Information-Theoretic
Key Agreement
from
Close Secrets

Leonid Reyzin

January 5, 2018
IISc
Close Secrets

Alice

$w_0$

assume these are “close” and partially secret

Bob

$w_1$
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

\[ w_0 \hspace{1cm} w_1 \]
basic paradigm
basic paradigm

Alice

$w_0$

Bob

$w_1$

Eve
basic paradigm: passive adversary

Conversation about their differences
also known as information reconciliation
basic paradigm: passive adversary

Conversation about their differences also known as information reconciliation

some information $E$ about $w$
basic paradigm: passive adversary

Conversation about their differences also known as information reconciliation

Conversation about removing Eve’s information also known as privacy amplification
basic paradigm: passive adversary

Conversation about their differences also known as information reconciliation

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

some information $E$ about $w$
privacy amplification

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

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

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

Solution: use a strong extractor

$$\text{minentropy } k$$

$$w \xrightarrow{\text{Ext}} r$$
privacy amplification

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

Solution: use a strong extractor

(e.g., Eve knows some $E$ about it)
If average min-entropy $H_{\min}(W | E)$ is sufficiently high, and Ext is an average-case strong extractor, this works!

Using universal hashing:

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 for any 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]:
poor quality randomness → Ext → indistinguishable from uniform

[Wyner]:
randomness (maybe uniform) → Ext → indistinguishable from uniform given leakage → Eve

The equivalence of these two views wasn’t obvious at first
basic paradigm: passive adversary

Conversation about their differences also known as information reconciliation

Conversation about removing Eve’s information also known as privacy amplification
Outline

• Passive adversaries
  – Privacy amplification
  – Information reconciliation

• Active adversaries, $\mathcal{W}$ has a lot of entropy
  – Privacy amplification
  – Information reconciliation

• Active adversaries, $\mathcal{W}$ has little entropy
  – Privacy amplification
  – Information reconciliation
basic paradigm: passive adversary

Conversation about their differences also known as information reconciliation

seed to a strong extractor
basic paradigm: passive adversary

Conversation about their differences also known as information reconciliation

seed to a strong extractor

Alice

Bob
basic paradigm: passive adversary

Conversation about their differences also known as information reconciliation

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

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

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

**Sketch:**

\[
\text{Sketch}(w_0) \\
\]

**Focus today:** single-message, starting with Bennett-Brassard-Robert 85
(interactive protocols more rare e.g., Brassard-Salvail 93)
Aside: chain rule for $H_{\text{min}}$

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

Lemma: $H_{\text{min}}(X | E) \geq H_{\text{min}}(X, E) - H_{\text{max}}(E)$

Proof: Reduction. Suppose $\text{Pr}_{(x,e)}[A(e) \rightarrow x] = p$.

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

$\text{Pr}_{(x,e)}[B \rightarrow (x,e)] \geq \text{Pr}_{(x,e,g)}[e=g \ \text{and } A(g) \rightarrow x]$

$= \text{Pr}_{(x,e,g)}[e=g \ \text{and } A(e) \rightarrow x]$

$= \text{Pr}_{(x,e,g)}[e=g] \ \text{Pr}_{(x,e,g)}[A(e) \rightarrow x]$

$= p/|\text{support}(E)|$

$\square$

Lemma: $H_{\text{min}}(X | E_1, E_2) \geq H_{\text{min}}(X, E_2 | E_1) - H_{\text{max}}(E_2)$
**definition:** secure sketch is a pair \((\text{Sketch}, \text{Rec})\)
definition: secure sketch is a pair (Sketch, Rec)

Alice

\[ w_0 \]

Sketch

\[ w_0 \rightarrow c \]

Bob

\[ w_1 \approx w_0 \]

same definition
for every notion of “\( \approx \)"
definition: secure sketch is a pair (Sketch, Rec)

Alice

\( w_0 \)

Sketch

\( c \)

Bob

\( w_1 \approx w_0 \)

same definition for every notion of “\( \approx \)”

\( c \)

Rec

\( w_1 \)

\( c \)

\( w_0 \)
**Definition:** secure sketch is a pair \((\text{Sketch}, \text{Rec})\)

\[(w_0, \text{Sketch}) \rightarrow c \] Alice

\[c \rightarrow w_1 \] Sketch

\[w_1 \rightarrow c \rightarrow w_0 \] Bob

same definition for every notion of “≈”

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

\((\text{Sketch}, \text{Rec})\) is a \((k, k - l)\)-secure sketch if

\[H_{\text{min}}(W_0 \mid E) \geq k \text{ implies } H_{\text{min}}(W_0 \mid E, \text{Sketch}(W_0)) \geq k - l\]
information-reconciliation + privacy amplification
information-reconciliation + privacy amplification

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

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

Thus can get \( m = k - l - 2 \log (1/\varepsilon) \)
information-reconciliation + privacy amplification

Alice

$w_0$

Sketch $\rightarrow c$

$w_0$

Ext $\rightarrow r$

Bob

$w_1$

Eve

$c, seed$

Rec $\rightarrow w_0$

Ext $\rightarrow r$

All in one message!
Let’s take another view of what we’ve built…
information-reconciliation + privacy amplification

- Alice:
  - $w_0$
  - Sketch
  - Ext
  - $w_0$
  - seed

- Bob:
  - $w_1$
  - Rec
  - Ext

$p = (c, seed)$
information-reconciliation + privacy amplification

Alice

\( w_0 \)

Sketch \( \rightarrow c \)

Ext

\( w_0 \) \( \rightarrow r \)

seed

Bob

\( w_1 \)

\( c, seed \)

Rec \( \rightarrow w_0 \)

Ext

\( w_1 \)

\( c \)

seed

\( r \)

\( p = (c, seed) \)
information-reconciliation + privacy amplification
information-reconciliation + privacy amplification
information-reconciliation + privacy amplification
Fuzzy Extractors

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

Definition of fuzzy extractors:

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

Security requirement: if $H_{\text{min}}(W_0|E) \geq k$ 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]

Advantages of this view:
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 $w_1, w_2 \ldots$ 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, $\mathcal{W}$ has a lot of entropy
  – Privacy amplification
  – Information reconciliation

• Active adversaries, $\mathcal{W}$ has little entropy
  – Privacy amplification
  – Information reconciliation
How to build a secure sketch?
How to build a secure sketch?

Want:
\[ H_{\min}(W_0 | E) \geq k \implies H_{\min}(W_0 | E, \text{Sketch}(W_0)) \geq k - l \]

Focus for now: \( \approx \) means Hamming distance

\( w_0 \) and \( w_1 \) are strings over \( GF(q) \) that differ in \( \leq t \) positions
background: error-correcting codes

\((n, \mu, \delta)_q\) code \(GF(q)^\mu \rightarrow GF(q)^n\)

- encodes \(\mu\)-symbol messages into \(n\)-symbol codewords
- any two codewords differ in at least \(\delta\) locations
  - fewer than \(\delta/2\) errors \(\Rightarrow\) unique correct decoding
background: error-correcting codes

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

- encodes \(\mu\)-symbol messages into \(n\)-symbol codewords
- any two codewords differ in at least \(\delta\) 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 \(n, \mu, \delta\) 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:

$$c = w_0 - \text{random codeword}$$
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:
  $c = w_0 - \text{random codeword}$
• Rec: $\text{dec}(w_1 - c) + c$
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:
  \[ c = w_0 - \text{random codeword} \]
- Rec: \( \text{dec}(w_1 - c) + c \)
- 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

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

c = w_0 - \text{ random codeword}

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

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

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

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

\(\downarrow\) entropy loss \(l\)
optimization for linear codes

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

\(c = w_0 - \text{random codeword}\)

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

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

\(c = \text{uniform choice from } \{w_0 - \text{Ker} H\}\)

Observe that \(\{w_0 - \text{Ker} H\} = \{x: Hx = Hw_0\}\)

(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 \(\text{GF}(q)^\mu \rightarrow \text{GF}(q)^n\)

\(c = w_0 - \text{random codeword}\)

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

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

\(c = \text{uniform choice from } \{w_0 - \text{Ker } H\}\)

Observe that \(\{w_0 - \text{Ker } H\} = \{x: Hx = Hw_0\}\)

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

\[H_{\text{min}}(W_0 \mid E, HW_0) \geq H_{\text{min}}(W_0 \mid E) - H_{\text{max}}(HW_0)\]

\[= H_{\text{min}}(W_0 \mid E) - (n - \mu) \log q\]
syndrome or code-offset construction

\[
\text{Sketch}(w) = Hw \text{ OR Sketch}(w) = w - \text{random codeword}
\]

- If ECC $\mu$ symbols $\rightarrow n$ symbols and has distance $\delta$:
  - 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,

- 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

Alice

Bob

\( w_0 \)

\( w_1 \)

**Sketch**

\( c \)

**Ext**

\( p = (c, seed) \)

**Rec**

\( w_0 \)

\( w_1 \)

**Ext**

**Eve**
1-message key agreement for passive adversaries

Alice

Bob

Gen

Rep

Eve

$w_0$

$r$

$p$

$w_1$

$r$
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, $\mathcal{W}$ has a lot of entropy
  – Privacy amplification
  – Information reconciliation

• Active adversaries, $\mathcal{W}$ 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 $\text{Eve}(p)$ produces $p' \neq p$

(with $1 - \text{negligible probability over } w_0$ & coins of Rep, Eve)
building robust extractors

Idea 0:

\[ w \rightarrow \text{Ext} \rightarrow r \]

\[ \text{seed} \rightarrow \text{MAC} \rightarrow \sigma \]

\[ p = (\text{seed}, \sigma) \]
building robust extractors

Idea 0:

\[ w \xrightarrow{\text{seed}} \text{Ext} \xrightarrow{} r \]

\[ \text{MAC} \xrightarrow{} \sigma \]

\[ p = (\text{seed}, \sigma) \]
building robust extractors

Idea 0:

\[ p = (\text{seed}, \sigma) \]

But if adversary changes \textit{seed}, then \( r \) will change
building robust extractors

Idea 0:

\[ p = (\text{seed}, \sigma) \]

\[ w \xrightarrow{\text{seed}} \text{Ext} \xrightarrow{r} \]

\[ \text{MAC} \xrightarrow{\sigma} \]

\[ \text{Key}?\]

\[ r? \text{ But if adversary changes } \text{seed}, \text{ then } r \text{ will change} \]

\[ w? \]

Circularity!

\[ \text{seed extracts from } w \]

\[ w \text{ authenticates } \text{seed} \]
**background: XOR-universal functions and MACs**

- Define $f_a (\cdot)$ 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 $\text{MAC}_\kappa (\cdot)$ to be a $\delta$-secure one-time message authentication code (MAC) if $\Pr[\text{Eve wins}]$ is at most $\delta$:
  - Pick a random $\kappa$; 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 $f_a (\cdot)$ is XOR-universal then
  $$\text{MAC}_{a,b}(i) = f_a(i) \oplus b$$
  is a $1/2^v$ secure MAC
  - Proof: guessing $\sigma_j \iff$ 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

- \( \Pr[\text{Eve wins}] = \mathbb{E}_{\kappa \text{ chosen uniformly}} \Pr[\text{Eve wins for key } = \kappa] \leq \delta \)
- What if \( \kappa \) is not uniform but has min-entropy \( k \)?

\( \mathbb{E}_{\kappa \text{ chosen from some entropy-} k \text{ distribution}} \frac{f(\kappa)}{\Pr[\kappa]} \leq \sum f(\kappa) 2^{-k} \)

(because \( f \) is nonnegative)

\[ \begin{align*}
&= 2^{|\kappa|-k} \sum f(\kappa) 2^{-|\kappa|} \\
&= 2^{|\kappa|-k} \mathbb{E}_{\kappa \text{ chosen uniformly}} f(\kappa) \\
&= 2^{|\kappa|-k} \delta
\end{align*} \]

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

• Passive adversaries
  – Privacy amplification
  – Fuzzy extractors
  – Information reconciliation

• Active adversaries, \( \omega \) has a lot of entropy
  – Message authentication codes
  – Privacy amplification
  – Information reconciliation

• Active adversaries, \( \omega \) has little entropy
  – Privacy amplification
  – Information reconciliation
building robust extractors

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

[Maurer-Wolf97] $w = $

\[
\begin{array}{c|c|c}
\hline
n/3 & n/3 & n/3 \\
\hline
a & b & c \\
\hline
\end{array}
\]

\[i \quad \times \]

\[ r = [ai]_1^m \]
building robust extractors

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

[Maure-Wolf97] $w = \begin{array}{ccc}
n/3 & n/3 & n/3 \\
a & b & c \\
\end{array}$

$$r = [ai]^m_1$$

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

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

[Maurer-Wolf97] $w = \begin{array}{c} a \\ b \\ c \end{array}$

Extract if $k > 2n/3$

$r = [ai]^m_1$

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

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

[Maurer-Wolf 97] $w = \begin{array}{ccc}
\frac{n}{3} & \frac{n}{3} & \frac{n}{3} \\
a & b & c \\
\times & \times & +
\end{array}$

Extract if $k > 2n/3$

$r = [ai]^m_1$

$\epsilon$-uniform if $n/3 > m + g + 2\log_2 \frac{1}{\epsilon}$

$\sigma = bi + c$
building robust extractors

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

[Maurer-Wolf97] $w = \begin{array}{c}
\times \\
\times \\
+ \\
\end{array}$

Extract if $k > 2n/3$

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

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

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

$[\text{Maurer-Wolf} \ 97]$  

Extract if $k > 2n/3$

$r = [ai]^m_1$

$i, \sigma = bi + c$

$p$
building robust extractors

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

[Maure-Wolf97] $w = \begin{array}{c}
\frac{n}{3} \\
\frac{n}{3} \\
\frac{n}{3}
\end{array}
\begin{array}{c}
a \\
b \\
c
\end{array}$

Extract if $k > 2n/3$

$r = [ai]^m_1$

$\sigma = bi + c$

[Dodis-Kanukurthi-Katz-Reyzin-Smith ’12] $w = \begin{array}{c}
n - v \\
v
\end{array}
\begin{array}{c}
a \\
b
\end{array}$

$i$

$r = [ai]^{n-v}_{v+1}$
Building robust extractors

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

[Maurer-Wolf97] $w = \frac{n}{3} \times \frac{n}{3} \times \frac{n}{3}$

Extract if $k > \frac{2n}{3}$

$\sigma = bi + c$

$[\text{Maurer-Wolf97}]$

$w = a \times b \times c$

$\times \quad i \quad \times \quad +$

$r = [ai]^m$

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

$w = \frac{n-v}{1} \times \frac{v}{1}$

$i \quad \times \quad +$

$r = [ai]^{n-v}$

$\sigma = [ai]^v + b$

[Dodis-Kanukurthi-Katz-Reyzin-Smith ’12]
building robust extractors

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

[Maurer-Wolf97] $w = \begin{array}{ccc} n/3 & n/3 & n/3 \\ a & b & c \end{array}$

Extract if $k > 2n/3$

$w =$

[Dodis-Kanukurthi-Katz-Reyzin-Smith ’12] $w = \begin{array}{c} n - \nu \\ a \end{array}$

jointly $\epsilon$-uniform if $\nu > g + 2\log \frac{1}{\epsilon}$

$\sigma = [ai]_{\nu+1} + b$

$\delta$-secure if $\nu > g + \log \frac{1}{\delta}$
building robust extractors

\[ w = a + b \]

Analysis:

- Extraction: \((R, \sigma) = ai + b\) is a universal hash family (few collisions) \((i \text{ is the key, } w = (a, b) \text{ is the input})\) [ok by leftover hash lemma]
- Robustness: \(\sigma = [ai]_{n-v}^+ + b\) is XOR-universal \((w = (a, b) \text{ is the key, } i \text{ is the input})\) [ok by Maurer-Wolf]
building robust extractors

\[ w = \begin{array}{c}
\text{a} \\
\text{b}
\end{array} \]

\[ i \rightarrow \times \rightarrow + \]

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

\[ r = [ai]^{n-v}_{v+1} \]

\[ \sigma = [ai]^{v}_{1} + b \]

\( k > n/2 \) is necessary [Dodis-Wichs09]
building robust extractors

\[ w = a \]

\[ i \]

\[ r = [ai]_{n-v}^{n-v+1} \]

\[ \sigma = [ai]_1^{v} + b \]

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

\( k > \frac{n}{2} \) is necessary [Dodis-Wichs09]
Outline

• Passive adversaries
  – Privacy amplification
  – Fuzzy extractors
  – Information reconciliation

• Active adversaries, $\mathcal{w}$ has a lot of entropy
  – Message authentication codes
  – Privacy amplification
  – Information reconciliation

• Active adversaries, $\mathcal{w}$ has little entropy
  – Privacy amplification
  – Information reconciliation
recall: secure sketch

Alice

\(w_0\)

Sketch

Bob

\(w_0 \approx w_1\)

\(c\)

\(c\)

Rec

\(w_1\)

\(c\)

\(w_0\)
building robust fuzzy extractors

How to MAC long messages? \( \sigma = [a^2c + ai]_1^γ + b \)

(recall \( w = a|b \))
building robust fuzzy extractors

\[ p = (i, c, \sigma) \]

How to MAC long messages? \( \sigma = [a^2c + ai]_1 + b \) (recall \( w = a|b \))

How to Rep
How to MAC long messages? \( \sigma = [a^2c + ai]^\gamma + b \) (recall \( w = a|b \))
building robust fuzzy extractors

\[ p = (i, c, \sigma) \]

How to MAC long messages? \[ \sigma = [a^2 c + ai]_1^\gamma + b \]
(recall \( w = a|b \))

How to Rep
building robust fuzzy extractors

\[ p = (i, c, \sigma) \]

How to MAC long messages? \[ \sigma = [a^2 c + ai]^\gamma + b \] (recall \( w = a|b \))

How to Rep
building robust fuzzy extractors

\[ p = (i, c, \sigma) \]

How to MAC long messages?
\[ \sigma = [a^2c + ai]^\gamma + b \]
(recall \( w = a|b \))

How to Rep
building robust fuzzy extractors

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

How to MAC long messages? $\sigma = [a^2c + ai] \gamma + b$
(recall $w = a|b$)

How to Rep
the MAC problem

Authentication:

\[ \sigma = \text{MAC}_w(i, c) = [a^2 c + ai]^\vee + b \]
(recall \( w = a|b \))

Verification:

\[ \frac{w_1}{w_0} \quad \frac{\land \land i}{c} \quad \frac{\text{Rec}}{\text{Ver}(\sigma)} \quad \text{ok/ } \perp \]

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

Observe: knowing \((w_1 \oplus w_0 \text{ and } c \oplus \land \land c)\)
gives knowledge of \(w_0 \oplus \land \land w_0 = u\)

Need: \( \forall u \), given \( \text{MAC}_w(i, c) \), hard to forge \( \text{MAC}_{w+u}(i, c) \)
the MAC problem

Authentication:
\[ \sigma = \text{MAC}_w(i, c) = [a^5 + a^2c + \gamma ai]_1 + b \]
(recall \( w = a|b \))

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

Observe: knowing \( (w_1 \oplus w_0 \text{ and } c \oplus \wedge c) \)
gives knowledge of \( w_0 \oplus \wedge w_0 = u \)

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

Hard to forge for any fixed \( u \)
The MAC problem

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

(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 $c \oplus ^\wedge c)$

gives knowledge of $w_0 \oplus ^\wedge w_0 = u$

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

Generalization [Padro et al. ‘05] if $i$ is public

Code that detects additive change

Generalization [Padro et al. ‘05] if $i$ is public

Code that detects additive change

Generalization [Padro et al. ‘05] if $i$ is public

Code that detects additive change
the MAC problem

Authentication  Alternative [Boyen et al. ‘05]
\[ \sigma = \text{MAC}_w(i, c) = \text{RandomOracle}(w, i, c) \]

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

Verification:

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

Observe: knowing \((w_1 \oplus w_0 \text{ and } c \oplus \overline{c})\)
gives knowledge of \(w_0 \oplus \overline{w_0} = u\)

Need: \( \forall u \), given \( \text{MAC}_w(i, c) \), hard to forge \( \text{MAC}_{w+u}(i, c) \)
building robust fuzzy extractors

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

Problem: \( c \) reveals \( l \) bits about \( w \) ⇒

\( k \) decreases, \( g \) increases ⇒

lose \( 2l \)

Can't avoid decreasing \( k \), but can avoid increasing \( g \)

\( c = \text{Sketch}(w_0) \) 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 $1 - \text{negligible probability over } w_0$ & coins of Rep, Eve)
Robustness: as long as $w_0 \approx w_1$, if Eve($p$) produces $p' \neq p$

$(\text{with } 1 - \text{negligible probability over } w_0 \text{ & coins of Rep, Eve})$
Summary: robust fuzzy extractors

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

$(with\ 1 - \text{negligible\ probability\ over\ } w_0\ &\ \text{coins\ of\ Rep,\ Eve})$
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

Entropy Deficiency ("gap")
Privacy Amplification

Alice

$w$

Protocol AUTH

authenticate seed

Entropy Deficiency ("gap")

Authentically receive seed

Bob

$w$
Privacy Amplification

Alice

\[ w \]

Entropy Loss

Authenticate seed

Protocol AUTH

Entropy Deficiency ("gap")

Authentically receive seed

Bob

\[ w \]
Privacy Amplification

Alice

\[ w \]

\[ \text{Entropic Loss} \]

\[ \text{Authenticate seed} \]

\[ w \rightarrow \text{Ext} \rightarrow r \]

\[ r \text{ looks uniform given } seed \]

Protocol AUTH

\[ \text{Authentically receive seed} \]

\[ w \rightarrow \text{Ext} \rightarrow r \]

\[ r \text{ looks uniform given } seed \]

Bob

\[ w \]
Privacy Amplification

Alice

\[ w \]

\[ w \]

Entropy Loss

Authenticate seed

\[ w \rightarrow \text{Ext} \rightarrow r \]

\[ seed \rightarrow \text{Ext} \rightarrow r \]

\[ r \text{ looks uniform given } seed \]

Entropy of \( r \)

Protocol AUTH

[Renner-Wolf ’03]

Bob

\[ w \]

\[ w \]

Entropy Deficiency (”gap”)

Authentically receive seed

\[ w \rightarrow \text{Ext} \rightarrow r \]

\[ seed \rightarrow \text{Ext} \rightarrow r \]

\[ r \text{ looks uniform given } seed \]

Goal: Increase length of \( r \) by minimizing entropy loss
**[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.
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**

Alice

\( w \)

Accept if \( \text{Ext}_x(w) \) is correct

\[ \text{challenge } x \rightarrow \text{response } y = \text{Ext}_x(w) \]

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

Bob

\( w \)

\( \text{Ext} \rightarrow y \)

\( x \rightarrow y \rightarrow x' \rightarrow y' \)

\( w \rightarrow \text{Ext} \rightarrow y' \)
**[RW03] Auth: Sub-protocol $\frac{1}{2}$ bit authentication**

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

- **Alice**
  - $w$
  - $y = \text{Ext}_x(w)$

- **Bob**
  - $w$
  - Generate random seed $x$
  - $(1, y)$ or $(0, \bot)$
  - If $b = 1$, verify $y = \text{Ext}_x(w)$

- **Eve**
  - bit-auth($b$)
Guarantees: if Bob receives bit $b = 1$, then Alice sent $b = 1$

- Problem: Eve can’t change 0 to 1, but can change 1 to 0
- Solution: make the string balanced (#0s = #1s)
[RW03] Auth: From $\frac{1}{2}$ 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 $\frac{1}{2}$ bit to string

For $2^{-\delta}$-security, each Ext output needs $\approx \delta$ bits. Loss $\approx 1.5 |\text{seed}| \delta$
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 $O(\delta)$ in [Chandran-Kanukurthi-Ostrovsky-Reyzin 2014] (but for practical values of $\delta$, 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

\[ c = \text{Sketch}(w_0) \]

Alice

Bob

Eve

recover \( w_0 \)

\( w_0 \)

\( w_1 \)
Information Reconciliation

$w_0$
$c = \text{Sketch}(w_0)$
$w_1$

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

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

Authenticate message

Protocol AUTH

Authentically receive message
To verify, Bob needs to recover $w_0$ from $w_1$ so Alice needs to send $c$, so need authentication protocol!
**Attempt 1: Error-Tolerant Authentication**

- 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**

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_0$</td>
<td>$w_1 \approx w_0$</td>
</tr>
<tr>
<td>$c = \text{Sketch}(w_0)$</td>
<td>$w^* = \text{Rec}(w_1, c)$</td>
</tr>
</tbody>
</table>

**Protocol AUTH**

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 $w$
Solution [Kanukurthi-Reyzin ‘09]: Reduce entropy loss using a MAC

- MAC needs a symmetric key $\kappa$
- Where does $\kappa$ come from? Generate random $\kappa$ and authenticate it
Solution [Kanukurthi-Reyzin ‘09]: Reduce entropy loss using a MAC

- MAC needs a symmetric key $\kappa$
- Where does $\kappa$ come from? Generate random $\kappa$ and authenticate it
**Solution [Kanukurthi-Reyzin ‘09]:** Reduce entropy loss using a MAC

- MAC needs a symmetric key $\kappa$
- Where does $\kappa$ come from? Generate random $\kappa$ and authenticate it
Solution [Kanukurthi-Reyzin ‘09]: Reduce entropy loss using a MAC

- MAC needs a symmetric key $\kappa$
- Where does $\kappa$ come from? Generate random $\kappa$ and authenticate it.

**Attempt 2: Error-Tolerant Authentication**

- Protocol $\text{AUTH}(\kappa)$
- Liveness Test
- Auth reveals $\kappa$!
Solution [Kanukurthi-Reyzin ‘09]: Reduce entropy loss using a MAC

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

By the time Eve learns $\kappa$, it is too late for Eve to come up with forgery!
**information-reconciliation + privacy amplification**

**Alice**

- $w_0$
- $c = \text{Sketch}(w_0)$
- Authenticate $c$

**Bob**

- $w_1 \approx w_0$
- $w_0 = \text{Rec}(w_1, c)$

**Protocol Auth**

- Send $seed$
- Receive $seed$

**Error-Tolerant Authentication**

- Send $seed$
- Receive $seed$

- $w_0 \rightarrow \text{Ext} \rightarrow r$
- $w_0 \rightarrow \text{Ext} \rightarrow r$

Use $r$ as a MAC key to send the real extractor seed.
Information-reconciliation + privacy amplification

---

**Alice**

- $w_0$
- $c = \text{Sketch}(w_0)$
- Authenticate $c$
- and seed

**Bob**

- $w_1 \approx w_0$
- $w_0 = \text{Rec}(w_1, c)$

---

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