How to extract useful randomness from unreliable sources

Divesh Aggarwal  Maciej Obremski  João Ribeiro  Luisa Siniscalchi  Ivan Visconti

CQT & National University of Singapore  Imperial College London  University of Salerno → Aarhus University  University of Salerno

Eurocrypt 2020
Randomness and cryptography

Perfect randomness

Cryptography

In practice, randomness sources $X$ are not perfect!

Weaker assumption: min-entropy lower bound

$$\max_x \Pr[X = x] \leq 2^{-k} \quad \quad \quad H_\infty(X) \geq k$$

$k$: bits of min-entropy
Randomness extraction

\[ \max_x \Pr[X = x] \leq 2^{-k} \quad \text{and} \quad H_\infty(X) \geq k \]

\( k \) bits of min-entropy

**IMPOSSIBLE!**

\[ X \xrightarrow{\text{Ext}} Y \approx_\epsilon \text{Unif} \]

(arbitrary weak \( k \)-source)  
(statistically close to uniform)

**Multi-source extraction:** combine several **independent** weak sources  
(e.g., sampled from different devices/locations)
Multi-source randomness extraction

Need to trust several devices at different locations!
(especially when dealing with public randomness!)

What happens if some sources are corrupted?
SHELA sources:
Multi-source randomness extraction without trust

\[(t, \ell, k)\]-SHELA source: Somewhere-Honest Entropic Look Ahead

\[X_1 \quad X_2 \quad X_3 \quad \cdots \quad X_{\ell-1} \quad X_\ell\]

1. Adversary chooses \(\ell - t\) blocks to corrupt
SHELA sources: Multi-source randomness extraction \textit{without trust} \\

\((t, \ell, k)\)-SHELA source: Somewhere-Honest Entropic Look Ahead \\

1. Adversary chooses \( \ell - t \) blocks to corrupt \\
2. Adversary fixes corrupted block based on previous samples \\
Adversary knows positions and distributions of honest blocks \\
Honest \( X_i \)'s are independent of each other and satisfy \( \mathbb{H}_\infty(X_i) \geq k \)
Some other adversarial source models

Old:

Santha-Vazirani sources

Bit-fixing sources

[Dodis 2001]: Bias-control limited sources

Recent:

[Austrin, Chung, Mahmoody, Pass, Seth 2014]: p-tampering attacks

[Bentov, Gabizon, Zuckerman 2016]: p-resettable sources

[Chattopadhyay, Goodman, Goyal, Li 2019]: Multi sources w/ local dependence

[Dodis, Vaikuntanathan, Wichs 2019]: Extractor-dependent sources

[Ball, Goldreich, Malkin 2019]: Somewhat-dependent sources
Can we extract perfect randomness from SHELA sources?

No

Regime of interest: \( t = \gamma \cdot \ell \) (constant fraction of corruptions), \( \ell \) larger than some constant

impossibility for \( p \)-resettable sources
[Bentov, Gabizon, Zuckerman 2016]

Follows from impossibility for special subset of Santha-Vazirani sources

impossibility for SHELA sources
must have error \( \epsilon = \Omega(1 - \gamma) \)

Holds even if honest blocks are uniform!

Can we extract “useful” randomness from SHELA sources?
The next best thing: somewhere-random sources

\[(t', \ell')\text{-SR source: Somewhere-Random}\]

\[
\begin{array}{cccc}
Y_1 & Y_2 & Y_3 & \cdots & Y_{\ell'}
\end{array}
\]

Guarantee: There exist \(i_1 < i_2 < \cdots < i_{\ell'}\) such that \((Y_{i_1}, Y_{i_2}, \ldots, Y_{i_{\ell'}}) = \text{unif}\)

Interested in **convex combinations** of SR sources \(\rightarrow\) **convSR** sources

convSR sources are very useful!

SHELA \(\rightarrow\) great convSR sources
SR sources and one-sided error

A randomized algorithm with one-side error

\[ x \in \mathcal{L} \quad \text{Always outputs YES} \]
\[ x \notin \mathcal{L} \quad \text{Outputs NO with probability 2/3, YES otherwise} \]

Only guaranteed under uniform randomness!

SR source

\[ \begin{array}{l}
Y_1 \rightarrow \mathcal{A}(x, Y_1) \quad \text{YES if all output YES} \\
Y_2 \rightarrow \mathcal{A}(x, Y_2) \quad \text{NO otherwise} \\
\vdots \\
Y_{\ell'} \rightarrow \mathcal{A}(x, Y_{\ell'}) \quad \text{Also one-sided error!}
\end{array} \]

Runtime: \( O(\ell' \cdot t_A) \)

wish to:

i) Minimize \( \ell' \)

ii) Maximize length of \( Y_i \)'s
Crypto applications of SR sources

We construct (from generic complexity assumptions):

• Non-interactive witness indistinguishable proof systems
• Non-interactive commitments

Elsewhere:

• Publicly-verifiable proof systems [Scafuro, Siniscalchi, Visconti 2019]

Overall: non-interactive primitives with a “somewhere-random CRS”
“Somewhere-extraction” from SHELA sources

**Goal:** Design $\text{Ext} : \{0, 1\}^{l \cdot n} \rightarrow \{0, 1\}^{l' \cdot m}$ such that for every $(t, l, k)$-SHELA source $X$,

$$\text{Ext}(X) \approx_\epsilon \text{convSR}$$

**Want:** #output blocks $l'$ and error $\epsilon$ **small**, output block length $m$ **large**

**Naive approach:** apply 2-source extractor $2\text{Ext}$ to every pair of blocks of $X$

**Why?** If $X_i$ and $X_j$ are honest, then $2\text{Ext}(X_i, X_j) \approx \text{unif}$

**Cons:**

i) $l' = \Omega(l^2)$

ii) Non-negligible error when $k < 0.44n$

*Can we do better?*
Better somewhere-extraction from SHELA sources

$X_i \in \{0, 1\}^n$

$k \geq 0.51n$
Better somewhere-extraction from SHELA sources

\[ X_i \in \{0, 1\}^n \]
\[ k \geq 0.51n \]

\[ X_1 \quad \text{emoji} \quad X_3 \quad \text{emoji} \quad X_5 \]

unbalanced 2-source extractors
(left source: low entropy, right source: high entropy)

\[ \text{Ext}_i : \{0, 1\}^{i \cdot n} \times \{0, 1\}^n \rightarrow \{0, 1\}^m \]
Better somewhere-extraction from SHELA sources

\[ X_i \in \{0,1\}^n \]
\[ k \geq 0.51n \]

Unbalanced 2-source extractors
(left source: low entropy, right source: high entropy)

\[ \text{Ext}_i : \{0, 1\}^{i \cdot n} \times \{0, 1\}^n \rightarrow \{0, 1\}^m \]

\[ Y_1 = \text{Ext}_1(X_1, \text{left source}) \]
Better somewhere-extraction from SHELA sources

\[ X_i \in \{0, 1\}^n \]
\[ k \geq 0.51n \]

unbalanced 2-source extractors
(left source: low entropy, right source: high entropy)

\[ \text{Ext}_i : \{0, 1\}^i \times \{0, 1\}^n \rightarrow \{0, 1\}^m \]

\[ Y_1 = \text{Ext}_1(X_1, \smiley) \]

\[ Y_2 = \text{Ext}_2(X_1 \frown, X_3) \]
Better somewhere-extraction from SHELA sources

\[ X_i \in \{0, 1\}^n \]
\[ k \geq 0.51n \]

unbalanced 2-source extractors
(left source: low entropy, right source: high entropy)

\[ \text{Ext}_i : \{0, 1\}^{i \cdot n} \times \{0, 1\}^n \rightarrow \{0, 1\}^m \]

\[ Y_1 = \text{Ext}_1(X_1, \text{DEVIL}) \]
\[ Y_2 = \text{Ext}_2(X_1 \text{DEVIL}, X_3) \]
\[ Y_3 = \text{Ext}_3(X_1 \text{DEVIL}, X_3, \text{DEVIL}) \]
Better somewhere-extraction from SHELA sources

\[ X_i \in \{0,1\}^n \]
\[ k \geq 0.51n \]

\( \epsilon = 2 - \Omega(n) \)

\( m = \Omega(n) \)

unbalanced 2-source extractors
(left source: low entropy, right source: high entropy)

\[ \text{Ext}_i : \{0,1\}^{i \cdot n} \times \{0,1\}^n \rightarrow \{0,1\}^m \]

\[ Y_1 = \text{Ext}_1(X_1, \medcat) \]
\[ Y_2 = \text{Ext}_2(X_1 \medcat, X_3) \]
\[ Y_3 = \text{Ext}_3(X_1 \medcat, X_3, \medcat) \]
\[ Y_4 = \text{Ext}_4(X_1 \medcat, X_3, \medcat, X_5) \]
Better somewhere-extraction from SHELA sources

\[ X_i \in \{0, 1\}^n \]
\[ k \geq 0.51n \]

unbalanced 2-source extractors
(left source: low entropy, right source: high entropy)

\[ \text{Ext}_i : \{0, 1\}^i \times \{0, 1\}^n \rightarrow \{0, 1\}^m \]

\[ Y_1 = \text{Ext}_1(X_1, \text{left source}) \]
\[ Y_2 = \text{Ext}_2(X_1, \text{left source}, X_3) \]
\[ Y_3 = \text{Ext}_3(X_1, \text{left source}, X_3, \text{left source}) \]
\[ Y_4 = \text{Ext}_4(X_1, \text{left source}, X_3, \text{left source}, X_5) \]
Better somewhere-extraction from SHELA sources

\[ X_i \in \{0, 1\}^n \]
\[ k \geq 0.51n \]

\[ X_1 \quad \text{left source} \]
\[ X_3 \quad \text{independent high min-entropy} \]
\[ X_5 \quad \text{right source} \]

unbalanced 2-source extractors
(Left source: low entropy, right source: high entropy)

\[ \text{Ext}_i : \{0, 1\}^{i \cdot n} \times \{0, 1\}^n \rightarrow \{0, 1\}^m \]

i) \[ Y_2 \approx U_m \]

ii) \[ \text{whp over fixing of } Y_2, \ Y_4 \approx U_m \]

\[ Y_1 = \text{Ext}_1(X_1, \text{smiley}) \]
\[ \text{contains enough min-entropy} \]

\[ Y_2 = \text{Ext}_2(X_1, X_3) \]
\[ \text{independent high min-entropy} \]

\[ Y_3 = \text{Ext}_3(X_1, X_3, \text{smiley}) \]
\[ \text{independent high min-entropy} \]

\[ Y_4 = \text{Ext}_4(X_1, X_3, X_5) \]
\[ \text{contains enough min-entropy given } Y_1, Y_2, Y_3 \]

\[ Y \text{ is } \epsilon\text{-close to } (t', \ell')\text{-convSR source in } \{0, 1\}^{\ell' \cdot m} \]
\[ \ell' = \ell - 1; \ t' = t - 1; \]
\[ \epsilon = 2^{-\Omega(n)}; \ m = \Omega(n); \]

works with only 2 honest blocks!
Somewhere-extraction from low-entropy SHELA sources

Want: Somewhere-extractor for \((t, \ell, k = \delta n)\)-SHELA, for \(\delta\) arbitrarily small constant

Idea: Combine previous high-entropy construction with somewhere-condensers

\[\text{Essentially the same parameters:}\]

\[
\ell' = O_\delta(\ell); \quad t' = t - 1; \quad \epsilon = 2^{-\Omega_\delta(n)}; \quad m = \Omega_\delta(n); \quad \text{works with only 2 honest blocks!}
\]

[Raz 2005], [Barak, Kindler, Shaltiel, Sudakov, Wigderson 2005],
[Zuckerman 2007], [Li 2011]
Can we extract useful convSR sources \textbf{without} exploiting structure of SHELA sources?

Problem: Superpolynomial \#blocks if error $\epsilon$ is negligible!

\textit{Can we do better?}
Somewhere-extraction from a weak source

Can we extract useful convSR sources without exploiting structure of SHELA sources? No!

Treat \((t, \ell, k)\)-SHELA source \(X \in \{0, 1\}^{\ell \cdot n}\) as weak \((n' = \ell \cdot n, k' = t \cdot k)\)-source

Somewhere-extractor for \((n', k')\)-sources with error \(\epsilon\), output block length \(m\)

\[
\text{#output blocks} \geq \frac{n' - k'}{\epsilon + 2^{-m}}
\]

If \(m\) isn’t small and \(\epsilon\) is negligible, need superpolynomial #output blocks

Proof: Somewhere-extractor \(\rightarrow\) disperser, so can apply well-known lower bounds

[Radhakrishnan, Ta-Shma 2000]

Open Q: Prove analogous result when \(m = 1\)
Summing up

• SHELA sources model multiple randomness sources corrupted by strong adversary

• Can’t extract perfect randomness

• **Can** extract great SR sources from low-entropy SHELA sources (**only need 2 honest blocks**)!

• SR sources are very useful (algorithms + crypto)

• Can’t extract useful SR sources **without** exploiting structure of SHELA source

Thanks for watching!