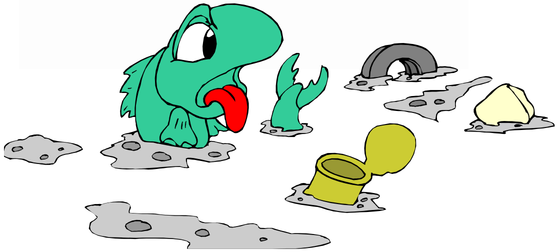
KEY DERIVATION WITHOUT ENTROPY WASTE

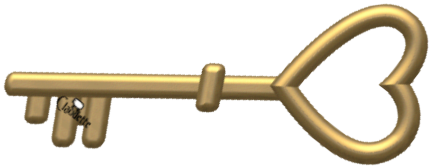


Yevgeniy Dodis

New York University

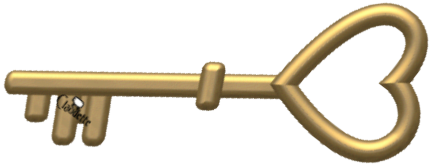
Joint work with Krzysztof Pietrzak and Daniel Wichs

Key Derivation



- Setting: application P needs m–bit secret key *R*
- Theory: pick *uniformly random* $R \leftarrow \{0,1\}^m$
- Practice: have "imperfect randomness" $X \in \{0,1\}^n$
  - physical sources, biometric data, partial key leakage, extracting from group elements (DH key exchange), …
- Need a "bridge": *key derivation function* (KDF) $h\colon \{0,1\}^n \rightarrow \{0,1\}^m$ s.t. $R = h(X)$ is "good" for P
  - … *only* assuming X has **"minimal entropy"** k

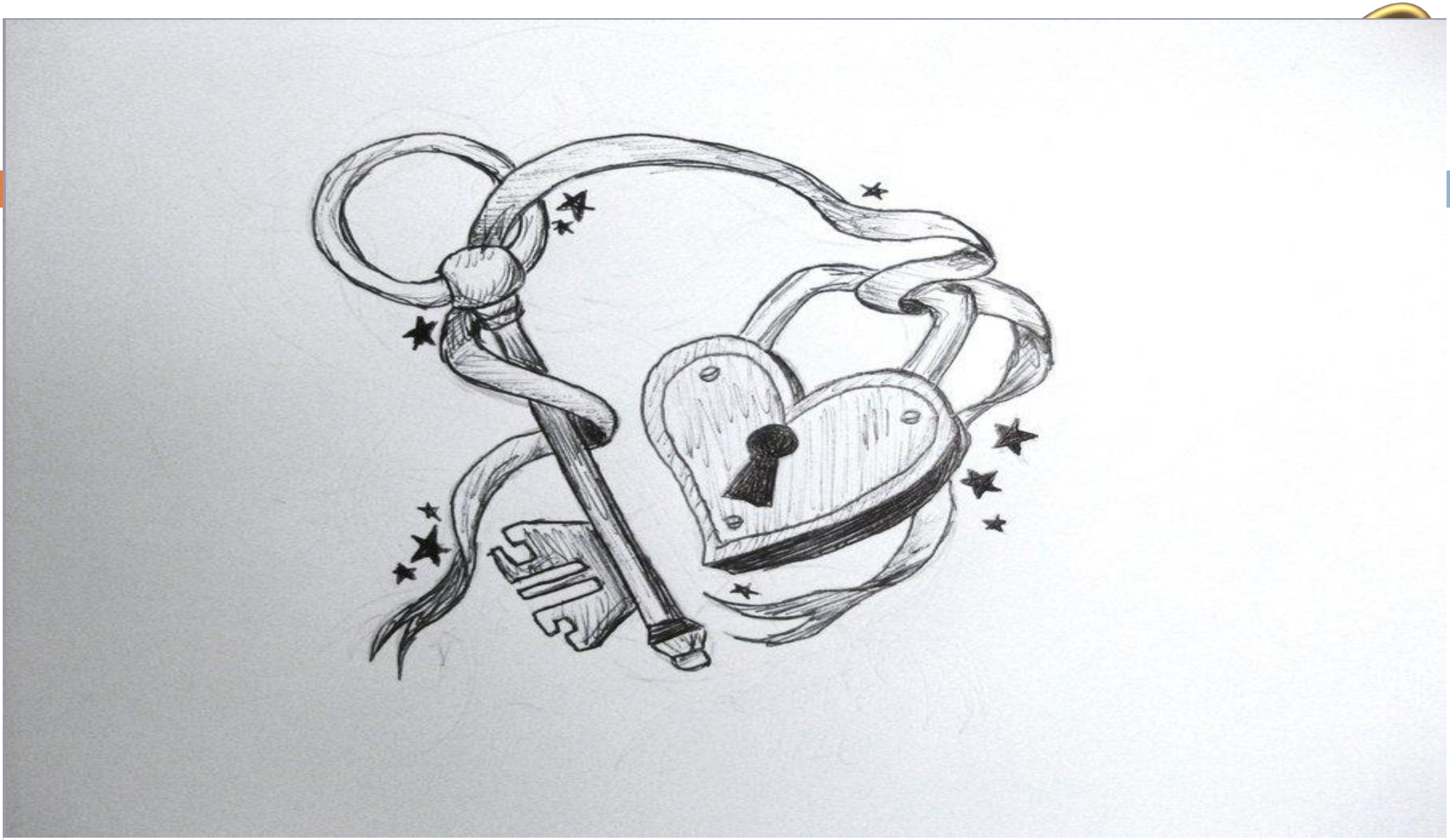
Formalizing the Problem



- Ideal Model: pick *uniform* $R \leftarrow U_m$ as the key
  - Assume P is ε–secure (against certain class of attackers ***A***)
- Real Model: use $R = h(X)$ as the key, where

Real Security ε' ≈ Ideal Security ε

- Goal: minimize k s.t. P is 2ε–secure using $R = h(X)$
  - Equivalently, minimize entropy loss $L = k - m$
  - (If possible, get information-theoretic security)
  - Note: we design h but must work for any (n, k)-source X



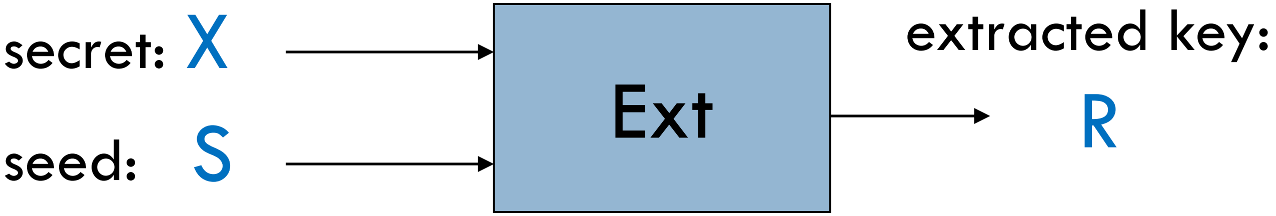
- Note: we design h but must work for any (n, k)-source X

Old Approach: Extractors

- <u>Tool:</u> Randomness Extractor [NZ96].
  - Input: a *weak secret* X and a *uniformly random seed* S.
  - Output: *extracted key* R = **Ext**(X; S).
  - R is uniformly random, even conditioned on the seed S.

$$(\mathbf{Ext}(X; S), S) \approx (\text{Uniform}, S)$$

- **Many uses in complexity theory and cryptography.**
  - Well beyond key derivation (de-randomization, etc.)



Old Approach: Extractors

- Tool: Randomness Extractor [NZ96].
  - Input: a *weak secret* X and a *uniformly random seed* S.
  - Output: *extracted key* R = **Ext**(X; S).
  - R is uniformly random, even conditioned on the seed S.

(**Ext**(X; S), S) ≈ (Uniform, S)

- (k,ε)-extractor: given any secret (n,k)-source X, outputs m secret bits "ε–fooling" any distinguisher ***D***:

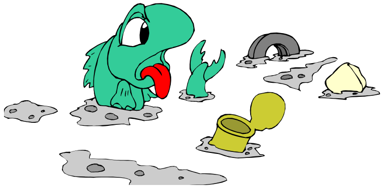
statistical distance

$$| \Pr[D(\mathbf{Ext}(X; S), S) = 1] - \Pr[D(U_m, S) = 1] | \le \varepsilon$$

Extractors as KDFs

- **Lemma**: for any ε-secure P needing an m–bit key, (k,ε)-extractor is a KDF yielding security $\varepsilon' \leq 2\varepsilon$
- **LHL** [HILL]: universal hash functions are (k,ε)-extractors where $\boxed{k = m + 2\log(1/\varepsilon)}$
- **Corollary**: For any P, can get entropy loss $2\log(1/\varepsilon)$

How Bad is $2\log(1/\varepsilon)$ Entropy Loss?



- Many sources do not have "extra" $2\log(1/\varepsilon)$ bits
  - Biometrics, physical sources, DH keys on elliptic curves
    - DH: low $k \Rightarrow$ smaller group size $\Rightarrow$ higher efficiency
  - AES-based P: $\varepsilon = 2^{-64}$, $m = 128 \Rightarrow k = 256 = 2m$ ☹
- Heuristic extractors have "no entropy loss": $k = m$
- <u>End Result</u>: practitioners prefer heuristic key derivation to provable key derivation [DGH$^+$,Kra]
- Can we provably match it, despite RT-bound?

Extractors as KDFs

- **Lemma**: for any ε-secure P needing an m–bit key, (k,ε)-extractor is a KDF yielding security $\varepsilon' \le 2\varepsilon$
- **LHL** [HILL]: universal hash functions are (k,ε)-extractors where $\boxed{k = m + 2\log(1/\varepsilon)}$
- **Corollary**: For any P, can get entropy loss $2\log(1/\varepsilon)$
- **RT-bound** [RT]: for any extractor, $k \ge m + 2\log(1/\varepsilon)$
  - entropy loss $2\log(1/\varepsilon)$ seems necessary ☹
  - … or is it?

Side-Stepping RT

- Do we need to derive statististically random *R*?
  - **Yes** for one-time pad ☹…
  - **No** for many (most?) other applications **P** ☺ !

this work

Series of works “beating” RT [BDK$^+$11,DRV12,DY13,DPW14]

For the first time match heuristic extractors!

Punch line: Difference between

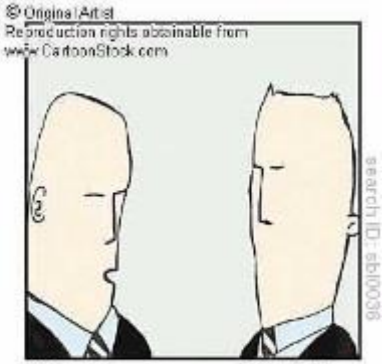
Extraction and Key Derivation !

New Approach/Plan of Attack

- **Step1.** Identify general class of applications P which work "well" with *any* high-entropy key R
  - Interesting in its own right !

- **Step2.** Build good *condenser*: relaxation of extractor producing high-entropy (but *non-uniform*!) derived key $R = h(X)$

Unpredictability Applications



"Massive unpredictablity is absolutely certain, maybe."

- Sig, Mac, OWF, … (*not* Enc, PRF, PRG, …)
- Example: unforgeability for Signatures/Macs
  - <u>Assume</u>: Pr[A forges with ≤ ε (= negl)
  - <u>Hope</u>: Pr[A forges with hig entropy key] ≤ ε'
- **<u>Lemma</u>**: for any ε-secu e unpredictability appl. P,

$$\mathbb{H}_\infty(R) \geq m - d \Rightarrow \boxed{\varepsilon' \leq 2^{d} \cdot \varepsilon}$$

Entropy deficiency

  - E.g., random *R* except first bit 0 ⇒ ε' ≤ 2ε

Plan of Attack

- ☑ **Step1.** Argue *any* unpredictability applic. P works well with (*only*) a high-entropy key R
  - $\mathbb{H}_\infty(R) \geq m - d \Rightarrow \varepsilon' \leq 2^d \cdot \varepsilon$

- ☐ **Step2.** Build good condenser: relaxation of extractor producing high-entropy (but *non-uniform*!) derived key $R = h(X)$

Randomness Condensers

THE CONDENSER

HUNTING DOWN THE ~~GOOD~~ random BITS

- (k,d,ε)-**condenser**: given $(n,\ k)$-source X, outputs m bits R "ε–close" to some $(m,\ m-d)$-source Y :

  $\boxed{(\mathbf{Cond}(X;\ S),\ S) \approx_{\varepsilon} (Y,\ S)}$ and $\boxed{\mathbb{H}_{\infty}(Y \mid S) \geq \boldsymbol{m} - \boldsymbol{d}}$

  - **Cond** + Step1 $\Rightarrow$ $\boxed{\varepsilon' \leq (1+2^{d}) \cdot \varepsilon}$
- $\boxed{\text{Extractors: } d = 0 \text{ but only for } k \geq m + 2\log(1/\varepsilon)}$ ☹
- <u>**Our Main Result**</u>: $\boxed{d = 1 \text{ with } k = m + \log\log(1/\varepsilon) + 4}$
  - KDF: $\log(1/\varepsilon)$-independent hash function works!

Balls and Bins

- Reduces to simple balls-and-bins game:
  - Throw 2^k balls into 2^m bins
  - Pick a random ball x
  - Lose if $|Bin(x)| > 2^d$.
- **Goal**: given d, m, $\varepsilon \Rightarrow$ min . $\Pr[\text{Lose}] \le \varepsilon$
- Easy calculation $\Rightarrow$ para eters of theorem if throw balls totally independently
- **Observation**: $\log(1/\varepsilon)$-independence suffices!

improve $|S|$ to $O(n \log k)$ using “gradual increase of independence” [CRSW11]

Unpredictability Extractors

- ☐ **Corollary**: provably secure KDF with entropy loss $\log\log(1/\varepsilon) + 4$ for all unpredictability applications

- ☐ Implicitly built $(k, \varepsilon, \varepsilon')$-unpredictability extractors:

  $\Pr[\boldsymbol{D}(U_m, S) = 1] \leq \varepsilon \Rightarrow \Pr[\boldsymbol{D}(\mathbf{UExt}(X;S), S) = 1] \leq \varepsilon'$

  - got $\varepsilon' = 3\varepsilon$ and $k = m + \log\log(1/\varepsilon) + 4$

Example: CBC-MAC, $\varepsilon = 2^{-64}$, $m = 128$

LHL: $k = 256$; **Now:** $k = 138 \cong 128$ **(RO)**

Unpredictability Extractors

- **Corollary**: provably secure KDF with entropy loss $\log\log(1/\varepsilon) + 4$ for all unpredictability applications

- Implicitly built $(k, \varepsilon, \varepsilon')$-unpredictability extractors:

  $\Pr[\boldsymbol{D}(U_m, S) = 1] \le \varepsilon \Rightarrow \Pr[\boldsymbol{D}(\mathbf{UExt}(X;S), S) = 1] \le \varepsilon'$

  - got $\varepsilon' = 3\varepsilon$ and $k = m + \log\log(1/\varepsilon) + 4$

- More generally, $\varepsilon' = \varepsilon \cdot (1 + \log(1/\varepsilon) \cdot 2^{m-k})$
  - E.g., $\varepsilon' = \varepsilon \cdot (1 + \log(1/\varepsilon))$ when $k = m$
  - CBC-MAC: $k = m = 128 \Rightarrow \varepsilon = 2^{-57.9}$ (vs. 2^{-63} RO)

Options for Avoiding RT

- ✓☐ Route 1: *implicitly* restrict ***D*** by considering special classes of applications P [BDK+11,DRV12,DY13,DPW14]
  - ☐ This paper: $L = \log\log(1/\varepsilon)$ for all unpredictability P
  - ☐ [BDK+11,DY13]: $L = \log(1/\varepsilon)$ for all "square-friendly" P (includes unpred. P, but also CPA enc, weak PRF, …)
- ☐ Route 2: efficiently samplable sources X [DGKM12]

Efficient Samplability?

- **Theorem** [DPW14]: efficient samplability of X does *not* help to improve entropy loss below
  - $2\log(1/\varepsilon)$ for all applications P (RT-bound)
    - Affirmatively resolves “SRT-conjecture” from [DGKM12]
  - $\log(1/\varepsilon)$ for all square-friendly applications P
  - $\log\log(1/\varepsilon)$ for all unpredictability applications P

Options for Avoiding RT

- ☑ Route 1: *implicitly* restrict ***D*** by considering special classes of applications P [BDK$^+$11,DRV12,DY13,DPW14]
  - This paper: $L = \log\log(1/\varepsilon)$ for all unpredictability P
  - [BDK$^+$11,DY13]: $L = \log(1/\varepsilon)$ for all "square-friendly" P (includes unpred. P, but also CPA enc, weak PRF, …)
- ☑ Route 2: efficiently samplable sources X [DGKM12]
- ☐ Route 3: computat. bounded ***D*** (pseudo-randomness)

Computational Assumptions?

- **Theorem** [DGKM12 ,DPW14]: ~~SRT-conjecture ⇒~~ efficient **Ext** beating RT-bound for all *computationally bounded* ***D*** ⇒ **OWFs exist**
- How far can we go with OWFs/PRGs/…?
  - One of the main open problems
- Current Best [DY13]: "computational" extractor with entropy loss $2\log(1/\varepsilon) - \log(1/\varepsilon_{prg})$
  - "Computational" condenser?

Summary

- Difference between extraction and KDF
  - $\log\log(1/\varepsilon)$ loss for all unpredictability apps
  - $\log(1/\varepsilon)$ loss for all square-friendly apps

    (+ motivation to study "square security")
- Efficient samplability does not help ☹
- Good computational extractors require OWFs ☹
- **Main challenge**: better "computational" KDFs

Questions?