Wyner's Wire-Tap Channel, Forty Years Later

Leonid Reyzin

BOSTON UNIVERSITY

These slides are a superset of the talks given at:

- Theory of Cryptography Conference on March 24, 2015 (parts I and II)
- École Normale Supérieure Crypto Seminar on March 26, 2015 (parts I and II)
- Charles River Crypto Day on April 17, 2015 (parts I and most of III)

Part I

History and Context

Two Events in October 1975 Enabled This Talk

1. I was born
2. Aaron D. Wyner published “The Wire-Tap Channel”

Aaron Wyner c. 1975
(courtesy of Adi Wyner)

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 54 | October 1975 | Number 8

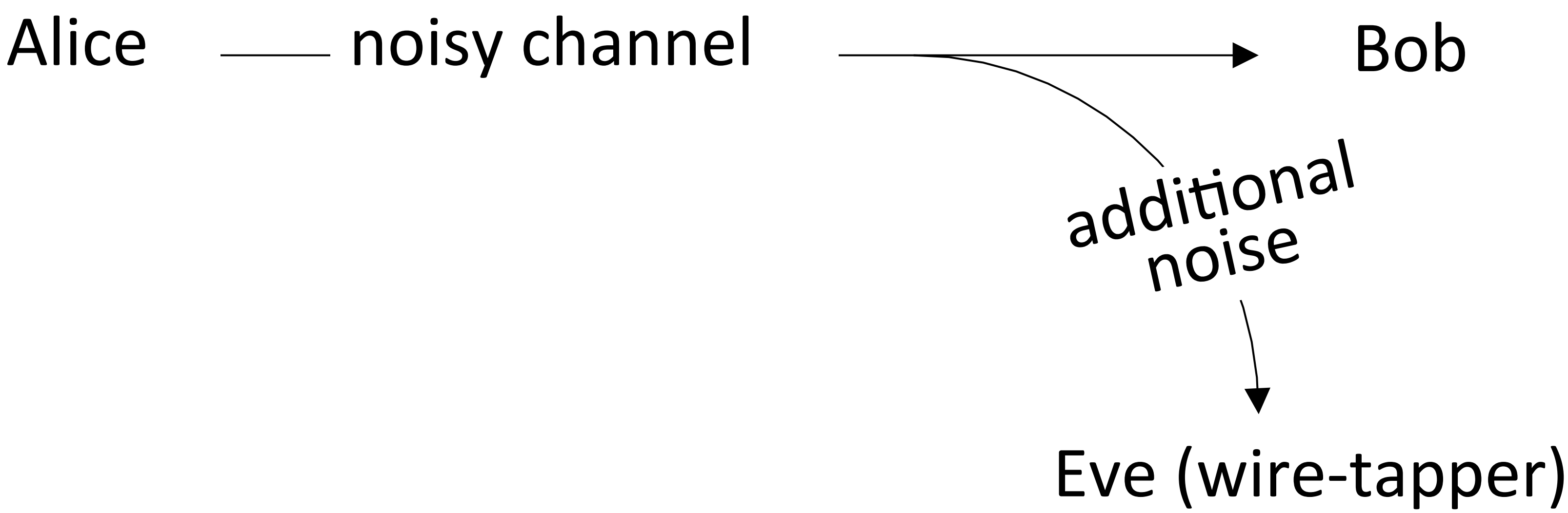
The Wire-Tap Channel

By A. D. WYNER

(Manuscript received May 9, 1975)

We consider the situation in which digital data is to be reliably transmitted over a discrete, memoryless channel (DMC) that is subjected to a wire-tap at the receiver. We assume that the wire-tapper views the channel output via a second DMC. Encoding by the transmitter and decoding by the the

Premise of "The Wire-Tap Channel"

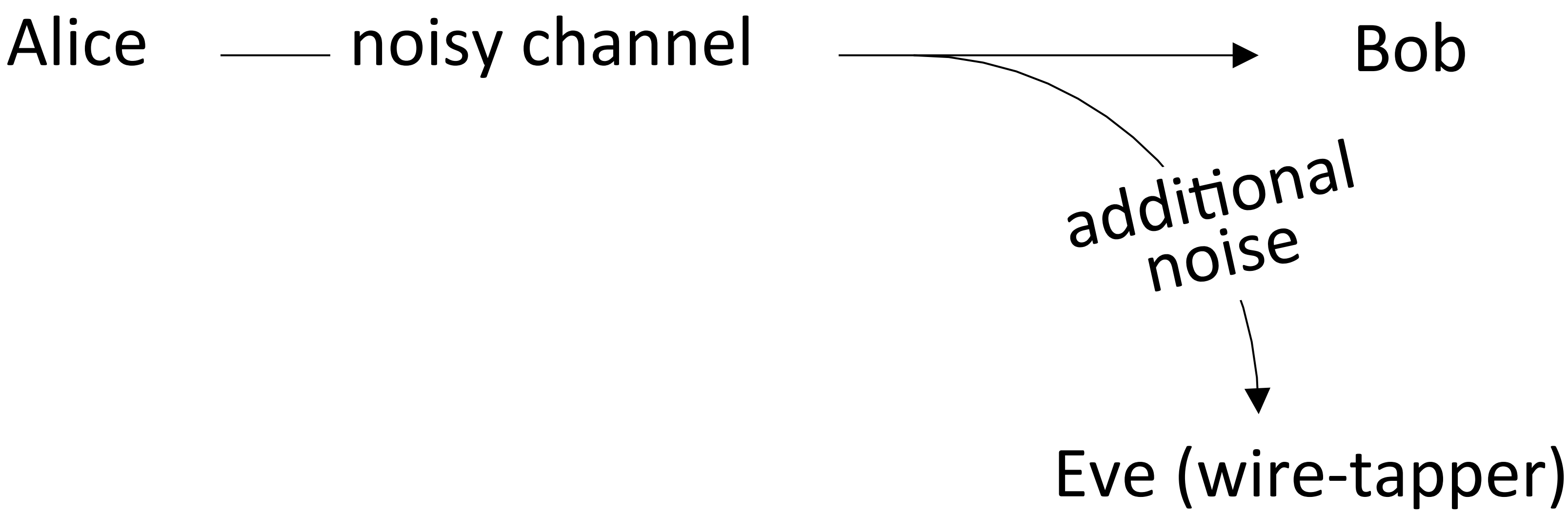


Goal: transfer information from Alice to Bob reliably while hiding it from Eve

Results: Upper/lower bounds on information rate ("secrecy capacity")

Constructive for a special case (in a few slides)

Context: mid-1970s surge of interest in crypto

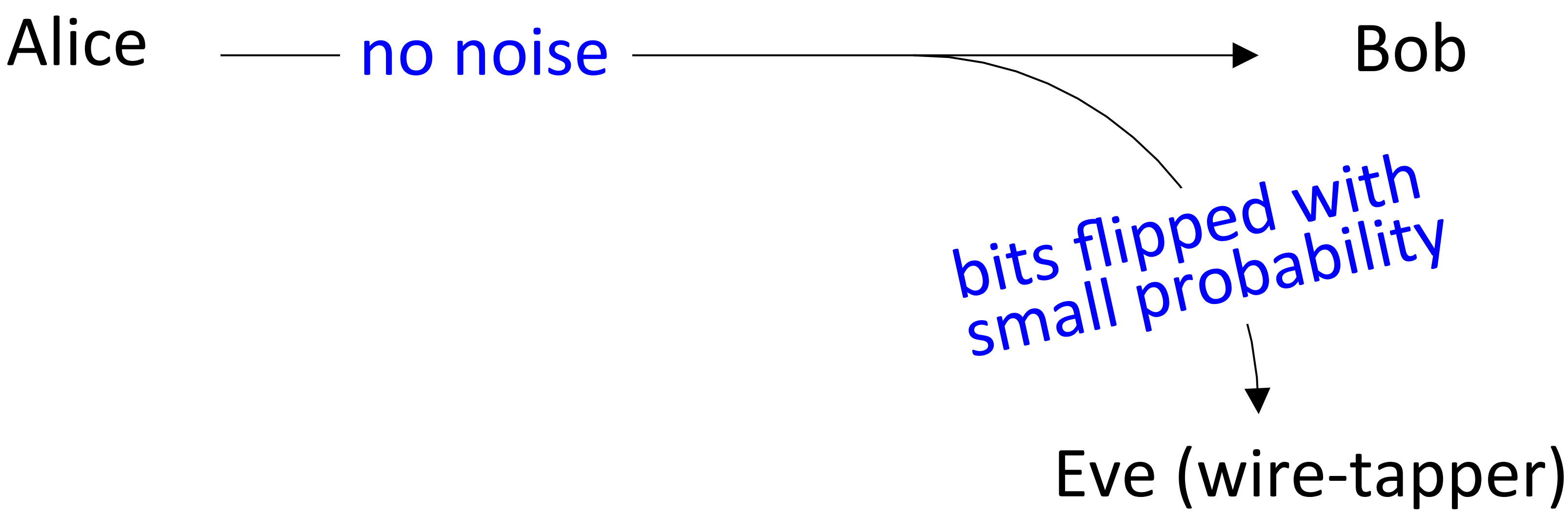


<u>Q: Where Does Secrecy Come From?</u>

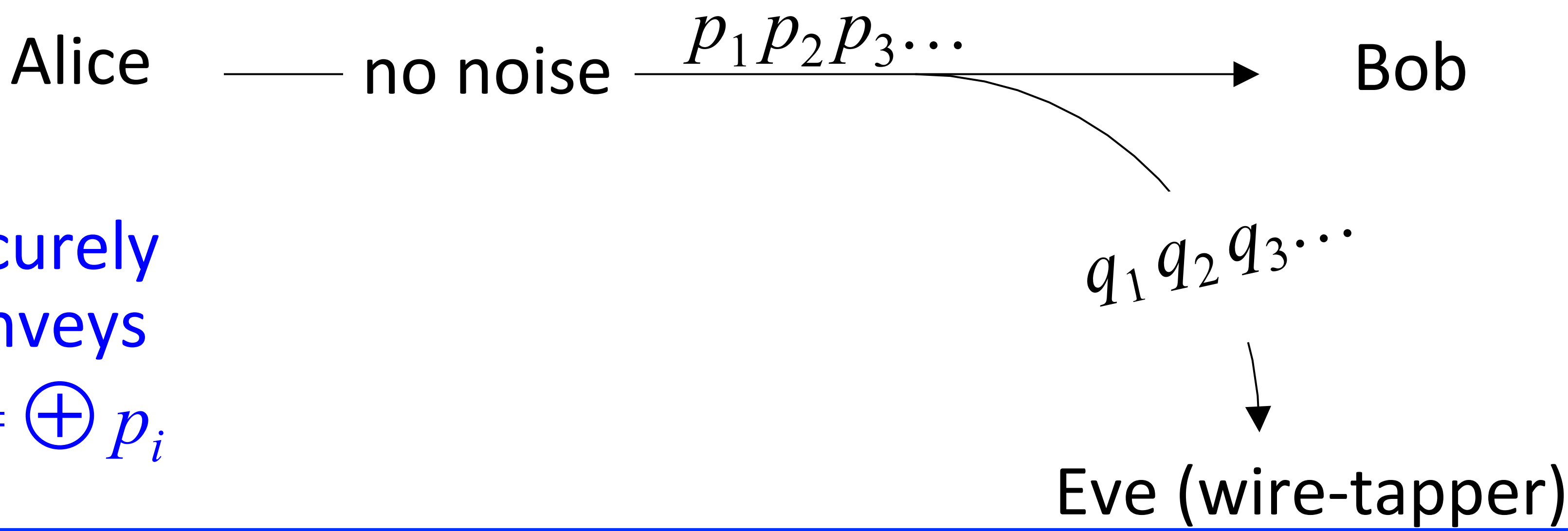
1. Public Keys [Diffie-Hellman 1976, Merkle 1978]
2. Secret Keys [Shannon 1949] … [Hellman 1974-77]
3. Nature [Wyner 1975]

   Only 5 references: 3 for basic info-theory background +

Simple Special Case



Simple Special Case



Securely conveys $b = \bigoplus p_i$

Wyner: XOR amplifies information-theoretic uncertainty

- Alice sends a string $p = p_1 p_2 p_3 \ldots$
- Eve will see $q_i = p_i \oplus \text{noise}$;
  most, but not all, $q_i = p_i$
- Given enough bits, parity of p looks uniform

Yao 1982: "Theory and Applications of Trapdoor Functions"

PK Alice —— $f(x_1), f(x_2), \ldots$ ——→ Bob SK

Perfect knowledge of $f(x_i)$, but "noisy" knowledge of $P(x_i)$

→ Eve (noiseless but bounded wire-tapper)

Securely conveys $b = \bigoplus P(x_i)$

Yao: XOR amplifies computational uncertainty

<u>Setup</u>: Weak TDP $f(x)$ with somewhat hardcore predicate $P(x)$
Alice has PK and Bob has SK

Yao 1982: "Theory and Applications of Trapdoor Functions"

PK Alice —— $f(x_1), f(x_2), \ldots$ ——→ Bob SK

Perfect knowledge of $f(x_i)$, but "noisy" knowledge of $P(x_i)$

↓

Eve (noiseless but bounded wire-tapper)

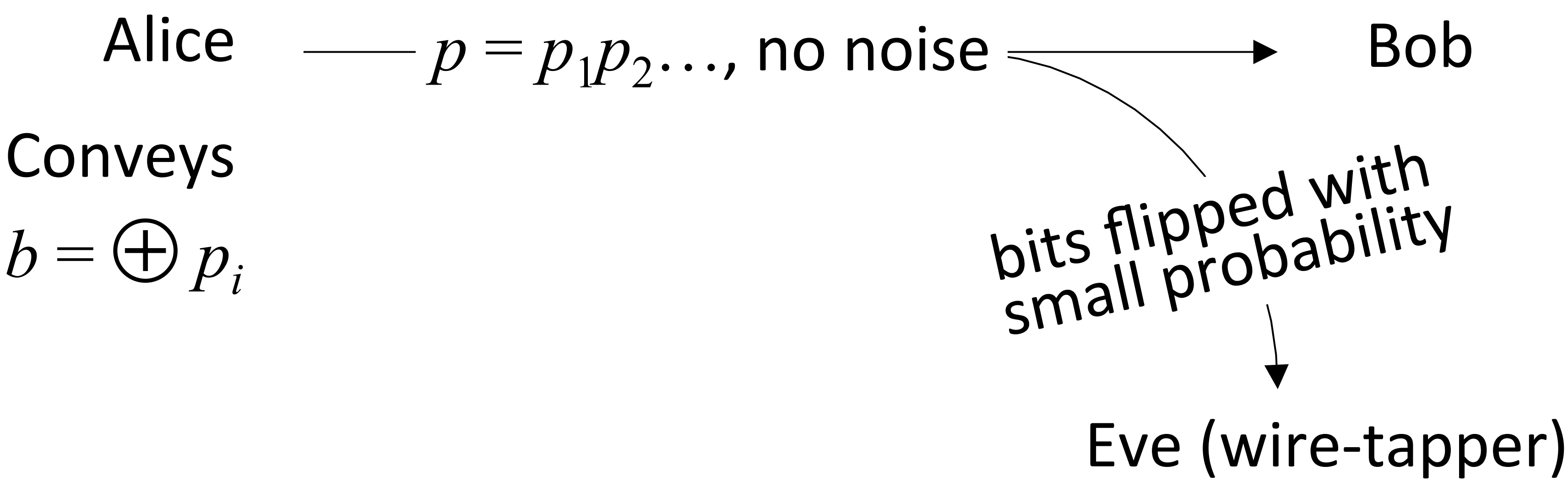
Securely conveys $b = \bigoplus P(x_i)$

"this situation has an exact analogue
in the classical information theory,
known as the Wyner wiretap channel [25].
Wyner showed that even when
the noise in [Eve's] channel is small,
[Alice] can magnify the noise
by properly encoding [her] messages."

Levin 1985,87
"One-Way Functions and Pseudorandom Generators"
(proof of Yao's XOR Lemma)

"One of the important ideas of [Yao 82] is
that the methods of [Wyner 75]
can be applied for computational
as well as for purely probabilistic unpredictability."

Back to Wyner



Consider the conditional distribution $p \mid \text{Eve's knowledge}$

($= U \mid U \oplus$ binary symmetric noise)

Wyner's observation (reformulated):

parity is a deterministic extractor from this distribution

Q: Can we do better? (i.e., extract more bits = increase rate)

A [Wyner]: Yes!

Back to Wyner

Alice —— $p = p_1 p_2 \ldots$, no noise ——→ Bob

bits flipped with small probability

↓

Eve (wire-tapper)

Conveys
$s = Hp$
(H = parity-check matrix for a linear error-correcting code)

Consider the conditional distribution $p \mid \text{Eve's knowledge}$
($= U \mid U \oplus$ binary symmetric noise)

Wyner's observation (reformulated):
parity is a deterministic extractor from this distribution

Q: Can we do better? (i.e., extract more bits = increase rate)

A [Wyner]: Yes!

"the coding scheme [above] is based on an idea of Mr. [Colin] Mallows"

Santha-Vazirani 1986

"Generating quasi-random sequences from semi-random sources"

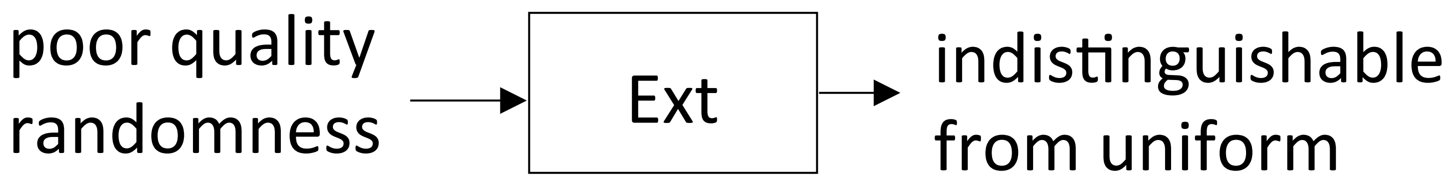
"Wyner shows how to achieve
optimal rate of communication,
using parity-check codes.
We show how to use the same method
to extract quasi-random sequences at a higher rate."

Thm: Hp is an extractor from any distribution where each bit has pre-selected bounded bias

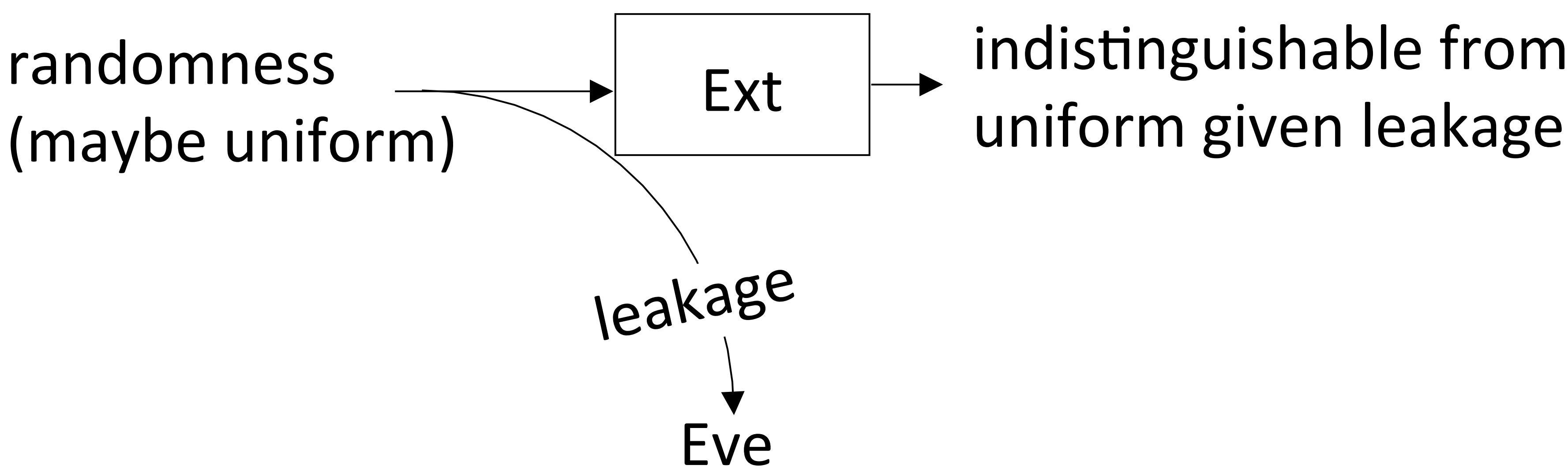
(under stronger demands on quality of output than Wyner)

Note the two views of extractors

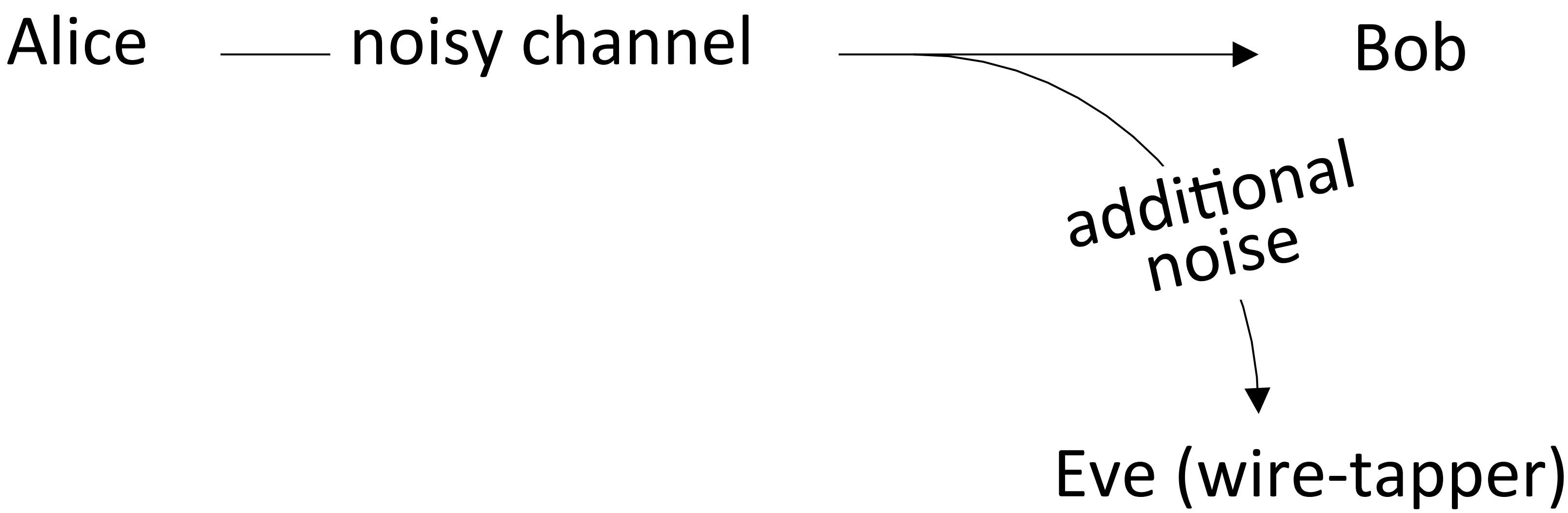
[Santha-Vazirani]:



[Wyner]:



Summary of Wyner's paper

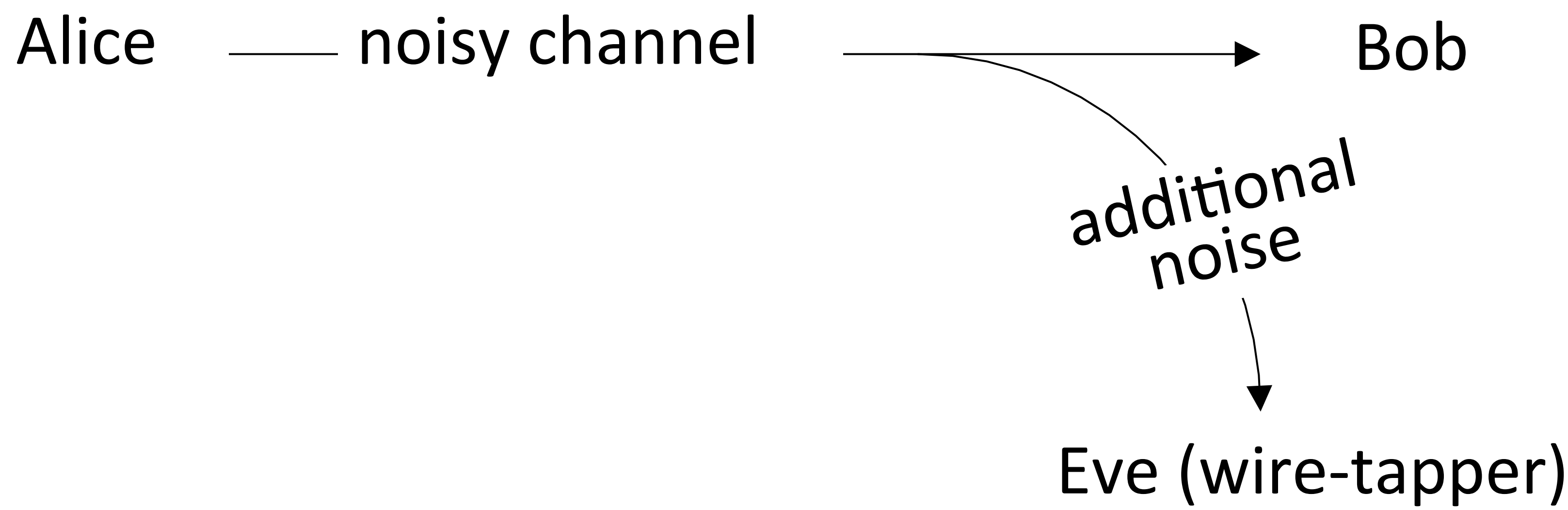


<u>Goal</u>: transfer information from Alice to Bob reliably while hiding it from Eve

<u>Results</u>:
- Derive best achievable "secrecy capacity"
- Achieve it constructively for the case above (assuming good linear codes)
- Achieve it nonconstructively in all other cases

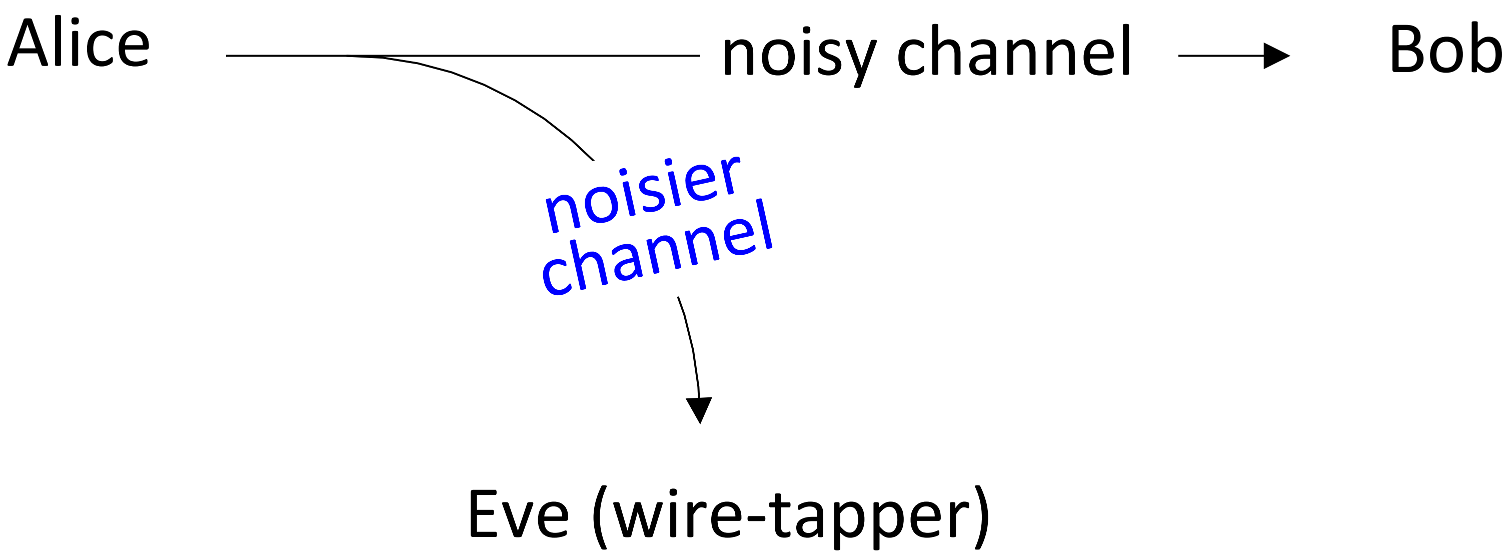
<u>Drawback</u>: Weak notion of security ("low rate of leakage")

Lessons



- There is interesting work to do in provable information-theoretic security
- Noise can be your friend
- Secrets can come from nature

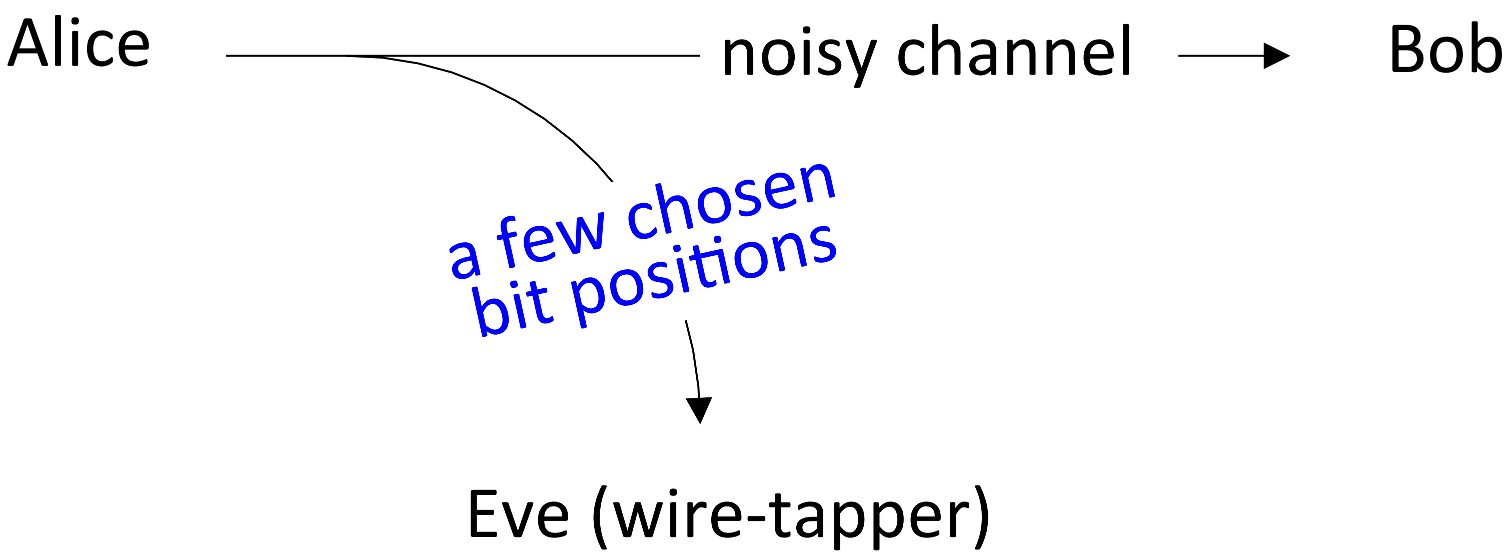
Early Generalization



[Csiszár-Körner 1978]:

no reason Eve's channel should be a degraded version of Bob's: it just needs to be worse

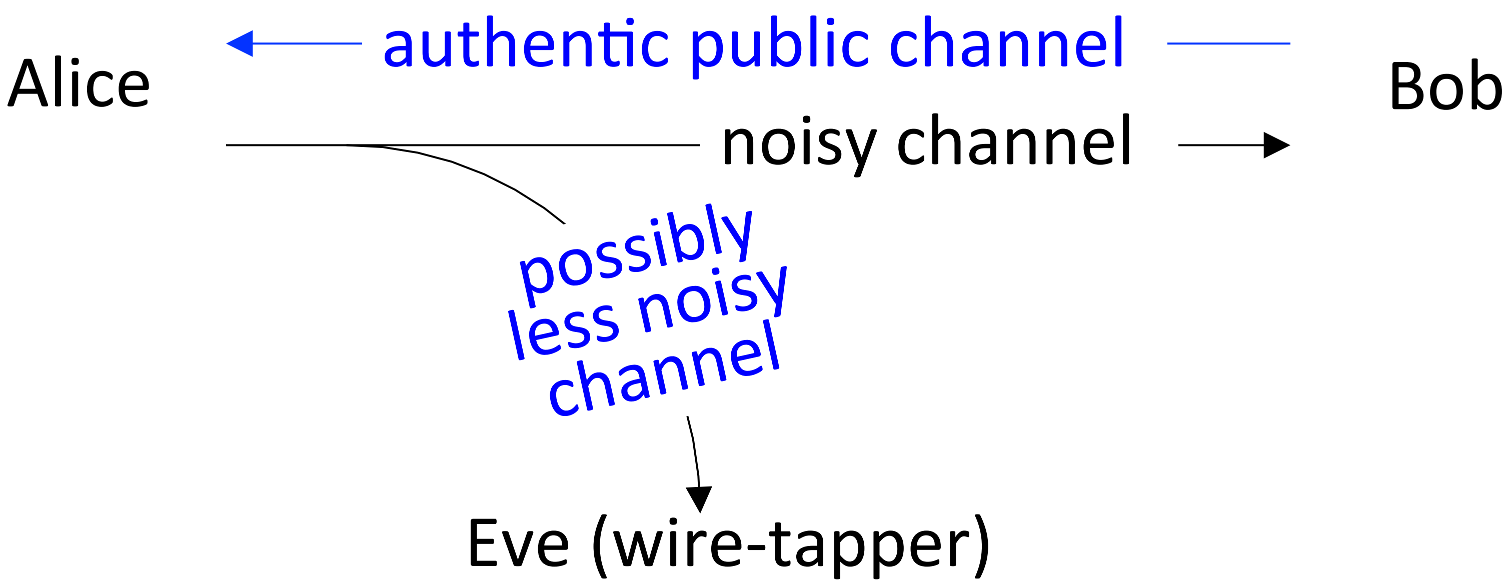
A Different Model for Eve's Knowledge



[Ozarow-Wyner 1985]: wire-tapper gets to choose specific symbols to see

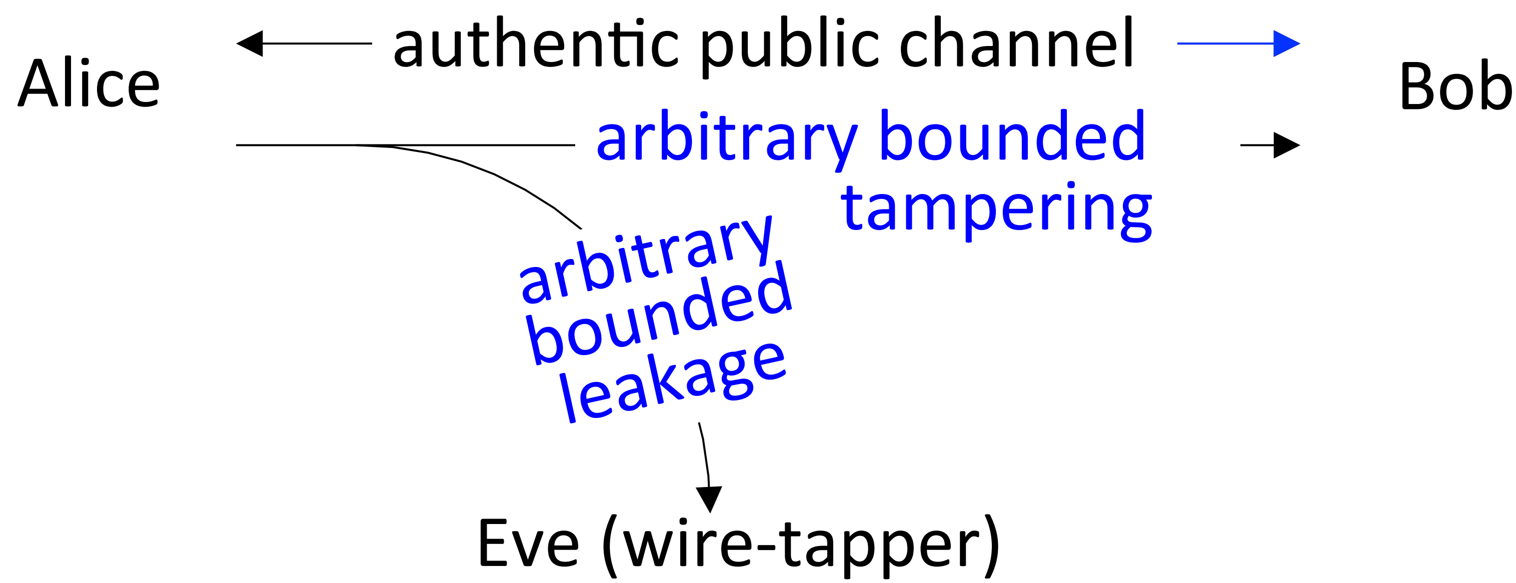
Result: Hp is still a good deterministic extractor

Adding a Feedback Channel



[Leung-Yan-Cheong 1976] (Ph.D. Thesis under Hellman), [Maurer 1993], [Ahlswede-Csiszár 1993]:
Eve’s channel need not be noisier if Bob has a feedback channel

Generalizing



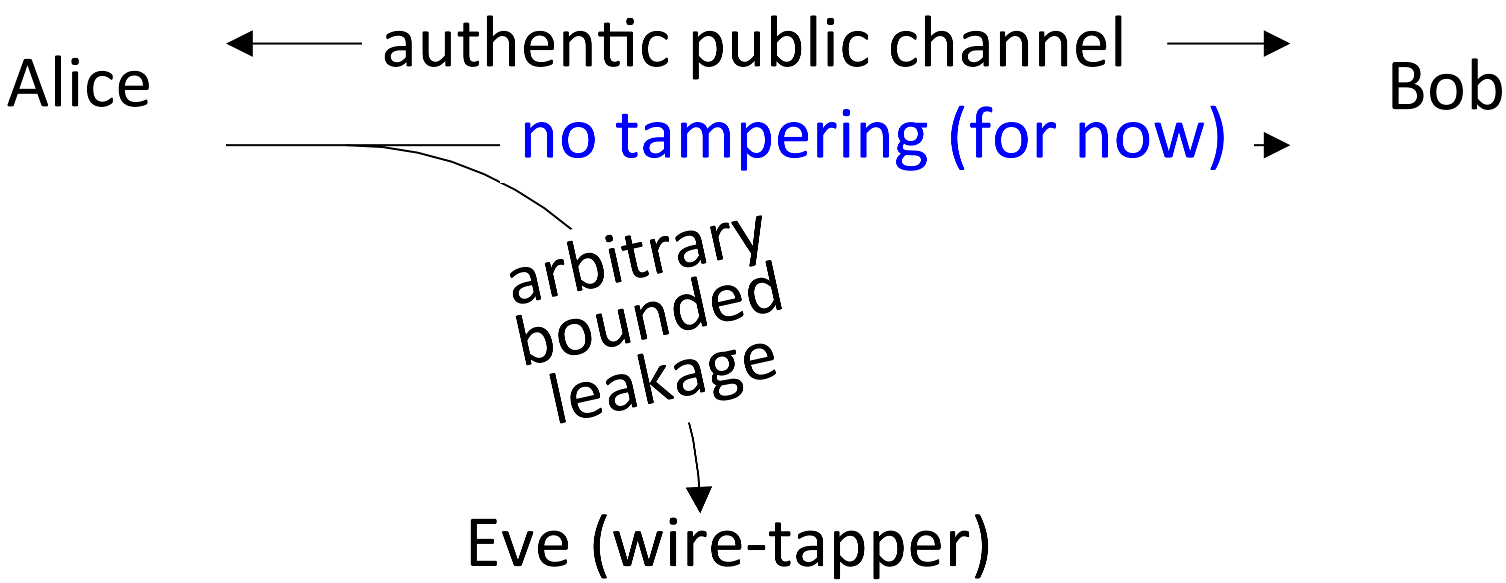
- [Bennett,Brassard,Robert 1985,88] (motivated by quantum):
  - errors can be adversarial
  - leakage can be arbitrary

information reconciliation

privacy amplification

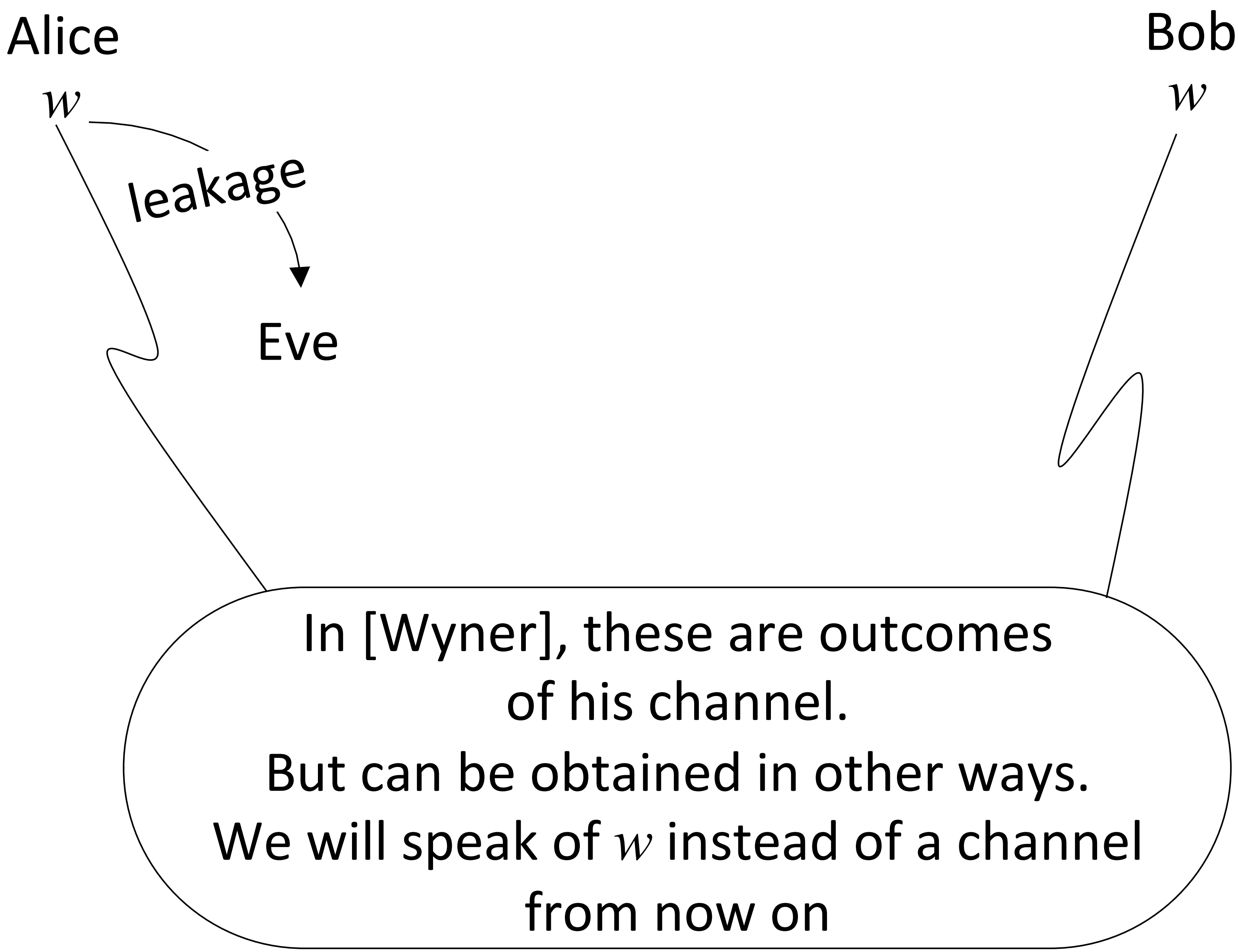
- Better security notion
- Generalized further, better notions of entropy in [Bennett,Brassard,Crépeau,Maurer 1995]

Privacy Amplification

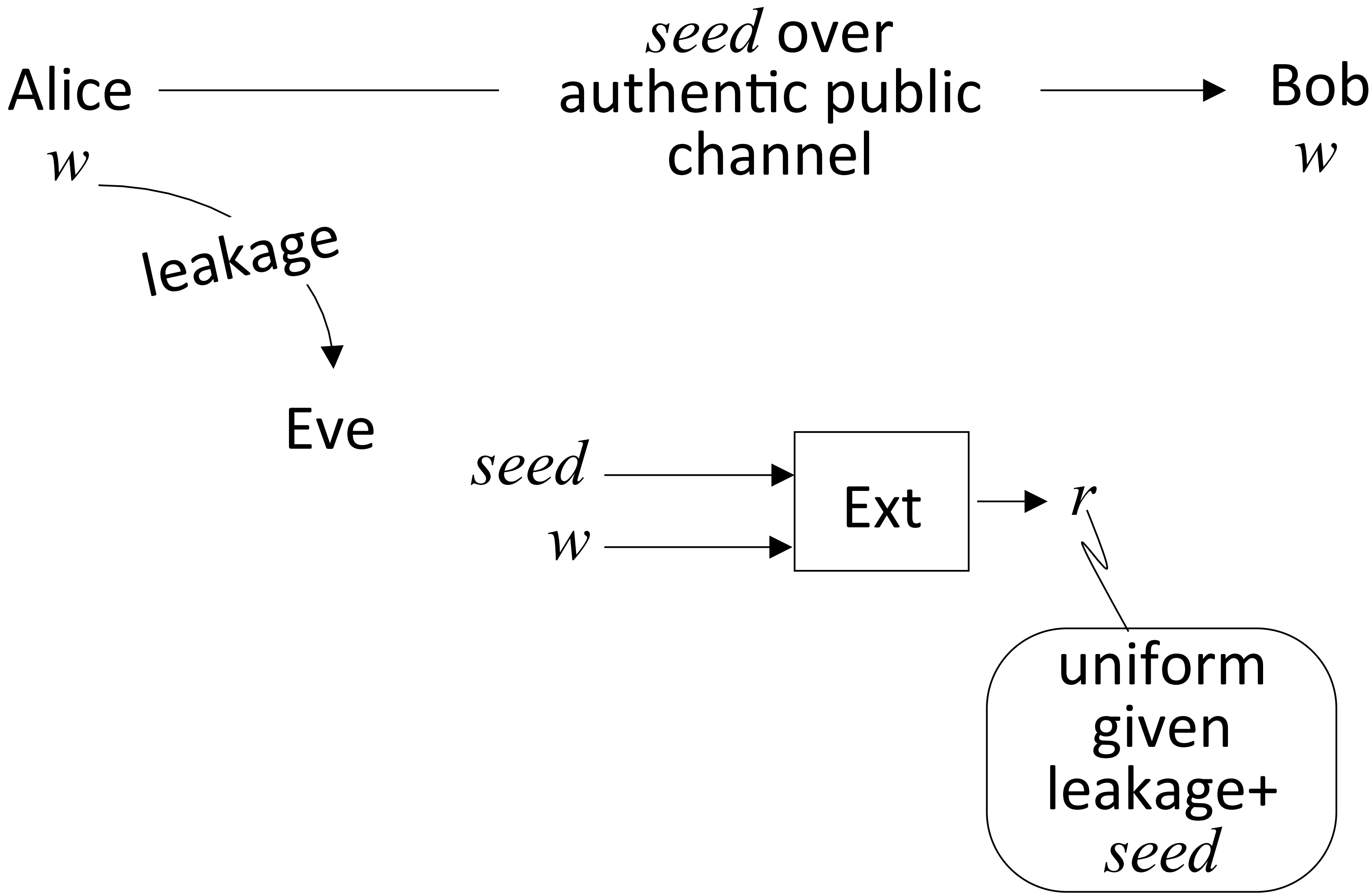


- Bennett-Brassard-Robert: no fixed function will work (as opposed to Wyner's specific kind of leakage)
- But: a random universal hash function is an extractor ("leftover hash lemma")
- Just send the choice of the function ("extractor seed") over the public channel

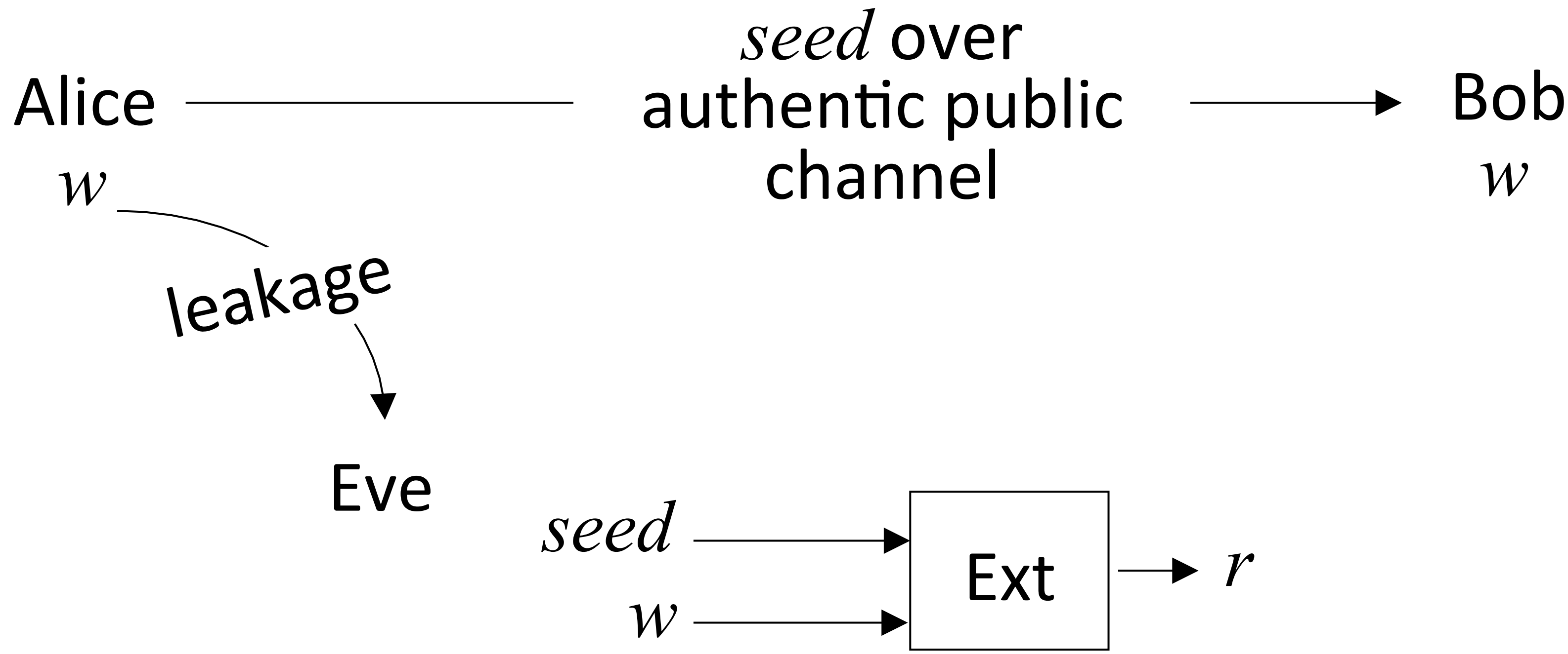
Privacy Amplification = Strong Extractor



Privacy Amplification = Strong Extractor



Privacy Amplification = Strong Extractor



- Note the power of the public channel:
  fewer assumptions on adversarial knowledge
- Modified goal: derive a good key
  (can use public channel once the key is derived)

~~Privacy~~ Amplification and Extractors
Hardness

Håstad-Impagliazzo-Levin-Luby 1989,99 (OWF⇒PRG)

another version of the same result:

universal hash function is an extractor

(thus, three chain from Wyner to extractors:

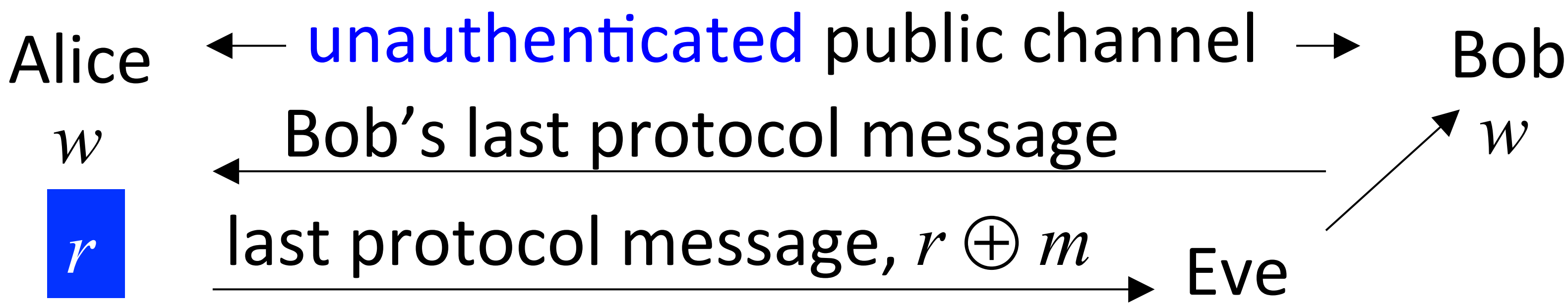
Wyner 75 → Yao 82 → Levin 87 → HILL

Wyner 75 → Santha-Vazirani

Wyner 75 → Bennett-Brassard-Robert)

New Model: no authentic channels (besides w)

[Maurer, Maurer-Wolf 1997, 2003]



- Can no longer simply send the extractor seed: adversary may tamper
- Single-message protocols still possible: "robust" extractors [Maurer-Wolf 97, Boyen-Dodis-Katz-Ostrovsky-Smith 05] but they exist only if w at least ½-entropic [Dodis-Wichs 09]
- Lots of work on multi-message protocols [RW03,KR09,DW09,CKOR10,DLWZ11,CRS12,Li12,Li15…]; important tool: "non-malleable" extractors [Dodis-Wichs 09]
- Two kinds of robustness: tampering before/after r is used (pre/post- application) [Dodis-Kanukurthi-Katz-Reyzin-Smith]

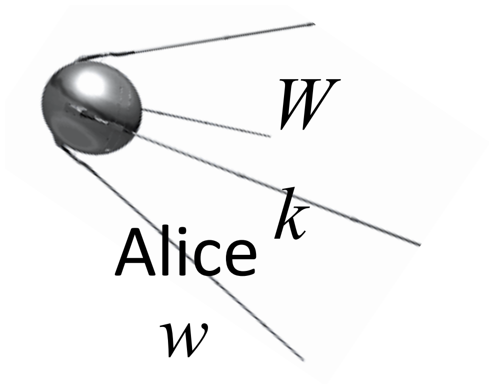
New Requirement: Source Privacy

Alice ⟵ unauthenticated public channel ⟶ Bob

w w

- Risk: tampering Eve may learn something about w by observing how the parties behave after she tampers
- Standalone security guarantees it won't be enough to cause problems
- But what about sequential security? Imagine w is obtained using a secret process; the next one uses the same process
- Need: "source private" extractors [Bouman,Fehr 2011] (Def'n: an active attack won't tell Eve anything about w)
- Relevant in, e.g., bounded storage model [Maurer 1990] and quantum key distribution

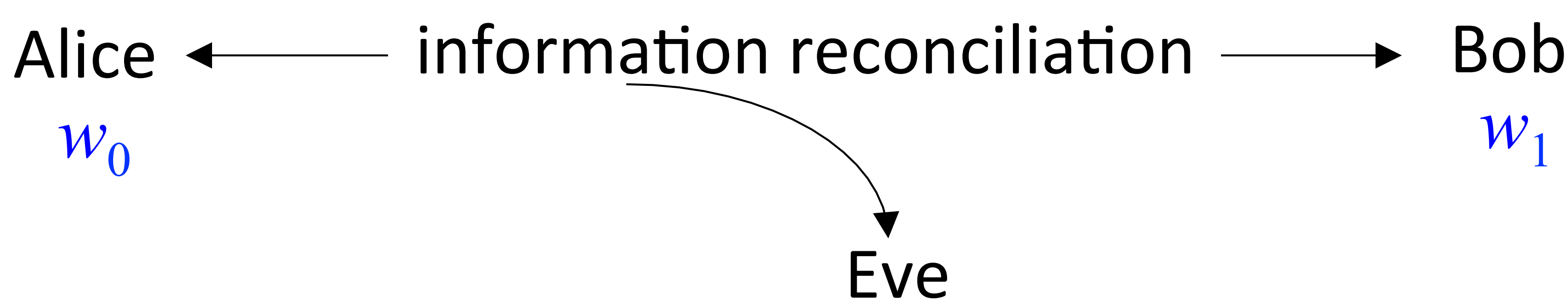
Bounded Storage Model



Bob k
w

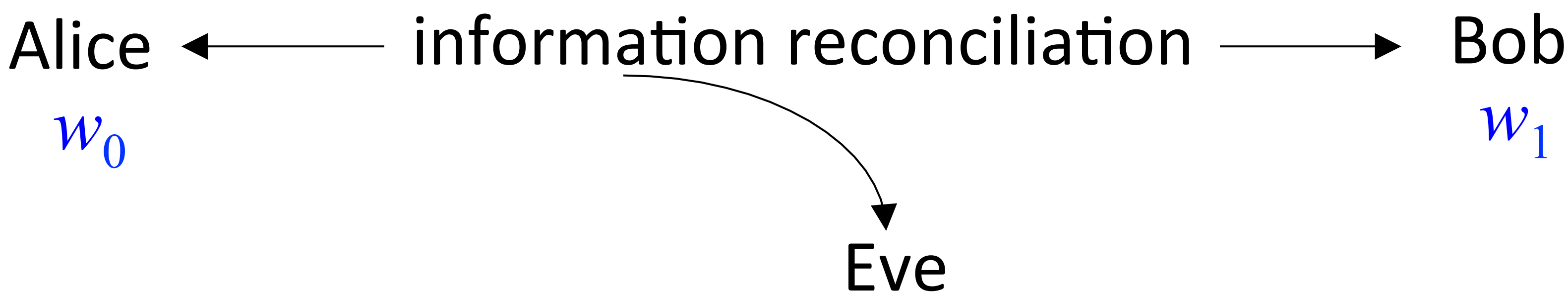
- [Maurer 1990, 92]:
  W is a HUGE (perhaps streaming) string no one can store
- Alice and Bob have a short shared secret k
  for probing W to get w
- Eve stores arbitrary information about W
- Need "locally computable" extractors [Lu 2002, 04]
  - Also used in "bounded retrieval model"
    [Dziembowski 06] [Di Crescenzo,Lipton,Walfish 06]
- k is the secret used to obtain w and needs to be reusable:
  hence need source privacy

Dealing With Errors



- Recall that Wyner's original paper had noise on the main channel, but no constructions for this case
- Common approach: reduce to the no-tampering case by performing information reconciliation over public channel
- Add whatever leaks during that process to Eve's knowledge
- Solve the no-tampering case using appropriate extractors
- Information reconciliation + extractors = "fuzzy extractors" [Dodis-Ostrovsky-Reyzin-Smith 2004, 08]

Applying to Biometrics



- Most of us have ≤10 fingers and ≤2 eyes
- Variants of w_0 derived from same iris scan may be used in different systems without coordination
- A single-protocol security guarantee doesn't extend: information reconciliation leakage may be additive
- Can we prevent multiple protocols from revealing w_0?
- Need: "reusable" extractors [Boyen 2004]

Many Kinds of Extractors

[robust/n-m] [local] [source-private] [fuzzy] [reusable]

Most combinations are interesting and valid as models

Many Kinds of Extractors

[robust/n-m] [local] [source-private] [fuzzy] [reusable]

Interaction: Deterministic / Single-Message / Interactive

Input constraints: what's minimum required entropy

Protocol quality: entropy loss, number of rounds if interactive

- Ex 1: active adversary, need to handle errors:
  robust + fuzzy
  [RW04,BDKOS05,DKKRS06,KR09,DW09,CKOR10, …]
- Ex 2: bounded storage model with errors
  local + fuzzy + source-private
  [Dodis-Smith 2005]
- Ex 3: active adversary, large secret, need to protect source
  post-app robust + source-private + local
  [Aggarwal-Dodis-Jafargholi-Miles-Reyzin 2014]

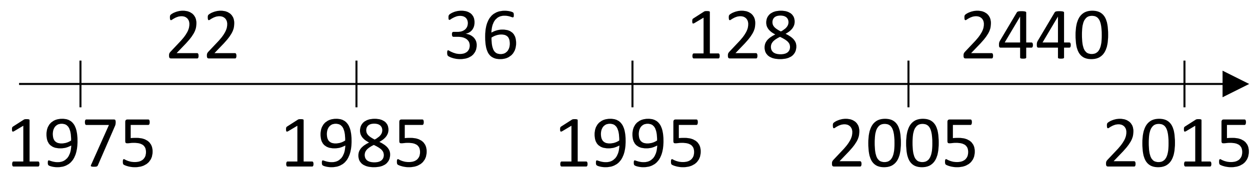
Information-Theoretic Protocols Beyond Key Agreement

- [Crépeau,Kilian 1988]
[Benett,Brassard,Crépeau,Skubiszweska 1991]
[Damgard, Kilian, Salvail 1999]
oblivious transfer and variations using noise/quantum

- [Crépeau 1997] bit commitment using noise

- Other works I probably don't know about, including many in quantum cryptography

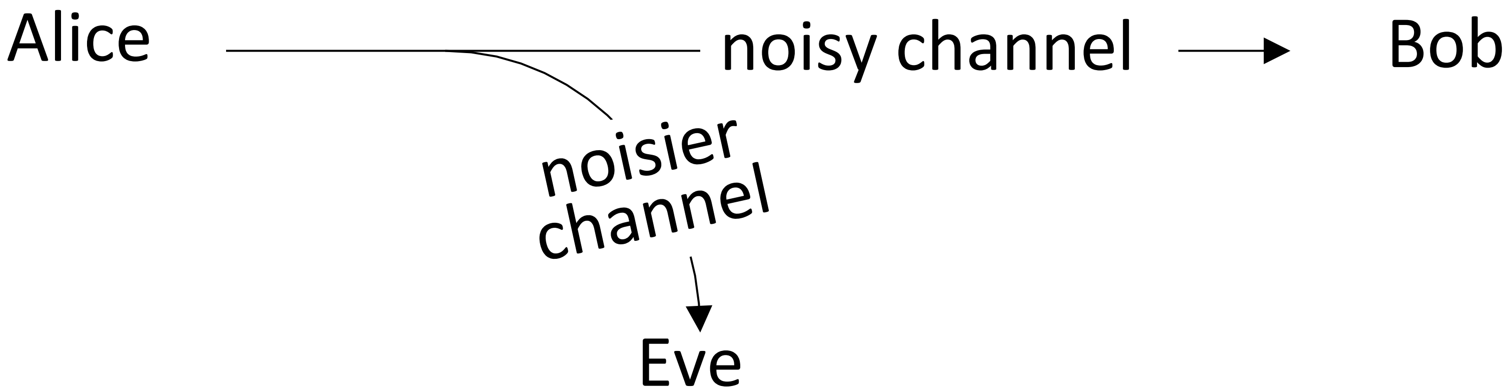
Given all this follow-up, what about Wyner's paper?



Google Scholar Citation Counts by Decade

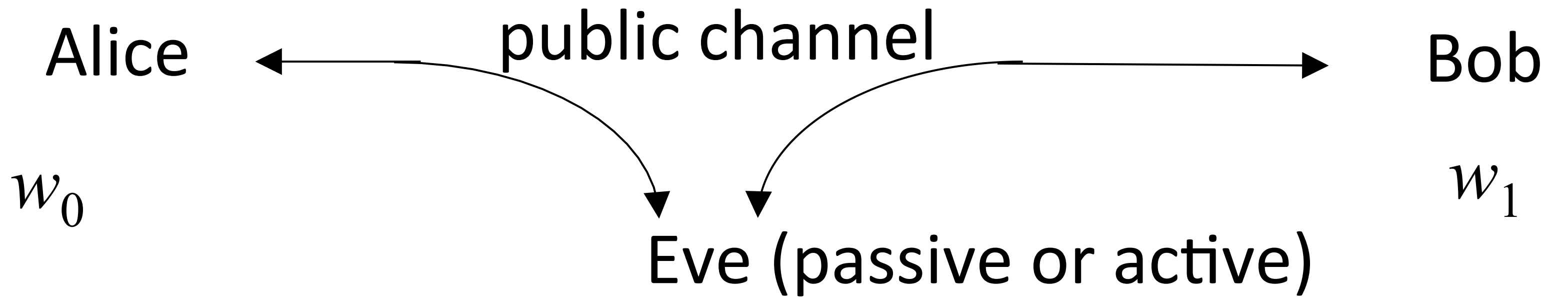
- Half the total citations are from the 2012-2015
- Most (?) are in the info-theory community, nonconstructive, with security definitions that won't make us happy
- Many are due to recent interest in physical-layer security

Given all this follow-up, what about Wyner's paper?



- Wyner's model is artificial for crypto community: we assume a free public channel and thus focus on key derivation
- Question: what can you do without a free channel (every bit counts – so "derive key + encrypt" loses half the rate)
- [Bellare Tessaro Vardy 2012]: first cryptographic treatment ("semantic security") and first optimal construction

Lessons

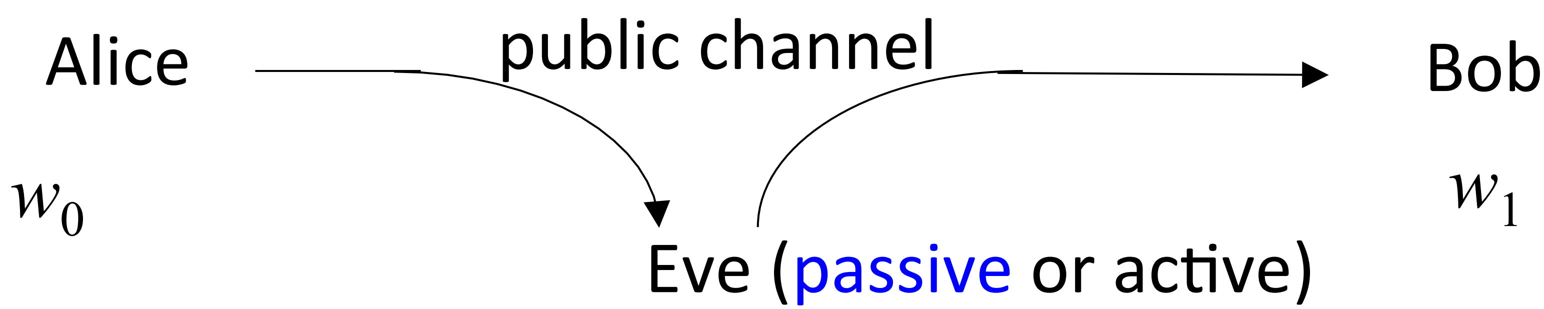


Secrets can come from nature, but we need to tame them

Research Directions:

- Finding the right notion of security
- Minimizing assumptions about adversarial knowledge
- Broadening sources of secrets
- Understanding fundamental bounds on what's feasible
  - Finding the right notion of input entropy
- Making it all efficient

Lessons



Secrets can come from nature, but we need to tame them

Research Directions:

- Finding the right notion of security
- Minimizing assumptions about adversarial knowledge
- Broadening sources of secrets
- Understanding fundamental bounds on what's feasible
  - Finding the right notion of input entropy
- Making it all efficient

Part II

Fuzzy Extractors for Noisy Sources with More Errors than Entropy

Ran Canetti, Benjamin Fuller, Omer Paneth, Leonid Reyzin, and Adam Smith

http://eprint.iacr.org/2014/243

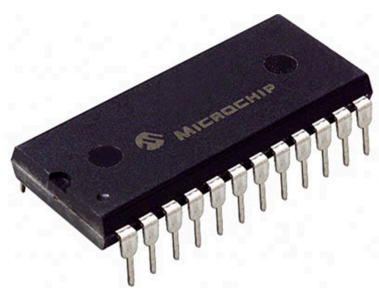
Our Setting

Alice w_0 → information reconciliation + privacy amplification → Bob w_1

↓

Non-Tampering Eve

- Single message:
  Alice and Bob can be the same person at different times
- Target application:
  key extraction from unique physical features

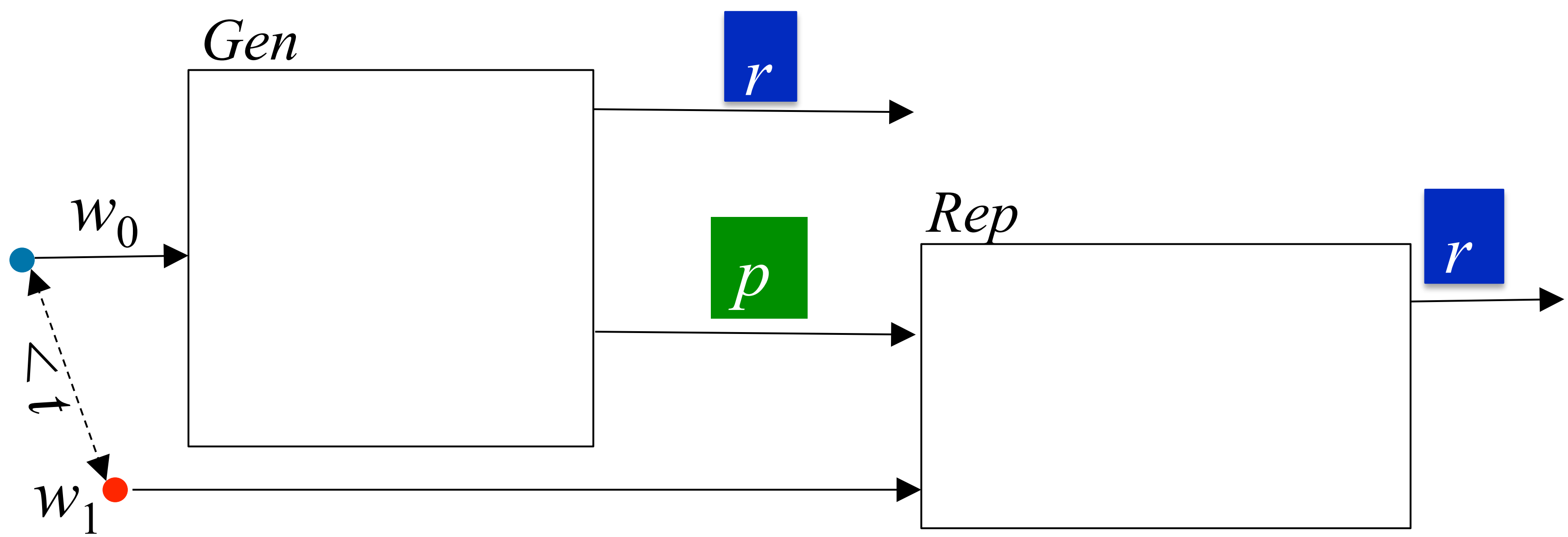


Physically Unclonable Functions (PUFs)

Biometrics

Notation

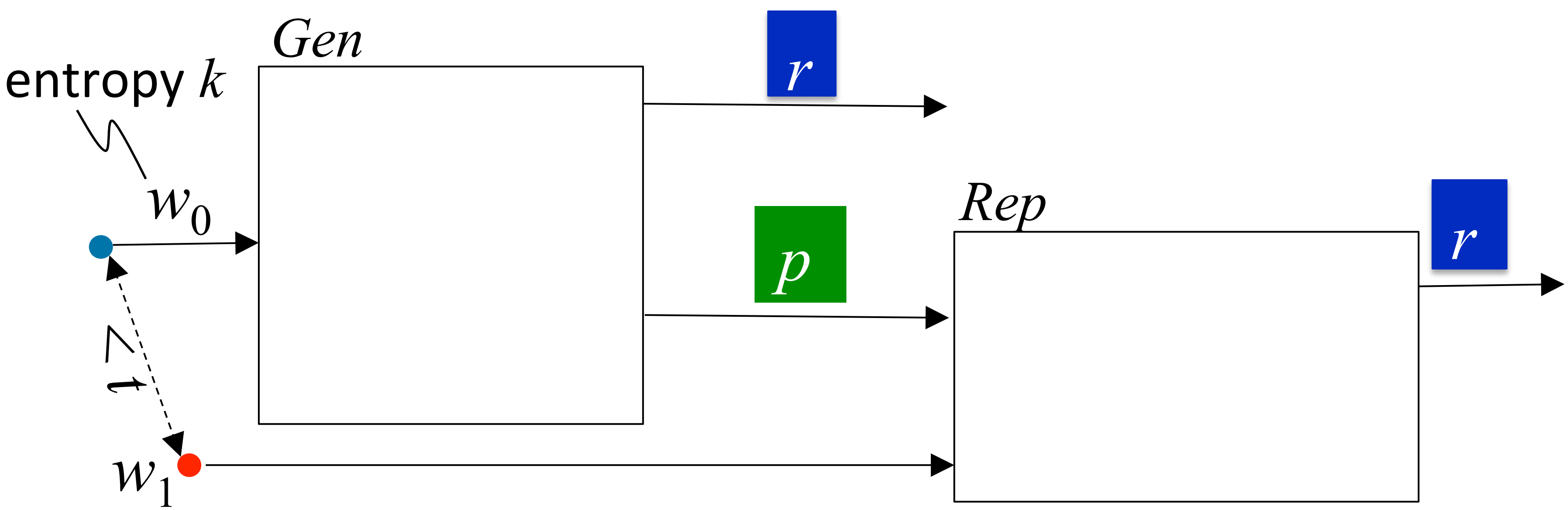
- Enrollment algorithm *Gen* (Alice):
  Take a measurement w_0 from the source.
  Use it to "lock up" a random output in a nonsecret value p.
- Subsequent algorithm *Rep* (Bob):
  give same output if $d(w_0, w_1) < t$
- Security: r looks uniform even given p,
  whenever the source is good enough



Fuzzy Extractors: Goals

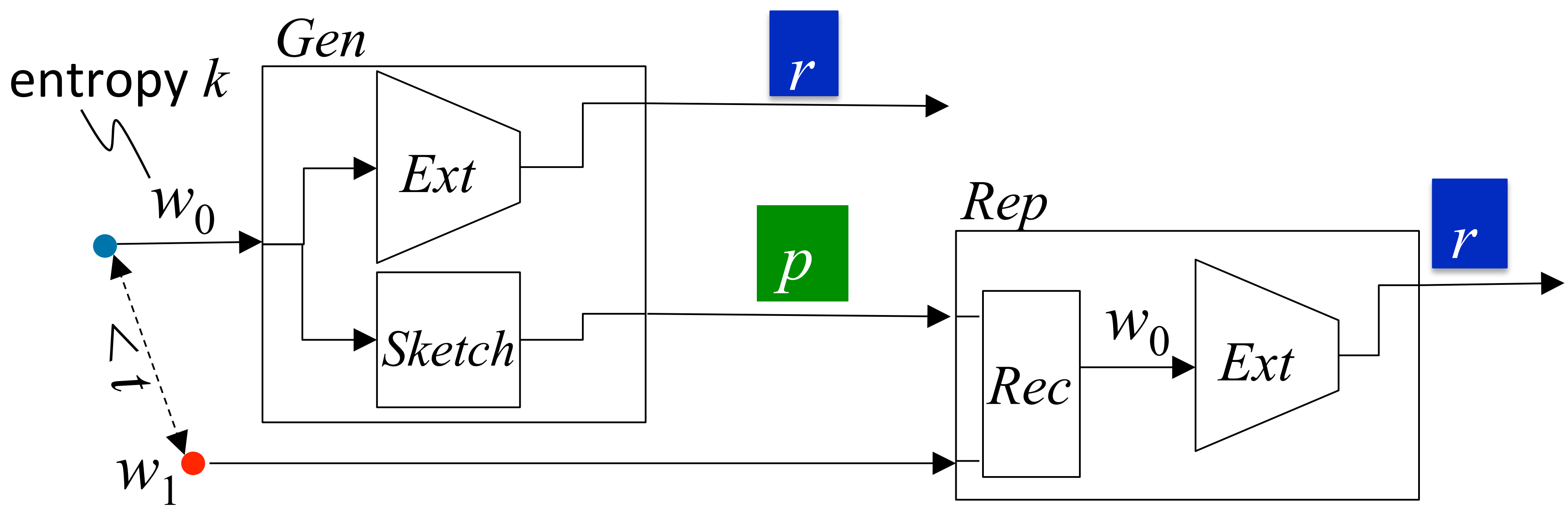
- Goal 1: handle as many sources as possible
(typically, any source in which w_0 is 2^k hard to guess)
- Goal 2: handle as much error as possible
(typically, any w_1 within distance t)
- Most previous approaches are analyzed in terms of t and k

This work: handle $t > k$



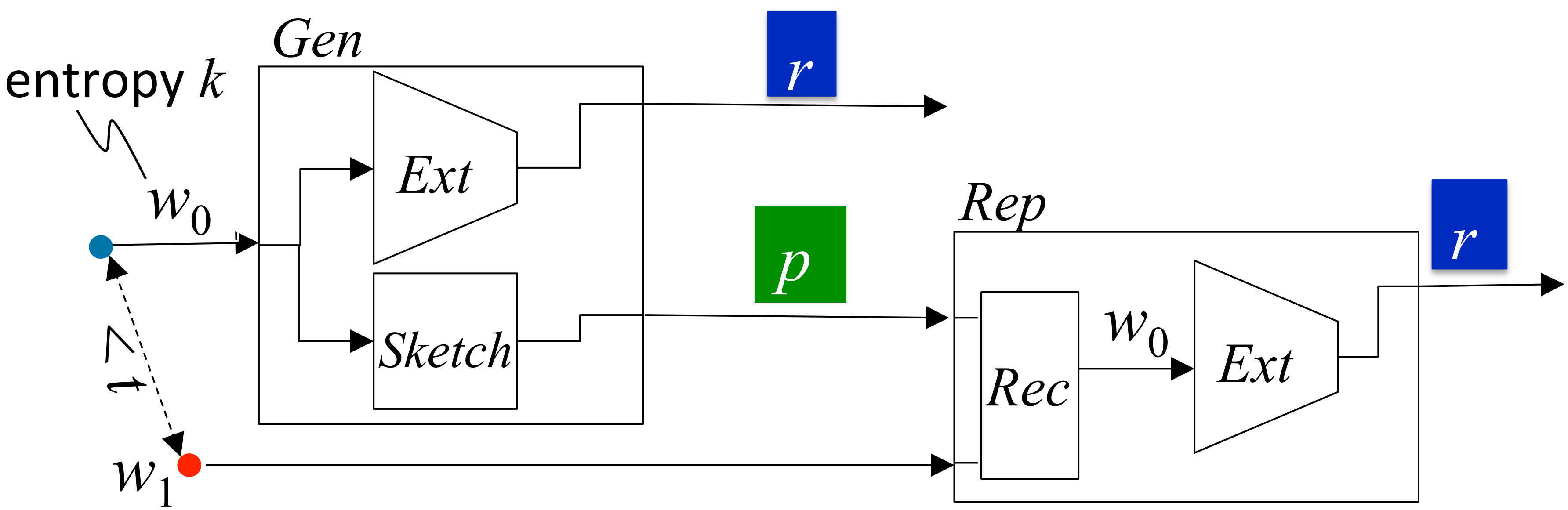
Fuzzy Extractors: Typical Construction

- derive r using a randomness extractor
  (converts high-entropy sources to uniform,
  e.g., via universal hashing)
- correct errors using a *secure sketch*
  (gives recovery of the original from a noisy signal
  e.g., via the "checksum" bits of an error-correcting code)



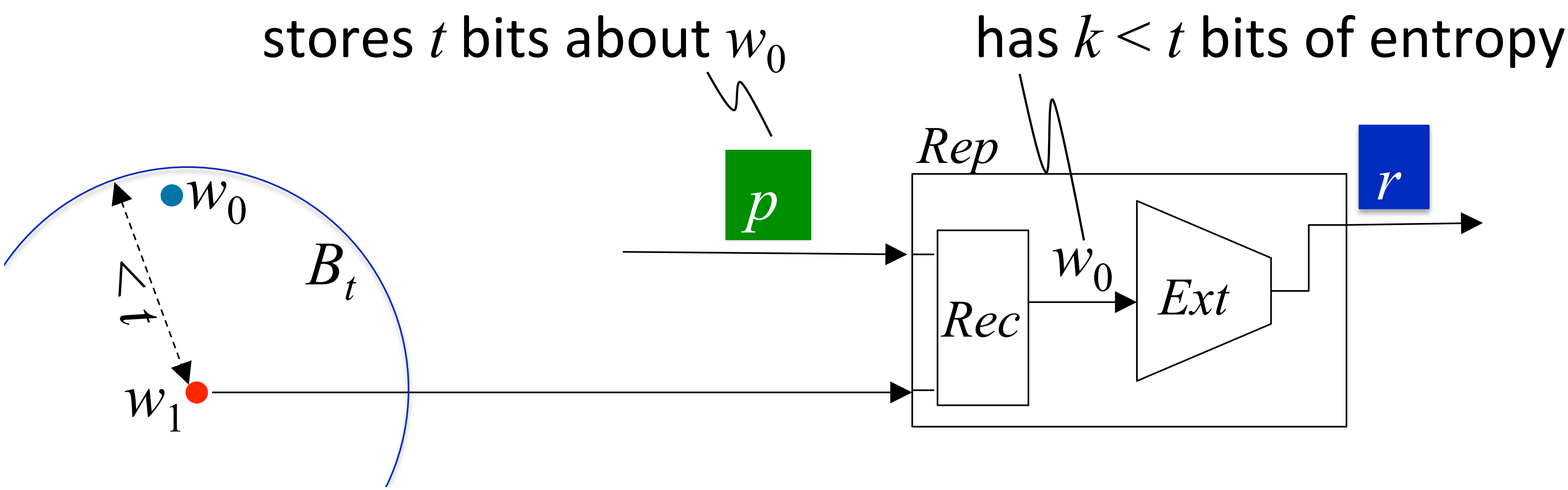
Problem with Secure Sketches

- p must store enough information to let you recover w_0
- How much information is that?



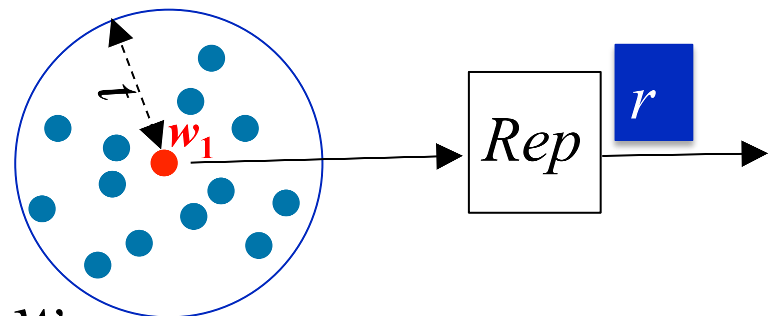
Problem with Secure Sketches

- p must store enough information to let you recover w_0
- How much information is that?
- w_0 could be anywhere within distance t, so $\log|B_t| > t$ bits
- No security left if $t > k$
  (can be made rigorous for large classes of sources)
- Observation: not necessary to recover w_0

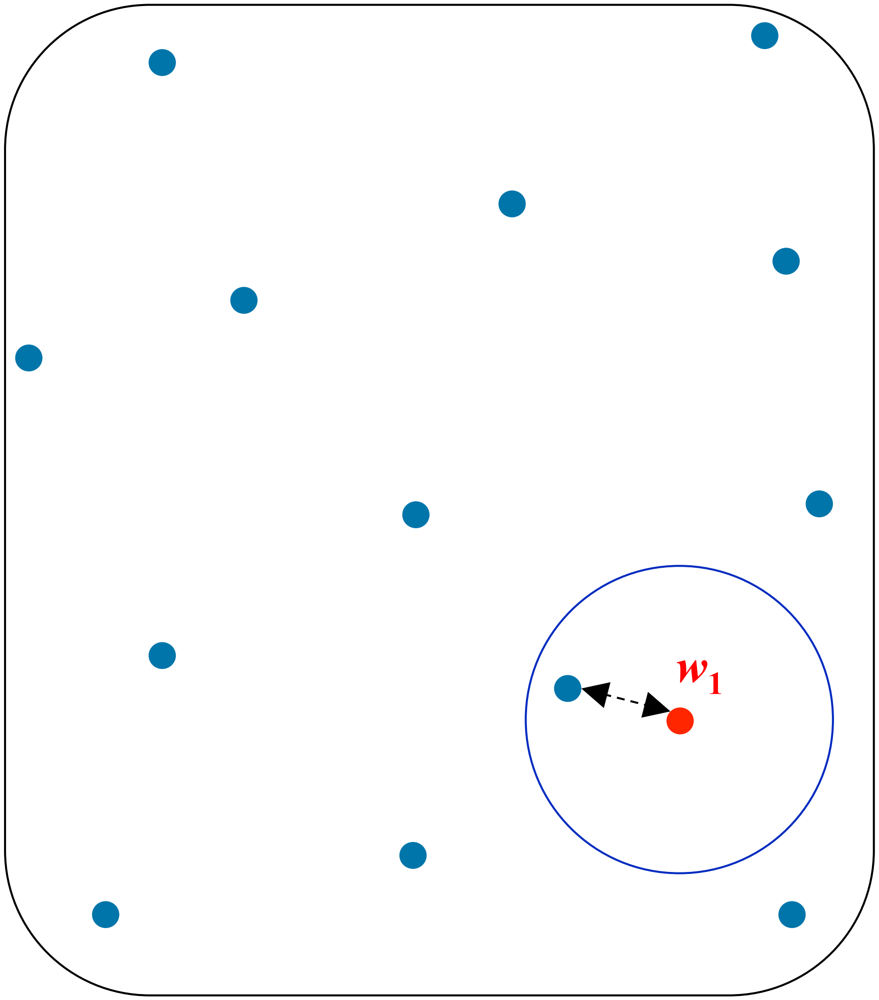


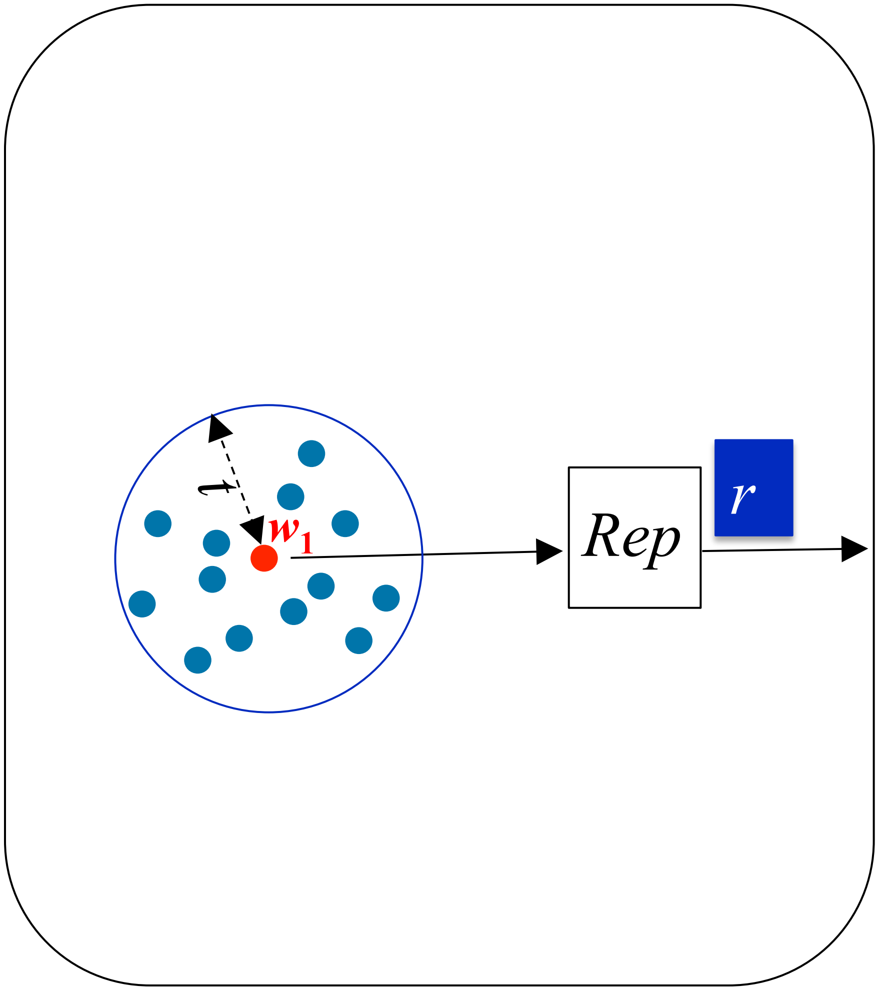
Is it possible to handle "more errors than entropy" ($t > k$)?

- Consider some distribution for w_0 with entropy k
- Suppose $t > k$
- Then $B_t > 2^k$
- Possibly $|B_t| >$ # of possibilities for w_0
- Possibly all w_0 lie in a single ball
- No matter what we do, adversary can get the output by running *Rep* on w_1 = center of that ball



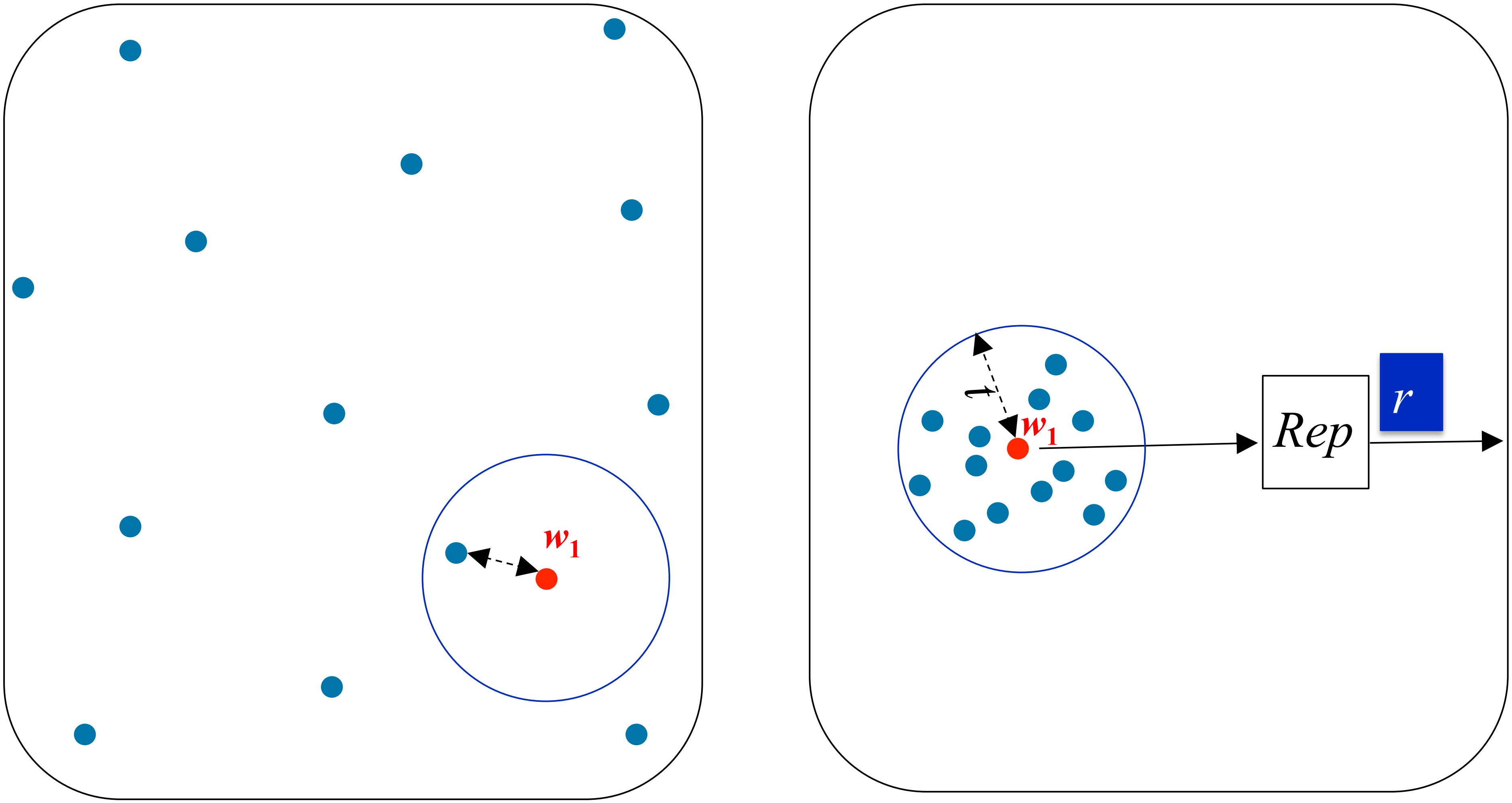
Is it possible to handle "more errors than entropy" ($t > k$)?





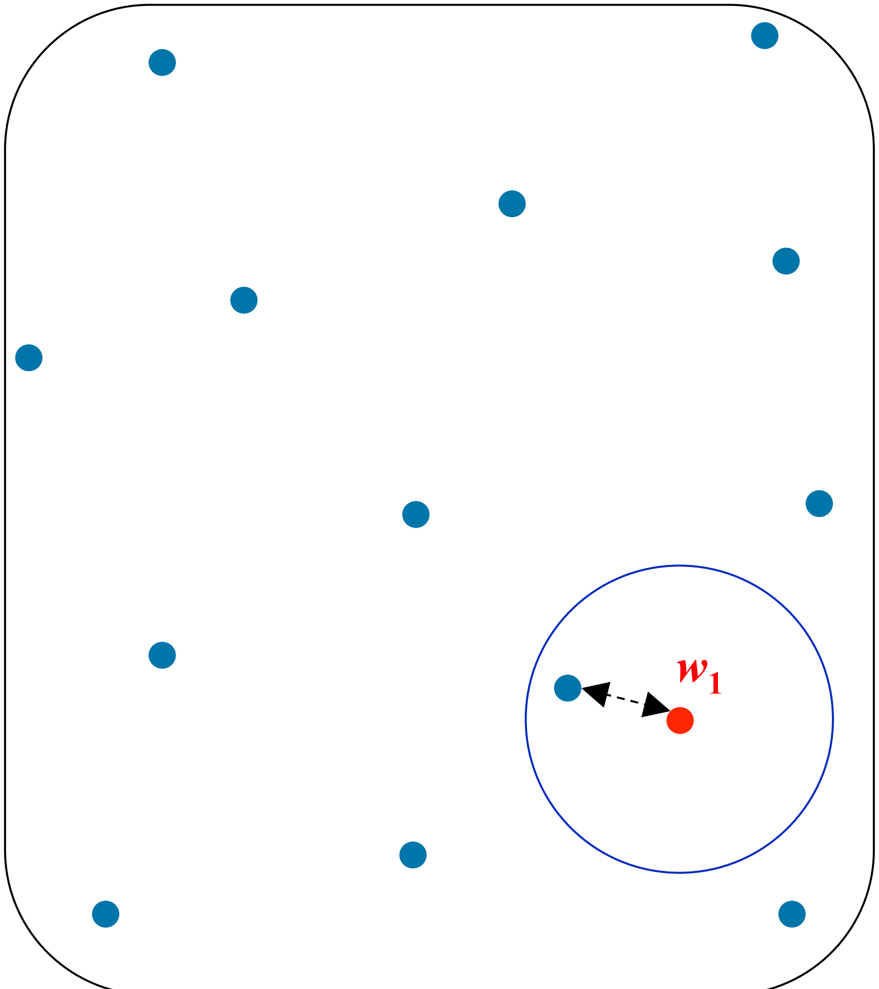
But if all the points are far apart, the problem is trivial!
(at least information-theoretically)

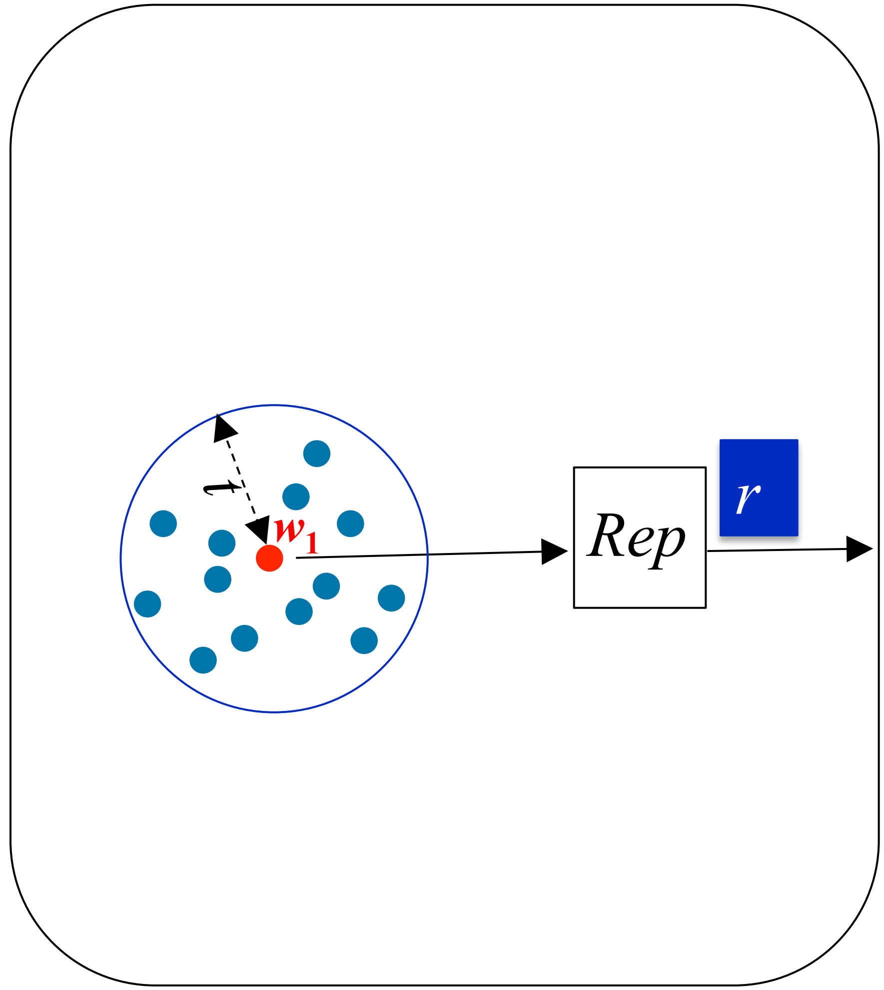
Is it possible to handle "more errors than entropy" ($t > k$)?



No construction that is analyzed
only in terms of t and k can distinguish the two cases

Is it possible to handle "more errors than entropy" ($t > k$)?





Moral: our constructions will exploit structure in the source (not "any source of a given k" like prior work)

What Sources Can We Handle?

1. Sources with high-entropy samples

$$w_0 = a_1\, \boxed{a_2}\, a_3\, a_4\, \boxed{a_5}\, a_6\, \boxed{a_7}\, a_8\, a_9$$

sample: $a_2\, a_5\, a_7$

We need: for some superlog sample size
you are guaranteed to get superlog entropy

Sufficient assumption: somewhat q-wise independence
for superlogarithmic q

E.g., IrisCode [Daugman] is redundant and noisy (t>>k):

$\log |B_t| \approx 900$ but $k \approx 250$

Yet this assumption is plausible

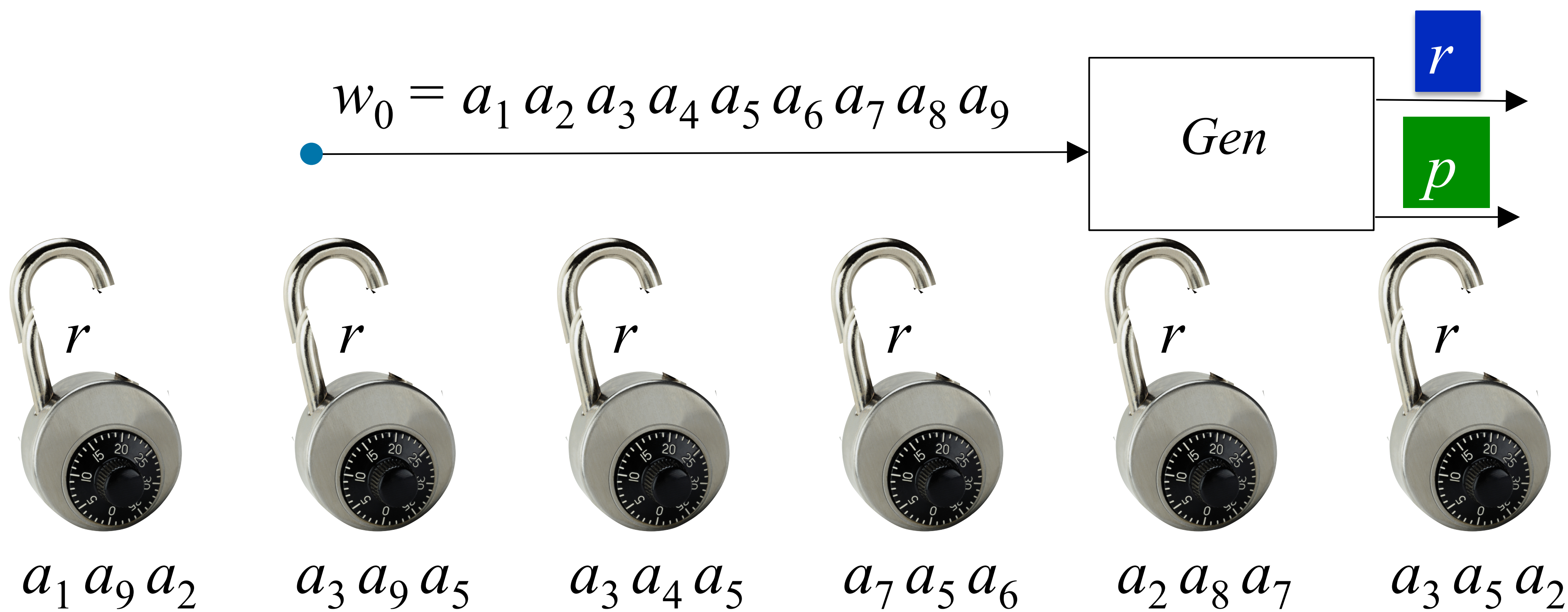
Idea: "encrypt" r using parts of w_0

Source: a string of symbols, arbitrary alphabet

Errors: Hamming

Gen: - get random combinations of symbols in w_0

- "lock" r using these combinations



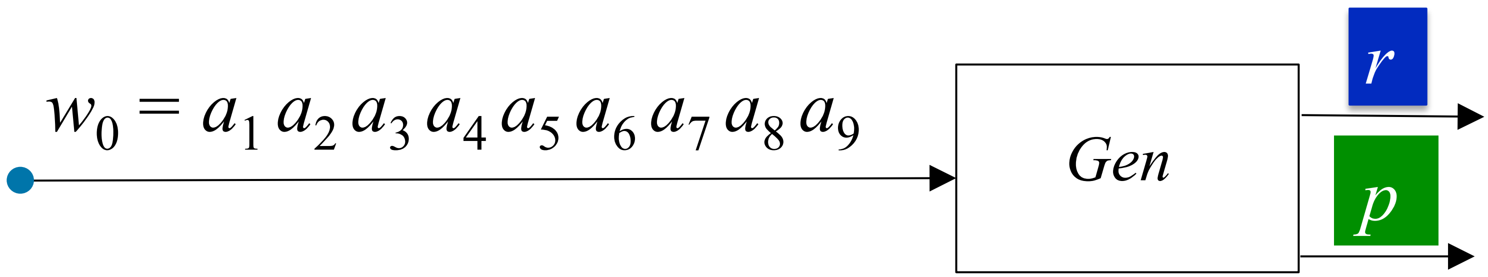
Idea: "encrypt" r using parts of w_0

Source: a string of symbols, arbitrary alphabet

Errors: Hamming

Gen: - get random combinations of symbols in w_0

- "lock" r using these combinations
- p = locks + positions of symbols needed to unlock



$a_1\, a_9\, a_2$ $a_3\, a_9\, a_5$ $a_3\, a_4\, a_5$ $a_7\, a_5\, a_6$ $a_2\, a_8\, a_7$ $a_3\, a_5\, a_2$

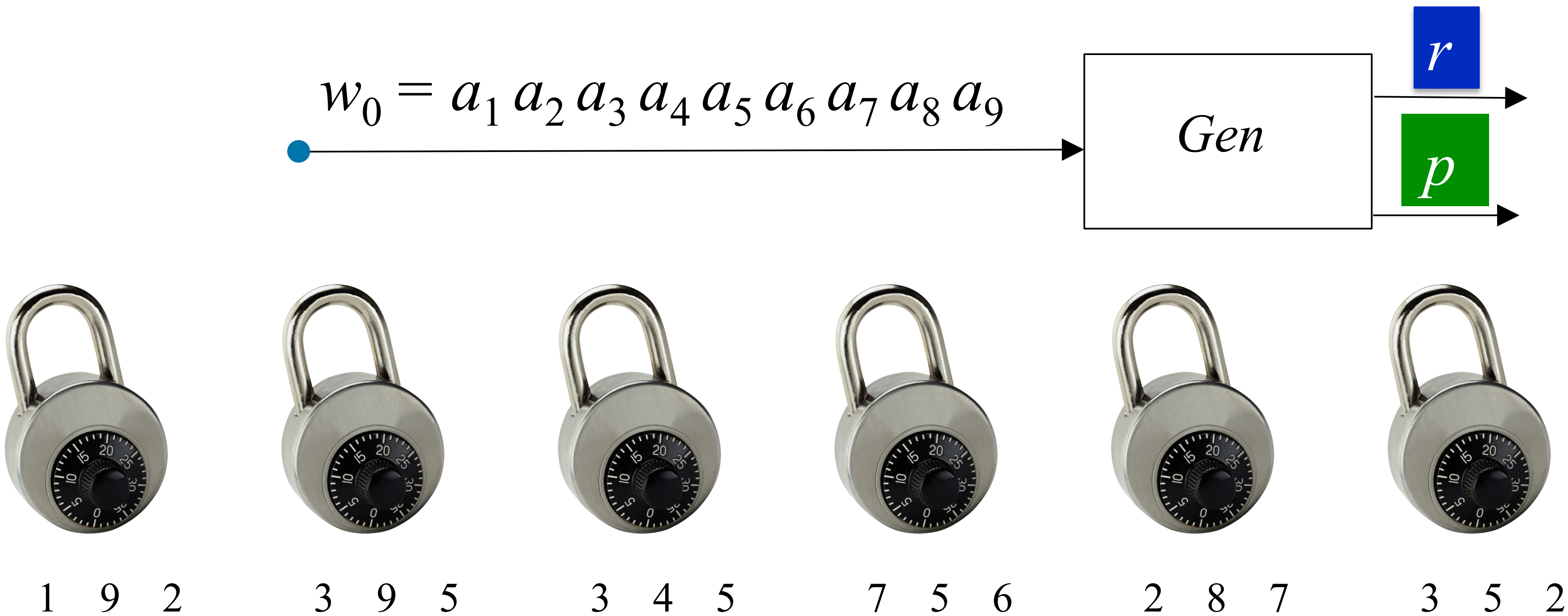
Idea: "encrypt" r using parts of w_0

Source: a string of symbols, arbitrary alphabet

Errors: Hamming

Gen: - get random combinations of symbols in w_0

- "lock" r using these combinations
- p = locks + positions of symbols needed to unlock



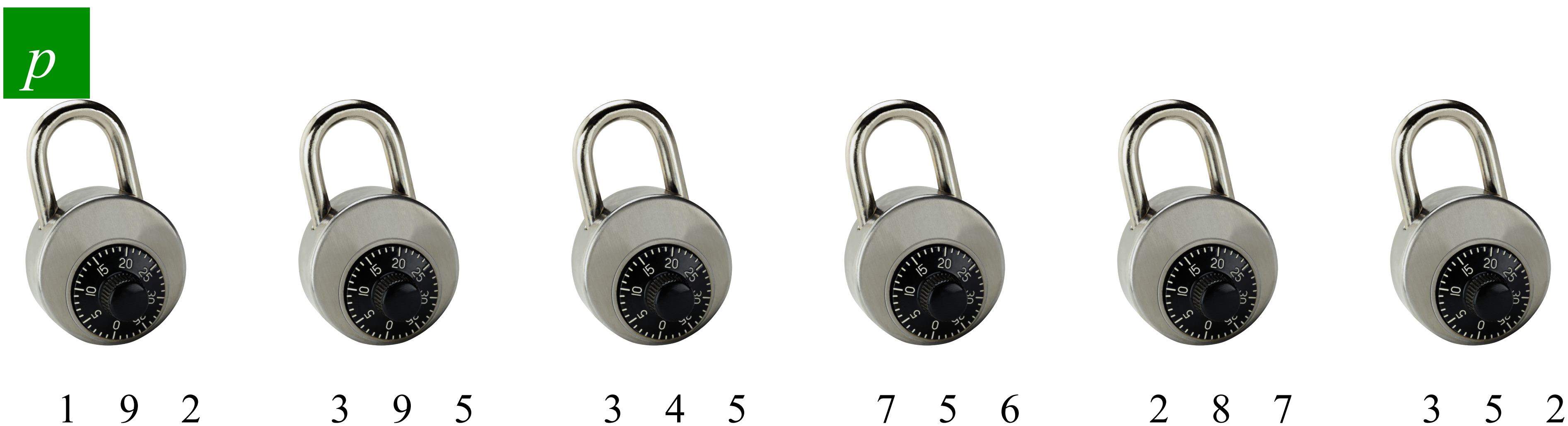
Idea: "encrypt" r using parts of w_0

Source: a string of symbols, arbitrary alphabet

Errors: Hamming

Gen: - get random combinations of symbols in w_0

- "lock" r using these combinations
- p = locks + positions of symbols needed to unlock



Idea: “encrypt” r using parts of w_0

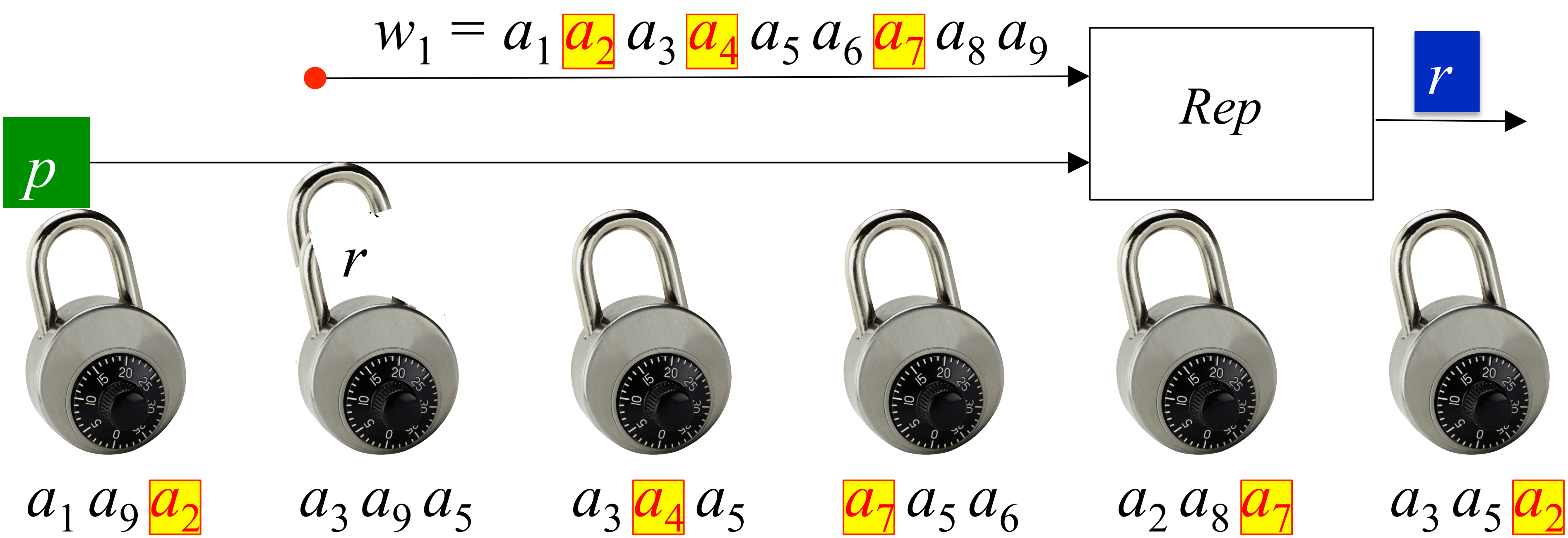
Source: a string of symbols, arbitrary alphabet

Errors: Hamming

Gen: - get random combinations of symbols in w_0

- “lock” r using these combinations
- p = locks + positions of symbols needed to unlock

Rep: Use the symbols of w_1 to open at least one lock



Idea: "encrypt" r using parts of w_0

Source: a string of symbols, arbitrary alphabet

Errors: Hamming

Gen: - get random combinations of symbols in w_0

- "lock" r using these combinations
- p = locks + positions of symbols needed to unlock

Rep: Use the symbols of w_1 to open at least one lock

Error-tolerance: as long as at least one combination is ok

Security: each combination must have enough entropy

$a_1\, a_9\, a_2$ $a_3\, a_9\, a_5$ $a_3\, a_4\, a_5$ $a_7\, a_5\, a_6$ $a_2\, a_8\, a_7$ $a_3\, a_5\, a_2$

How to implement locks?

R.O. model [Lynn Prabhakaran Sahai 04]:

lock = nonce, Hash(nonce, $a_1\, a_9\, a_2$) $\oplus$ (r||00…0)

How to implement locks?

- A lock is the following program:
  - If input = $a_1\, a_9\, a_2$, output r
  - Else output $\perp$

$a_1\, a_9\, a_2$

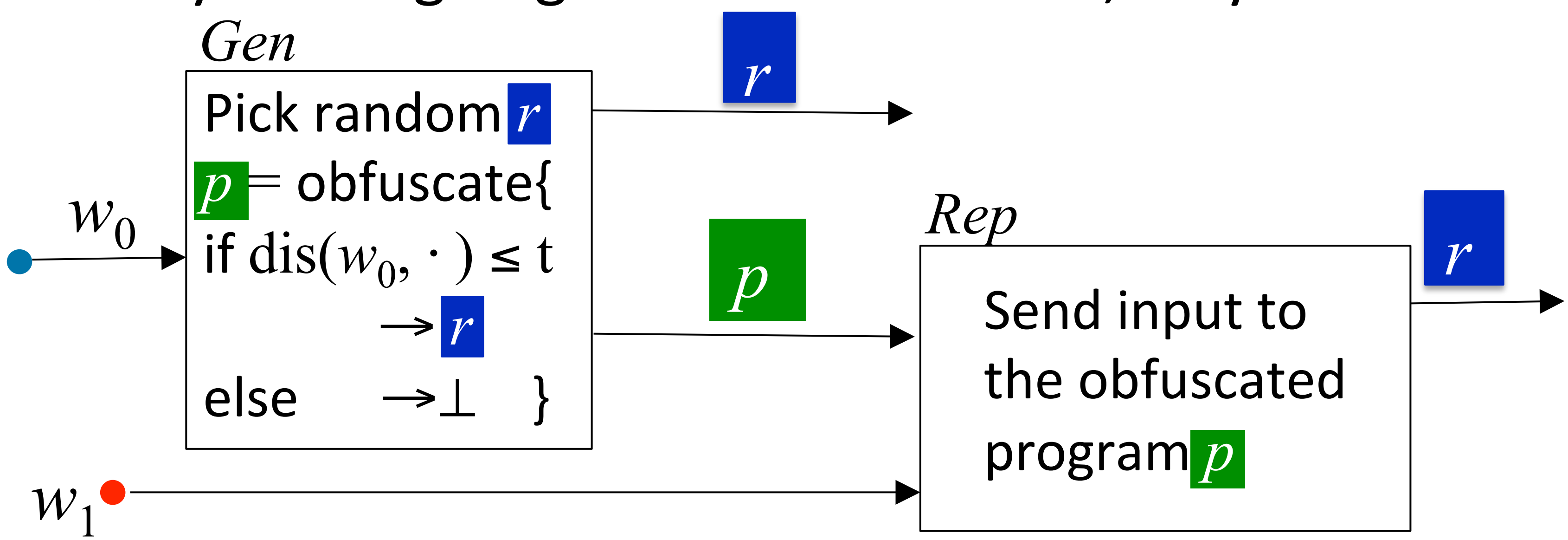
- Obfuscate this program!
  - Obfuscation: preserve functionality, hide the program
  - Obfuscating this specific program gives a "digital locker": encryption of r that is secure even multiple times with correlated and weak keys [Canetti Kalai Varia Wichs 10]
  - For this specific program: obfuscation is practical (R.O. or DL-based) [Canetti Dakdouk 08], [Bitansky Canetti 10]
  - Hiding r as long as the input can't be exhaustively searched (superlogarithmic entropy)

How to implement locks?

- A lock is the following program:
  - If input = $a_1\, a_9\, a_2$, output r
  - Else output ⊥

$a_1\, a_9\, a_2$

- Q: if you are going to use obfuscation, why not this:



- A: you can do that [Bitansky Canetti Kalai Paneth 14], except it's very impractical + has a very strong assumption

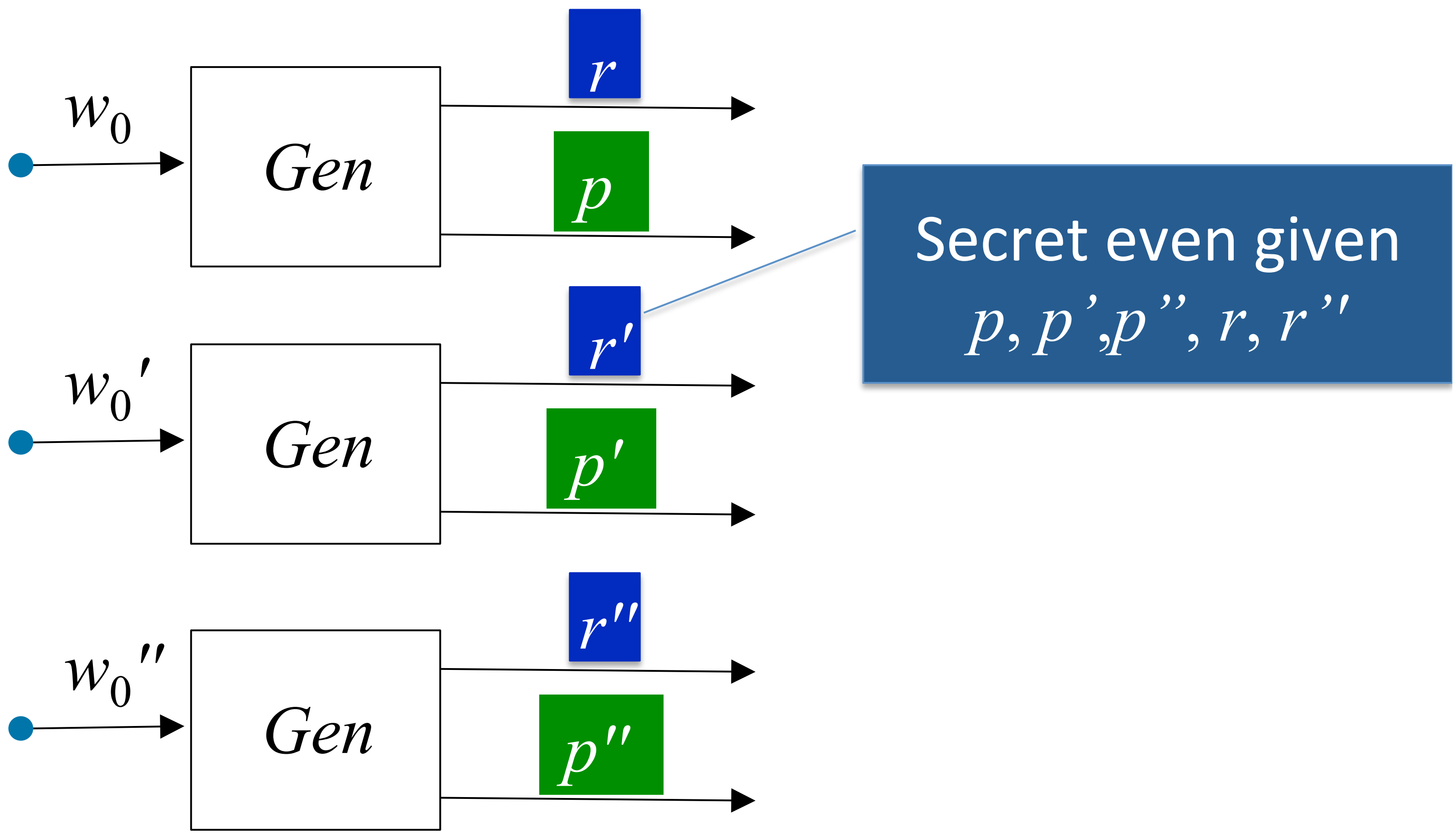
How good is this construction?

- We can correct more errors than entropy!
- For correctness: need sublinear error
- Note: computational, not information-theoretic, security

$a_1\, a_9\, a_2$ $\quad$ $a_3\, a_9\, a_5$ $\quad$ $a_3\, a_4\, a_5$ $\quad$ $a_7\, a_5\, a_6$ $\quad$ $a_2\, a_8\, a_7$ $\quad$ $a_3\, a_5\, a_2$

How good is this construction?

- It is reusable!
  - Same source can be enrolled multiple times with multiple independent services



How good is this construction?

- It is reusable!
  - Same source can be enrolled multiple times with multiple independent services
  - Follows from composability of obfuscation
  - In the past: difficult to achieve, because typically new enrollments leak fresh information
  - Previous constructions: non-fuzzy [Dodis Kalai Lovett 09] or all readings must differ by fixed constants [Boyen 2004]
  - Our construction: each reading individually must satisfy our conditions

What Sources Can We Handle?

1. Sources with high-entropy samples (with reusability)!
2. Sources with sparse high-entropy marginals
   (requires large alphabets)

$$w_0 = \sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup\sqcup$$

OR

$$w_0 = \underline{\quad\quad}\,\underline{\quad\quad}\,\underline{\quad\quad}\,\underline{\quad\quad}\,\underline{\quad\quad}\,\underline{\quad\quad}$$

What Sources Can We Handle?

1. Sources with high-entropy samples (with reusability)!

2. Sources with sparse high-entropy marginals (requires large alphabets)

   Constraint: individual symbols have high entropy (but no independence assumed)

$$w_0 = \underline{a_1}\ \underline{a_2}\ \underline{a_3}\ \underline{a_4}\ \underline{a_5}\ \underline{a_6}$$

a_1 a_2 a_3 a_4 a_5 a_6

Construction for Sparse High-Entropy Marginals

Problem: each low-entropy symbol reveals one bit of r

Solution: use a randomness extractor at the end

$$w_0 = \underline{a_1}\ \underline{a_2}\ \underline{a_3}\ \underline{a_4}\ \underline{a_5}\ \underline{a_6}$$

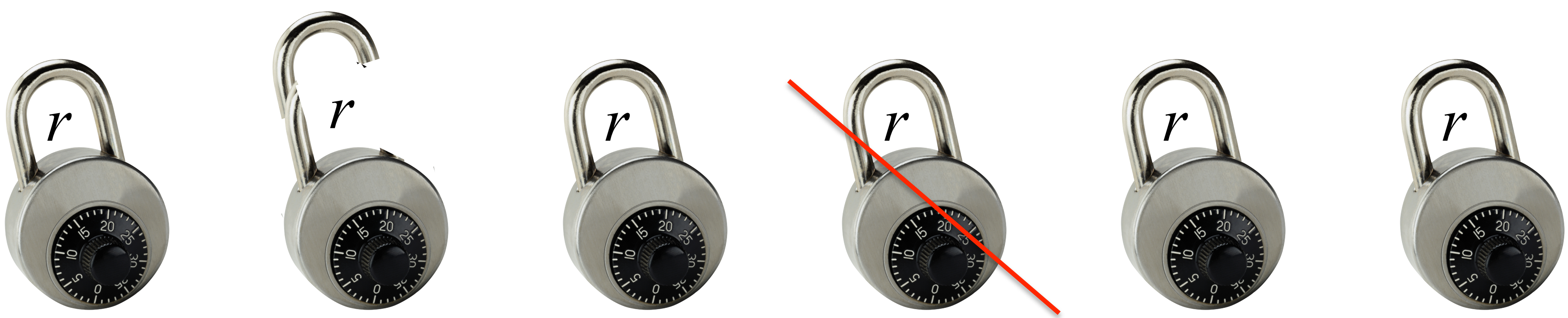
a_1 a_2 a_3 a_4 a_5 a_6

Construction for Sparse High-Entropy Marginals

Problem: differences in w_1 will make us miss some bits

Solution: use an error-correcting code

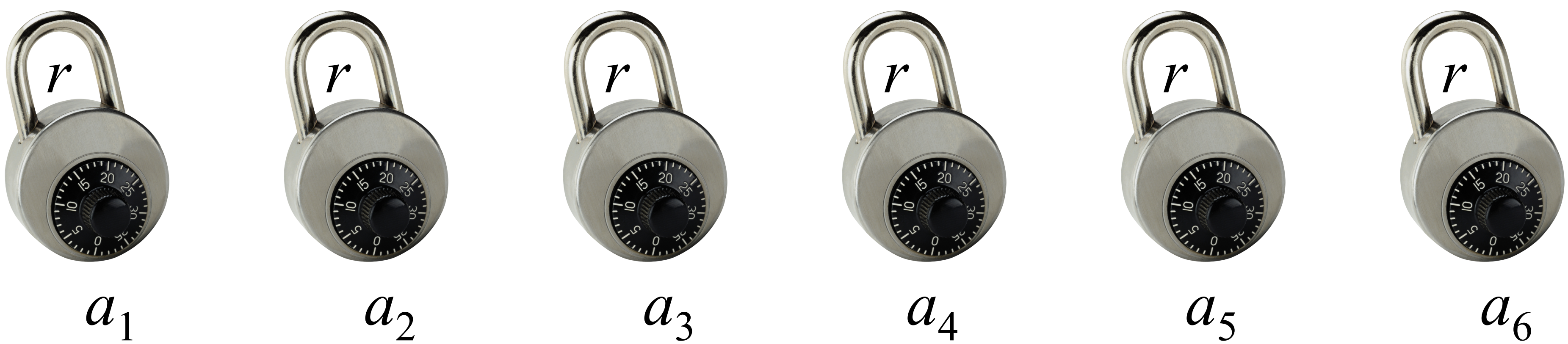
$$w_1 = \underline{a_1}\ \underline{a_2}\ \underline{a_3}\ \underline{a_4}\ \underline{a_5}\ \underline{a_6}$$



a_1 a_2 a_3 a_4 a_5 a_6

What Sources Can We Handle?

1. Sources with high-entropy samples:
   with reusability! (subconstant error rate)
2. Sources with sparse high-entropy marginals:
   with constant error rate! (requires large alphabets)
3. Sparse block sources
   information theoretically! (stricter entropy condition)



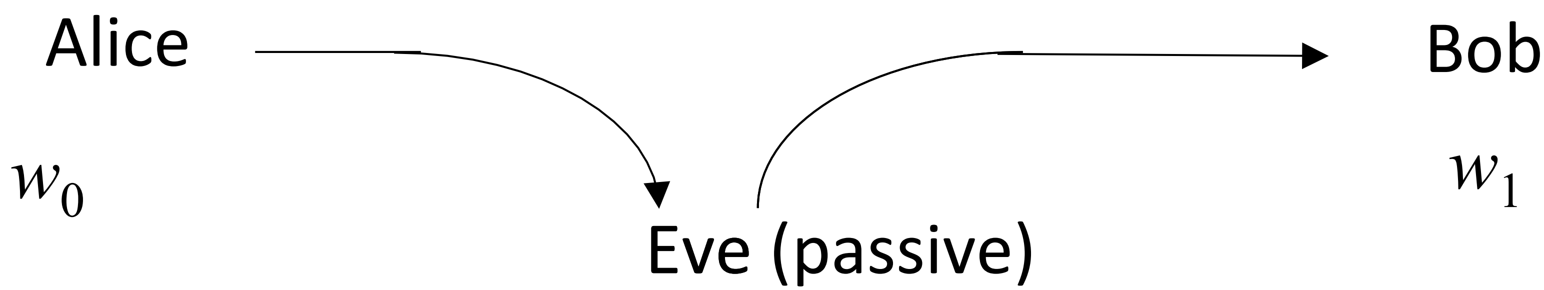
What Sources Can We Handle?

1. Sources with high-entropy samples:
   with reusability! (subconstant error rate)
2. Sources with sparse high-entropy marginals:
   with constant error rate! (requires large alphabets)
3. Sparse block sources
   information theoretically! (stricter entropy condition)

<u>Ideas</u>:

- For sources with more errors than entropy:
  - avoid information reconciliation
  - exploit the source structure
- For reusability:
  - use computational security

What I Just Showed

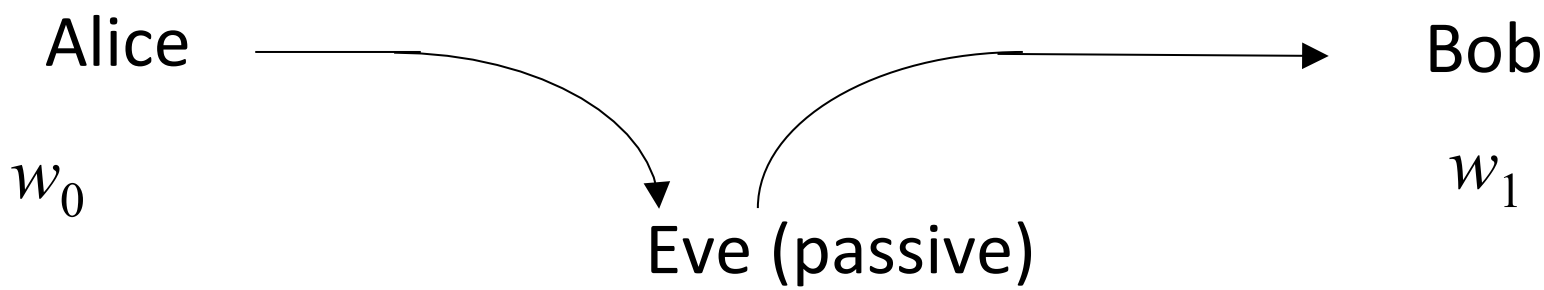


Secrets can come from nature, but we need to tame them

Research Directions:

- Finding the right notion of security
- Minimizing assumptions about adversarial knowledge
- Broadening sources of secrets
- Understanding fundamental bounds on what's feasible
  – Finding the right notion of input entropy
- Making it all efficient

What I will show next



Secrets can come from nature, but we need to tame them

Research Directions:

- Finding the right notion of security
- Minimizing assumptions about adversarial knowledge
- Broadening sources of secrets
- Understanding fundamental bounds on what's feasible
  - Finding the right notion of input entropy
- Making it all efficient

Fir0002/Flagstaffotos

Part III

When are Fuzzy Extractors Possible?

Benjamin Fuller, Leonid Reyzin, and Adam Smith

http://eprint.iacr.org/2014/961

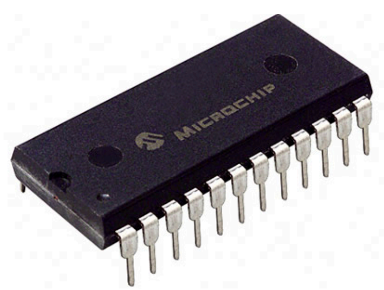
Our Setting (same as Part II)

Alice w_0 ——— information reconciliation + privacy amplification ———→ Bob w_1

↓

Non-Tampering Eve

- Single message:
  Alice and Bob can be the same person at different times
- Target application:
  key extraction from unique physical features

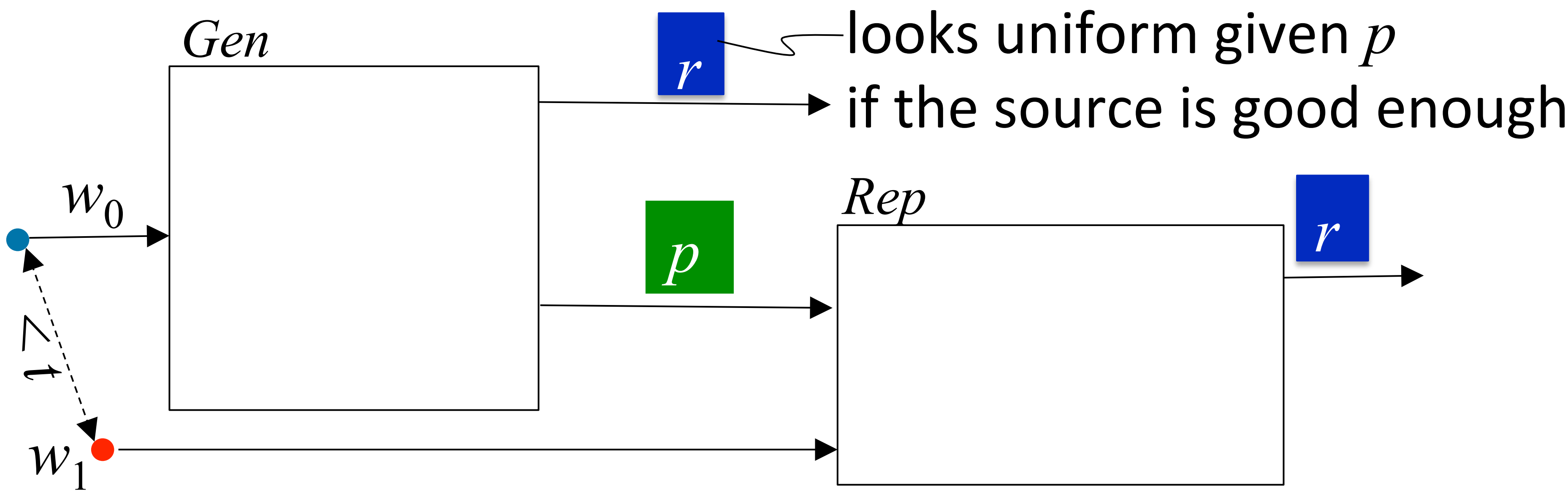


Physically Unclonable Functions (PUFs)

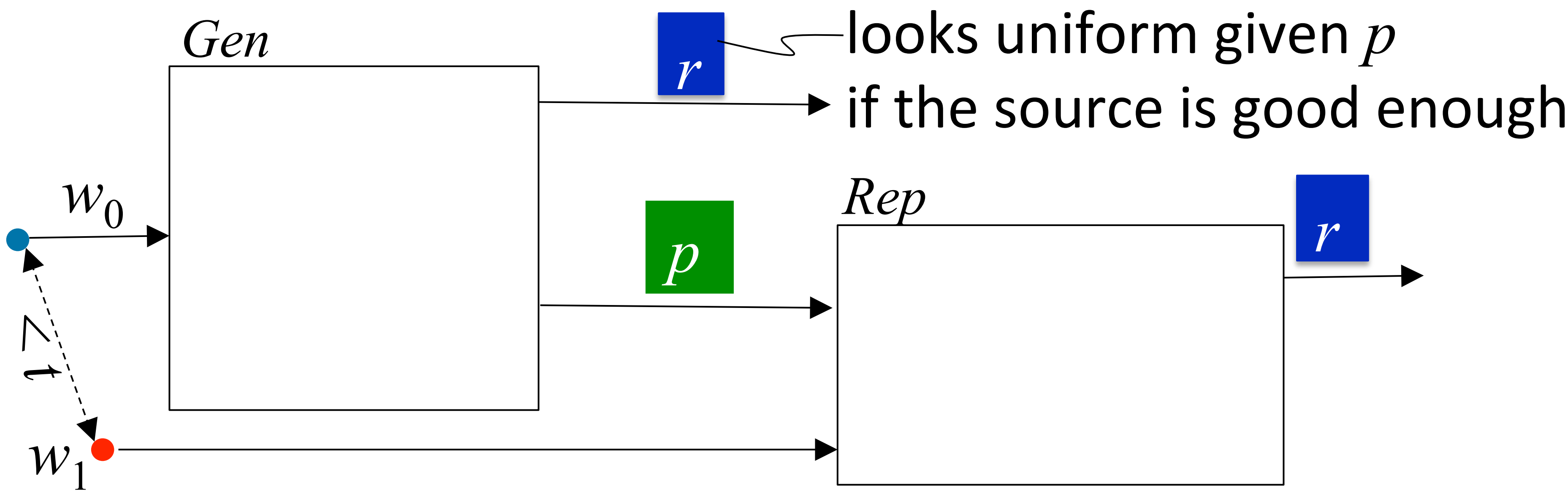
Biometrics

Notation (same as Part II)

- Enrollment algorithm *Gen* (Alice):
  Take a measurement w_0 from the source W.
  Use it to "lock up" a random output in a nonsecret value p.
- Subsequent algorithm *Rep* (Bob):
  give same output if $d(w_0, w_1) < t$

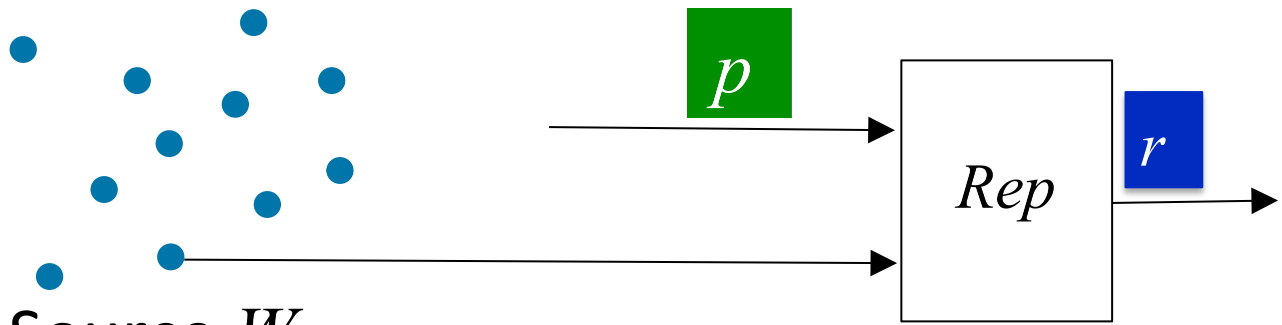


When are ~~fuzzy~~ extractors possible?



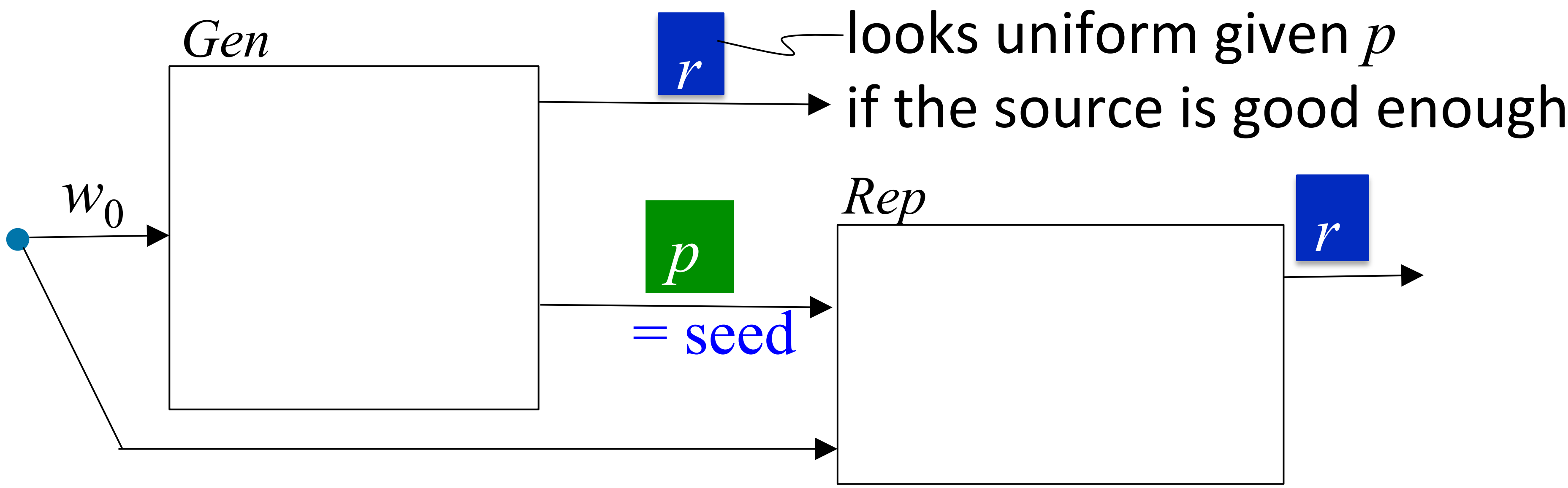
When are ~~fuzzy~~ extractors possible?

An adversary can always try a guess w_0 in W



Source W

Minimum requirement: every w_0 has low probability



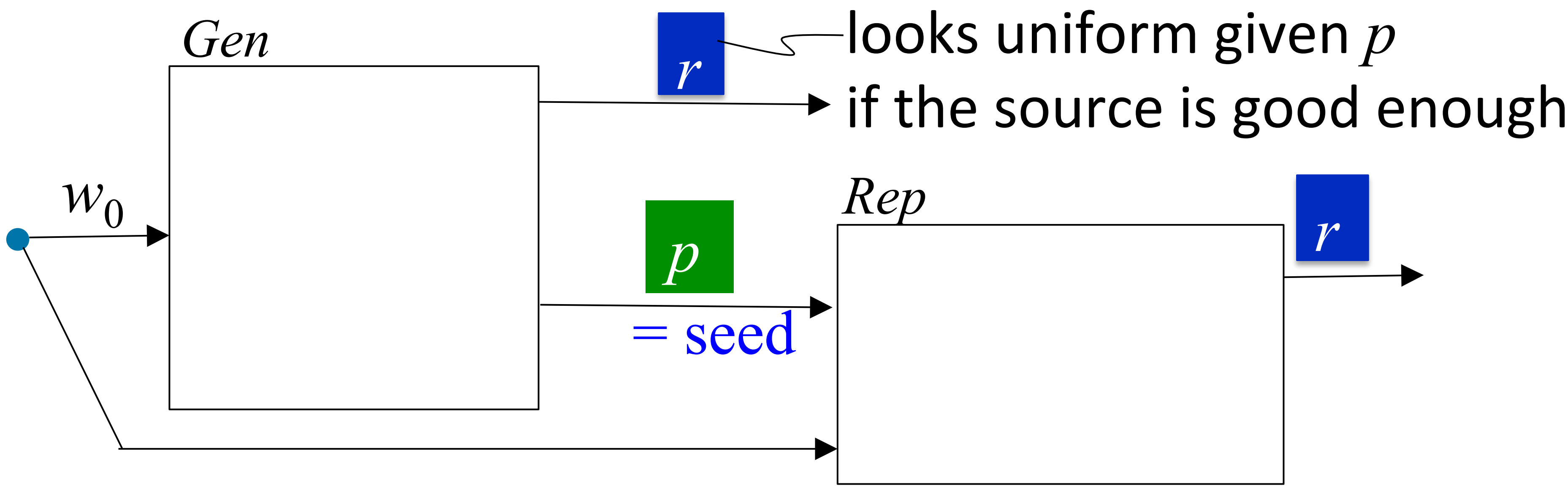
When are ~~fuzzy~~ extractors possible?

Define min-entropy: $H_\infty(W) = \min -\log \Pr[w]$

Necessary: $H_\infty(W) >$ security parameter

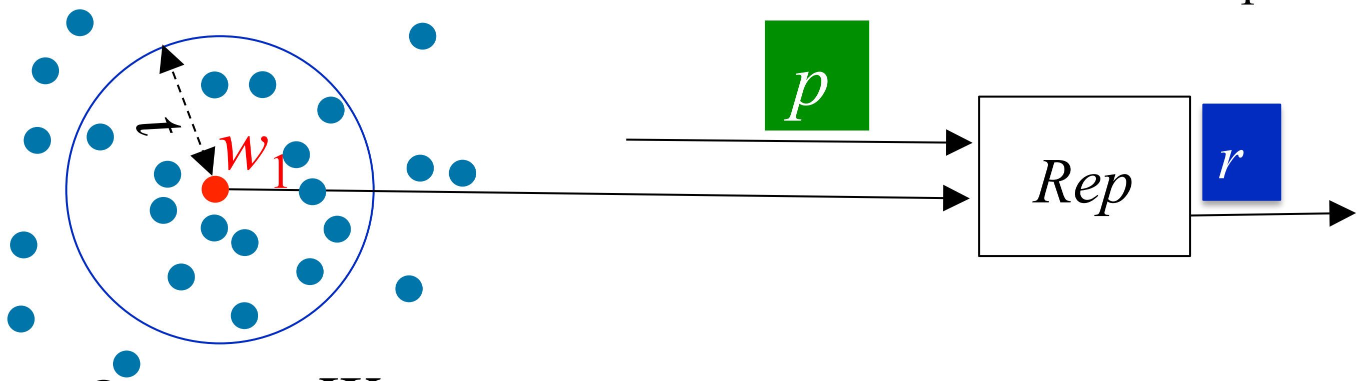
And sufficient by Leftover Hash Lemma

Minimum requirement: every w_0 has low probability



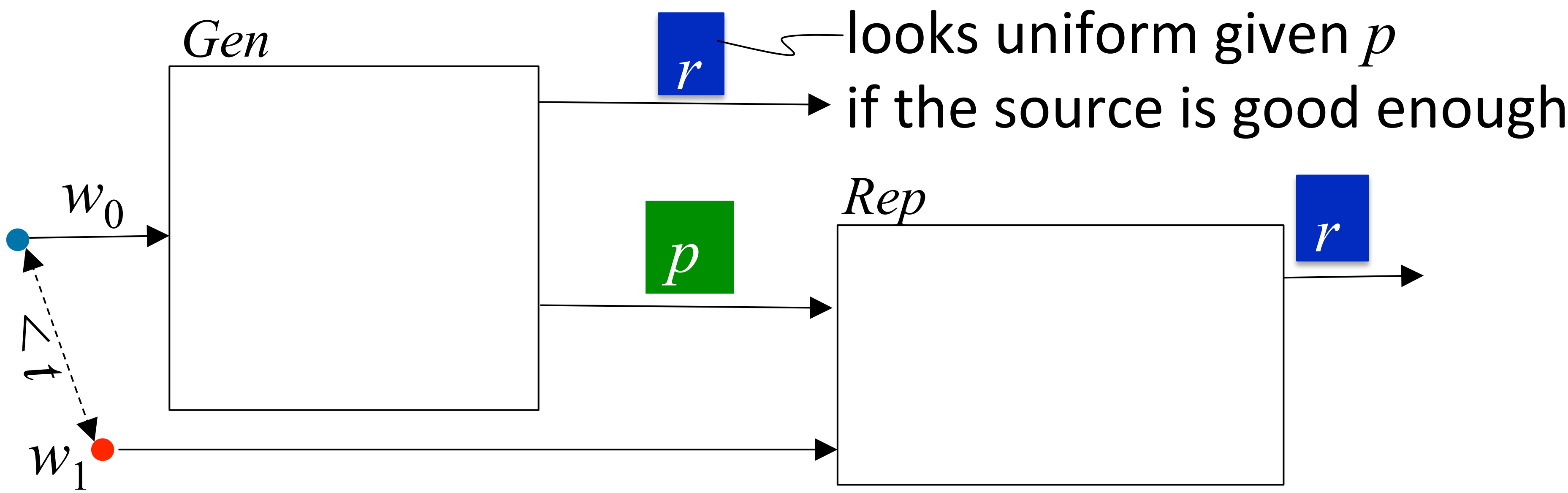
When are fuzzy extractors possible?

An adversary can always try a guess w_1 near many w_0



Source W

Minimum requirement: every B_t has low probability



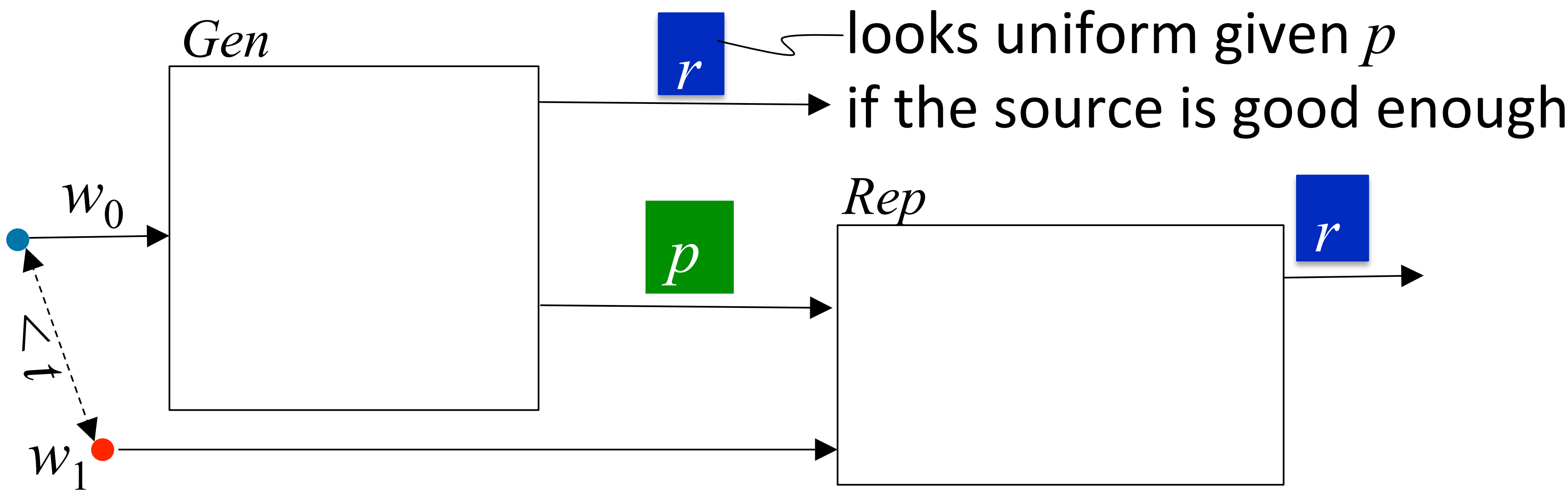
When are fuzzy extractors possible?

Define fuzzy min-entropy: $H_{\text{fuzz}}(W) = \min_{B_t} -\log \Sigma_{w \in B_t} \Pr[w]$

Necessary: $H_{\text{fuzz}}(W) >$ security parameter

Sufficient?

Minimum requirement: every B_t has low probability



$$H_{\text{fuzz}}(W) = \min_{B_t} -\log\text{ (mass inside } B_t)$$

Why bother with this new notion? Can't we use the old one?

$$H_{\text{fuzz}}(W) \geq H_\infty(W) - \log |B_t|$$

(since mass inside $B_t \leq \max \Pr[w] \cdot |B_t|$)

$$H_{\text{fuzz}}(W) = \min_{B_t} -\log\,(\text{mass inside } B_t)$$

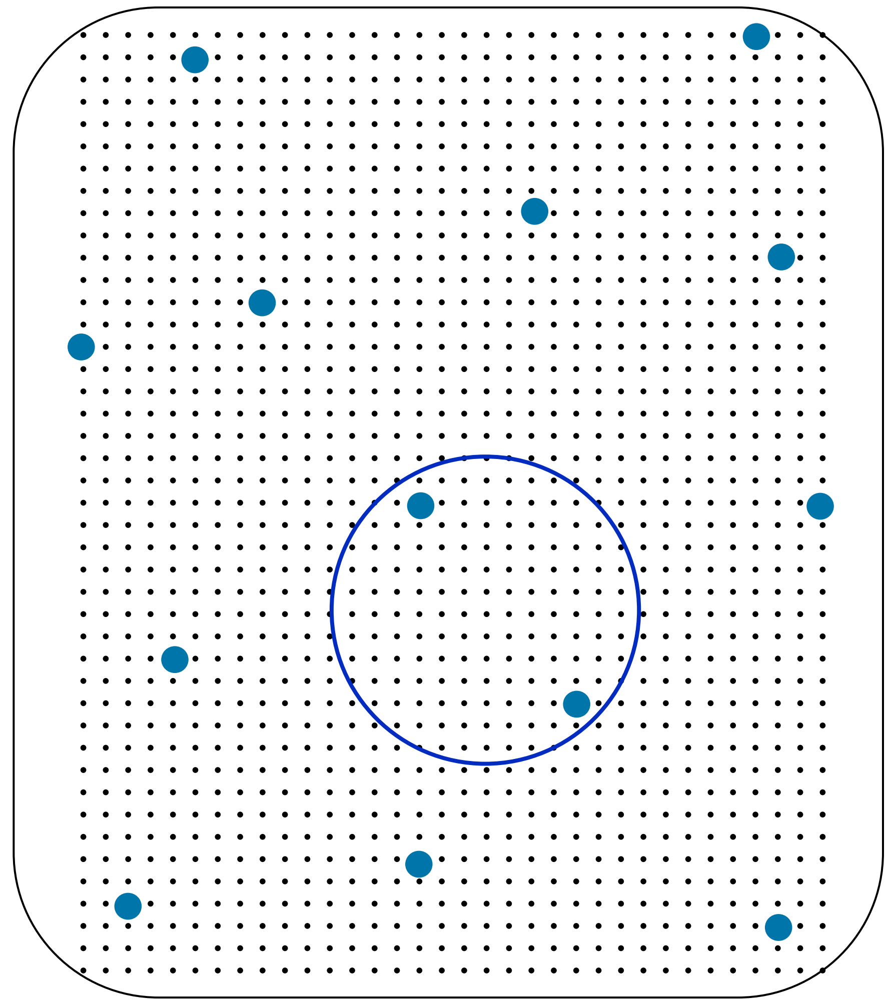
Why bother with this new notion? Can't we use the old one?

$$H_{\text{fuzz}}(W) \geq H_\infty(W) - \log|B_t|$$

Because there are distributions "with more errors than entropy"

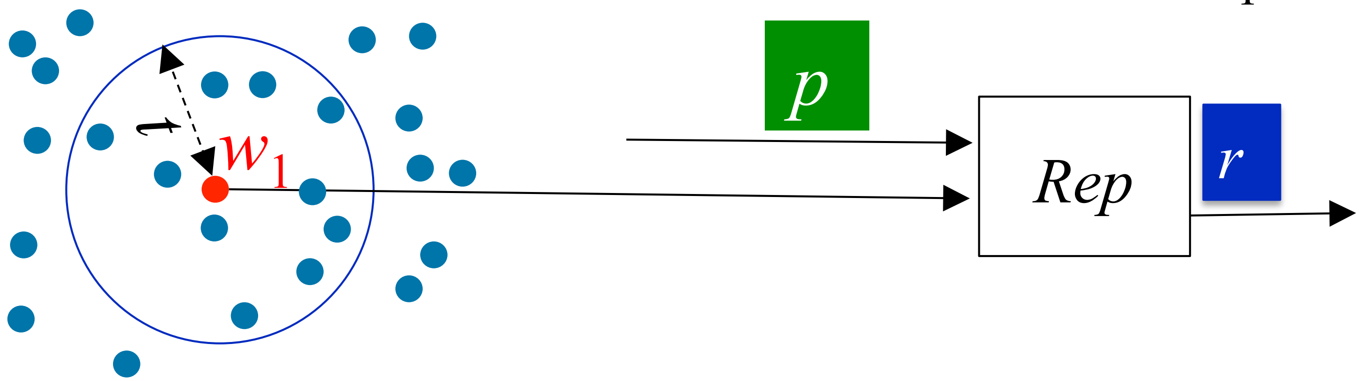
$$(\ \log|B_t| > H_\infty(W)\ )$$

But perhaps $H_{\text{fuzz}}(W) > 0$



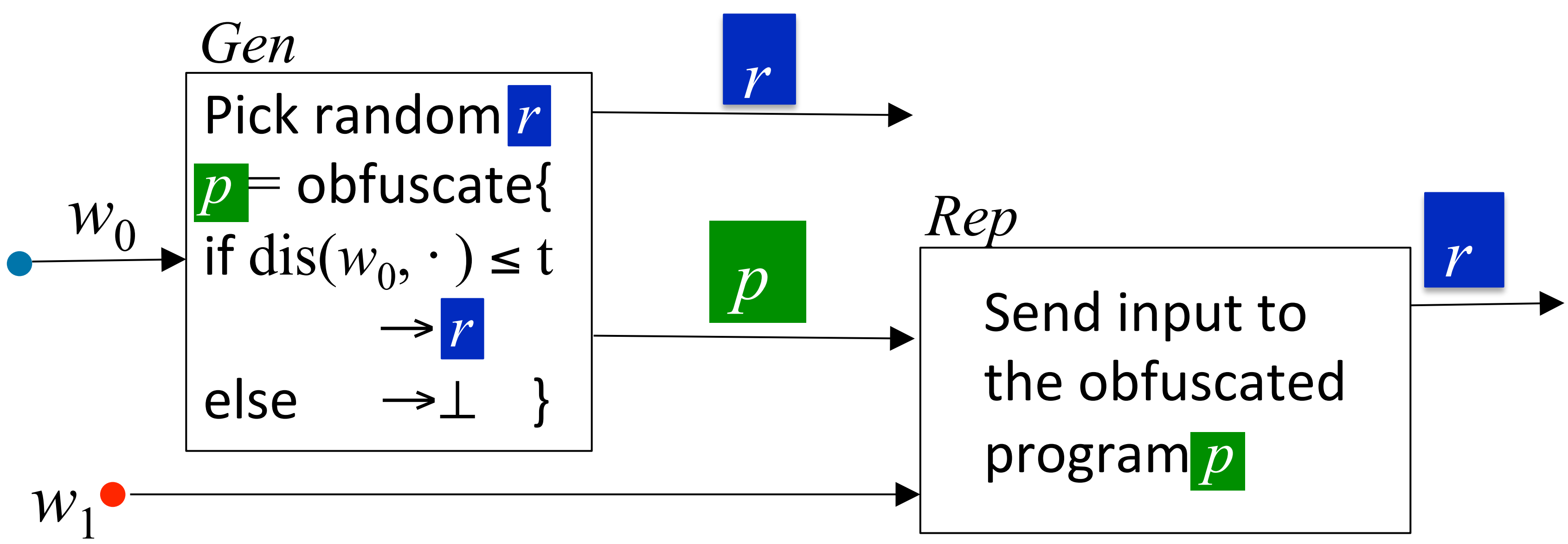
Why is H_{fuzz} the right notion?

An adversary can always try a guess w_1 near many w_0



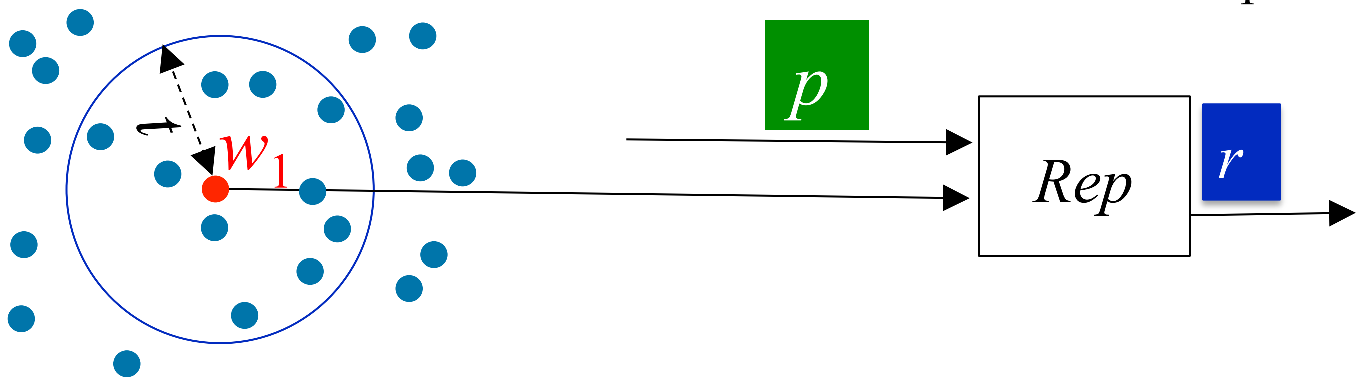
Q: can we make sure that's all the adversary can do?

A: yes, using obfuscation! [Bitansky Canetti Kalai Paneth 14]



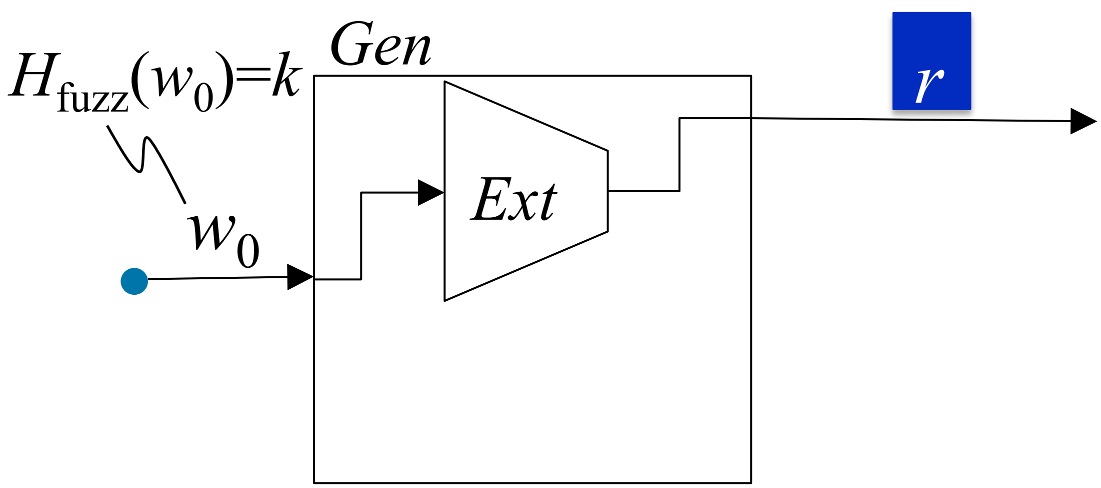
Why is H_{fuzz} the right notion?

An adversary can always try a guess w_1 near many w_0

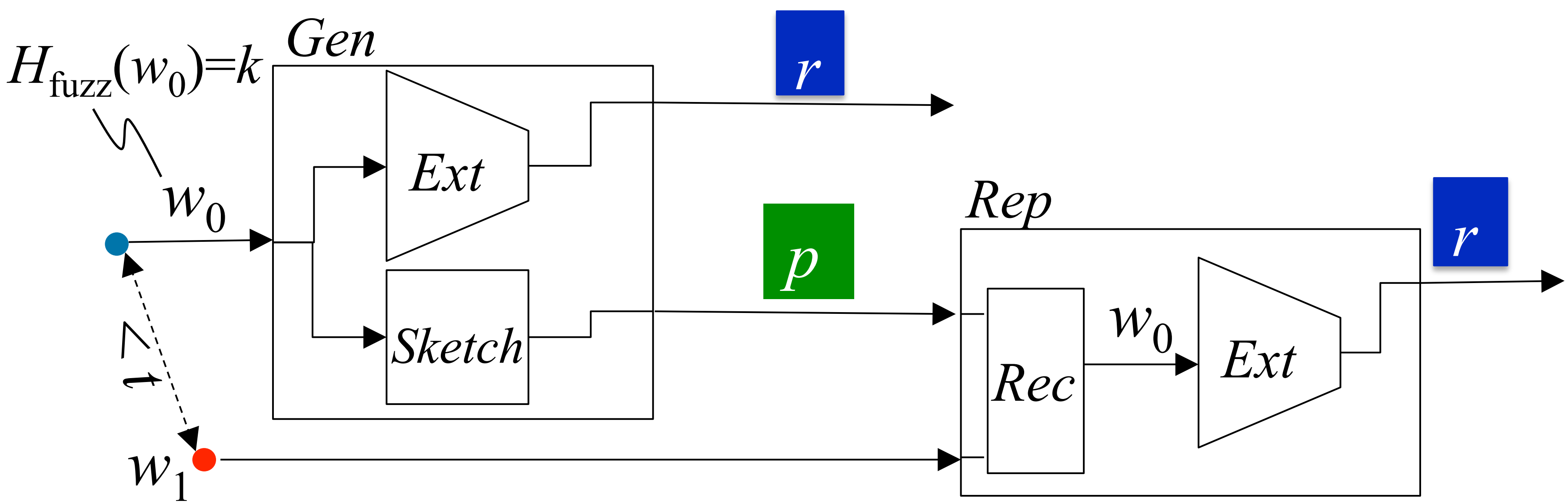


- H_{fuzz} is necessary
- H_{fuzz} is sufficient against computational adversaries
- What about information-theoretic adversaries?

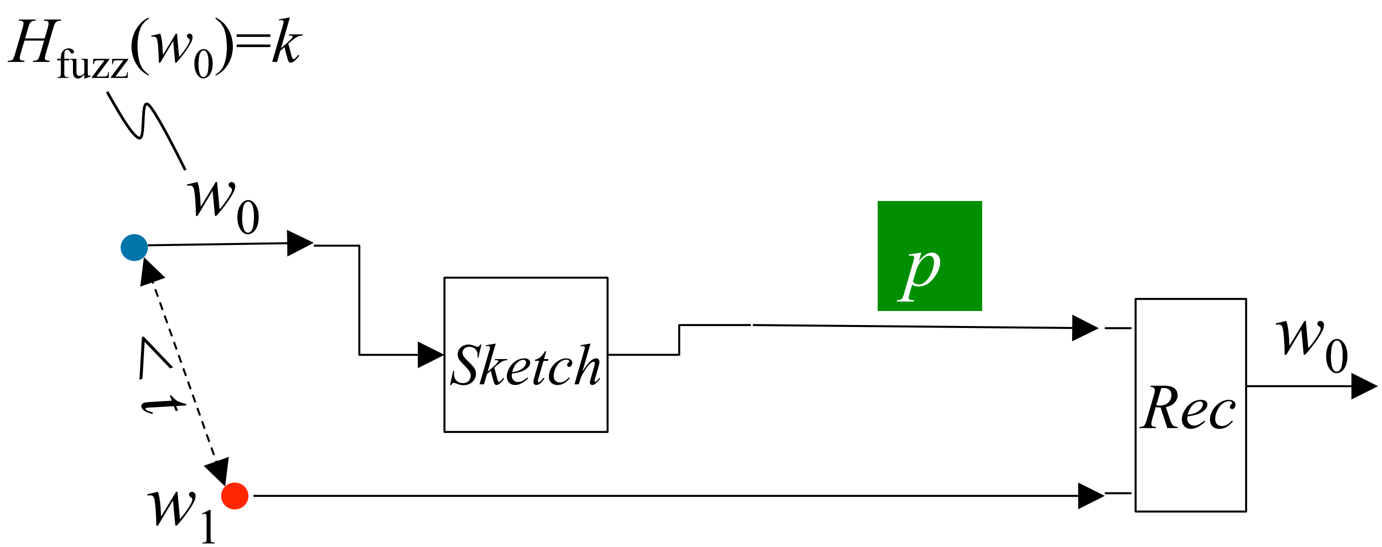
Claim: we can extract $\sim H_{\text{fuzz}}$ bits



Claim: we can extract ~H_{fuzz} bits



Claim: we can extract ~H_{fuzz} bits

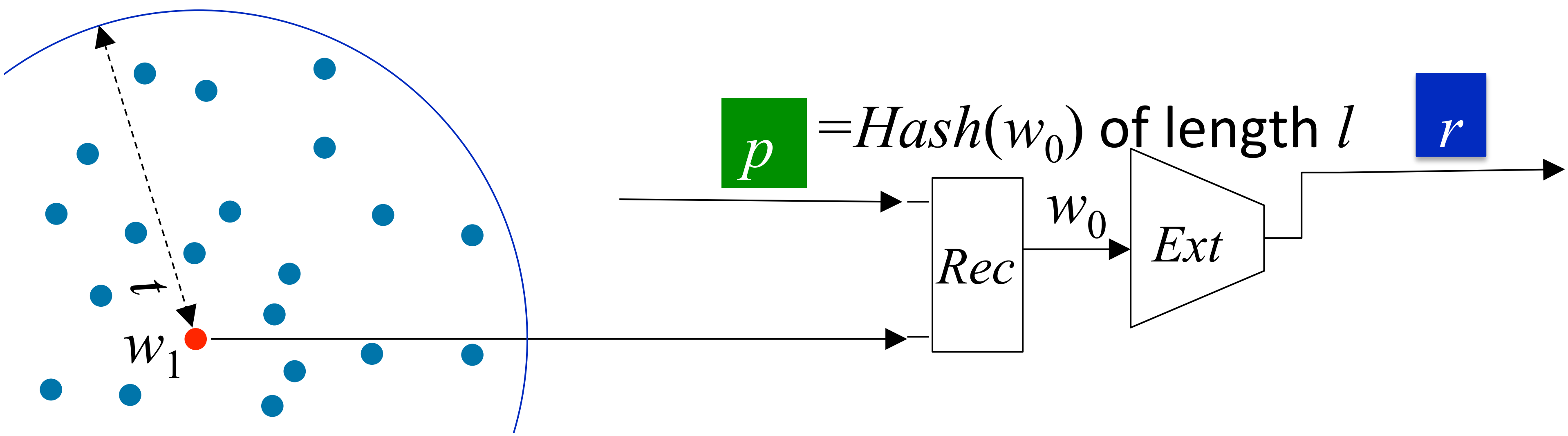


Claim: we can extract ~H_{fuzz} bits

- p needs to disambiguate the possible points in $B_t(w_1)$
- suppose all w_0 are equiprobable
- $l \approx \log(\max \#\, w_0 \text{ in } B_t) = \log \dfrac{(\text{max mass in } B_t)}{\Pr[w_0]}$

$$= H_\infty(W) - H_{\text{fuzz}}(W)$$

- $|r| \approx H_\infty(W) - l = H_{\text{fuzz}}(W)$



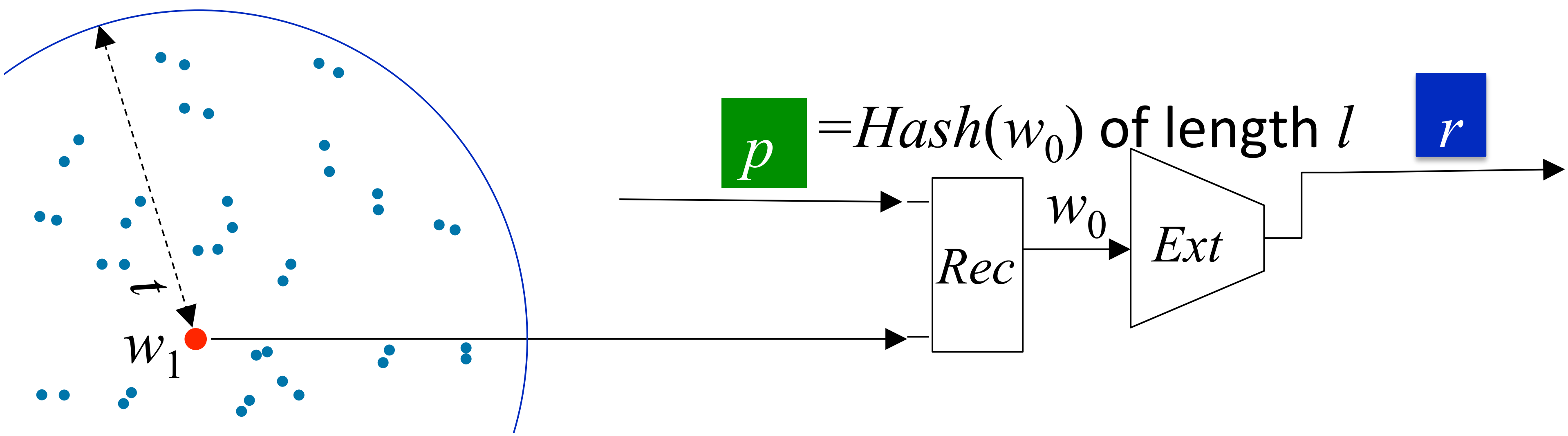
Claim: we can extract ~H_{fuzz} bits

- p needs to disambiguate the possible points in $B_t(w_1)$
- suppose all w_0 are equiprobable
- $l \approx \log(\max \# \, w_0 \text{ in } B_t) = \log \dfrac{(\text{max mass in } B_t)}{\Pr[w_0]}$

$$= H_\infty(W) - H_{\text{fuzz}}(W)$$

- $|r| \approx H_\infty(W) - l = H_{\text{fuzz}}(W)$

stays grows grows

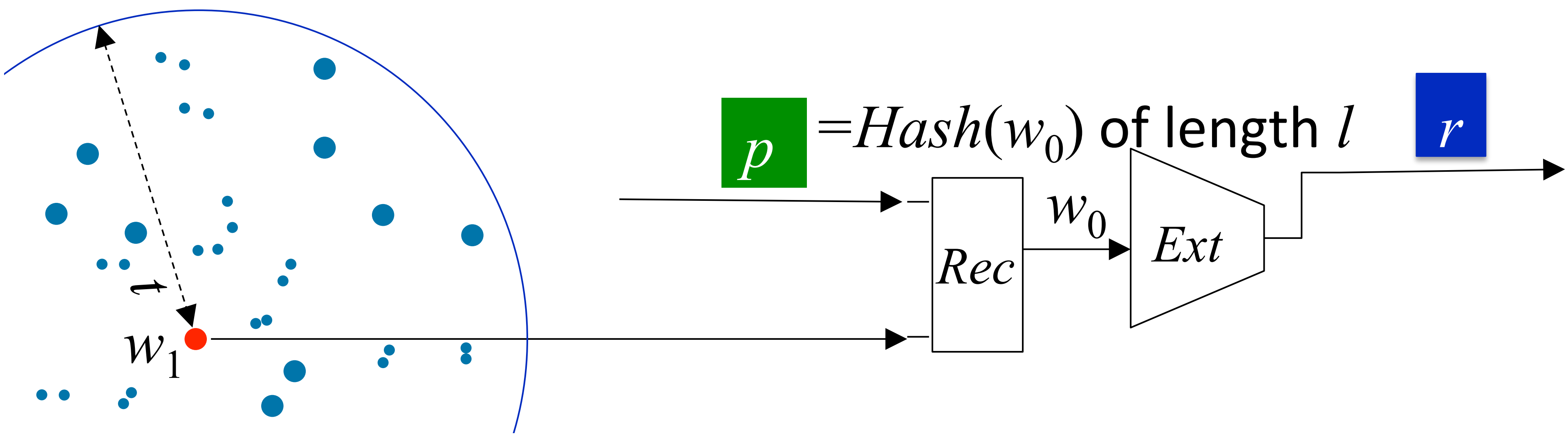


Claim: we can extract ~H_{fuzz} bits

- p needs to disambiguate the possible points in $B_t(w_1)$
- ~~suppose all w_0 are equiprobable~~
- $l \approx \log(\max \# w_0 \text{ in } B_t) \neq \log \frac{(\text{max mass in } B_t)}{\Pr[w_0]}$

$$> H_\infty(W) - H_{\text{fuzz}}(W)$$

- $|r| \approx H_\infty(W) - l = H_{\text{fuzz}}(W)$

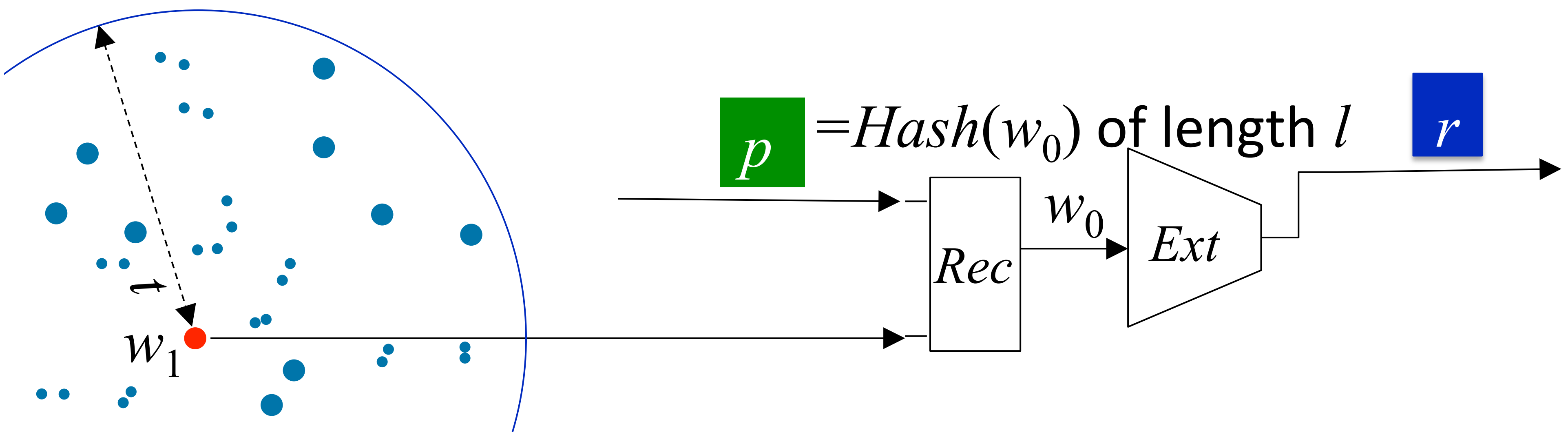


Claim: we can extract ~H_{fuzz} bits

- p needs to disambiguate the possible points in $B_t(w_1)$
- ~~suppose all w_0 are equiprobable~~
- $l \approx$ ~~$\log$ (max # w_0 in B_t)~~

variable, grows with $\log 1/\Pr[w_0]$
reveals $\lfloor \log 1/\Pr[w_0] \rfloor$, a value between 1 and $\log |W|$

- $|r| \approx$ ~~$H_\infty(W) - l = H_{\text{fuzz}}(W)$~~



Claim: we can extract ~H_{fuzz} bits

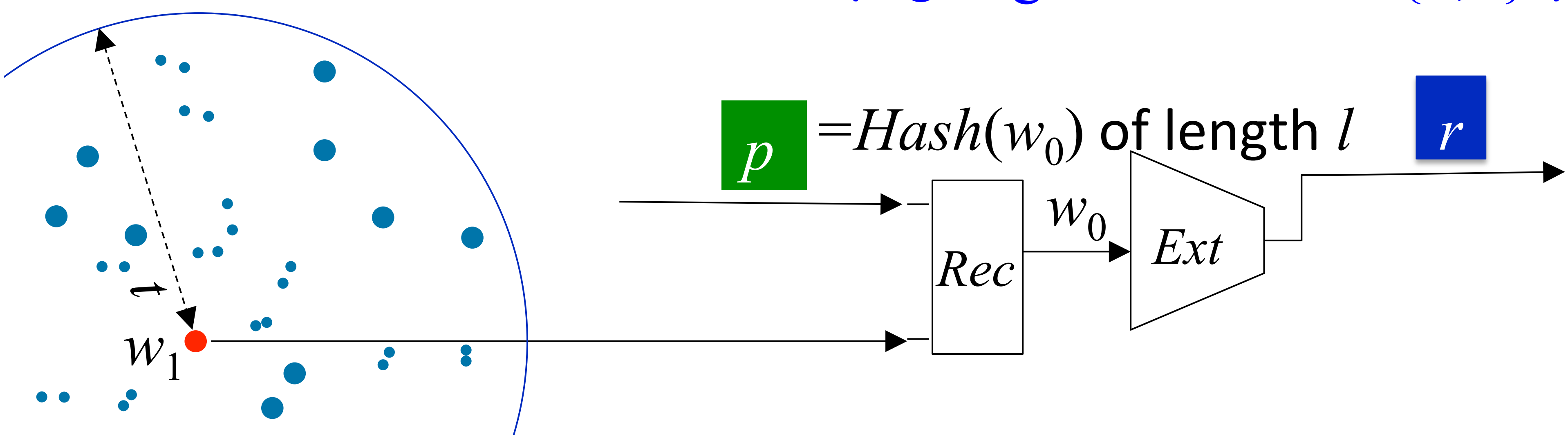
- p needs to disambiguate the possible points in $B_t(w_1)$
- ~~suppose all w_0 are equiprobable~~
- $l \approx$ ~~$\log(\max \# w_0 \text{ in } B_t)$~~

variable, grows with $\log 1/\Pr[w_0]$

reveals $\lfloor \log 1/\Pr[w_0] \rfloor$, a value between 1 and $\log |W|$

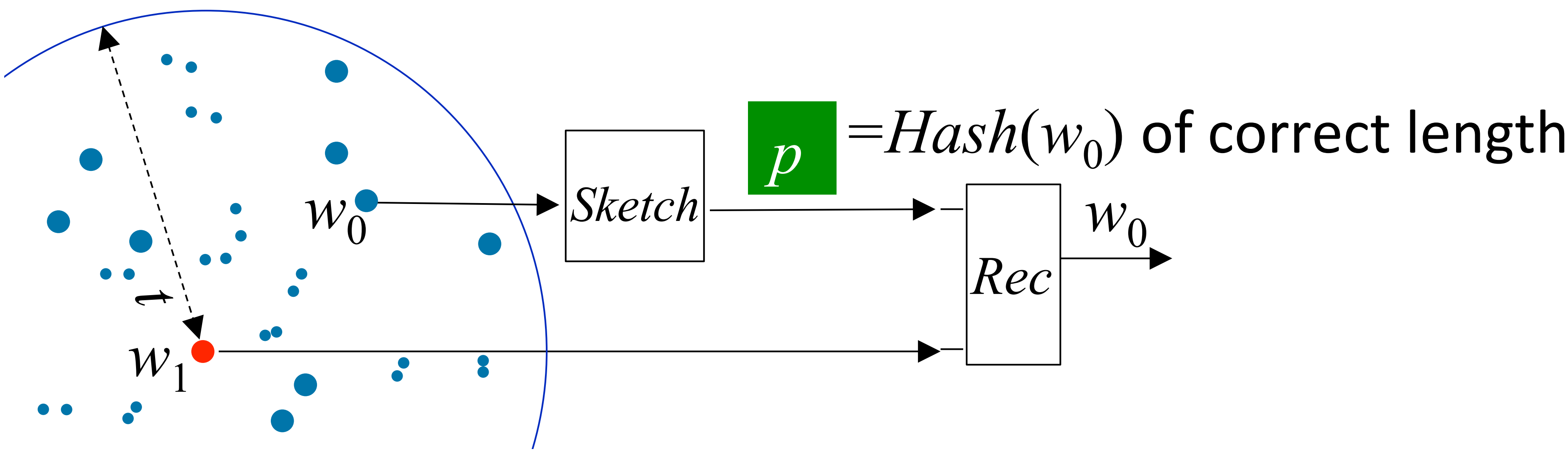
- $|r| \approx$ ~~$H_\infty(W) - l$~~ $= H_{\text{fuzz}}(W) - \log\log|W|$

(e.g. $\log n$ if W is over $\{0,1\}^n$)



Claim: we can extract ~H_{fuzz} bits

- Feasibility-only result!
- *Sketch* needs to know $\log \Pr[w_0]$
- *Rec* is not efficient in general.
- *Rec* needs to know W (to know candidate w_0 values)



What if we don't know W ?

Common design goal: one construction for family of sources
(e.g., all sources of a given H_{fuzz})

Recall:

Family of sources with
poor quality randomness $\Leftrightarrow$ randomness
(maybe uniform)

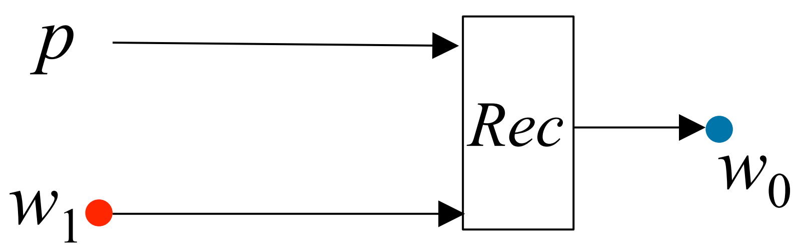
leakage
unknown
to Gen, Rep

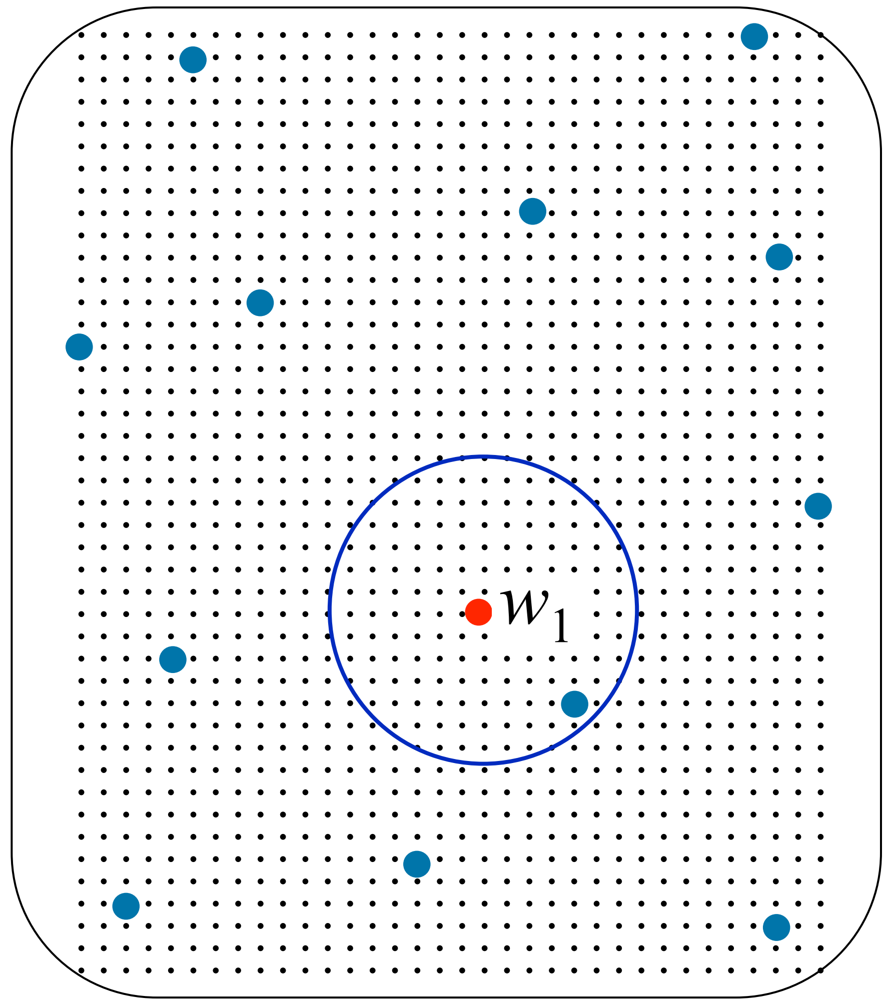
Eve

(e.g, Eve gets $z = Gw$ for random linear G)

What if we don't know W ?

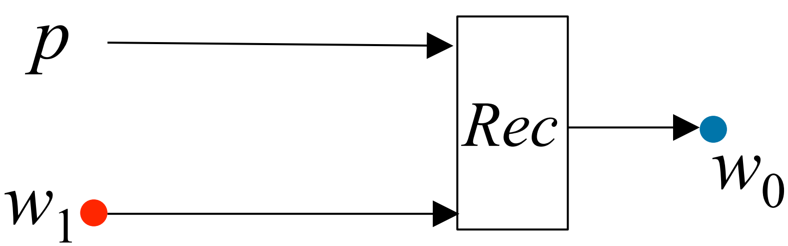
Common design goal: one construction for family of sources (e.g., all sources of a given H_{fuzz})

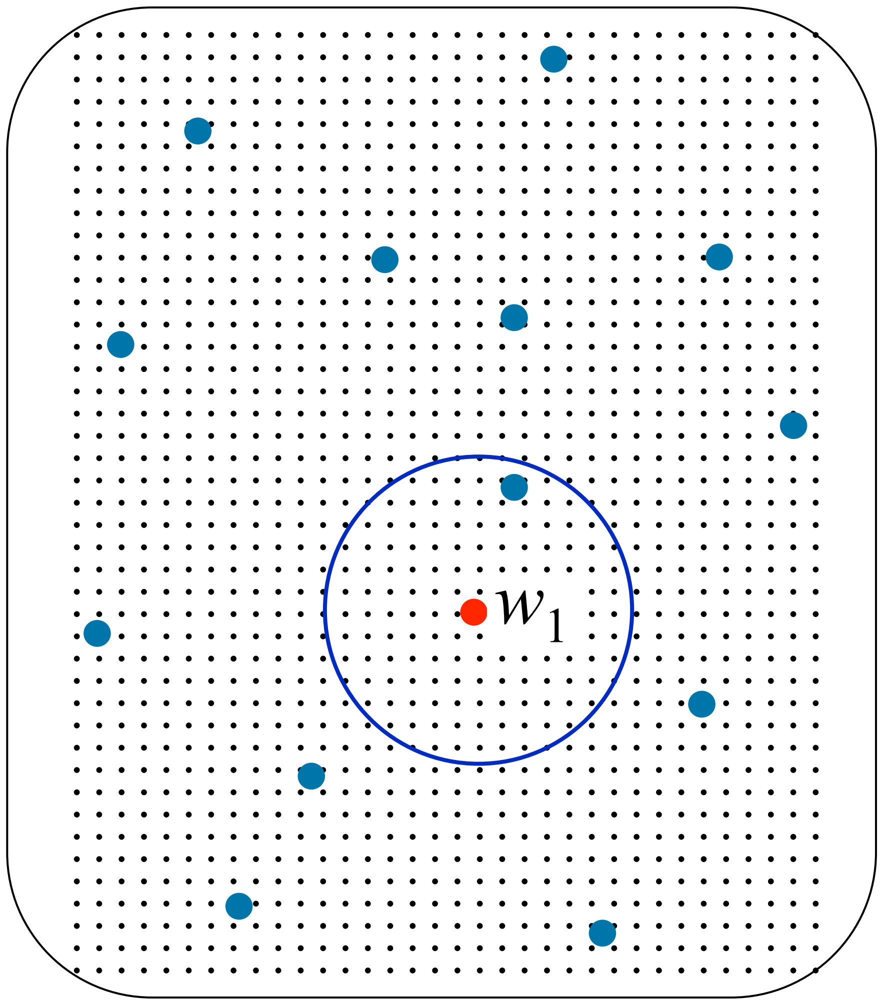




What if we don't know W ?

Common design goal: one construction for family of sources (e.g., all sources of a given H_{fuzz})

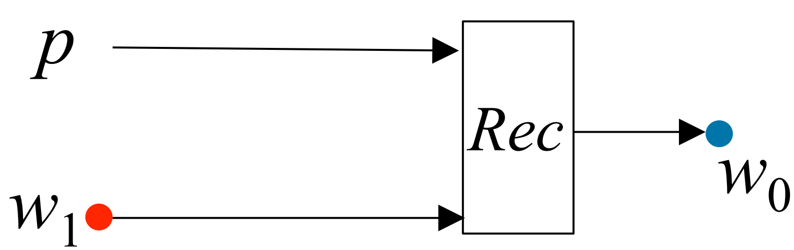


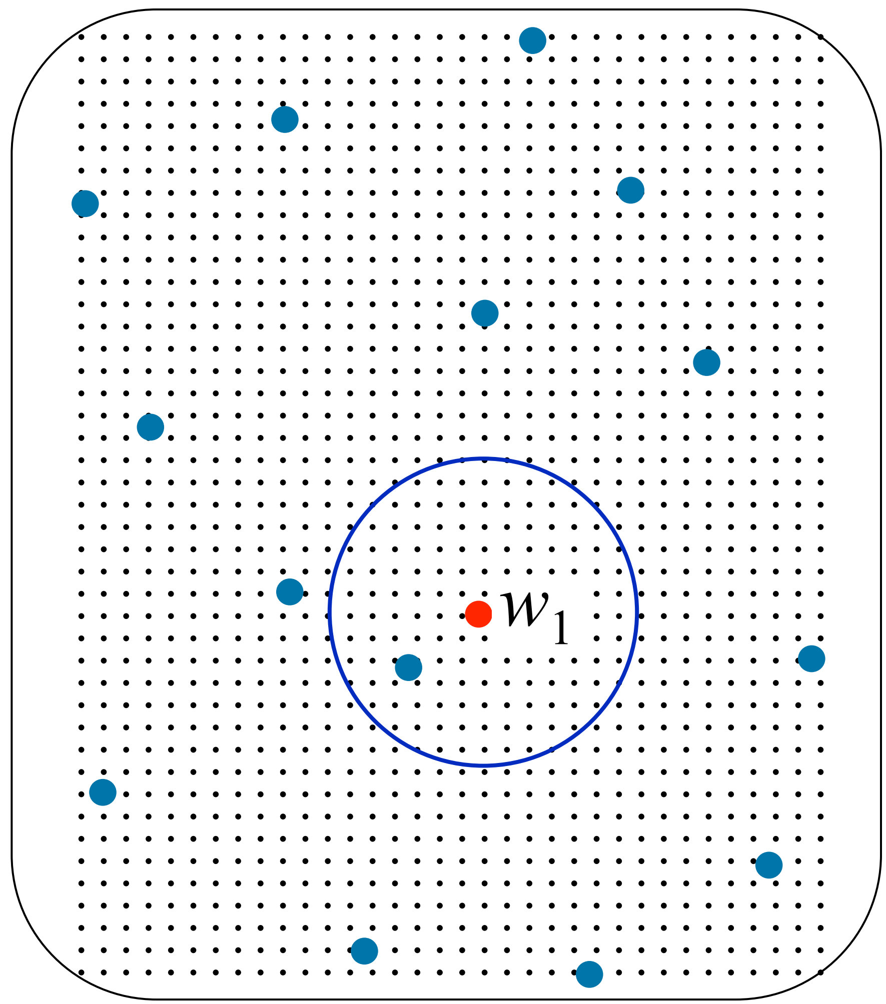


What if we don't know W ?

Common design goal: one construction for family of sources (e.g., all sources of a given H_{fuzz})

Rec needs to recover from w_1 regardless of which W the original w_0 came from





What if we don't know W ?

Common design goal: one construction for family of sources (e.g., all sources of a given H_{fuzz})

Rec needs to recover from w_1 regardless of which W the original w_0 came from

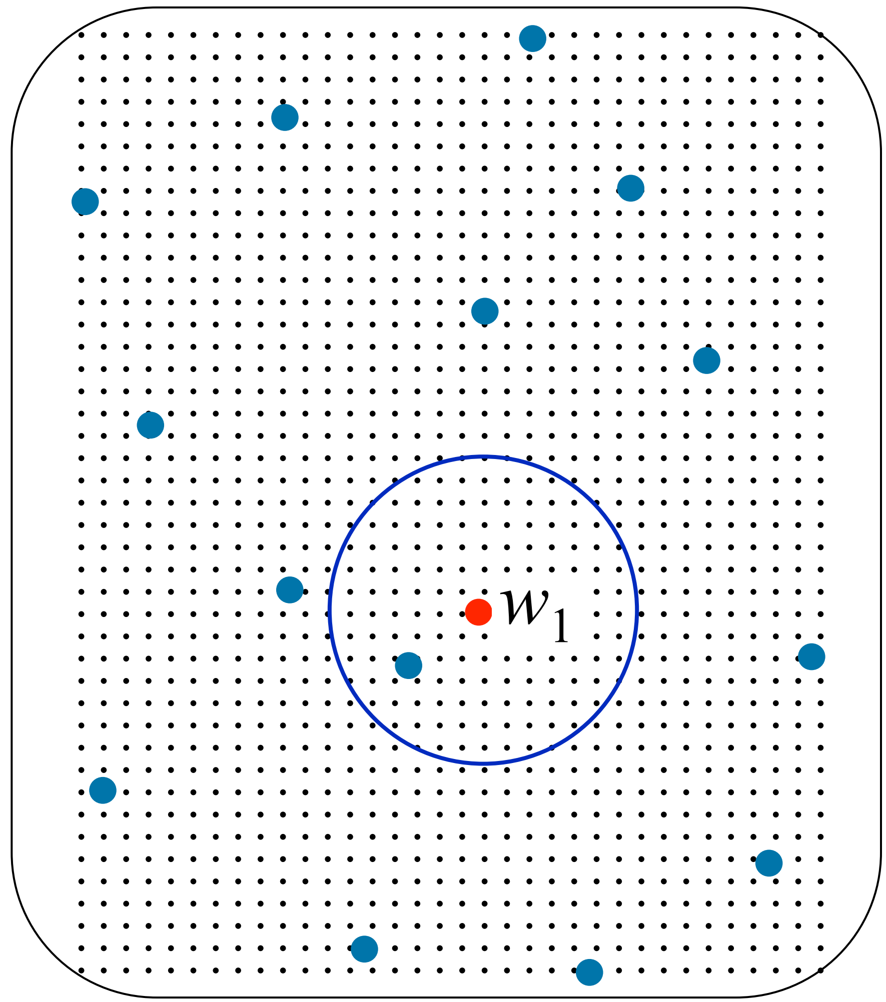
$\Downarrow$

p has a lot of information about w_0

$\Downarrow$

combined with Eve's $z = Gw_0$ it's too much

(because p is generated without knowledge of G)



What if we don't know W ?

Common design goal: one construction for family of sources (e.g., all sources of a given H_{fuzz})

Rec needs to recover from w_1
regardless of which W
the original w_0 came from
$\Downarrow$
p has a lot of
information about w_0
$\Downarrow$
combined with Eve's $z = Gw_0$
it's too much
(because p is generated
without knowledge of G)

<u>Theorem</u>: ∃ a family $\{W\}$ with superlog $H_\infty(W)$ s.t. any $Sketch$, Rec that corrects 4 Hamming errors with prob. > 1/4 will have $H_{\text{fuzz}}(W \mid p)$ < 2

What if we don't know W ?

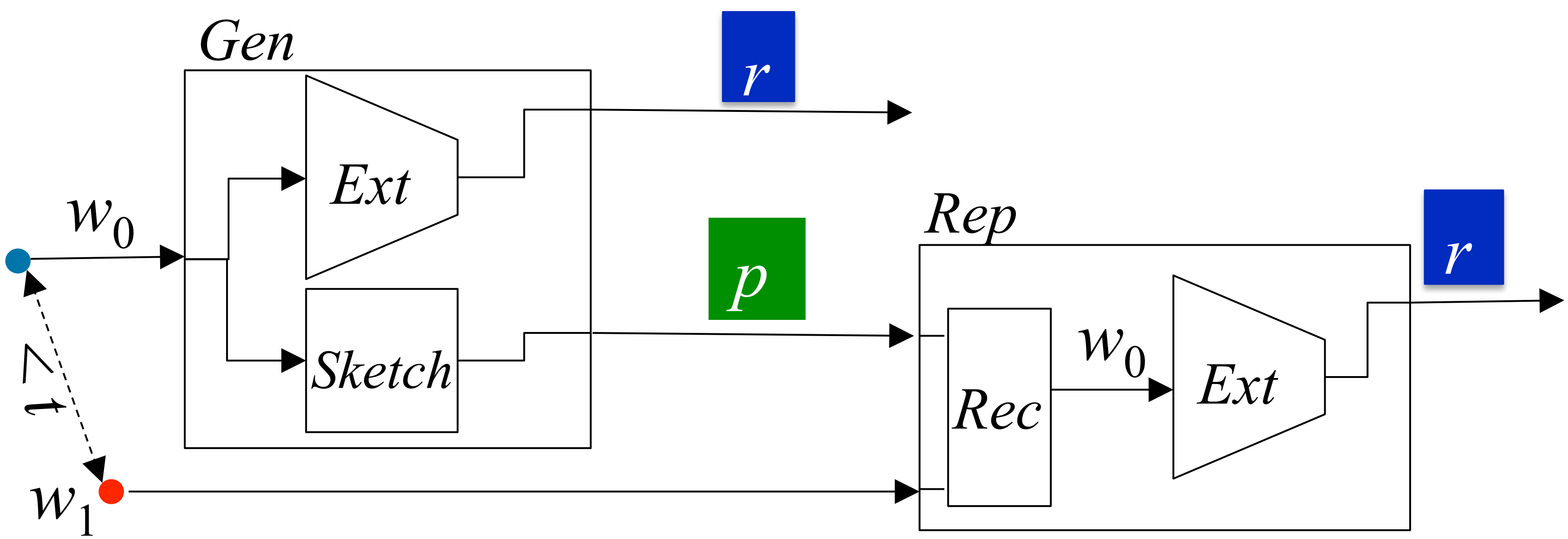
Common design goal: one construction for family of sources (e.g., all sources of a given H_{fuzz})

But we don't have to recover w_0!

What if we don't know W ?

Common design goal: one construction for family of sources (e.g., all sources of a given H_{fuzz})

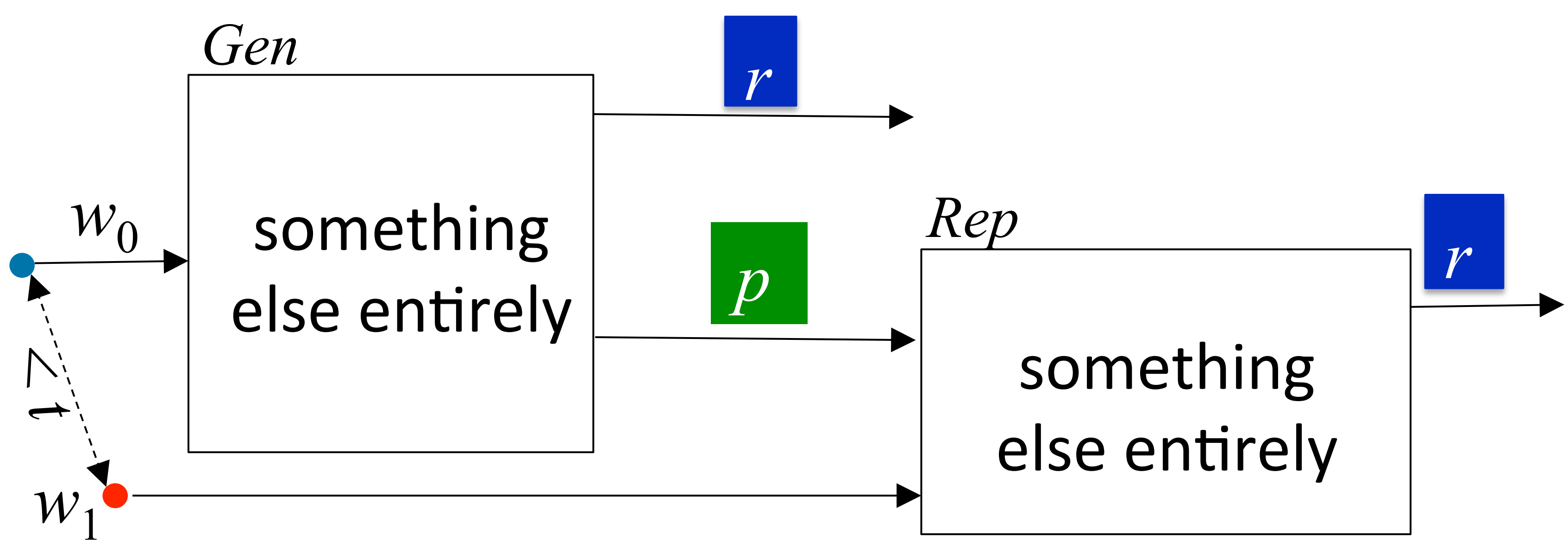
But we don't have to recover w_0!



What if we don't know W ?

Common design goal: one construction for family of sources (e.g., all sources of a given H_{fuzz})

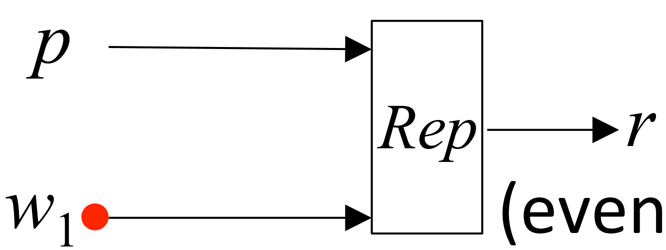
But we don't have to recover w_0!



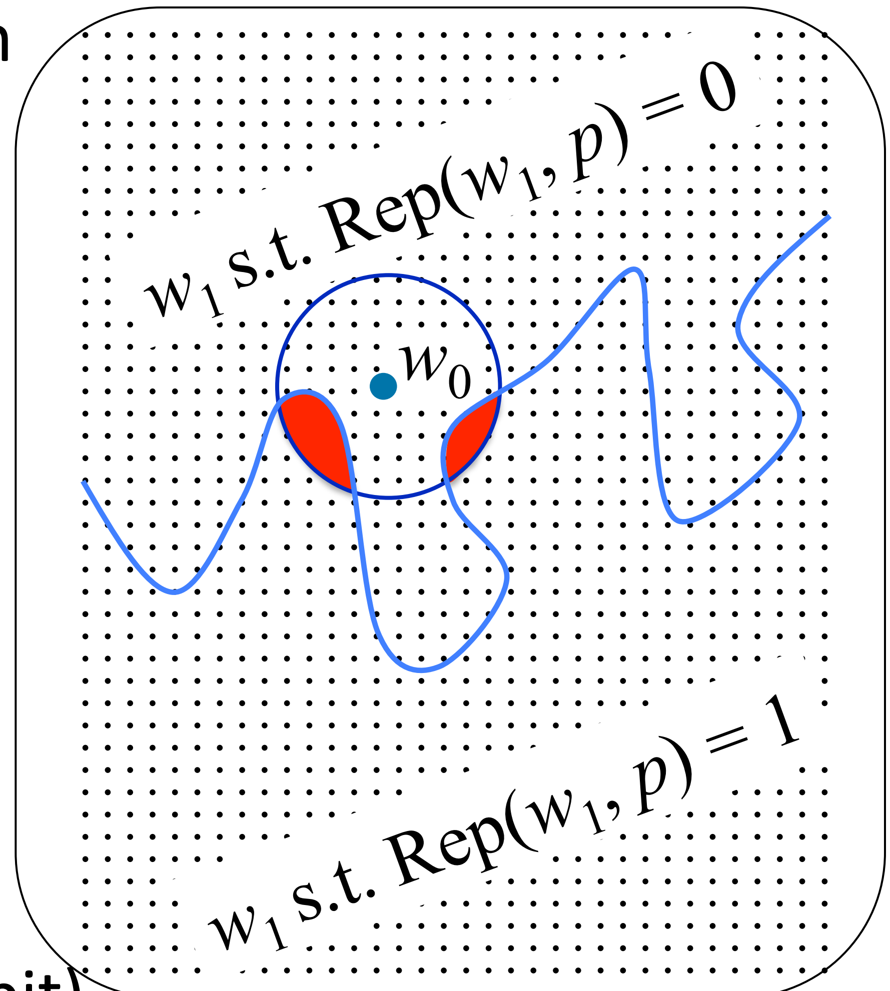
What if we don't know W ?

Common design goal: one construction for family of sources (e.g., all sources of a given H_{fuzz})

- Given p, this partition is known
- Nothing near the boundary can have been w_0 (else Rep wouldn't be 100% correct)



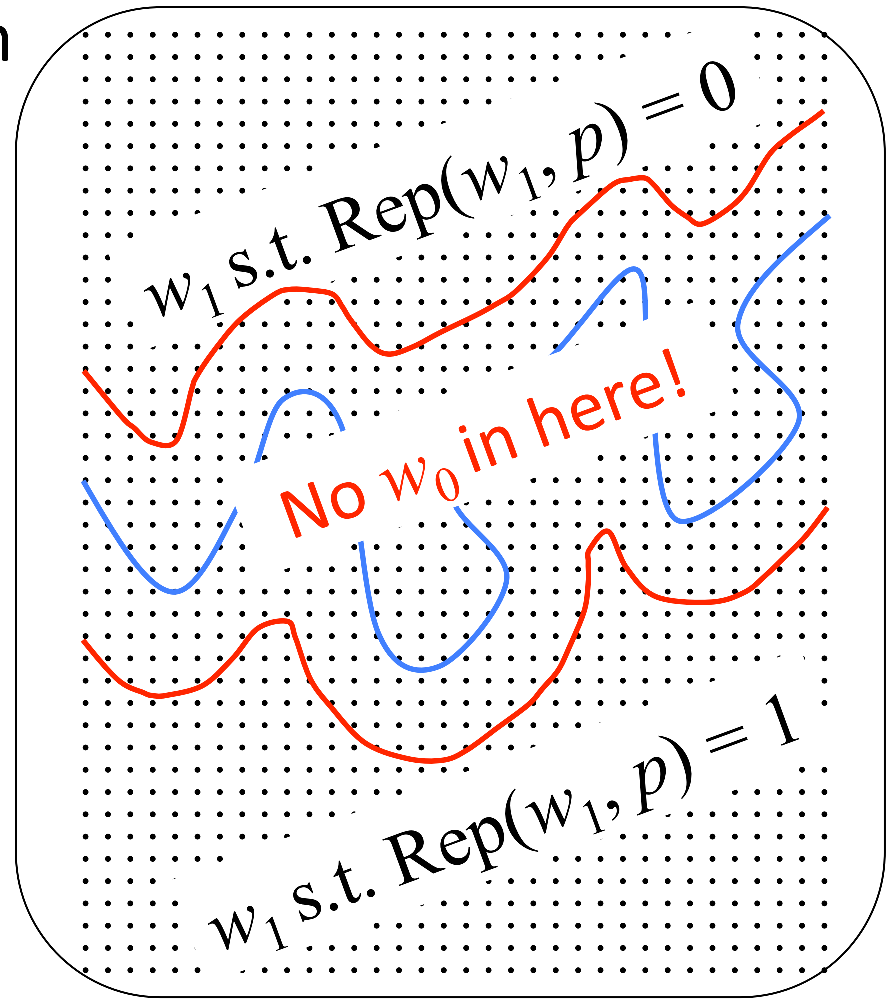
(even just one bit)



What if we don't know W ?

Common design goal: one construction for family of sources (e.g., all sources of a given H_{fuzz})

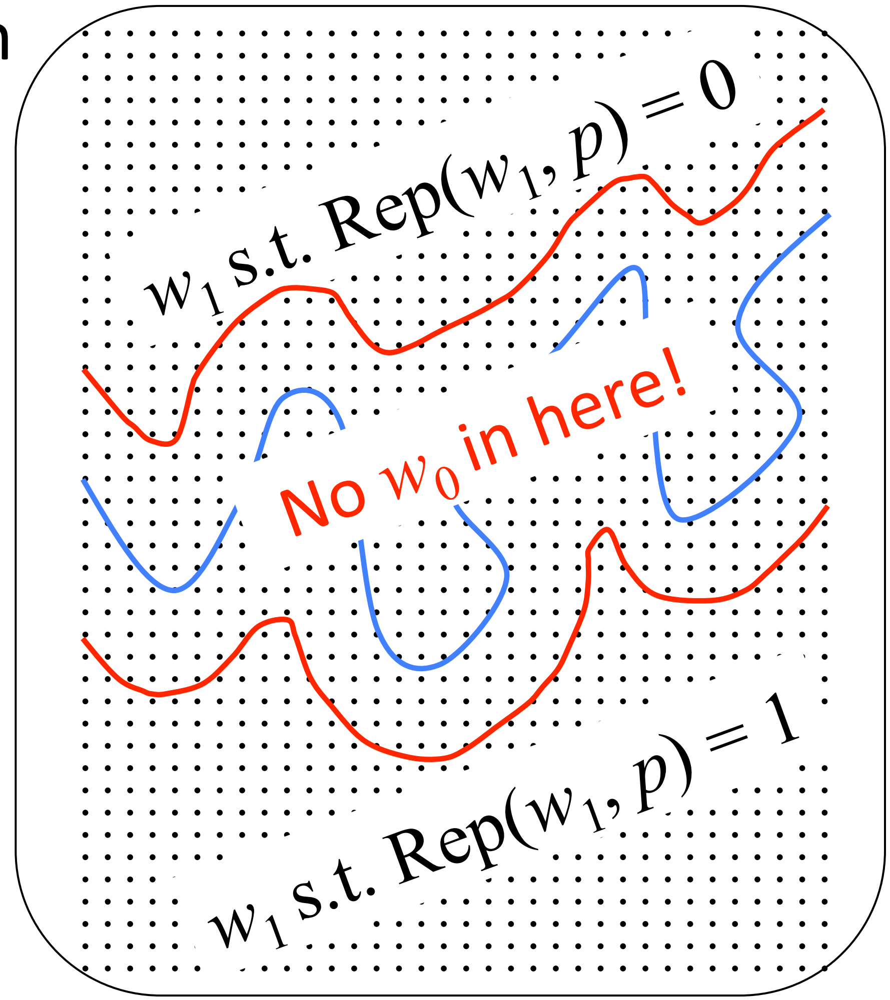
- Given p, this partition is known
- Nothing near the boundary can have been w_0 (else Rep wouldn't be 100% correct)



What if we don't know W ?

Common design goal: one construction for family of sources (e.g., all sources of a given H_{fuzz})

- Given p, this partition is known
- Nothing near the boundary can have been w_0 (else Rep wouldn't be 100% correct)
- Leaves little uncertainty about w_0 (high-dimensions $\Rightarrow$ everything near boundary)
- Combined with Eve's $z = Gw_0$ no uncertainty left (because p is generated without knowledge of G)



What if we don't know W ?

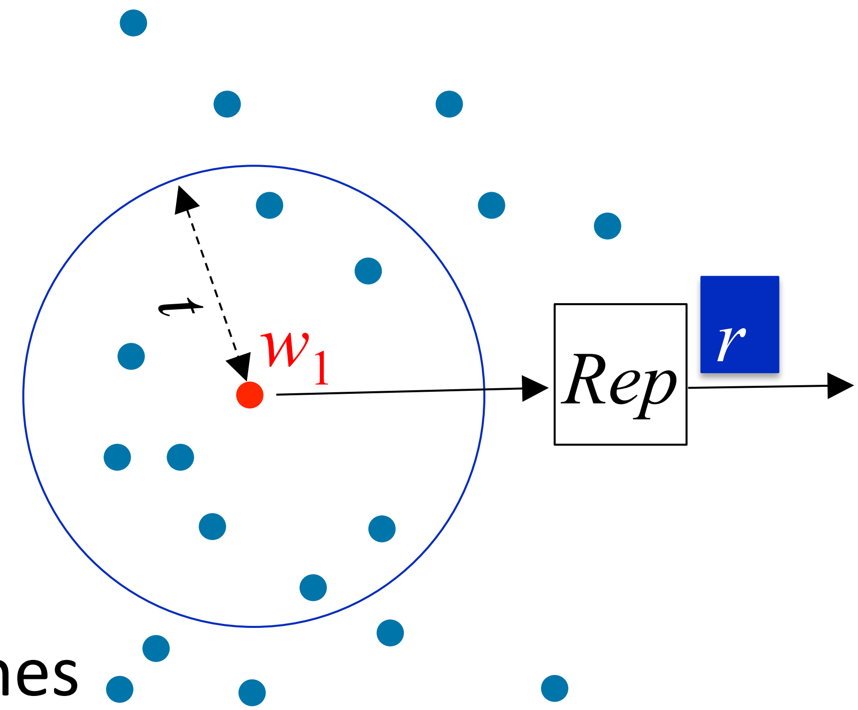
Common design goal: one construction for family of sources (e.g., all sources of a given H_{fuzz})

- Given p, this partition is known
- Nothing near the boundary can have been w_0 (else Rep wouldn't be 100% correct)
- Leaves little uncertainty about w_0 (high-dimensions $\Rightarrow$ everything near boundary)
- Combined with Eve's $z = Gw_0$ no uncertainty left (because p is generated without knowledge of G)

Theorem:
$\exists$ a family $\{W\}$ over $\{0,1\}^n$ with superlog $H_\infty(W)$ s.t. any Gen, Rep that handles at least $n^{1/2}\log n$ errors can't output even 2 bits (if Rep is 100% correct)

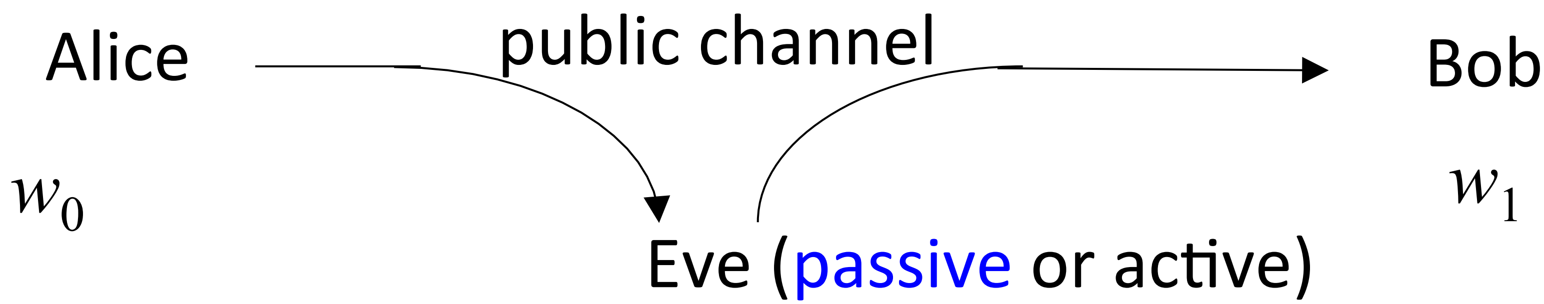
Summary

- Natural and necessary notion: $H_{\text{fuzz}}(W) = \log\,(1/\max \text{wt}(B_t))$
- Sufficient under computational assumptions
- Sufficient if the distribution is known
- In case of distributional uncertainty:
  - Insufficient for secure sketches
  - Insufficient for perfectly correct fuzzy extractors in high dimensions



- Open: removing perfect correctness limitation

What I just showed

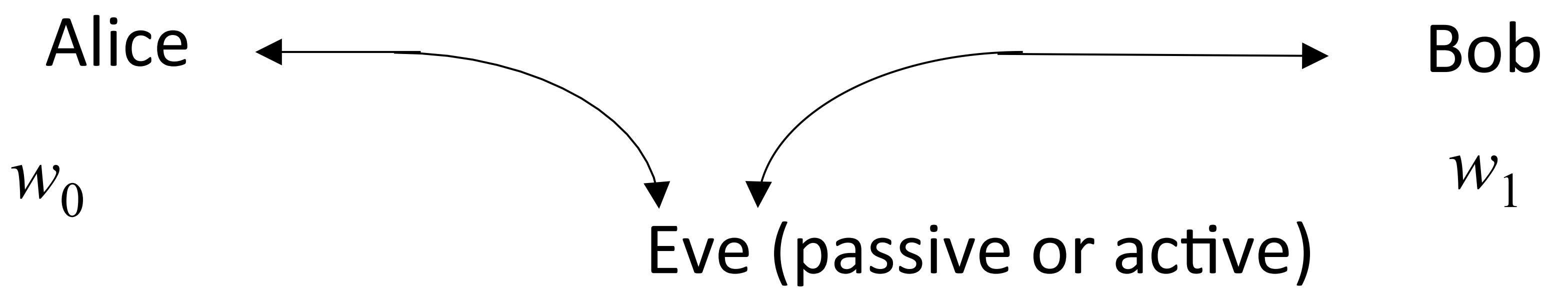


Secrets can come from nature, but we need to tame them

Research Directions:

- Finding the right notion of security
- Minimizing assumptions about adversarial knowledge
- Broadening sources of secrets
- Understanding fundamental bounds on what's feasible
  - Finding the right notion of input entropy
- Making it all efficient

Lots more to be done!



Secrets can come from nature, but we need to tame them

Research Directions:

- Finding the right notion of security
- Minimizing assumptions about adversarial knowledge
- Broadening sources of secrets
- Understanding fundamental bounds on what's feasible
  - Finding the right notion of input entropy
- Making it all efficient

Aaron Wyner c. 1975
(courtesy of Adi Wyner)

THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 54 | October 1975 | Number 8

The Wire-Tap Channel

By A. D. WYNER

(Manuscript received May 9, 1975)

We consider the situation in which digital data is to be reliably transmitted over a discrete, memoryless channel (DMC) that is subjected to a wire-tap at the receiver. We assume that the wire-tapper views the channel output via a second DMC. Encoding by the transmitter and decoding by the