Evaluating Entropy for True Random Number Generators

Maciej Skórski[1]

IST Austria

WR0NG 2017, 30th April, Paris

[1] Supported by the European Research Council consolidator grant (682815-TOCNeT)

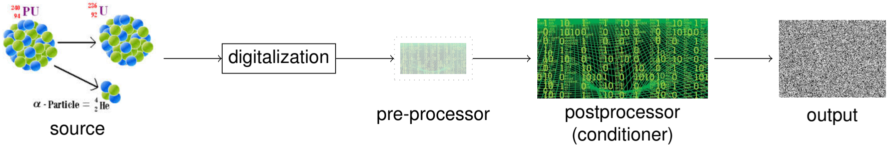
Plan

What is this talk about?

- overview of entropy estimation, in the context of TRNGs
- theoretical justification for some heuristics / explanation for subtle issues

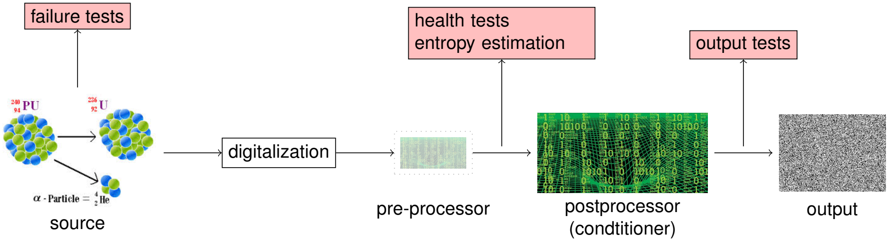
Plan

True Random Number Generators



(a) physical source generates noise (somewhat unpredictable)
(b) noise converted to digital form (may introduce extra bias)
(c) (little) preprocessing decreases bias (e.g. ignoring less variable bits)
(d) postprocessing eliminates bias and dependencies (e.g. extractor)
(e) output should be uniform

New paradigm: real-time monitoring



- standards [KS11,TBKM16]: monitor the source and digitalized raw numbers
- sometimes one implements also online output tests [VRV12].

Real-time testing necessary

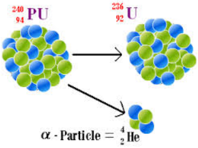
Need to evaluate the whole construction, no black-box outputs tests!

(a) biased functions may pass outputs tests

(b) sources may be bit different outside of lab (environmental influences)

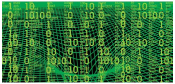
Theoretical framework

weak source:
entropy + *assumptions* to learn it from samples

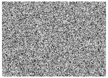


preprocessor: *condenser*

postprocessor: *extractor*
optionally: + hashing (extra masking)



output: *indistinguishable from random*



weak source + *online* entropy estimation + *calibrating* postprocessor ≈ TRNG

Evaluating security - criteria

Standards for Random Number Generators

Two popular and well documented (examples+justifications) recommendations

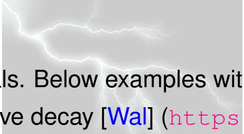
- AIS 31 - German Federal Office for Information Security (BSI)
- SP 800-90B - U.S. National Institute for Standards and Technology (NIST)

Randomness tests

Most popular: NIST, DieHard, DieHarder, TestU01

Plan

Examples of sources



Many proposals. Below examples with public (web) interfaces

- Radioactive decay [Wal] (https://www.fourmilab.ch/hotbits/)
- Atmospheric noise [Haa] (http://www.random.org/)
- Quantum vacuum fluctuations [SQCG] (http://qrng.anu.edu.au)

Necessary properties of sources

$$X = X_1, X_2, X_m \longrightarrow \boxed{f(X)} \longrightarrow \overset{\text{indistinguishability}}{\approx} \quad b_1 b_2 \ldots b_n$$

raw bits — post-processing — random bits

Theorem (Min-entropy in sources necessary [RW04])

If $X \in \{0,1\}^m$ is such that $f(X) \approx U_n$ then $X \approx Y$ s.t. $H_\infty(Y) \geqslant n$ where

$$H_\infty(X) = \min_x \log \frac{1}{P_X(x)}$$

is the min-entropy of the source (also when conditioned on the randomness of f).

Can we use Shannon entropy?

- many papers estimate Shannon entropy in the context of TRNGs (easier)
- best available tests utilize Shannon entropy (compression techniques)
- standards put more emphasize on min-entropy only recently

Shannon entropy is bad in one-shot regimes...

Shannon entropy is a bad estimate even for (less restrictive) collision entropy

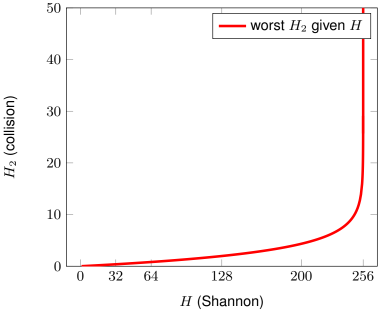


Figure: Worst bounds on collision entropy when Shannon entropy is fixed (256 bits).

Example

Even with $H(X) = 255.999$ we could have only $H_2(X) = 35.7$.
Construction: a heavy unit mass mixed with the uniform distribution.

... but ok for repeated experiments!

Asymptotic Equiparition Property

If the source produces $X_1, X_2, X_3 \ldots$ then for $x \leftarrow X_1, \ldots, X_n$ we have

$$\frac{1}{n}\log\frac{1}{P_{X^n}(x)} = \frac{1}{n}H(X^n) + o(1) \quad \text{w.p. } 1 - o(1)$$

Under reasonable restrictions on the source (e.g. iid or stationarity and ergodicity).

Essentially: almost all sequences are roughly equally likely.

Shannon is asymptotically good

We conclude that for $n \to \infty$

$$\frac{1}{n}H_{\infty}(X_1, \ldots, X_n|E) \approx \frac{1}{n}H(X_1, \ldots, X_n|E), \quad \Pr[E] = 1 - o(1)$$

this demonstrates the *entropy smoothing* technique [RW04,HR11,STTV07,Kog13].

How big is the error?

- can quantify the convergence in the AEP (Holenstein, Renner [HR11]...
- ... much better when entropy per bit is high - relevant to TRNGs [Sko17]

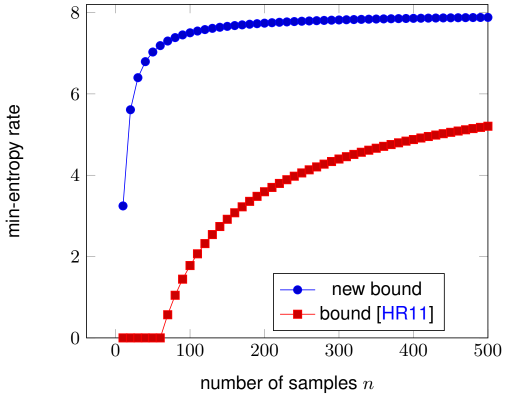


Figure: (smooth) min-entropy per bit, independent 8-bit samples with Shannon rate 0.997 per bit

Sources - conclusion

Shannon approximation

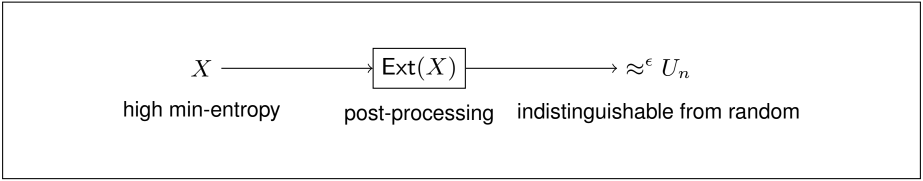
- min-entropy necessary for post-processing, but hard to estimate
- we have simple Shannon entropy estimators (compression techniques [Mau92])
- under (practically reasonable) restrictions on the source, one can approximate by Shannon entropy; the justification is by entropy smoothing+AEP
- convergence even better in high-entropy regimes (relevant to TRNGs)

What about Renyi entropy?

One can also use collision entropy (between min-entropy and Shannon entropy), which is faster to estimate [AOST15] (at least for iid sources).

Plan

Instantiating Postprocessors



Here $\approx^{\epsilon}$ means ϵ-closeness in total variation (statistical distance).

Implementing postprocessors

- Randomness extractors, like Teoplitz Matrices or the Trevisan extractor (implemented in quantum TRNGs [MXXTQ+13]).
- CBC-MAC (inside Intel's IvyBridge; TRNG is part of hybrid design!)
- other cryptographic functions (e.g. early Intel RNGs used SHA-1)

Postprocessors - Drawbacks

Disadvantages of post-processing

- entropy waste (input $>$ output, necessary!)
  - (a) best extractors: $2\log(1/\epsilon)$ bits
  - (b) other: half of input entropy as the practical rule of thumb [TBKM16,HKM12])
- slowdown
  - (a) Quantis: the bit rate goes down from about 4Mbps to approximately 75Kbps [Qua].

Security with insufficient entropy?

What if entropy estimates fail?

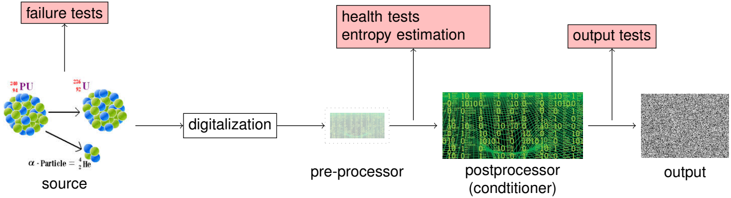
Key derivation - security under weak keys

- some cryptographic applications remain (somewhat) secure when fed with insufficient entropy [BDKPP+11,DY13,DPW14].
- entropy defficiency may be ”obscured” by the hash function and not easy to exploit in practice [TBKM16]

Plan

Plan

What to evaluate



test	feature	cathegory
source breakdown	zero-entropy alarm	health-test
source failure	low-entropy alarm	health-test
source rate	entropy level	entropy estimation
output uniformity	bias-alarm	randomness tests

How to evaluate security from samples?

Hypothesis testing

We use the statistical framework

- null Hyp_0: "generator is good"
- alternative Hyp_a: "generator is bad"

Can never confirm Hyp_0!

Absence of evidence is no evidence of absence

Can commit two errors

$\alpha = \Pr[\text{reject Hyp}_0 | \text{Hyp}_0]$ reject good generator = Type I Error

$\beta = \Pr[\text{accept Hyp}_0 | \text{Hyp}_a]$ accept bad generator = Type II Error

Note: often Type I is of interest (validating theories in empirical sciences)

Our priority: minimize Type II (first), keep Type I reasonably small (second).

Plan

Error testing - methodological issues (I)

type II errors ignored in standards and implementations?

Documents and packages refer to type I instead! Is the methodology correct?

type II errors for testing randomness are hard

Consider deciding the output uniformity

- type I errors can be computed precisely
  (”good” = uniform output, can give concrete bounds!)
- type II errors are hard
  (need state what ”bad” means; how to quantfify all ”bad” possibilities?)

Error testing - methodological issues (II)

Practical solution to Type II error testing

Since alternative is "amporphic":

1. develop tests for Type I error, but keep α not too small (e.g. $\alpha \in (0.01, 0.001)$)!
2. cover a range of assumptions by different tests

Rationale:

- too small α makes β big
- different tests cover different "pathologies"
- for some tests β is provably small under mild assumptions [Ruk11]

This approach used in standards and software packages.

Test batteries

Statistics of the observed data should be close to the ideal behavior

$$\forall T \in \text{Battery} \quad \Pr[T(obs) \gg T(ideal))] \approx 0$$

Multiple testing issues

The rejection power of a battery is bigger than a power of individual tests.

$$\Pr[\text{battery rejects}] \lesssim \#\text{tests} \cdot \Pr[\text{single test rejects}] \quad \text{union bound}$$

$$\Pr[\text{battery rejects}] \lesssim (\Pr[\text{single test rejects}])^{\#\text{tests}} \quad \text{positive dependency}$$

- BSI standard - addressed

$$\text{output uniformity}(\alpha = 10^{-3}) = 1258 \times \text{basic tests}(\alpha = 10^{-6})$$

- NIST standard - not addressed; criticized [DB16,MS15]
- not addressed in many batteries for randomness testing

multiple hypothesis not properly addressed?

- in output testing NIST rejects more $\Longrightarrow$ type II error smaller !
- consult the statistical literature when tailoring tests
- see [Ruk11] for more about the NIST methodology

Plan

Real-time tests on hardware

Why testing on hardware? Isolation from software!

- security countermeasure (against software attacks)
- efficiency (want real-time solution)

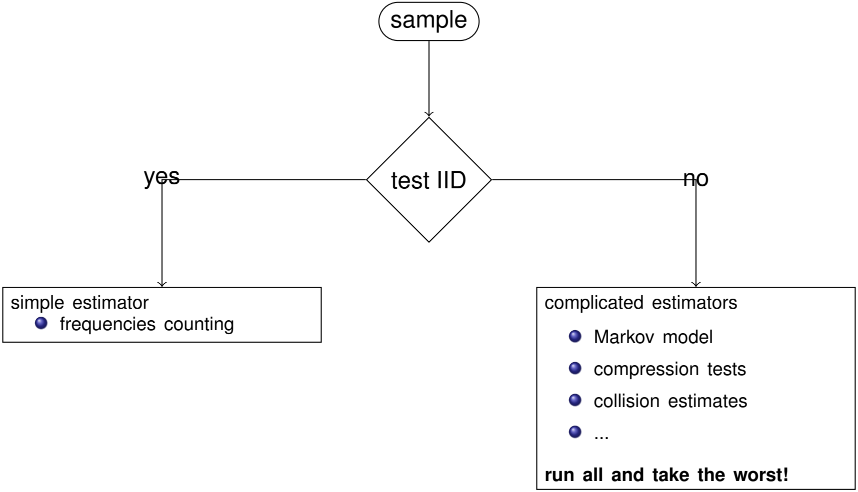
Can embed on-the-fly tests into small pieces of hardware?

- only relatively simple tests can be implemented (minimizing chip area)
- need to optimize variables (e.g. less storage for bounded quantities)
- need to precompute "heavy" functions (e.g. gaussian tails in CLT)
- implemented estimators may influence the source!

Some implementations have been done for FPGAs [SSR09].

Plan

Entropy estimation: overview



Entropy estimation: IID

Some physical sources can be modeled as IID (memoryless) [BL05]

- simplest: counting frequencies [KS11,TBKM16]
- possible low-memory implementations (online estimators [LPR11])
- further improvements possible, by comnbining concepts from streaming algorithms (frequency moments estimates) [AOST15] and entropy smoothing

Entropy estimation: testing IID

Testing the iid assumption roughly consists of the following steps

1. seek for bias
2. seek for long-term correlations
3. seek for short-term dependencies (stationarity)

Entropy estimation: non-IID - Markov model

- assume bits with k-th order dependencies (alphabet size = 2^k)
- estimate the initial distribution p_i (counting frequencies)
- estimate transition probabilities of the form

$$p_{i,j} \stackrel{def}{=} \Pr[X_n = i | X_{n-1} = j] =?$$

(counting occurrences of pairs j, i)

- address multiple testing $\alpha' = 1 - (1-\alpha)^{k^2}$ (transition probabilities)
- address sampling errors

$$p_{i,j} := \min(1, p_{i,j} + \delta_{i,j})$$

$\delta_{i,j}$ depends on occurrences of j, i, the sample size, the significance

- calculate entropy per sample using $(p_i))_i$ and $(p_{i,j}))_{i,j}$
  - Shannon Entropy in small chain $H = -\sum_i p_i \sum_j p_{i,j} \log p_{i,j}$
  - Renyi Entropy in small chain - transition matrix + dynamic programming [TBKM16]
  - Renyi Entropy in limit - eigenvalues of transition matrix powers [RAC99]

Entropy estimation under Markov model (II)

Estimation problems [TBKM16]

- can only capture small alphabets; for $k = 16$ bits, the matrix has 2^{32} entries to estimate! extensive lab tests use $k = 12$ [HKM12]
- give close bounds only for large probabilities (e.g. $p_{i,j} > 0.1$); estimates for small probabilities are crude (sampling issue: cannot easily hit a tiny set)

Practcal solution

Mitigate the sample size issues by preprocessing (e.g. ignorng less variable bits [TBKM16]).

Plan

Health tests

Required features of health tests

We expect the tests to be [KS11,TBKM16]

- efficient
- report failures quickly
- avoid false-alaram rates (the hypothesis: entropy decrease)
- cover major failures

- source gets stuck - many repetitions locally [TBKM16]
- big entropy decrease - too high frequencies of a block [TBKM16]
- frequencies of 4-bit words [KS11], genaralized [Sch01]

Low entropy detection

How to speed up health tests?

Frequency counting works under iid (otherwise 0101010101... passes the test). In this setting one can improve low-entropy detection by using Renyi entropy!

Estimators taylored to low-entropy regimes

Consider iid samples with at most k bit of collision entropy. Then estimating collision entropy per sample up to constant accuracy at the error probability ϵ needs

$$N = O\left(2^{k/2}\epsilon^{-2}\right)$$

samples [OS17]. This quantfies type II error under iid!
The result utilizes ideas developed in streaming algorithms.

Healt tests - summary

- online health tests: a new paradigm
- in practice: only simple tests requiring not too many samples
- not much literature on it

Plan

Conclusion

- Shannon entropy, under reasonable assumptions, may be used to approximate min-entropy; the higher entropy rate, the smaller error;
- in statistical tests, is almost impossible to quantify errors of type II (wrong TRNG); instead one develops many tests to cover a variety of ”bad” behaviors
- for health tests, one can take advantage of faster estimators for Renyi entropy

Research directions?

- implementing (provable secure) hardware-specific health tests and entropy evaluation
- theoretical analysis of health tests?
- more sophisticated approaches than well-known statistics (chi-squared, central limit theorem)?

Note: For a survery about security of TRNGs see also [Fis12].

Plan

References I

J. Acharya, A. Orlitsky, A. T. Suresh, and H. Tyagi. “The Complexity of Estimating Rényi Entropy”. In: *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*. 2015.

B. Barak, Y. Dodis, H. Krawczyk, O. Pereira, K. Pietrzak, F.-X. Standaert, and Y. Yu. “Leftover hash lemma, revisited”. In: *Annual Cryptology Conference*. Springer. 2011.

M. Bucci and R. Luzzi. “Design of testable random bit generators”. In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2005.

H. DEMIRHAN and N. BITIRIM. “Statistical Testing of Cryptographic Randomness”. In: *Journal of Statisticians: Statistics and Actuarial Sciences* 9 (1 2016).

Y. Dodis, K. Pietrzak, and D. Wichs. “Key derivation without entropy waste”. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2014.

References II

Y. Dodis and Y. Yu. “Overcoming weak expectations”. In: *Theory of Cryptography*. Springer, 2013.

V. Fischer. “A Closer Look at Security in Random Number Generators Design”. In: *Proceedings of the Third International Conference on Constructive Side-Channel Analysis and Secure Design*. COSADE’12. Darmstadt, Germany: Springer-Verlag, 2012.

M. Haahr. *random.org homepage*. Online; accessed 01-July-2016.

M. Hamburg, P. Kocher, and M. E. Marson. *Analysis of Intel’s Ivy Bridge digital random number generator*. 2012.

T. Holenstein and R. Renner. “On the Randomness of Independent Experiments”. In: *IEEE Transactions on Information Theory* 57.4 (2011).

H. Koga. “Characterization of the smooth Rényi Entropy Using Majorization”. In: *2013 IEEE Information Theory Workshop (ITW)*. 2013.

References III

W. Killmann and W. Schindler. *A proposal for: Functionality classes for random number generators. AIS 20 / AIS31*. 2011.

C. Lauradoux, J. Ponge, and A. Roeck. *Online Entropy Estimation for Non-Binary Sources and Applications on iPhone*. Research Report RR-7663. INRIA, June 2011.

U. M. Maurer. “A Universal Statistical Test for Random Bit Generators”. In: *J. Cryptology* 5.2 (1992).

K. MARTON and A. SUCIU. “On the interpretation of results from the NIST statistical test suite”. In: *SCIENCE AND TECHNOLOGY* 18.1 (2015).

X. Ma, F. Xu, H. Xu, X. Tan, B. Qi, and H.-K. Lo. “Postprocessing for quantum random-number generators: Entropy evaluation and randomness extraction”. In: *Phys. Rev. A* 87 (6 2013).

M. Obremski and M. Skorski. *Rényi Entropy Estimation, Revisited*. accepted to APPROX 2017. 2017.

.

References IV

- Z. Rached, F. Alajaji, and L. Campbell. *Rényi's Entropy Rate For Discrete Markov Sources*. 1999.
- Rukhin. Chapter 3 in: Randomness Through Computation: Some Answers, More Questions. 2011.
- R. Renner and S. Wolf. "Smooth Renyi entropy and Applications". In: *International Symposium on Information Theory, 2004*. 2004.
- W. Schindler. "Efficient online tests for true random number generators". In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2001.
- A. Secure Quantum Communication Group. *ANU homepage*. Online; accessed 01-April-2017.
- R. Santoro, O. Sentieys, and S. Roy. "On-the-Fly Evaluation of FPGA-Based True Random Number Generator". In: *2009 IEEE Computer Society Annual Symposium on VLSI*. ISVLSI '09. Washington, DC, USA: IEEE Computer Society, 2009.

References V

B. Schoenmakers, J. Tjoelker, P. Tuyls, and E. Verbitskiy. "Smooth Rényi Entropy of Ergodic Quantum Information Sources". In: *2007 IEEE International Symposium on Information Theory*. 2007.

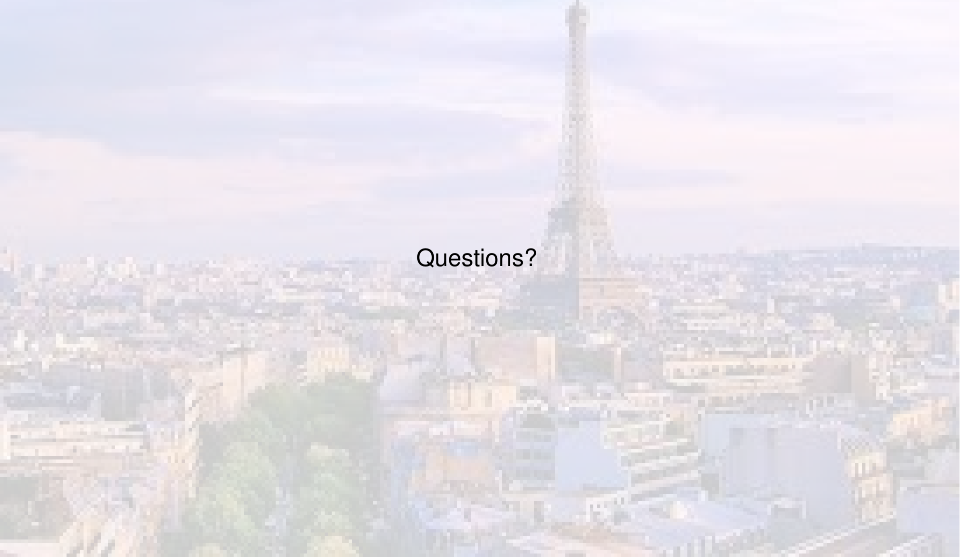
M. S. Turan, E. Barker, J. Kelsey, and K. McKay. "NIST DRAFT Special Publication 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation". In: http://csrc.nist.gov/publications/drafts/800-90/sp800-90b_second_draft.pdf. 2016.

F. Veljković, V. Rožić, and I. Verbauwhede. "Low-cost Implementations of On-the-fly Tests for Random Number Generators". In: *Proceedings of the Conference on Design, Automation and Test in Europe*. DATE '12. Dresden, Germany: EDA Consortium, 2012.

J. Walker. *HotBits homepage*. Online; accessed 01-July-2016.

M. Skorski. *Entropy of Independent Experiments, Revisited*. Apr. 2017. arXiv: 1704.09007 [cs.IT].

Thank you for your attention!



Questions?