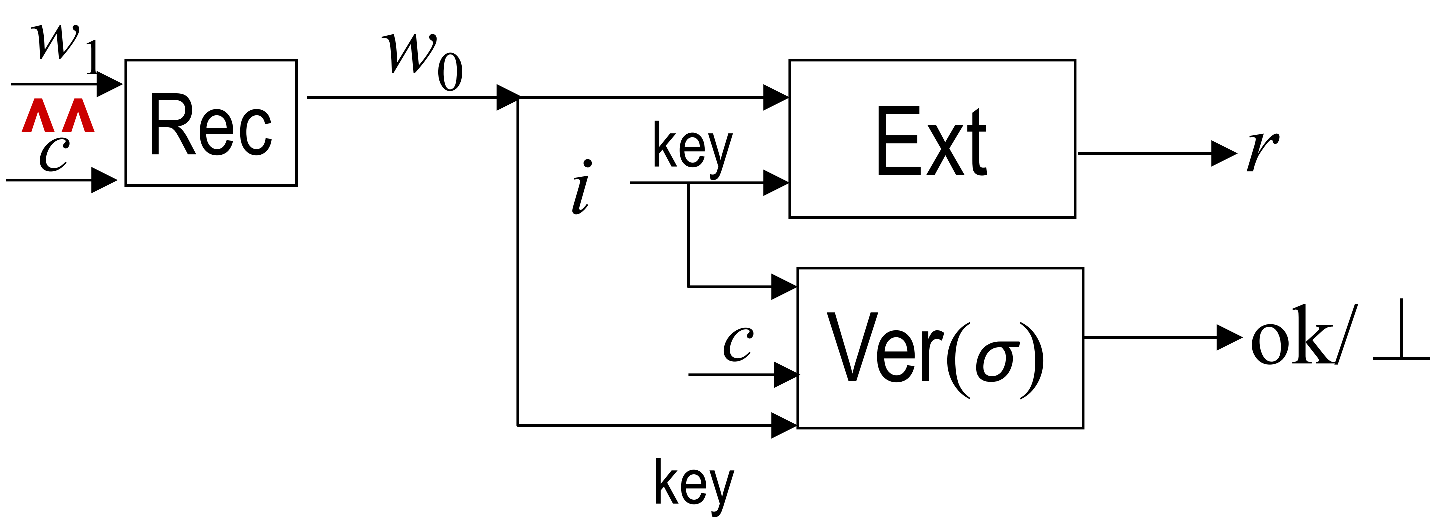


$p = (i, c, \sigma)$

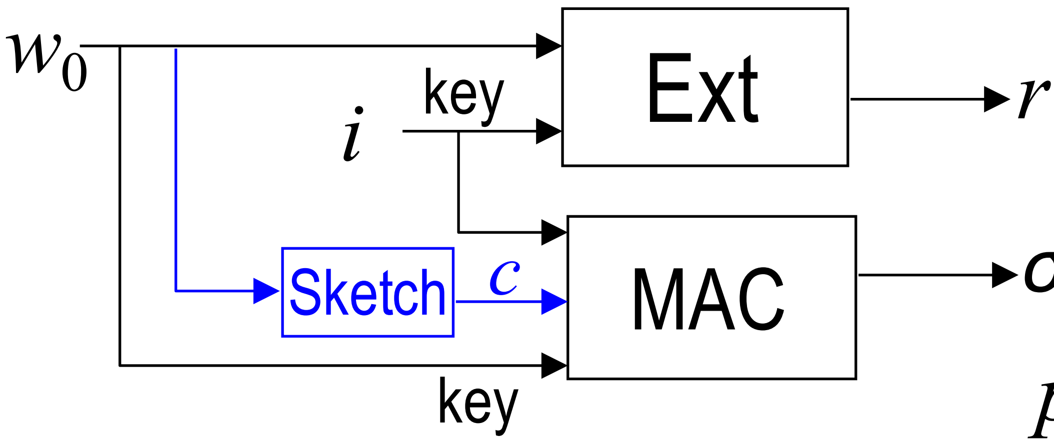
How to MAC long messages? $\sigma = [a^2c + ai]_1^v + b$

(recall $w = a|b$)

How to Rep



building robust fuzzy extractors



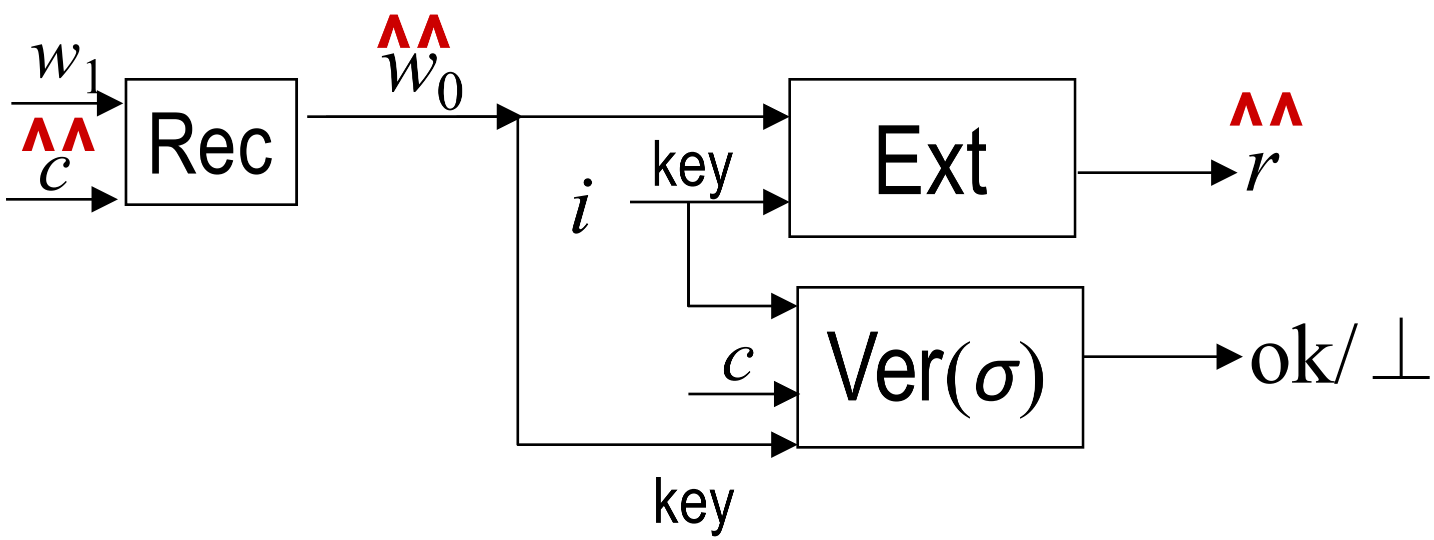
$\sigma = \text{MAC}_{w_0}(i, c)$

$p = (i, c, \sigma)$

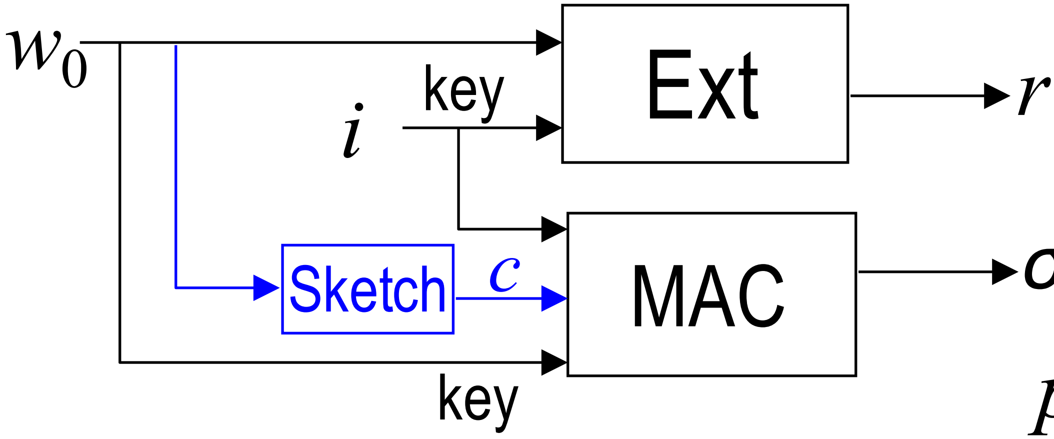
How to MAC long messages? $\sigma = [a^2 c + ai]_1^v + b$

(recall $w = a|b$)

How to Rep



building robust fuzzy extractors



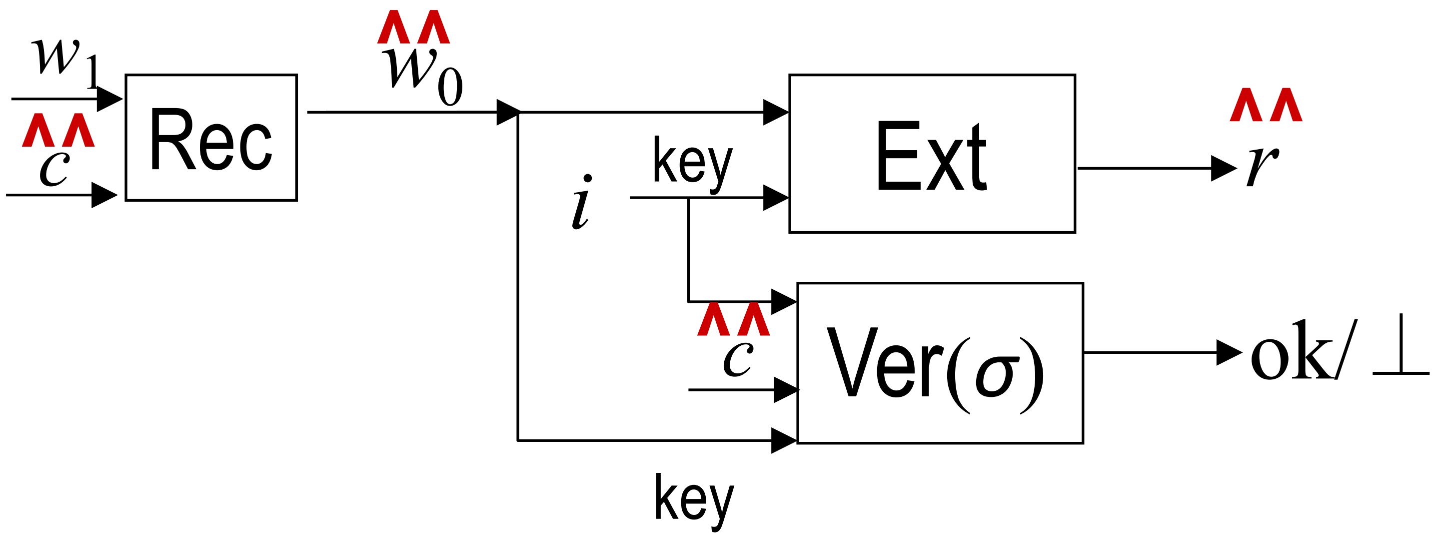
$\sigma = \text{MAC}_{w_0}(i, c)$

$p = (i, c, \sigma)$

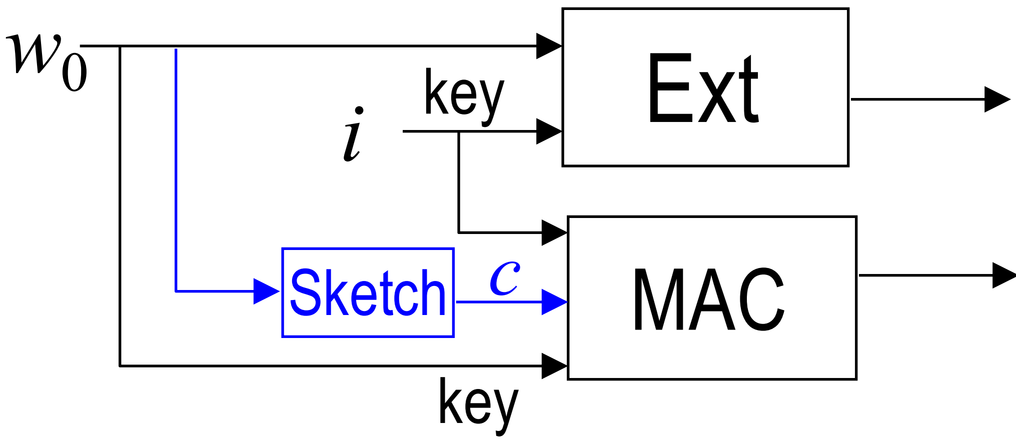
How to MAC long messages? $\sigma = [a^2 c + ai]_1^v + b$

(recall $w = a|b$)

How to Rep



building robust fuzzy extractors



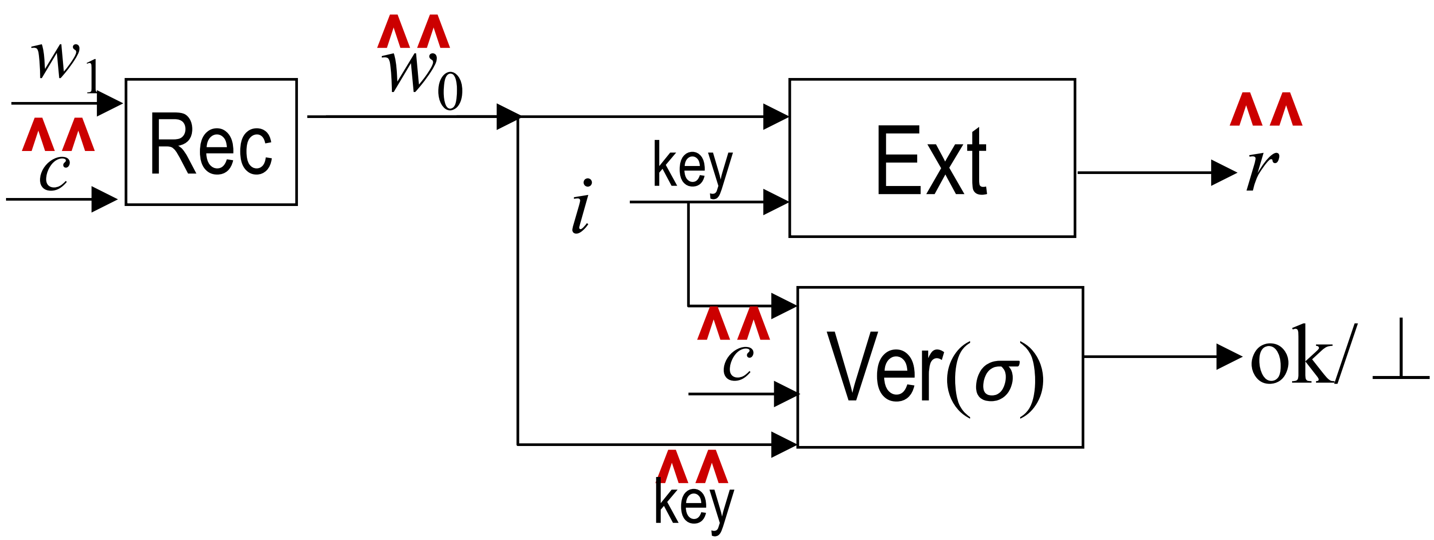
$\sigma = \mathrm{MAC}_{w_0}(i, c)$

$p = (i, c, \sigma)$

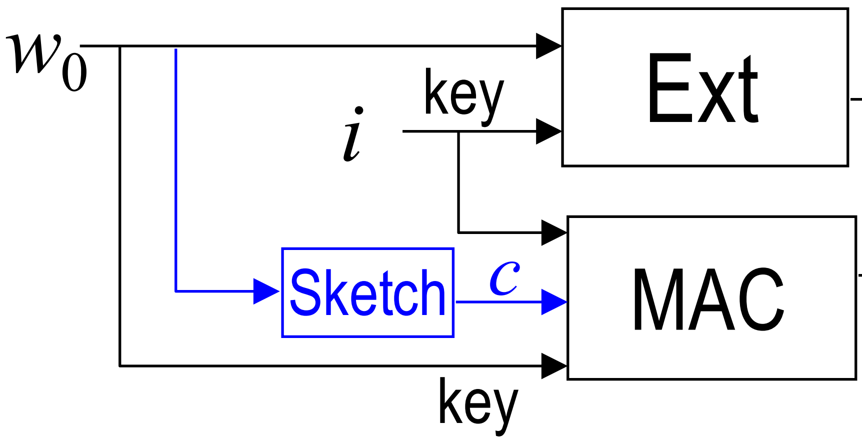
How to MAC long messages? $\sigma = [a^2c + ai]_1^v + b$

(recall $w = a|b$)

How to Rep



building robust fuzzy extractors



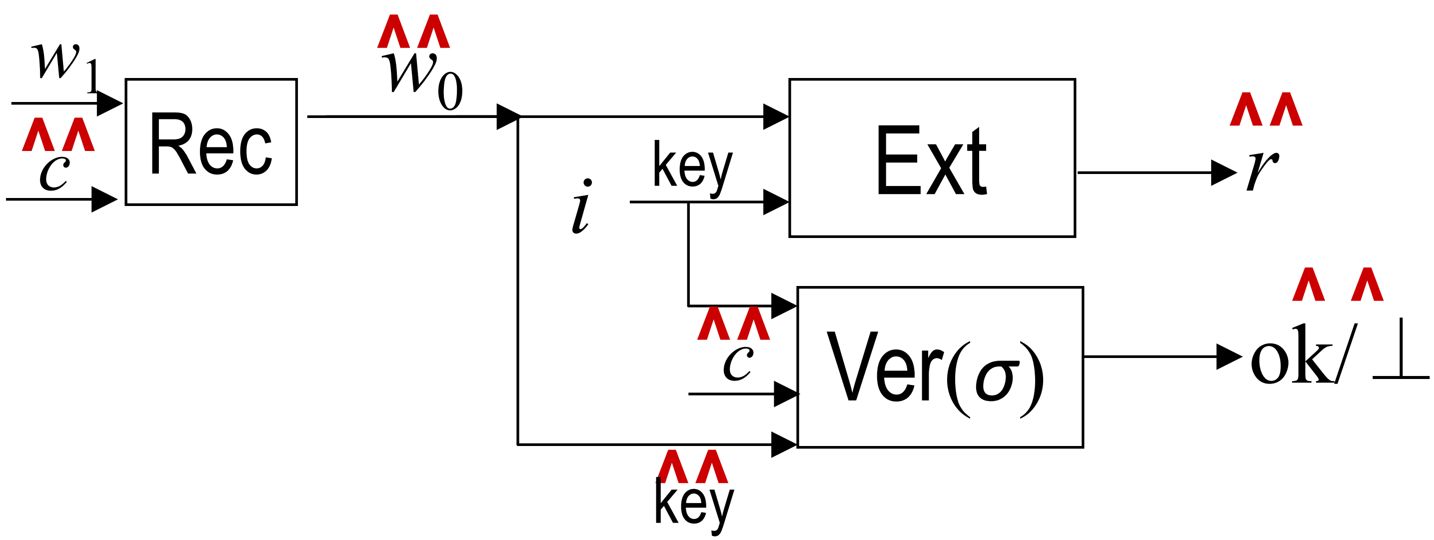
$\sigma = \text{MAC}_{w_0}(i, c)$

$p = (i, c, \sigma)$

How to MAC long messages? $\sigma = [a^2 c + ai]_1^v + b$

(recall $w = a|b$)

How to Rep



oops…

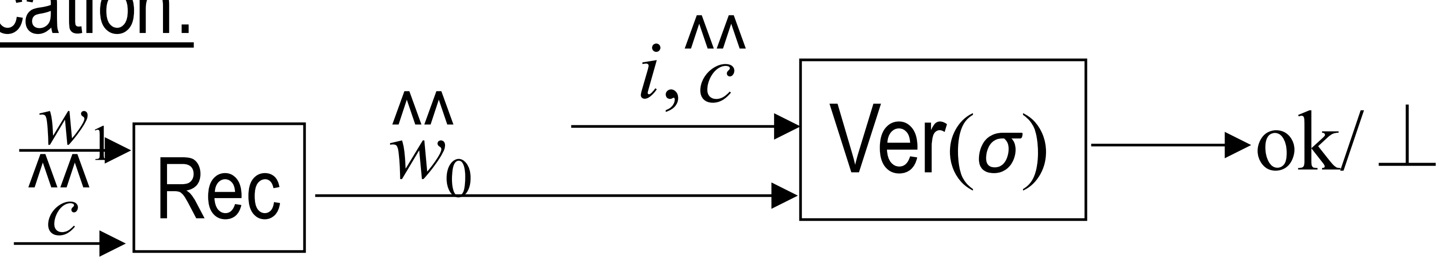
the MAC problem

Authentication:

$$\sigma = \mathrm{MAC}_w(i,c) = [a^2c + ai]_1^v + b$$

(recall $w = a|b$)

Verification:



Problem: circularity (MAC key depends on c, which is being authenticated by the MAC)

Observe: knowing ($w_1 \oplus w_0$ and $c \oplus \hat{c}$) gives knowledge of $w_0 \oplus \hat{w}_0 = u$

Need: $\forall\, u$, given $\mathrm{MAC}_w(i,c)$, hard to forge $\mathrm{MAC}_{w+u}(\hat{i}, \hat{c})$

the MAC problem

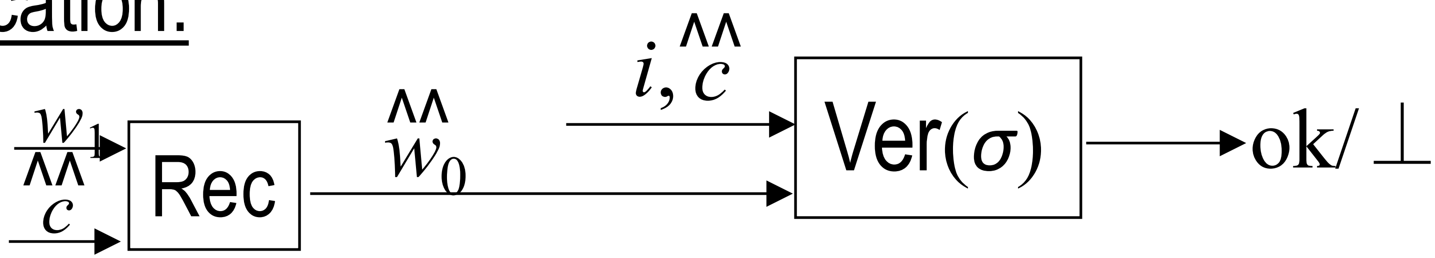
Hard to forge for any fixed u

Authentication:

$$\sigma = \mathrm{MAC}_w(i, c) = [a^5 + a^2 c + ai]_1 + b$$

(recall $w = a|b$)

Verification:



Problem: circularity (MAC key depends on c, which is being authenticated by the MAC)

Observe: knowing ($w_1 \oplus w_0$ and $c \oplus \hat{c}$) gives knowledge of $w_0 \oplus \hat{w}_0 = u$

Need: $\forall\, u$, given $\mathrm{MAC}_w(i, c)$, hard to forge $\mathrm{MAC}_{w+u}(\hat{i}, \hat{c})$

the MAC problem

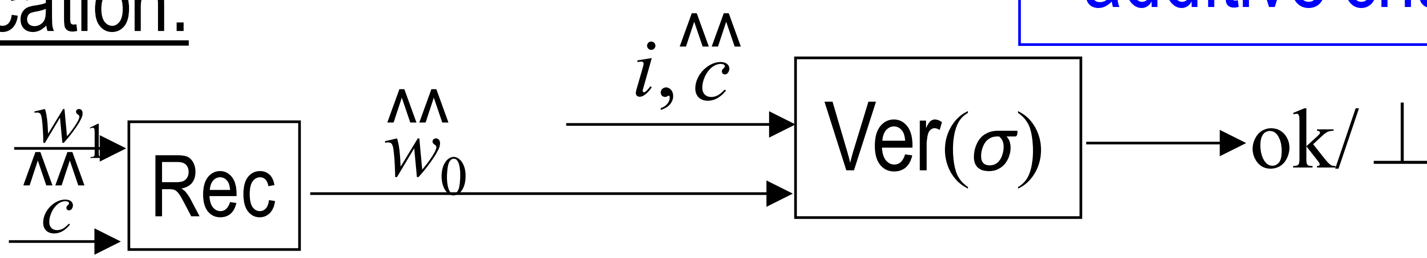
Authentication   Generalization [Padro et al. '05] if i is public

$\sigma = \mathrm{MAC}_w(i, c) = \text{AMD-Code}(a, c)+b$

(recall $w = a|b$)

Code that detects additive change

Verification:



Problem: circularity (MAC key depends on c, which is being authenticated by the MAC)

Observe: knowing ($w_1 \oplus w_0$ and $c \oplus \hat{c}$) gives knowledge of $w_0 \oplus \hat{w}_0 = u$

Need: $\forall u$, given $\mathrm{MAC}_w(i, c)$, hard to forge $\mathrm{MAC}_{w+u}(\hat{i}, \hat{c})$

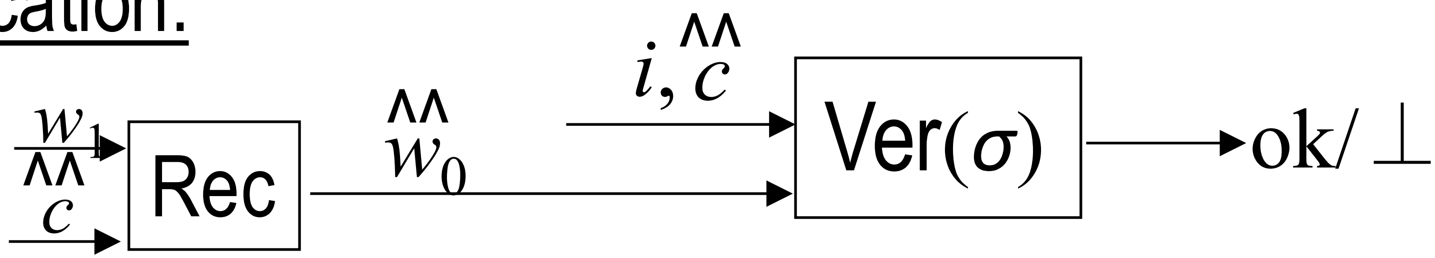
the MAC problem

Authentication   Alternative [Boyen et al. '05]

$\sigma = \mathrm{MAC}_w(i, c) = \mathrm{RandomOracle}(w, i, c)$

Advantage: works even when $H_{\min}(w) < n/2$

Verification:

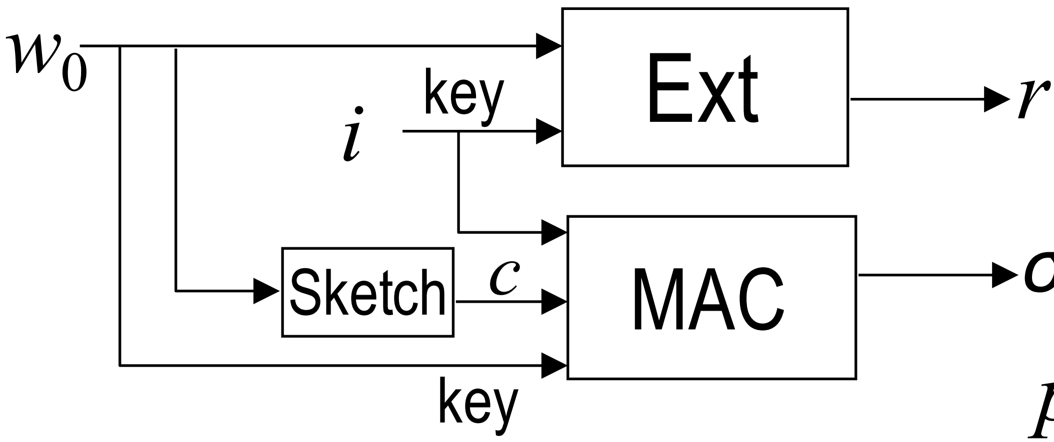


Problem: circularity (MAC key depends on c, which is being authenticated by the MAC)

Observe: knowing ($w_1 \oplus w_0$ and $c \oplus \hat{c}$) gives knowledge of $w_0 \oplus \hat{w}_0 = u$

Need: $\forall u$, given $\mathrm{MAC}_w(i, c)$, hard to forge $\mathrm{MAC}_{w+u}(\hat{i}, \hat{c})$

building robust fuzzy extractors



$\sigma = \text{MAC}_{w_0}(i, c)$

$p = (i, c, \sigma)$

Recall: without errors, extract $k - g - 2\log\frac{1}{\varepsilon}$

Problem: c reveals l bits about $w \Rightarrow$

k decreases, g increases $\Rightarrow$

lose $2l$

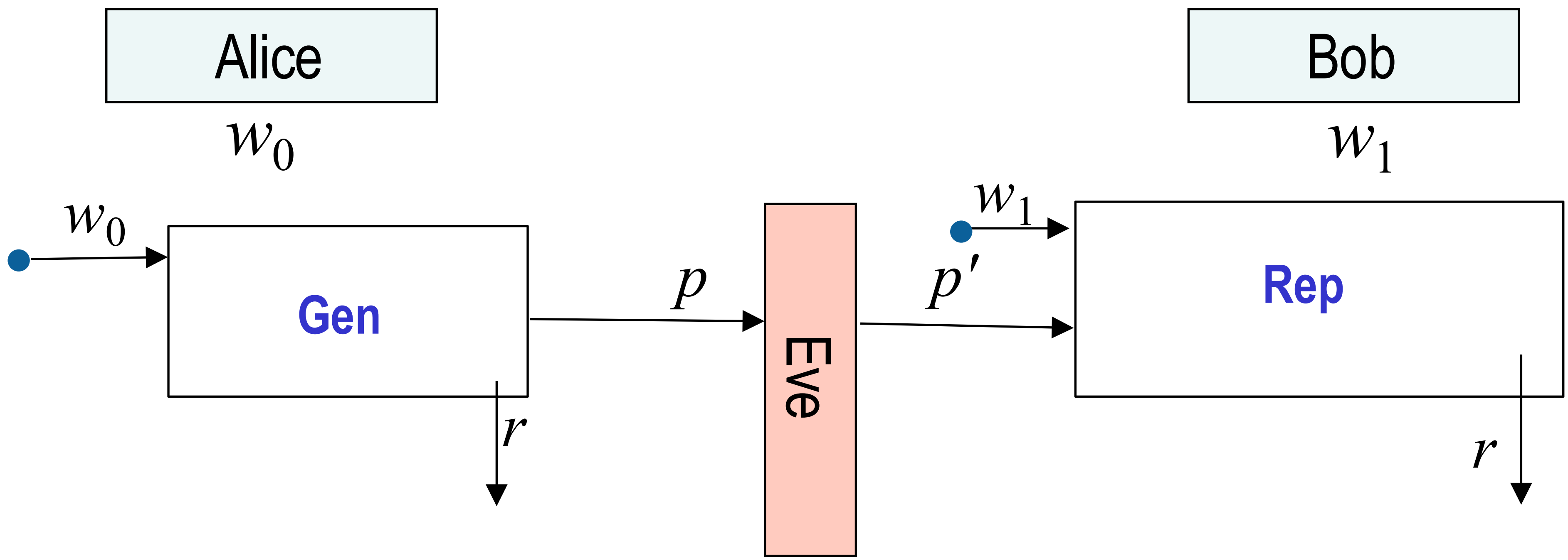
Can't avoid decreasing k, but can avoid increasing g

c = Sketch(w_0) is linear. Let $d = \text{Sketch}^{\perp}(w_0)$.

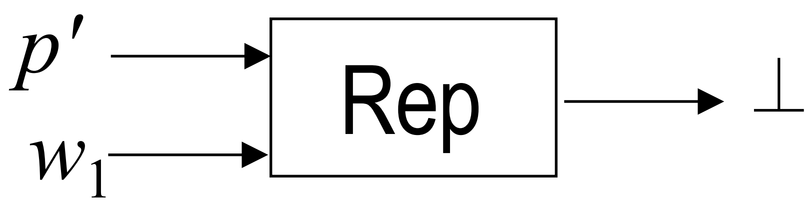
$|d| = |w| - l$, but d has entropy $k - l$. Use d instead of w_0.

Result: extract $k - l - g - 2\log\frac{1}{\varepsilon}$

Summary: robust fuzzy extractors

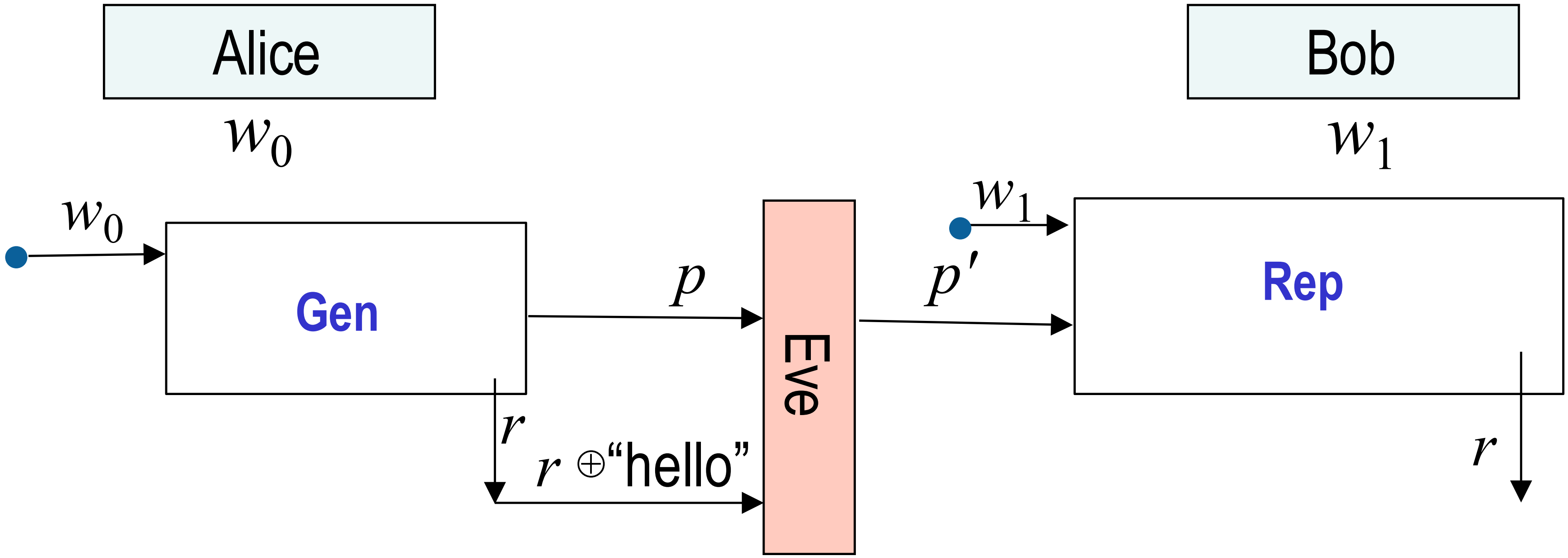


Robustness: as long as $w_0 \approx w_1$, if Eve(p) produces $p' \neq p$

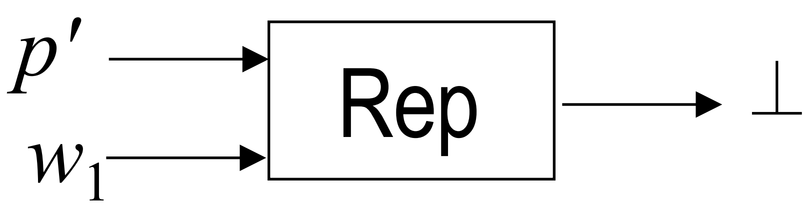


(with 1 – negligible probability over w_0 & coins of Rep, Eve)

Summary: robust fuzzy extractors

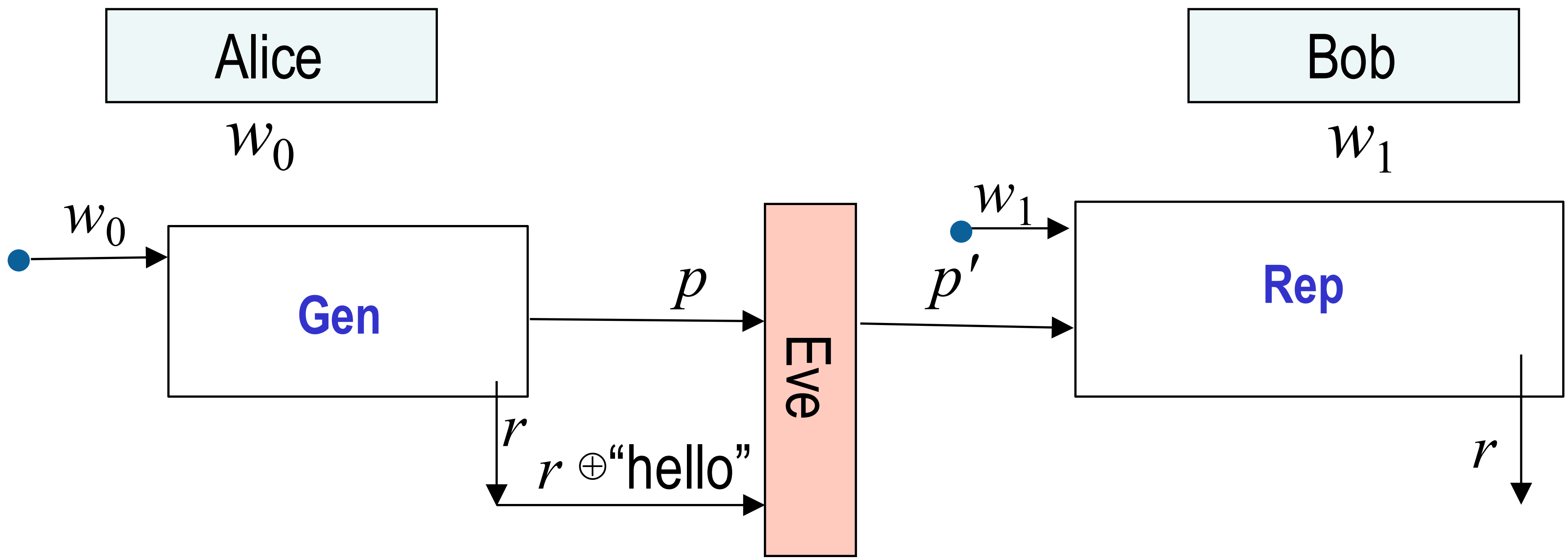


Robustness: as long as $w_0 \approx w_1$, if Eve(p) produces $p' \neq p$



(with 1 – negligible probability over w_0 & coins of Rep, Eve)

Summary: robust fuzzy extractors



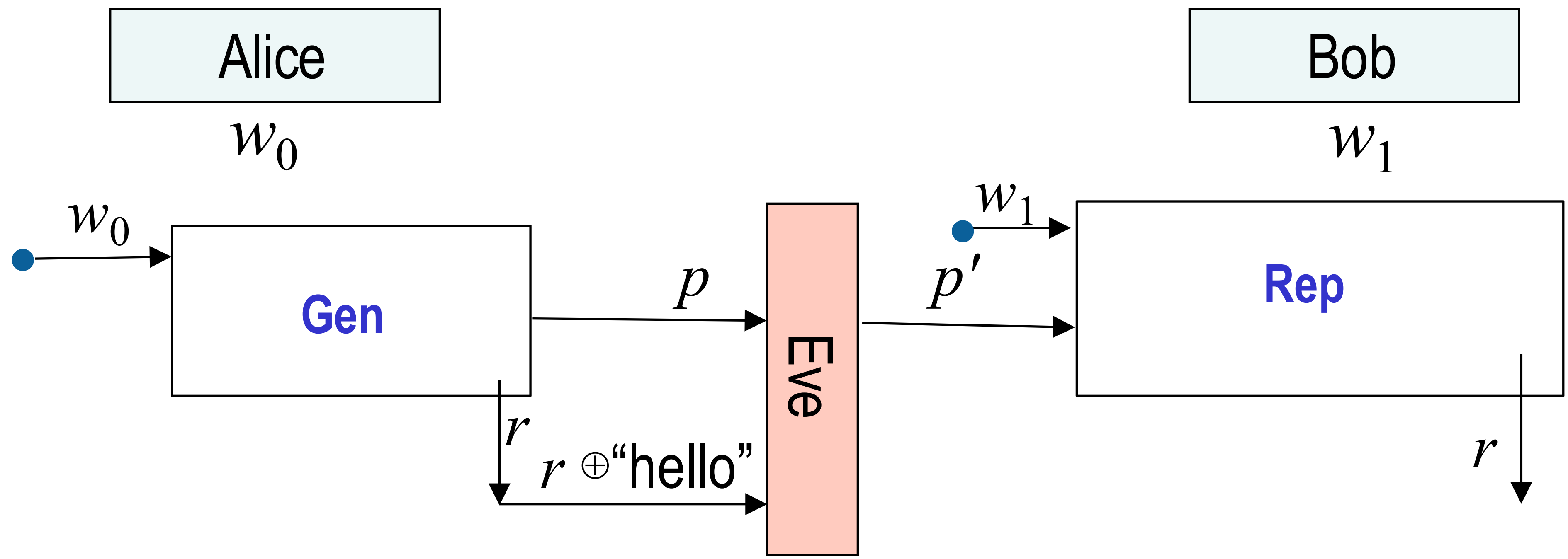
Post-Application

Robustness: as long as $w_0 \approx w_1$, if Eve(p,r) produces $p' \neq p$

$$p',\ w_1 \longrightarrow \boxed{\text{Rep}} \longrightarrow \perp$$

(with 1 – negligible probability over w_0 & coins of Rep, Eve)

Post-application robustness



Post-Application
Robustness:

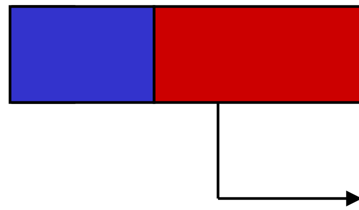
[DKKRS12]: a similar construction extracts about $(k–l–g)/2$
(half as much as pre-application)

Outline

- Passive adversaries
  - Privacy amplification
  - Fuzzy extractors
  - Information reconciliation
- Active adversaries, w has a lot of entropy
  - Message authentication codes
  - Privacy amplification only when $H_{\min}(w) > |w|/2$
  - Information reconciliation
  - Two security notions (pre-application vs. post-application)
- Active adversaries, w has little entropy
  - Privacy amplification
  - Information reconciliation

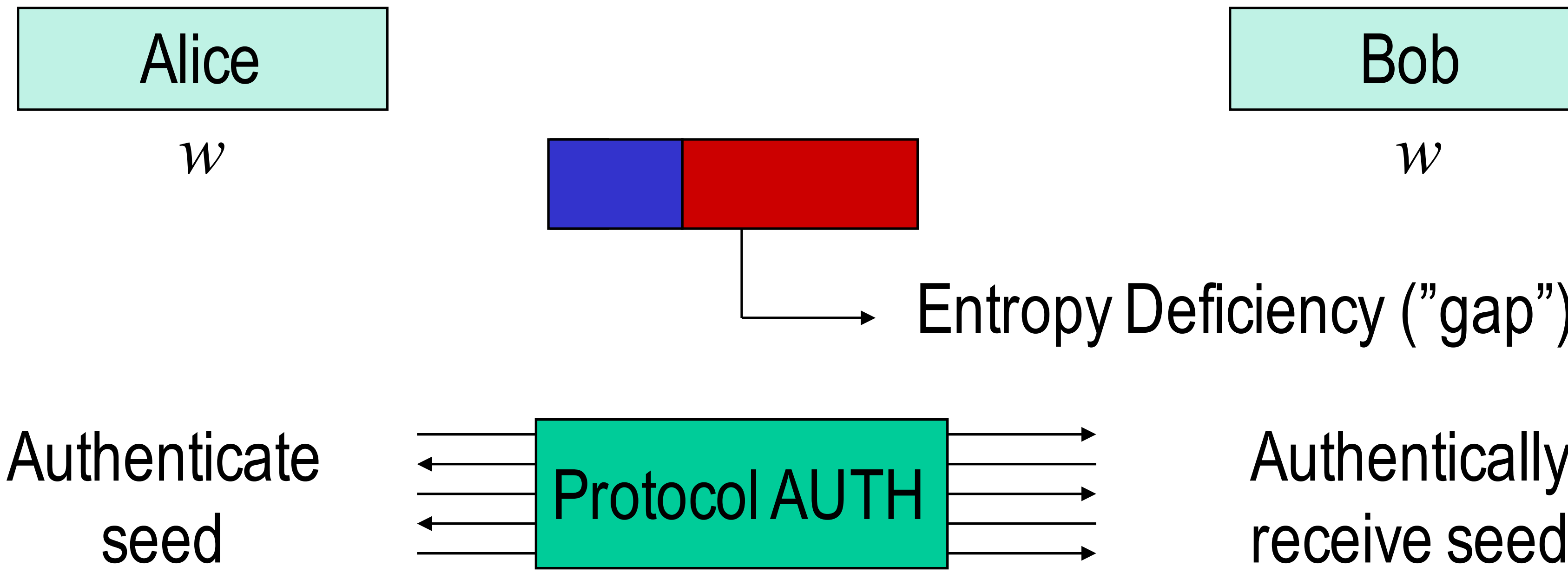
Privacy Amplification

Alice	Bob
w	w

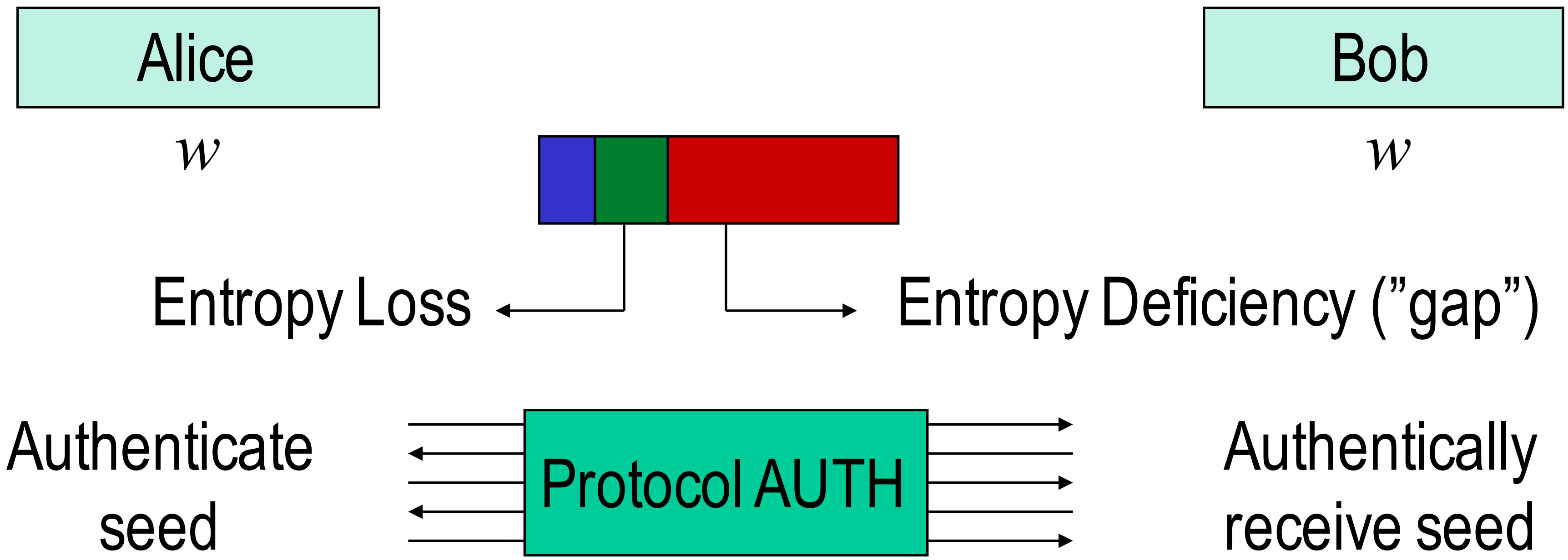


Entropy Deficiency (”gap”)

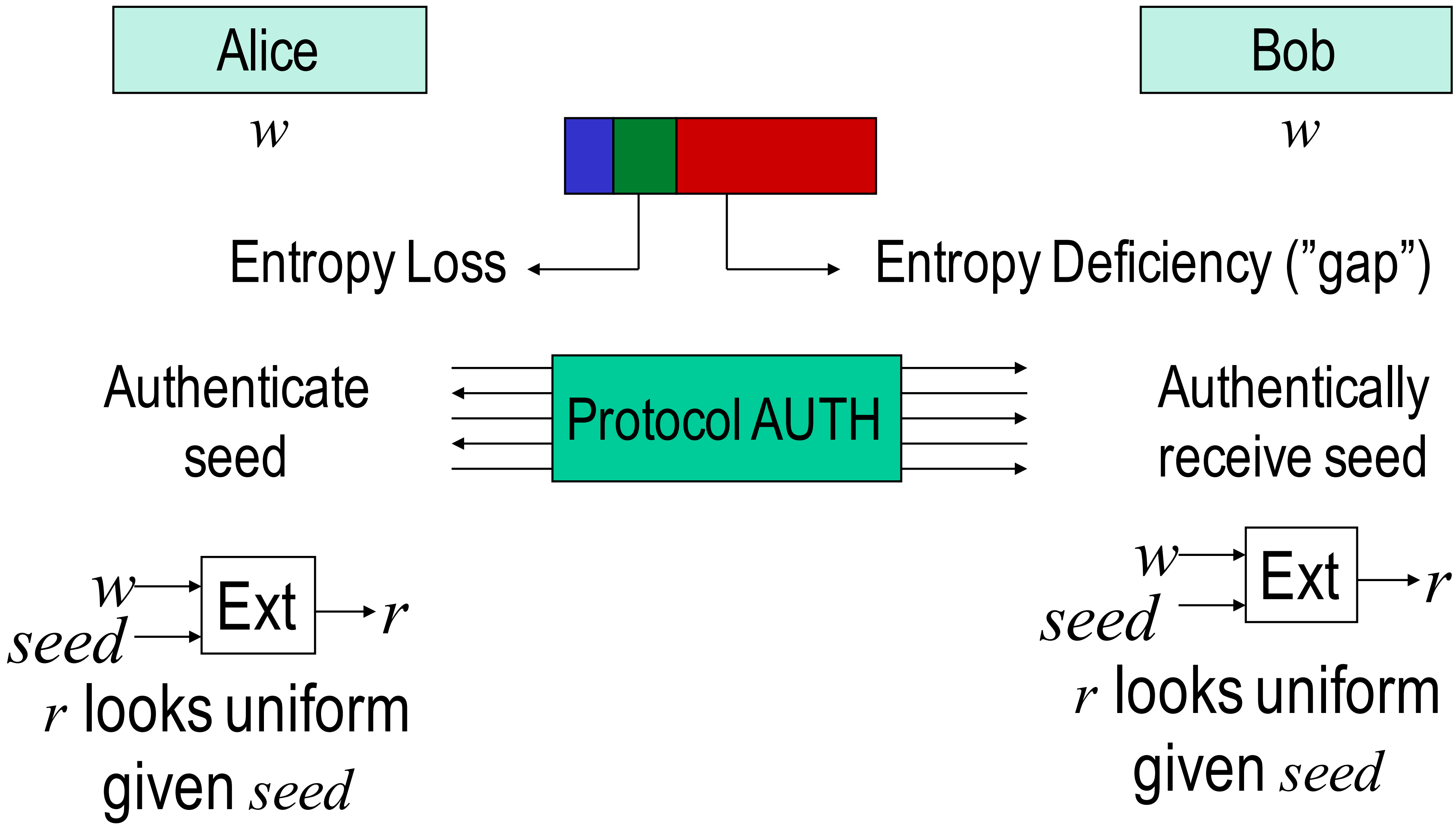
Privacy Amplification



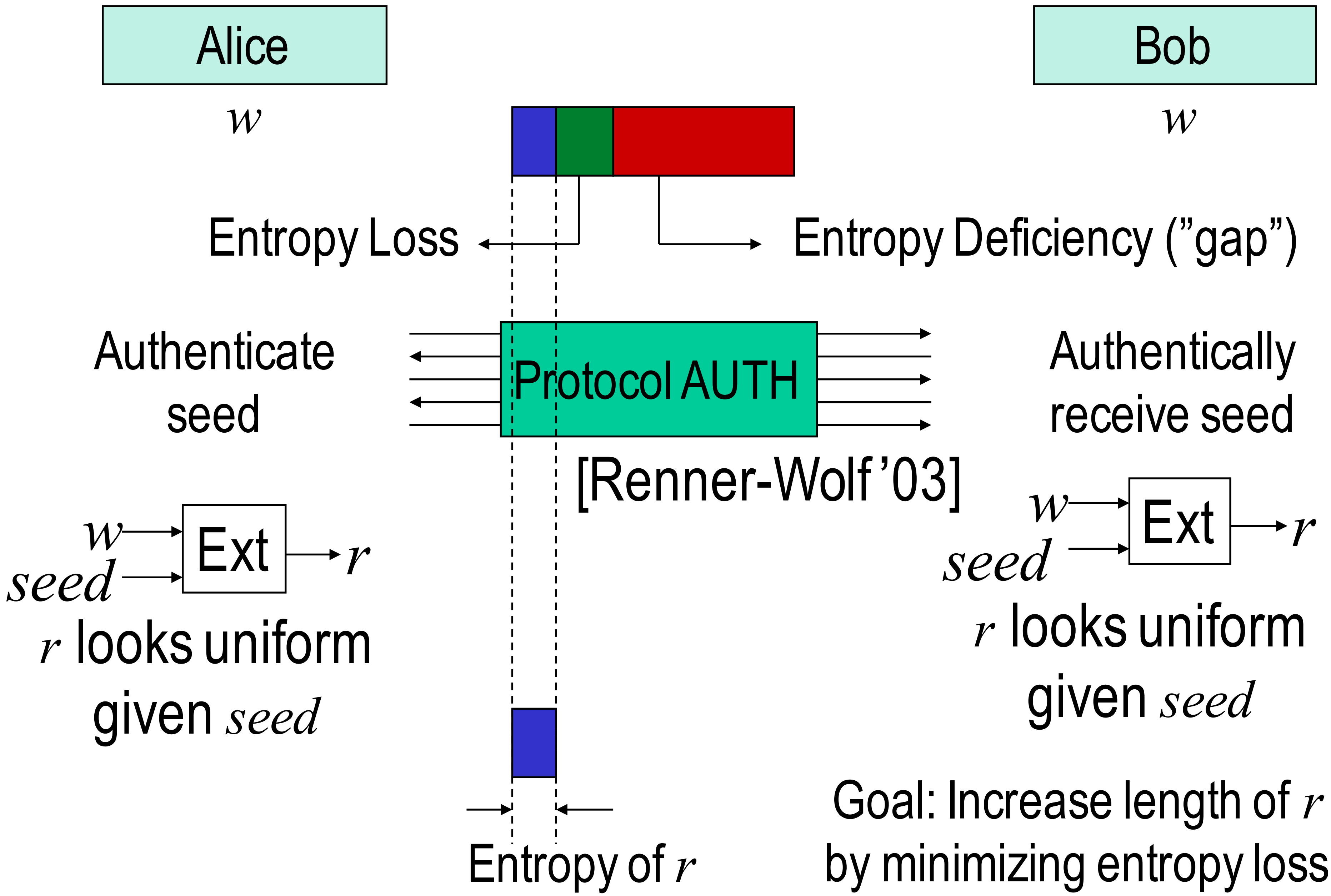
Privacy Amplification



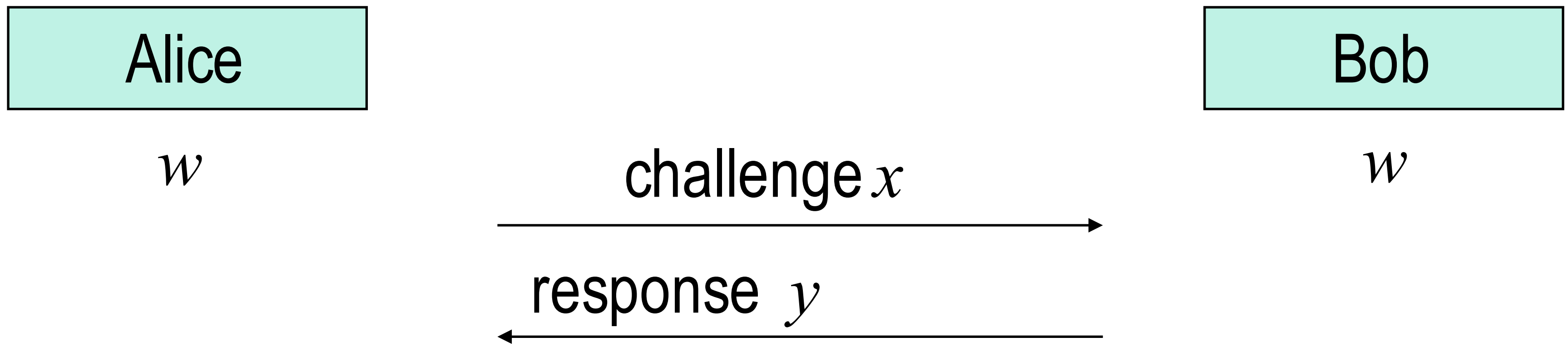
Privacy Amplification



Privacy Amplification



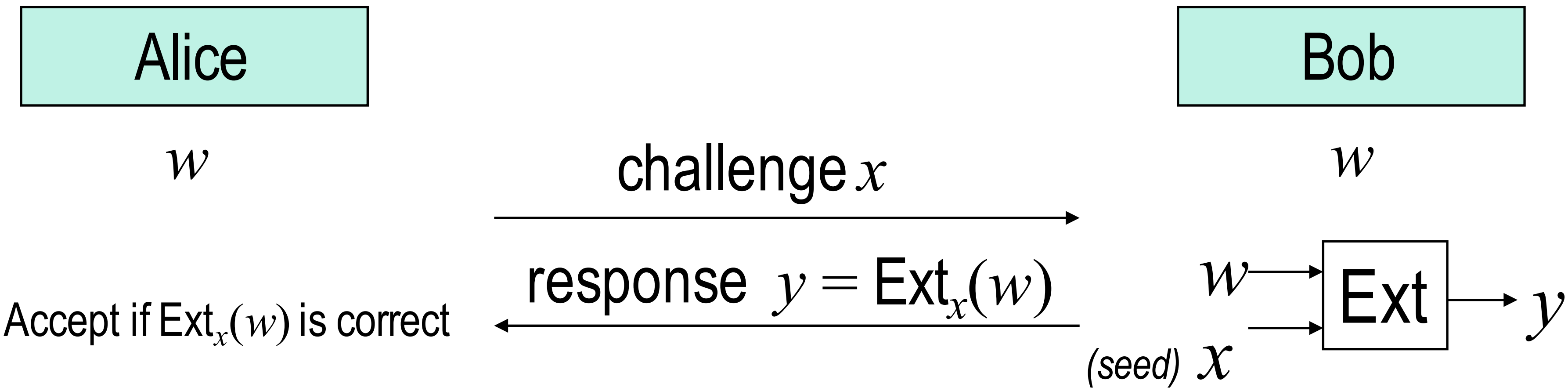
[RW03] Auth: Sub-Protocol Liveness Test



Want: If Alice accepts response, then Bob responded to *a challenge* and is, therefore, still "alive" in the protocol

Idea: "Response" should be such that Eve cannot compute it herself

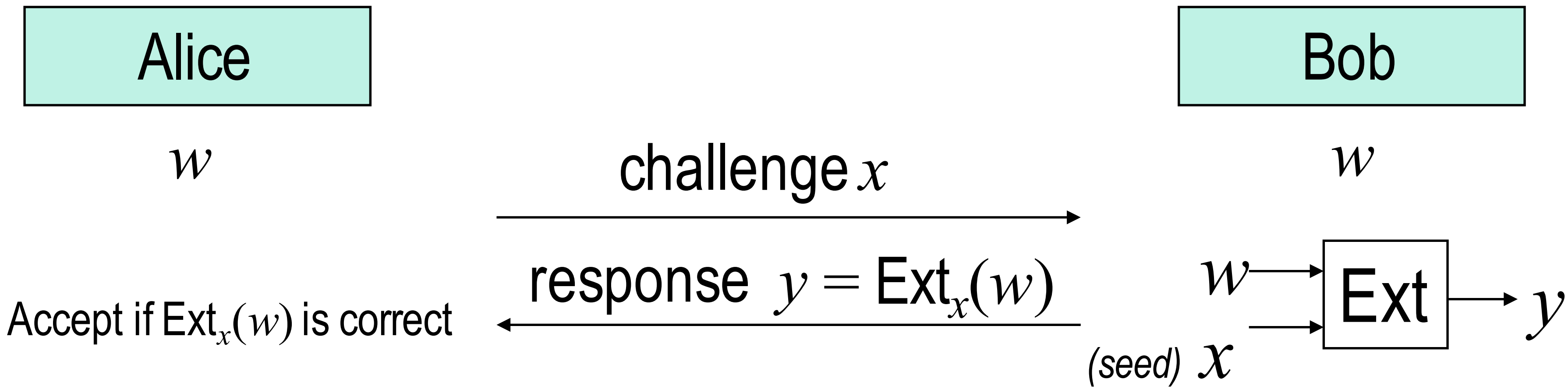
[RW03] Auth: Sub-Protocol Liveness Test



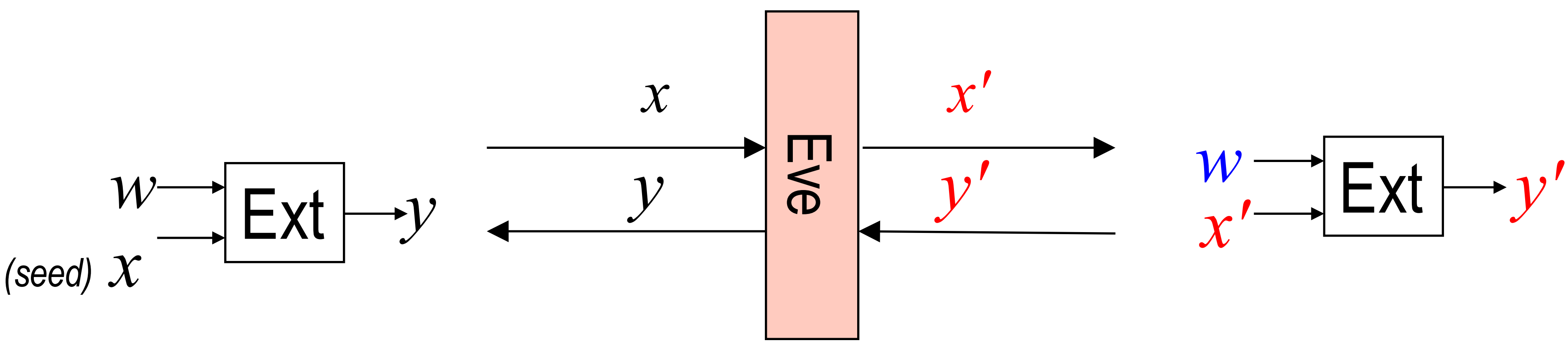
Want: If Alice accepts response, then Bob responded to *a challenge* and is, therefore, still "alive" in the protocol

Idea: "Response" should be such that Eve cannot compute it herself

[RW03] Auth: Sub-Protocol Liveness Test



Note: Active attack doesn't help Eve defeat liveness test



[RW03] Auth: *Sub-protocol ½ bit authentication*

Guarantees: if Bob receives bit $b = 1$,
then Alice sent $b = 1$

Alice	Eve	Bob
w		w
	$\xleftarrow{\quad x \quad}$	Generate random seed x
$y = \text{Ext}_x(w)$	$(1, y)$ or $(0, \perp)$ $\longrightarrow$	If $b = 1$, verify $y = \text{Ext}_x(w)$

bit-auth(b)

[RW03] Auth: *From ½ bit to string*

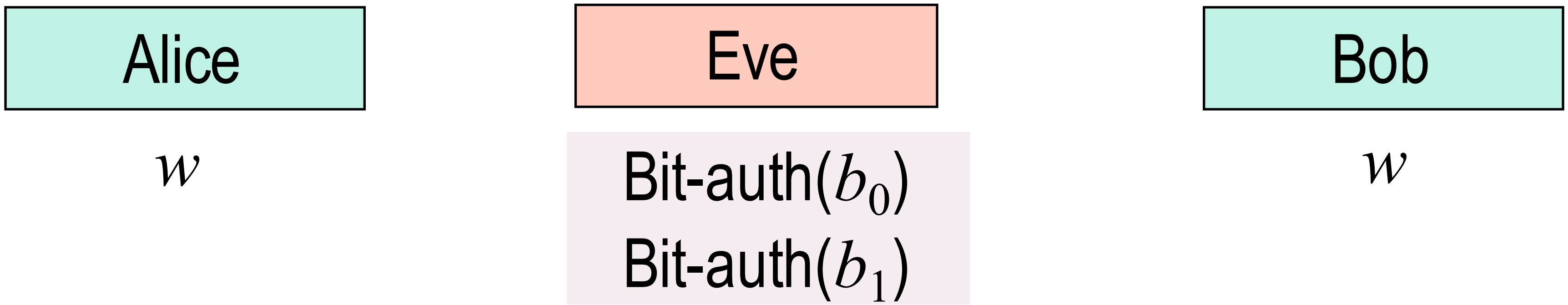
Guarantees: if Bob receives bit $b = 1$,
then Alice sent $b = 1$

Alice	Eve	Bob
w		w
	$\xleftarrow{\quad x \quad}$	Generate random seed x
$y = \text{Ext}_x(w)$	$\xrightarrow[(0,\ \perp)]{(1,\, y) \text{ or}}$	If $b = 1$, verify $y = \text{Ext}_x(w)$

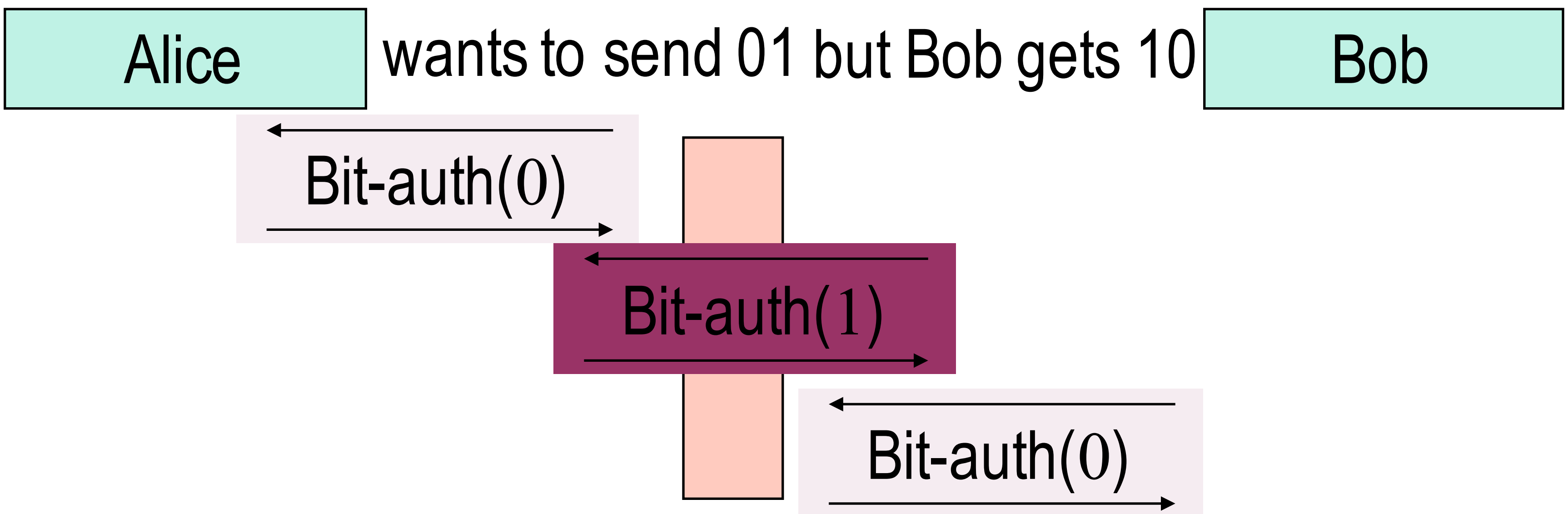
bit-auth(b)

- Problem: Eve can't change 0 to 1, but can change 1 to 0
- Solution: make the string balanced (#0s = #1s)

[RW03] Auth: From ½ bit to string

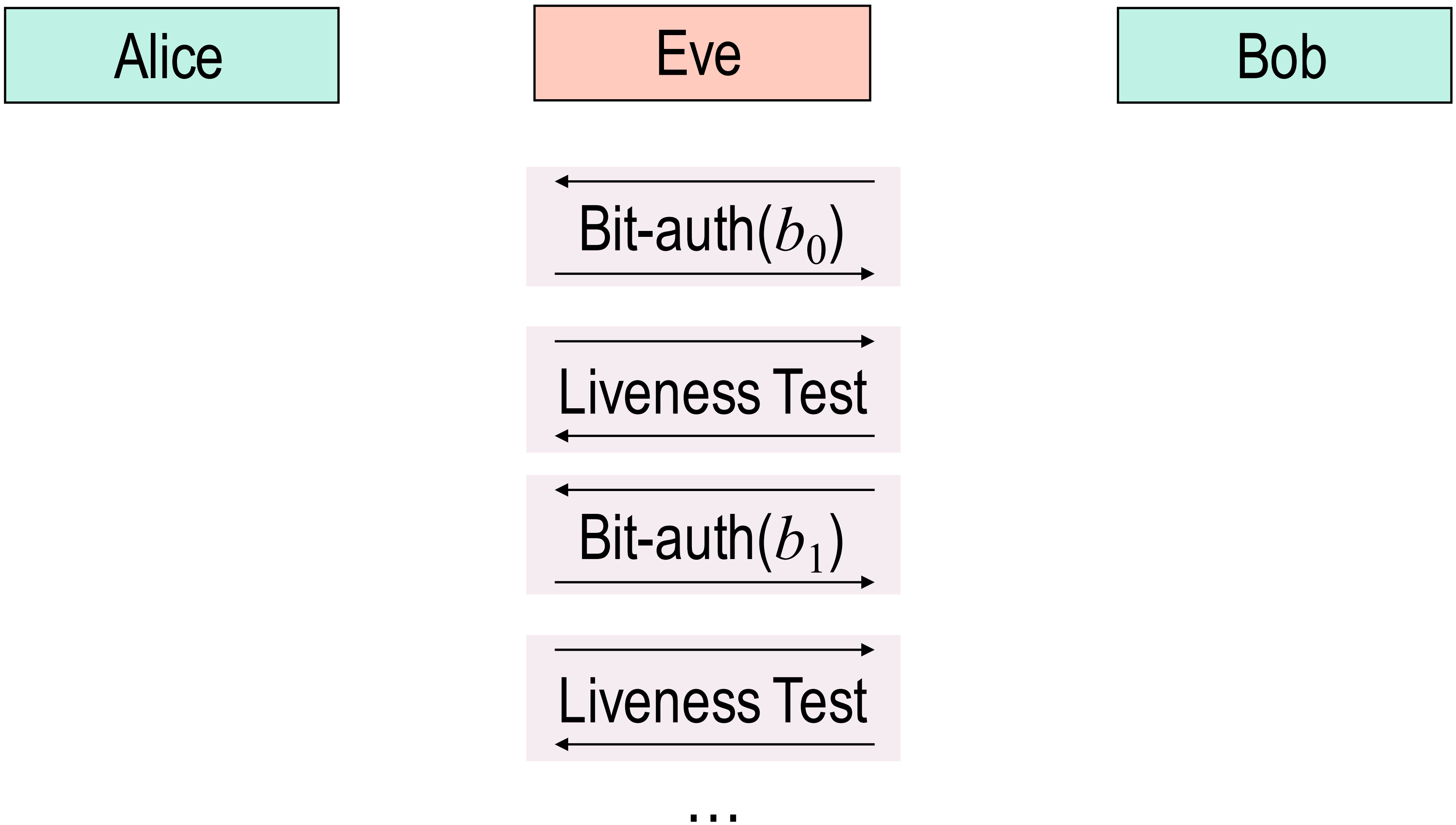


- Problem: Eve can delete any bit (and insert a 0 bit)



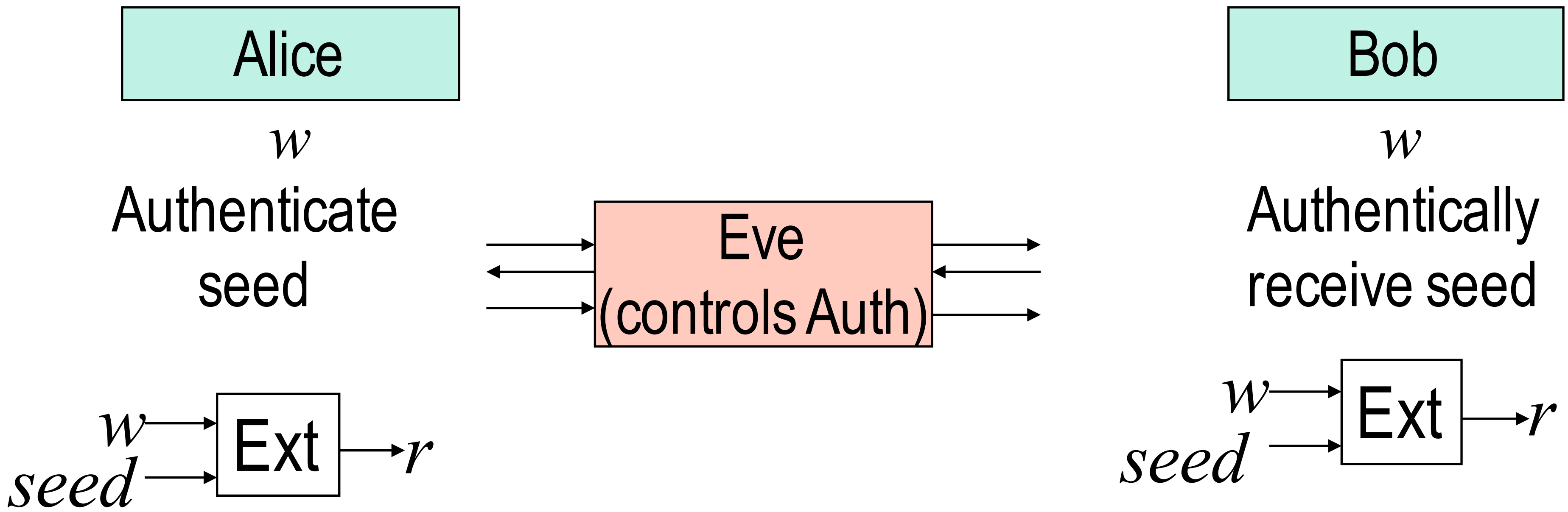
- Solution: add a liveness test after each bit to check that Bob got it

[RW03] Auth: From ½ bit to string



For $2^{-\delta}$-security, each Ext output needs ≈δ bits. Loss ≈ 1.5 |seed| δ

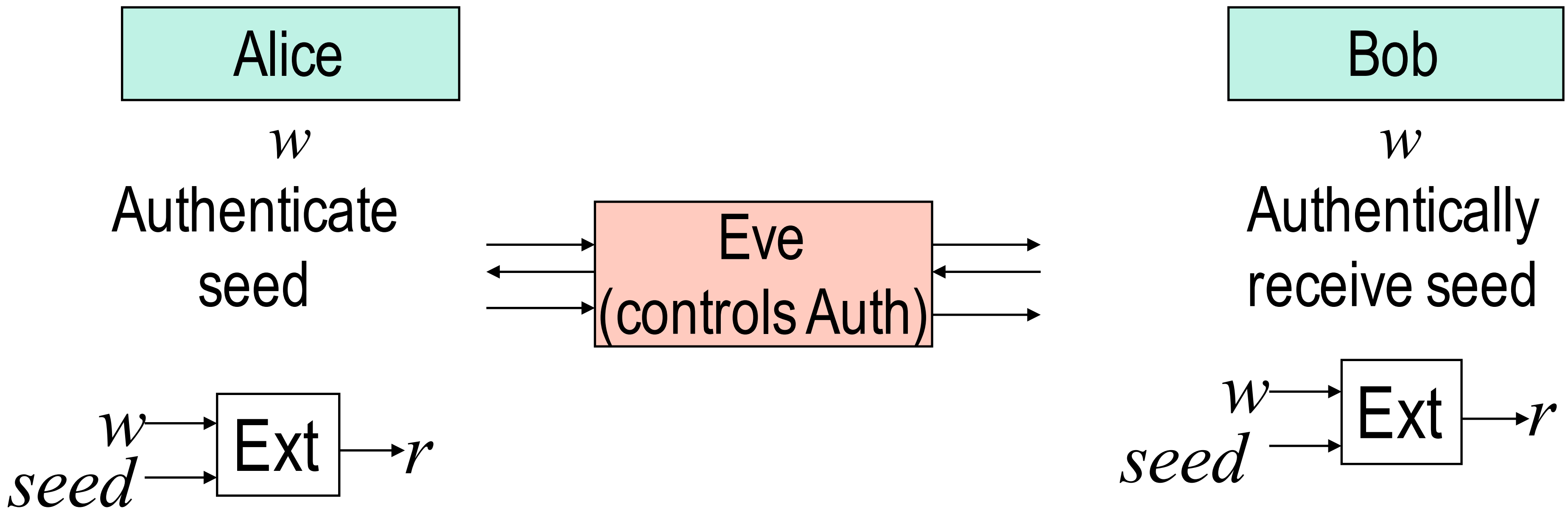
Privacy Amplification



- Does r look uniform given $seed$?

- Need: $seed$ independent of w
- Problem: Active Eve can play with AUTH to learn something correlated to $seed$ during AUTH
- Solution: If $|r| > 2|\text{Auth}|$, then r is >half entropic
- Use r as MAC key to authenticate the actual (fresh!) $seed'$

Privacy Amplification



- Total entropy loss (after some improvements from [Kanukurthi-Reyzin 2009]): about $\delta^2/2$
- Theoretical improvement to O(δ) in [Chandran-Kanukurthi-Ostrovsky-Reyzin 2014] (but for practical values of δ, constants make it worse than $\delta^2/2$)

Outline

- Passive adversaries
  - Privacy amplification
  - Fuzzy extractors
  - Information reconciliation
- Active adversaries, w has a lot of entropy
  - Message authentication codes
  - Privacy amplification only when $H_{\min}(w) > |w|/2$
  - Information reconciliation
  - Two security notions (pre-application vs. post-application)
- Active adversaries, w has little entropy
  - Privacy amplification
  - Information reconciliation

Information Reconciliation

Alice

w_0

$c = Sketch(w_0)$

Eve

Bob

w_1

recover w_0

Information Reconciliation

Alice

w_0

$c = \mathit{Sketch}(w_0)$

Eve

Bob

w_1

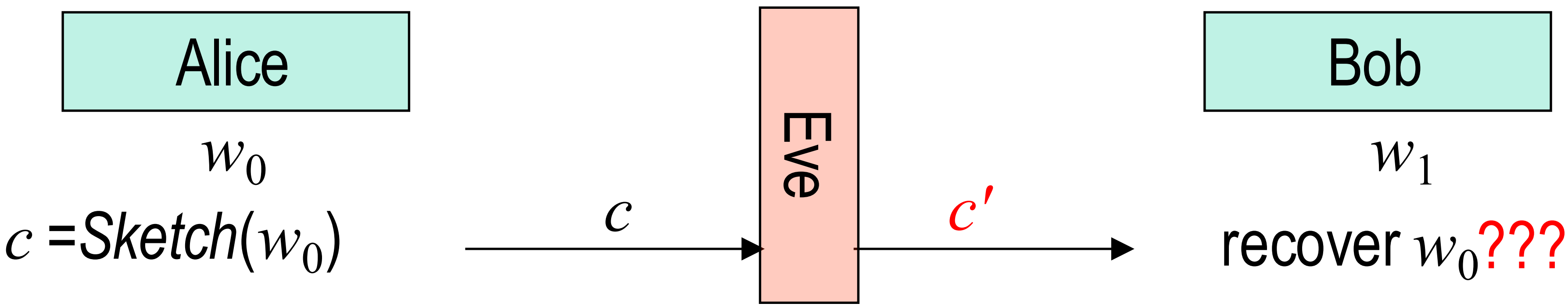
recover w_0

To verify, Bob needs to recover w_0 from w_1
so Alice needs to send c,

Authenticate message

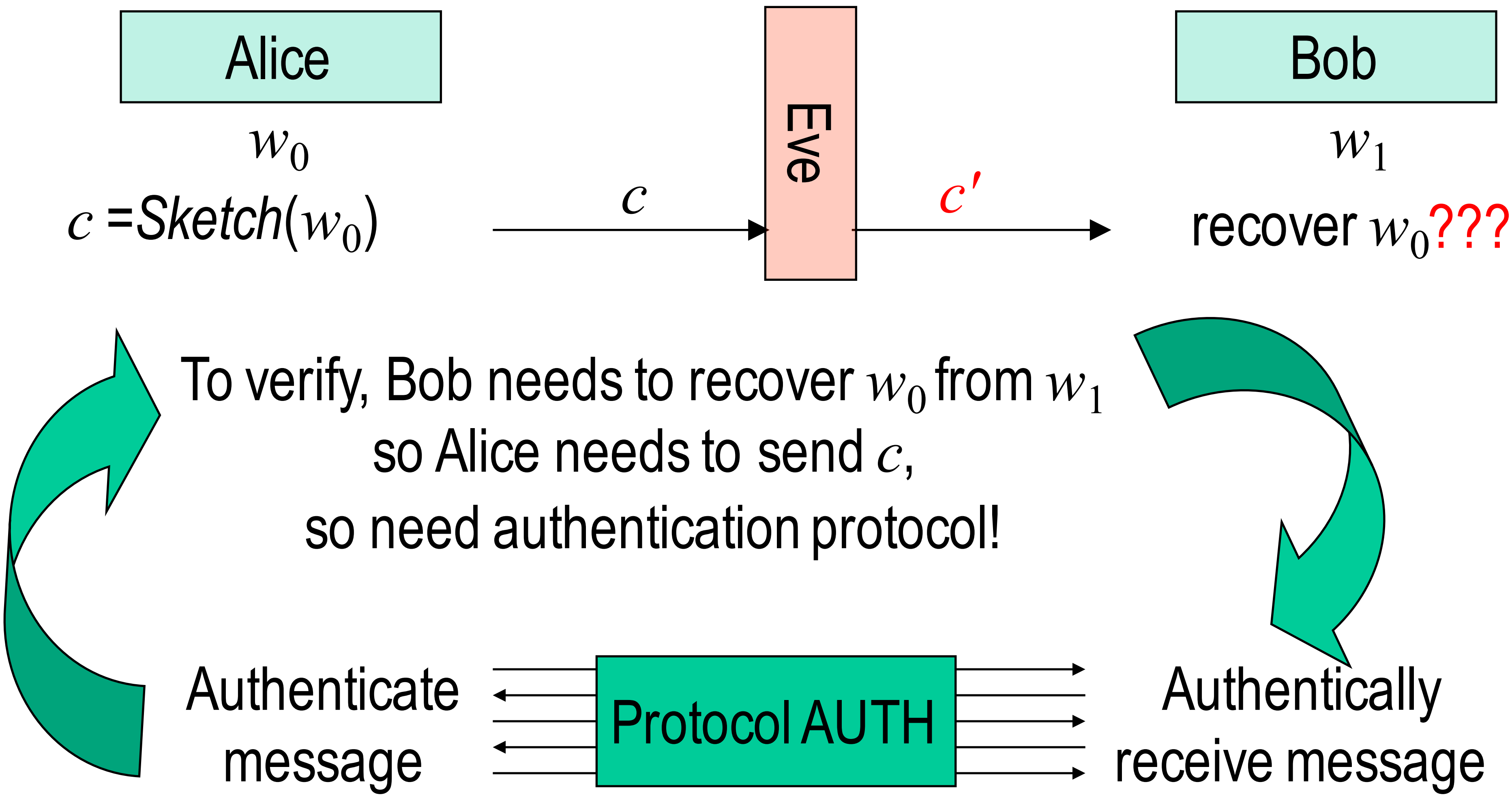
Authentically receive message

Information Reconciliation



To verify, Bob needs to recover w_0 from w_1
so Alice needs to send c,

Information Reconciliation



Attempt 1: *Error-Tolerant Authentication*

Alice		Bob
w_0		$w_1 \approx w_0$
c =*Sketch*(w_0)	$\xrightarrow{\quad c \quad}$	$w^* = \text{Rec}(w_1, \hat{\hat{c}})$
Authenticate c using w as key	Protocol AUTH	Authentically receive c using w^* as key

- Alice runs Auth using w_0 as key and Bob runs Auth using w^* key
- Auth Guarantees: For Eve to change even a single bit of the message authenticated, she needs to respond to an extractor query. (Either $\text{Ext}_x(w)$ or $\text{Ext}_x(w^*)$).
- If Bob runs protocol Auth on w^* (of high entropy, which Rec provides), Eve cannot change the message authenticated.

Attempt 1: *Error-Tolerant Authentication*

Alice		Bob
w_0		$w_1 \approx w_0$
$c = \mathit{Sketch}(w_0)$	$\xrightarrow{\quad c \quad}$	$w^* = \mathrm{Rec}(w_1, \hat{c})$
Authenticate c using w as key		Authentically receive c using w^* as key

Problem: Even if Eve's errors constitute a small fraction of w, Auth will lose more entropy than length of w

Attempt 2: Error-Tolerant Authentication

Alice		Bob
w_0		$w_1 \approx w_0$

Solution [Kanukurthi-Reyzin '09]: Reduce entropy loss using a MAC

- MAC needs a symmetric key κ
- Where does κ come from? Generate random κ and authenticate it

Attempt 2: *Error-Tolerant Authentication*

Alice

w_0

Bob

$w_1 \approx w_0$

Protocol AUTH(κ)

c, MAC$_\kappa(c)$ $\longrightarrow$

Solution [Kanukurthi-Reyzin '09]: Reduce entropy loss using a MAC

- MAC needs a symmetric key κ
- Where does κ come from? Generate random κ and authenticate it

Attempt 2: Error-Tolerant Authentication

Alice		Bob
w_0		$w_1 \approx w_0$

Protocol AUTH(κ)

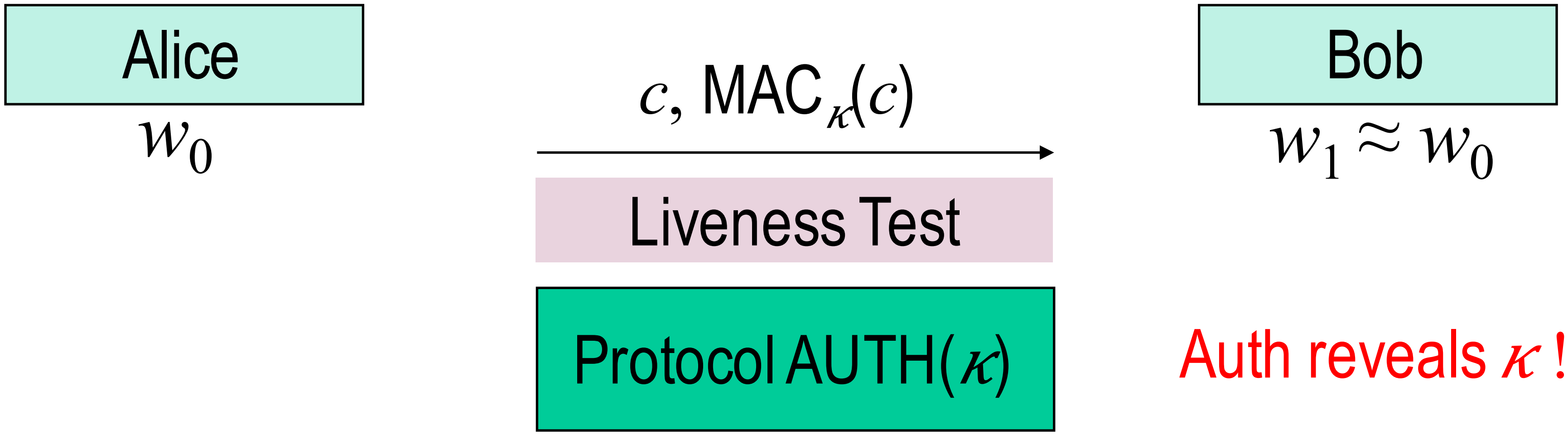
Auth reveals κ !

$$c,\ \mathrm{MAC}_{\kappa}(c) \longrightarrow$$

Solution [Kanukurthi-Reyzin '09]: Reduce entropy loss using a MAC

- MAC needs a symmetric key κ
- Where does κ come from? Generate random κ and authenticate it

Attempt 2: *Error-Tolerant Authentication*



Solution [Kanukurthi-Reyzin '09]: Reduce entropy loss using a MAC

- MAC needs a symmetric key κ
- Where does κ come from? Generate random κ and authenticate it

Attempt 2: Error-Tolerant Authentication

Alice
w_0

c, MAC$_\kappa(c)$ →

Liveness Test

Protocol AUTH(κ)

Bob
$w_1 \approx w_0$

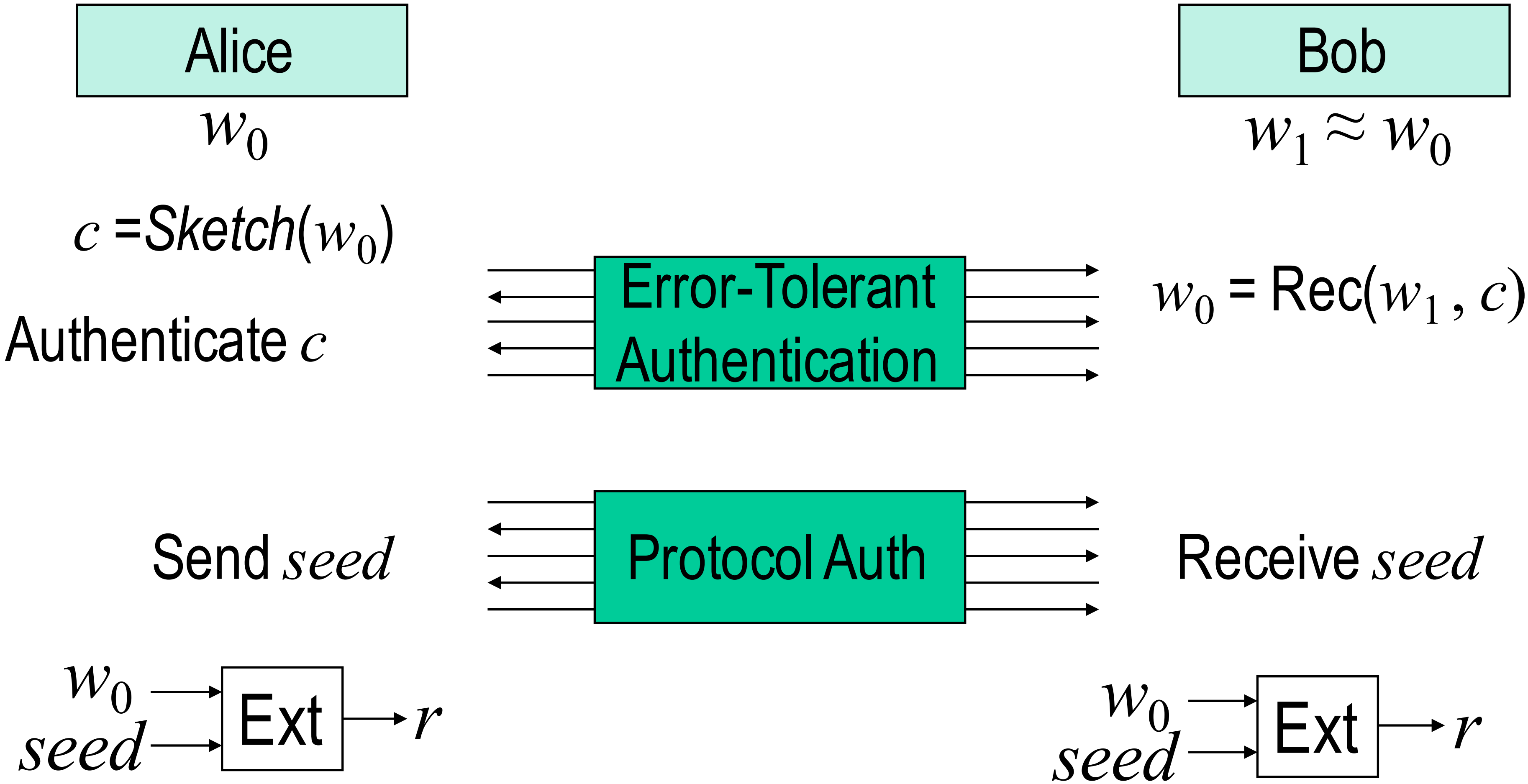
Auth reveals κ !

By the time Eve learns κ, it is too late for Eve to come up with forgery!

Solution [Kanukurthi-Reyzin '09]: Reduce entropy loss using a MAC

- MAC needs a symmetric key κ
- Where does κ come from? Generate random κ and authenticate it

information-reconciliation + privacy amplification



Use r as a MAC key to send the real extractor seed

information-reconciliation + privacy amplification

Alice

w_0

$c = \mathit{Sketch}(w_0)$

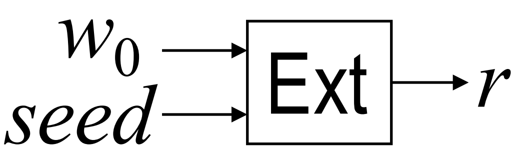
Authenticate c

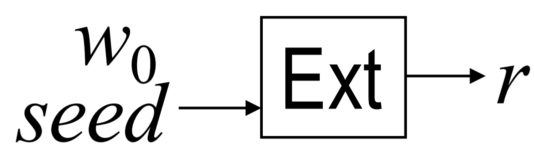
and seed

Bob

$w_1 \approx w_0$

$w_0 = \mathrm{Rec}(w_1, c)$





Use r as a MAC key to send the real extractor seed

Outline

- Passive adversaries
  - Privacy amplification
  - Fuzzy extractors
  - Information reconciliation
- Active adversaries, w has a lot of entropy
  - Message authentication codes
  - Privacy amplification only when $H_{\min}(w) > |w|/2$
  - Information reconciliation
  - Two security notions (pre-application vs. post-application)
- Active adversaries, w has little entropy
  - Privacy amplification
  - Information reconciliation