T-79.515 Cryptography: Special Topics

February 24th, 2005

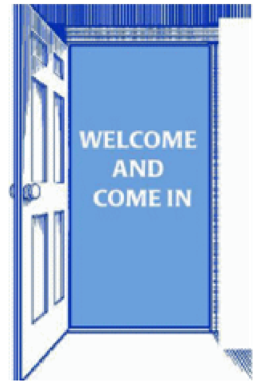
Fuzzy Extractors: Generating Strong Keys From Noisy Data

Mikko Kiviharju
Helsinki University of Technology
mkivihar@cc.hut.fi

Overview

- Motivation and introduction
- Preliminaries and notation
- General theory
- Examples (constructions)
- Conclusion

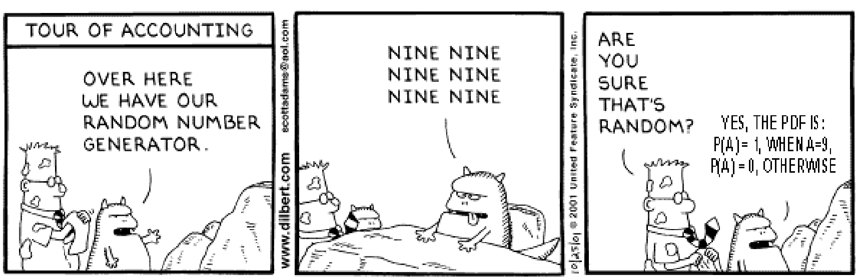
Motivation: Noisy Data



Motivation: Noisy Data

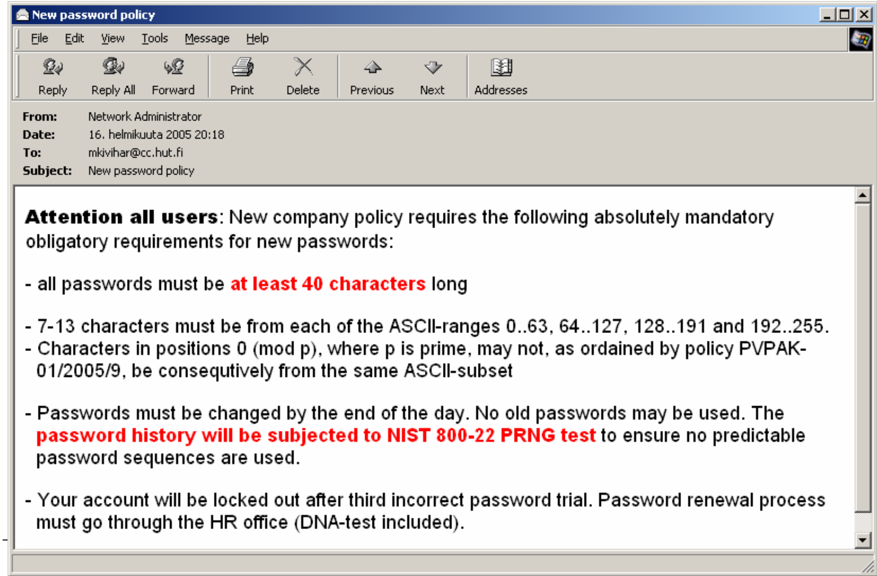
Motivation: Noisy Data

Motivation: non-uniform distributions



Randomness for cryptographic applications needs to be distributed nearly uniformly – unpredictability is lost otherwise.

Noisy Data AND non-uniform distributions



New password policy

File Edit View Tools Message Help

Reply | Reply All | Forward | Print | Delete | Previous | Next | Addresses

From: Network Administrator
Date: 16. helmikuuta 2005 20:18
To: mkivihar@cc.hut.fi
Subject: New password policy

Attention all users: New company policy requires the following absolutely mandatory obligatory requirements for new passwords:

- all passwords must be **at least 40 characters** long

- 7-13 characters must be from each of the ASCII-ranges 0..63, 64..127, 128..191 and 192..255.
- Characters in positions 0 (mod p), where p is prime, may not, as ordained by policy PVPAK-01/2005/9, be consequtively from the same ASCII-subset

- Passwords must be changed by the end of the day. No old passwords may be used. The **password history will be subjected to NIST 800-22 PRNG test** to ensure no predictable password sequences are used.

- Your account will be locked out after third incorrect password trial. Password renewal process must go through the HR office (DNA-test included).

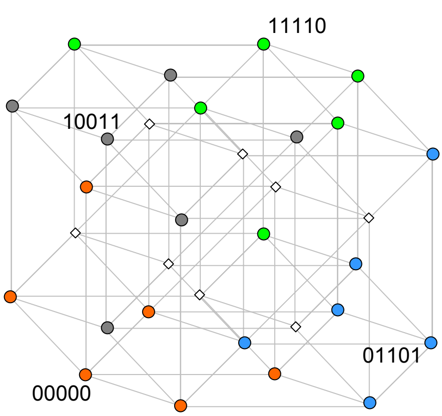
Introduction

- Natural world and applications of cryptology into real world noisy and non-uniform
- *Coding theory* deals with noisy data
- *Extractors* handle nonuniformity of random variables.
- *Fuzzy extractors* combine elements from both => error-tolerant extractors
- Applications
  - Biometric data, user-friendly passwords, privacy amplification, fast authentication (short seeds)

Introduction: concepts

- *Biometric embedding:* a function to construct F.E:s to another metric space from its ”home space” (metric space)
- *Secure Sketch:* function to produce error-tolerant public values from private data with upper bounds for entropy loss.
- *Strong Extractor:* prob. function to extract uniform randomness from a random variable.
- *Key-encapsulation:* technique of PKCs of agreeing over a secret key by not directly communicating the secret key
- *Random pairwise independent hash functions:* hash functions with the property that the *r.v.*s associated with them are both independent and have uniform distribution

Preliminaries: coding theory



n = 5 (five-bit strings)

K = 4 (four classes, four codewords)

k = $\log_2 K$ = 2 (dimensions)

d = 3 (minimum distance of codewords, 3-1 is the largest number of errors that can always be detected)

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor$$ largest number of errors that can always be corrected

For Hamming metric: [n,k,2t+1] = [5,2,3]-code

Preliminaries: probability and entropy

- Joint probability of variables noted as $\langle \cdot, \cdot, \cdot, \ldots \rangle$
- Entropies
  - Shannon entropy H (not used here)
  - Renyi entropy H_2 (not used here)
  - Minimum entropy $H_\infty(X) = -\log_2(\max_x P(X = x))$
  - Average (conditional) min entropy:

    $$\tilde{H}_\infty(X \mid Y) = -\log_2\left(E_{y \leftarrow Y}\left[2^{-H_\infty(X|Y=y)}\right]\right)$$

    (modified version in use because of statistical distance from U_ℓ)

Notes on: *"Preliminaries: probability and entropy"*

Average min-entropy of A given B is at most l lower than min-entropy of A.

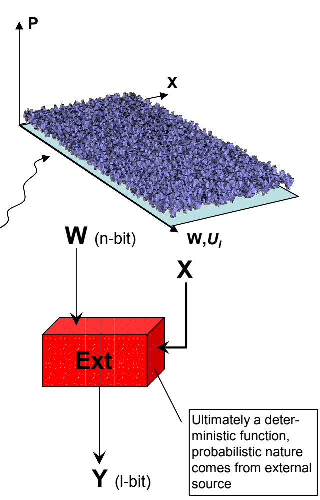
The statistical distance from uniform distribution has a so-called left-over has lemma, which upper-bounds the SD of pairwise independent hash functions, and this bound has exponentials.

Preliminaries: metric spaces

- Metric on probability distributions / random variables: $d(X,Y) = \mathbf{SD}(X,Y) = \frac{1}{2}\sum_v \left|P(X=v) - P(Y=v)\right|$
- Hamming metric on binary strings:
  $$d(x,y) = weight(x \oplus y)$$
- Set metric on any finite sets:
  $$d(X,Y) = \frac{1}{2}\left|X \Delta Y\right|$$
- Edit distance:
  - The number of `Ins` and `Del` – operation required to transform a (binary) string to another

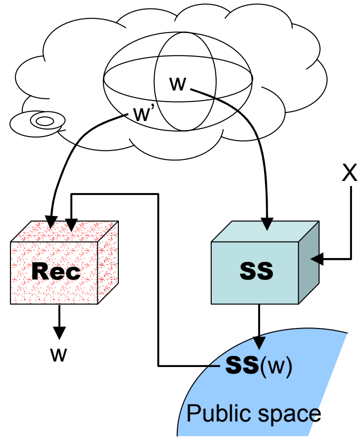
Preliminaries: extractors

- (Efficient) strong extractors: prob. polytime functions $Ext : \{0,1\}^n \to \{0,1\}^l$
- Four params:
  - source and extracted string lengths,
  - lower bound *m'* on min-entropy of W
  - upper bound ε on difference to U_ℓ
- Restriction on extracted strings:
  $\mathbf{SD}\left(\langle Ext(W;X), X\rangle, \langle U_\ell, X\rangle\right) \le \varepsilon$
- Upper bound on # of nearly random bits extracted (Radhakrishnan):
  $m' - 2\log_2(1/\varepsilon) + O(1)$



General theory: secure sketches

- Two functions:
  - probabilistic **SS** to produce a public "sketch" from a private value, i.e. a password
  - deterministic **Rec** to recover the original value with the help of the sketch and a value reasonably close to the original
- Limits the amount of information revealed with the sketch



General theory: secure sketches

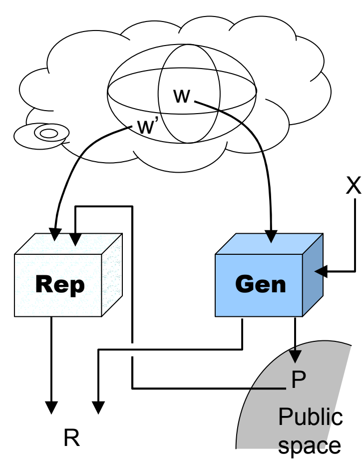
- ($\mathcal{M}$,*m*,*m'*,*t*)-secure sketch is a randomized map $\mathbf{SS}:\mathcal{M}\rightarrow\{0,1\}^*$, such that
  - there is a function $\mathbf{Rec}:\mathcal{M}\times\{0,1\}^*\rightarrow\mathcal{M}$ for which $\forall(w,w'\in\mathcal{M},d(w,w')\leq t):\mathbf{Rec}(w',\mathbf{SS}(w))=w$
  - for every *r.v* W over $\mathcal{M}$, for which $H_\infty(W)=m$, $\tilde{H}_\infty(W|\mathbf{SS}(W))\geq m'$ (*m'*<*m*)
- Example: for some code C and uniform random variable X, define $\mathbf{SS}(X;W)=W\oplus C(X)$

Notes on: *"General theory: secure sketches"*

Here, W is taken over the private metric space, and X is the usual "external" randomness inherent in the probabilistic function SS. The error-tolerance comes from the coding function – the error-correction capabilities are transmitted to the actual private string via the XOR-operation.

General theory: fuzzy extractors

- Two procedures:
  - probabilistic **Gen** to produce a public string and an extracted string (used i.e. as a key in key-encapsulation mechanisms)
  - deterministic **Rep** to recover the extracted string with the help of the public value and a value reasonably close to the original
- Constrains the distribution of the extracted string close to uniform.
- Does not, per se, limit the information given out in the public string



General theory: fuzzy extractors

- ($\mathcal{M}$,*m*,*l*,*t*,*ε*) fuzzy extractor is given by two procedures (**Gen**, **Rep**).
- $\mathbf{Gen}: \mathcal{M} \rightarrow \{0,1\}^l \times \{0,1\}^p$ and for any p.d *W* over $\mathcal{M}$, with $H_\infty(W) = m$ and $\mathbf{Gen}(W) \rightarrow \langle R, P \rangle$, it holds that $\mathbf{SD}(\langle R, P \rangle, \langle U_\ell, P \rangle) \leq \varepsilon$
- $\mathbf{Rep}: \mathcal{M} \times \{0,1\}^p \rightarrow \{0,1\}^l$ and $\forall (w, w' \in \mathcal{M}; d(w, w') \leq t)$ $\mathbf{Rep}(w', P) = R$
- Example: in constructions…

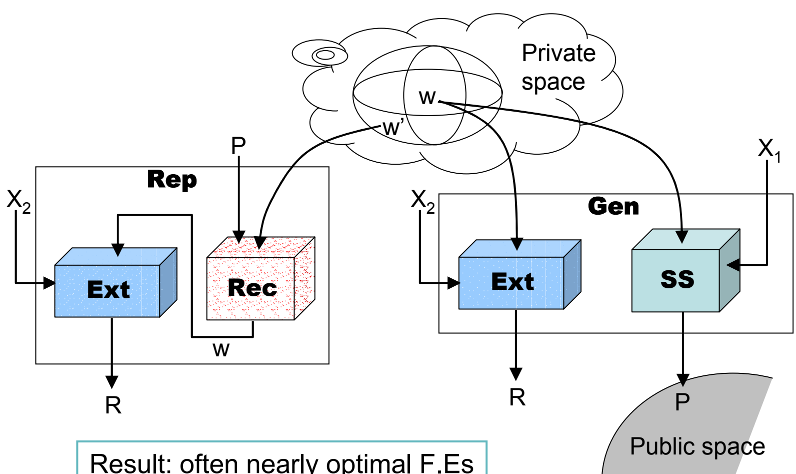
Notes on: "General theory: fuzzy extractors"

Actually, P is not fixed to any particular set. In practice, it could be a binary string, e.g. coming from a secure sketch.

Theory: constructing F.Es

- Fuzzy extractors do not restrict the amount of information revealed in the public string P.
- Utilize secure sketches and strong extractors
- Idea:
  - secure sketches to produce the public string P
  - strong extractors to produce the ”key material”, R
- To produce ($\mathcal{M},m,l,t,\varepsilon$) fuzzy extractor (where . $w \in \mathcal{M}$ can be represented with n bits), pick
  - ($\mathcal{M},m, l+2\log(1/\varepsilon), t$)-secure sketch
  - ($n, l+2\log(1/\varepsilon), l,\varepsilon$)-strong extractor (2 instances)
  - Entropy loss of $2\log(1/\varepsilon)$ is minimal, and due to pairwise-independent hash functions

Theory: constructing F.Es



Result: often nearly optimal F.Es
(w.r.t entropy loss; proof omitted here)

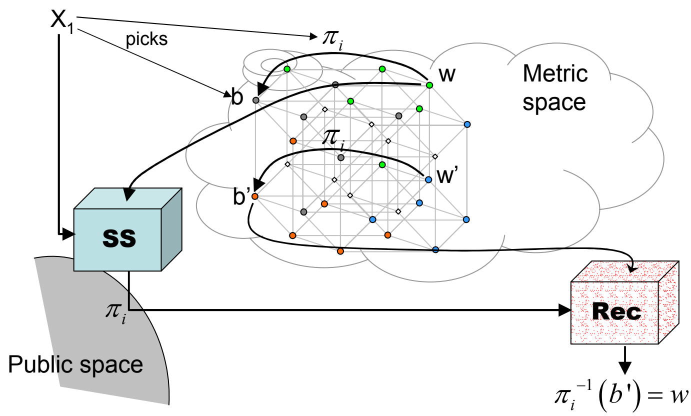
Theory: transitive metric spaces

- Define a set of isometric permutations $\Pi = \{\pi_i\}_{i \in I}$ on a metric space $\mathcal{M}$
- If $\forall (a, b \in \mathcal{M}) \exists (\pi_i \in \Pi) : \pi_i(a) = b$, both $\mathcal{M}$ and π are called *transitive:* If $(\pi_i(a) = b \wedge \pi_k(b) = c) \Rightarrow (\exists m : \pi_m(a) = c)$
- Example: Hamming spaces with the set of all shifts: $\pi_x(w) = w \oplus x$
- Secure sketches can be built on any transitive metric spaces:
  - a random permutation of a random codeword as the sketch function
  - recovery function is the inverse permutation of the decoded trial word
  - entropy loss: $|"\pi"| - \log_2 K$

Notes on: *"Theory: transitive metric spaces"*

K is the number of legal codewords in the code, "pi" is the representation on the permutation in canonical format (in cycles, lowest-numbered first, encoded as bits). This quantity is small if the family of transitive isometries is small and the code is dense.
Entropy loss is from counting: one gives out information about pi (which reduces entropy with the number of bits used in its encoding), but one would still have to guess b' such that it belongs to the right codeword-ball – and there are K codewords, encoded in log(K) bits.
Here, as in the Hamming code, the efficiency very much depends on the efficiency of the underlying code. Linear codes are fast and good in this respect.

Theory: transitive metric spaces



Notes on: *"Theory: transitive metric spaces"*

This works, because when d(w,w')<t, and due to isometry d(pi(w),pi(w'))= d(b,b')<t, which can be corrected by the code, thus giving out the original w

Theory: biometric embeddings

- How to construct fuzzy extractors, if the metric space is not transitive?
- Solution: *embed* the problematic space into a more friendly one
- Limit the min-entropy and deviations from uniform distribution of the resulting F.E
- Note: particular embeddings do not necessarily work for secure sketches (embedding function needs to be efficiently invertible to return the output of **Rec** to source space)

Theory: biometric embeddings

- Defined by $f:\mathcal{M}_1 \rightarrow \mathcal{M}_2$ to be a (t_1,t_2,m_1,m_2)-biometric embedding, if
  - $\forall\left(w_1, w_1' \in \mathcal{M}_1, d\left(w_1, w_1'\right) \leq t_1\right): d\left(f\left(w_1\right), f\left(w_1'\right)\right) \leq t_2$
  - For any W_1 on $\mathcal{M}$: $H_\infty\left(W_1\right) \geq m_1 \Rightarrow H_\infty\left(W_2\right) \geq m_2$
- Now, if (**Gen**(*), **Rep**(*,*)) is a ($\mathcal{M}_2,m_2,l,t_2,\varepsilon$)-F.E, then (**Gen**(f(*)), **Rep**(f(*),*)) is a ($\mathcal{M}_1,m_1,l,t_1,\varepsilon$)-F.E

Constructions: Hamming (1/3)

- Fuzzy commitment (Juels, Wattenberg) → directly applicable for secure sketches: $\mathbf{SS}(X;W) = W \oplus C(X)$
- When C is linear → syndrome (of *n-k* bits) revealed → information leak (entropy loss) = *n-k*
- Show that this is true of nonlinear codes as well:
  - Define a [n,k,2t+1] code C with decoder D, any m, **SS** as above, and let $v = \mathbf{SS}(w, X) = w \oplus C(x)$
  - If $d(w,w') \le t$, then $D(w' \oplus v) = D(w' \oplus w \oplus C(x)) = x$ since D can correct up to t errors
  - Thus $\mathbf{Rec}(w',v) = v \oplus C(D(w' \oplus v)) = w \oplus C(x) \oplus C(x) = w$

Constructions: Hamming (2/3)

- (cont'd) for entropy, let $H_\infty(W) = m$
- Then for (*X,W*) the min-entropy is *m+k, k* is from the number of code-words in C.
- **SS**(*W*) is n-bit → reveals n bits of information
- *W* and **SS**(*W*) uniquely determine the value of X → the presence of X does not increase the average entropy
- $\tilde{H}_\infty(W \mid \mathbf{SS}(W)) = \tilde{H}_\infty(W, X \mid \mathbf{SS}(W)) \ge m + k - n$
- Yields a ($\mathcal{M}$,*m*,*m*+*k*-*n*,*t*)-secure sketch

Constructions: Hamming (3/3)

- How about F.Es?
- A straightforward from ”fuzzy commitment”, by setting *R*=*X*, *P*=*V*, $V = W \oplus C(X)$ and $\mathbf{Rep}(W',V) = D(V \oplus W')$
- *W* must be uniform, though (revealed *V* is tied to *R* via *W* and the **Gen**-procedure)
- However, using **SS**, we can have a general F.E for any [n,k,2t+1]-code with parameters $\left(\mathcal{M}, m, m+k-n-2\log_2\left(\frac{1}{\varepsilon}\right), t, \varepsilon\right)$

Constructions: Set difference (1/4)

- Metric can be viewed as Hamming distance, if the ”weight” of the representation of the set is not too ”small”. (Size of the universe of the set is small)
- For small universes, several constructions work:
  - ”Fuzzy vaults” by Juels and Sudan
  - Encoding as bitstrings – reverting to Hamming
  - Using the transitivity of the `SetDiff`-metric for a permutation-based sketch
- Permutation based sketch allows optimal entropy loss but is in practice not implemented
- Fuzzy vaults achieve poor parameters: practice currently favors conversion to Hamming

Notes on: "Constructions: Set difference (1/4)"

Efficient implementations of constant-weight-codes are not known yet. In general, the whole concept, or limitation of codes to constant weight seems to be new area of research.

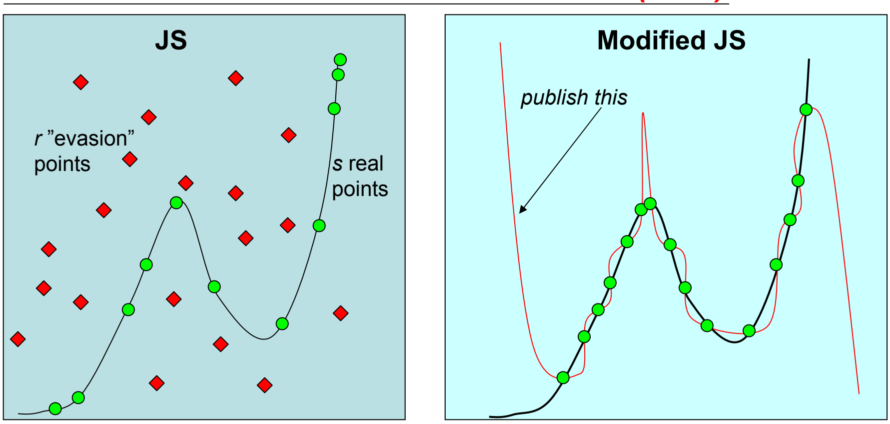
Constructions: Set difference (2/4)

- Permutation based sketch
  - use the set of all permutations as the isometric transitive transformation
  - choose any [*n,k,d*]-code, where *n* is the size of the universe
  - for a given set *A* of size *s*, choose a random *B* from the *selected code*.
  - choose a random matching between A and B and their complements → a random permutation $\pi : [n] \to [n]; \pi(A) = B$
  - output $\mathbf{SS}(A) = \pi$
  - Set $\mathbf{Rec}(\pi, A') = D\left(\pi^{-1}(A')\right)$
- Results in a $\left(\mathcal{M}, m, m-k+\log_2\binom{n}{s}, t\right)$ -secure sketch

Constructions: Set difference (3/4)

- Large universes: permutation finding inefficient
  (have to find a suitable images for the *complements* as well)
- Three main sketches: fuzzy vault (JS-scheme), modified JS-scheme and BCH-codes (omitted here)
- Both JS-based schemes encode the members of the universe as members of GF(p^k) ([n] is assumed to have exactly p^k members)
- The public sketch is information about a random polynomial (over the field) evaluated on the members of the private set

Constructions: Set difference (4/4)



Entropy loss for JS: $2t\log_2(n)+\log_2\binom{r}{s}-\log_2\binom{n}{s}; n \gg r$

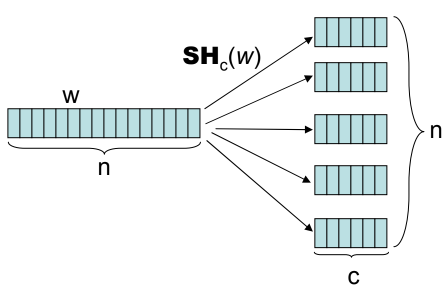
Entropy loss for modified JS: $2t\log_2(n)$

Constructions: Edit distance (1/2)

- Edit metric is not known to be transitive → normal sketch constructions do not work
- Embedding edit metric with relative distance-preserving embeddings (such as low-distortion embeddings into Hamming metric) are not known (in fact, some lower distortion bounds are even proven (by Andoni et al.))
- Solution *biometric embeddings*
- Looser restrictions on preserving the distances; for F.Es it is sufficient that ”close” points do not become ”distant”.

Constructions: Edit distance (2/2)

- A suitable biometric embedding is the c-shingling map $\mathbf{SH}_c(w)$:



Biometric embedding:

$$\left(t, ct, m, m - \frac{n \log_2 n}{c}\right)$$

Resulting fuzzy extractor (optimized): $\left(\mathbf{Edit}(n), m_1, \frac{m_1}{2} - 2\log\frac{1}{\varepsilon}, \frac{m^3}{16n^2 \log_2^2 n}, \varepsilon\right)$

Conclusion

- Error-tolerant extractors are very useful in natural settings, especially authentication
- Fuzzy extractors combine two important properties: uniformity and error-tolerance
- *Efficiency* stressed throughout the construction, but the theory doesn't contribute anything for efficiency, instead relies on efficiency of the underlying primitives
- More research needed in actual constructions and different metrics
- Other constructions beyond fuzzy extractors combining even more useful properties?