Algorithms of Information Security: Error-correcting codes III

Faculty of Information Technology
Czech Technical University in Prague

October 19, 2022

Fuzzy extractors - motivation

- Fuzzy extractors present an approach for handling secret biometric data in cryptographic applications.
- Fuzzy extractor extracts a uniformly random string R from its input w in a noise-tolerant way.
- If the input w changes to w', which is only "slightly" different from w, the string R can be reproduced exactly.
- Fuzzy extractors are used for encryption and authentication, using biometric input as a key.

Fuzzy extractors - basic definitions and notations

- U_ℓ denotes the uniform distribution $\{0,1\}^\ell$.
- If a function f is randomized, we denote by $f(x,r)$ the result of computing f on input x with randomness r.
- *Predictability* of a random variable A is $\max_a \mathbb{P}[A=a]$.
- *min-entropy* $H_\infty(A)$ is $-\log(\max_a \mathbb{P}[A=a])$. $H_\infty(A)$ can be viewed as the “worst-case” entropy.
- A random variable with min-entropy at least m is called an m-source.

Fuzzy extractors - basic definitions and notations

- Consider now a pair of (possibly correlated) random variables A and B. If the adversary finds out the value b of B, then the predictability of A becomes $\max_a \mathbb{P}[A = a|B = b]$.
- On average, the adversary's chance of success in predicting A is $\mathbb{E}_{b \leftarrow B}\left[\max_a \mathbb{P}[A = a|B = b]\right]$. (We are taking the average over B (which is not under adversarial control), but the worst case over A).
- *Conditional min-entropy*
$$\widetilde{H}_\infty(A|B) \stackrel{\text{def}}{=} -\log \mathbb{E}_{b \leftarrow B}\left[\max_a \mathbb{P}[A = a|B = b]\right] =$$
$$= -\log \mathbb{E}_{b \leftarrow B}\left[2^{-H_\infty(A|B=b)}\right]$$

Conditional min-entropy

- Conditional min-entropy satisfies a *weak chain rule*, namely, revealing any λ bits of information about A can cause its entropy to drop by at most λ.
- The definition of conditional min-entropy is suitable for cryptographic purposes and, in particular, for extracting "nearly" uniform randomness from A.
- "nearly" here corresponds to the *statistical distance* between two probability distributions A and B, defined as $SD[A, B] = \frac{1}{2}\sum_v |\mathbb{P}[A = v] - \mathbb{P}[B = v]|$.
- SD can be interpreted as a measure of distinguishability. We write $A \approx_\varepsilon B$ to say that A and B are at distance at most ε.

Strong extractor

Definition

A randomized function $Ext : \mathcal{M} \to \{0,1\}^\ell$ with randomness of length r is an (m, ℓ, ε)-strong extractor if for all m-sources W on $\mathcal{M}$, $(Ext(W; I), I) \approx_\varepsilon (U_\ell, U_r)$, where $I = U_r$ is independent of W.

We think of the output of the extractor as a key generated from $w \leftarrow W$ with the help of a seed $i \leftarrow I$.

Lemma

Strong extractors can extract at most $\ell = m - 2\log(1/\varepsilon) + \mathcal{O}(1)$ *bits from (arbitrary)* m*-sources.*

Properties of strong extractor

Definition

$Ext(w;i)$ with an ℓ-bit output is universal if for each $w_1 \neq w_2$, $\mathbb{P}_i[Ext(w_1;i) = Ext(w_2;i)] = 2^{-\ell}$.

If elements of $\mathcal{M}$ can be represented as n-bit strings, *universal hash functions* can be built using seeds of the length n: for instance, simply view w and x as members of $GF(2^n)$ and let $Ext(w;x)$ be ℓ least significant bits of wx.
Using universal hash functions we can extract
$\ell = m + 2 - 2\log(1/\varepsilon)$ bits:

Lemma

Let for any E (possibly dependent on W), if $\widetilde{H}_\infty(W|E) \geq m$ and $\ell = m + 2 - 2\log(1/\varepsilon)$, then $(Ext(W;I), I, E) \approx_\varepsilon (U_\ell, I, E)$.

Secure sketches and fuzzy extractors (Fig. from [1])

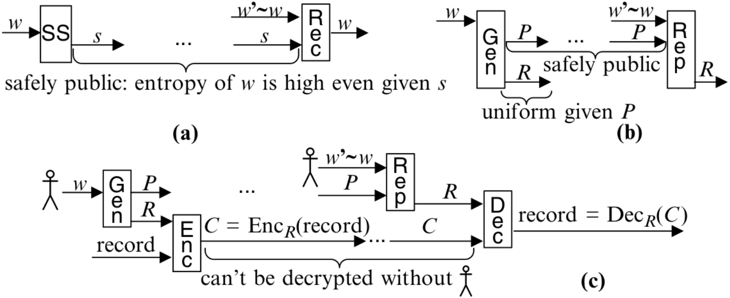


Fig. 5.1. (a) Secure sketch; **(b)** fuzzy extractor; **(c)** a sample application. The user encrypts a sensitive record using a key R extracted from biometric w via a fuzzy extractor; both P and the encrypted record may be sent or stored in the clear.

Secure sketch

Let $\mathcal{M}$ be a metric space with distance function dis. Informally, a *secure sketch* enables recovery of a string $w \in \mathcal{M}$ from any "close" string $w' \in \mathcal{M}$ without leaking too much information about w.

Definition

An $(m, \widetilde{m}, t)$-secure sketch is a pair of efficient randomized procedures (SS, Rec) ("sketch" and "recover") such that the following hold:

1. The sketching procedure SS on input $w \in \mathcal{M}$ returns a string $s \in \{0,1\}^*$. The recovery procedure Rec takes an element $w' \in M$ and $s \in \{0,1\}^*$.
2. Correctness: If $dis(w, w') \leq t$, then $Rec(w', SS(w)) = w$.
3. Security: For any m-source W over $\mathcal{M}$, the min-entropy of W given s is high: For any (W, E), if $\widetilde{H}_\infty(W|E) \geq m$, then $\widetilde{H}_\infty(W|SS(W), E) \geq \widetilde{m}$.

Fuzzy extractor -informal

Fuzzy extractors do not recover the original input but, rather, enable generation of a close-to-uniform string R from w and its subsequent reproduction given any w' close to w.

The reproduction is done with the help of the helper string P produced during the initial extraction; yet P need not remain secret, because R is nearly uniform even given P.

Fuzzy extractor

Definition

An $(m, \ell, t, \varepsilon)$-fuzzy extractor is a pair of efficient randomized procedures (Gen, Rep) ("generate" and "reproduce") such that the following hold:

1. Gen, given $w \in \mathcal{M}$, outputs an extracted string $R \in \{0,1\}^\ell$ and a helper string $P \in \{0,1\}^*$. Rep takes an element $w' \in \mathcal{M}$ and a string $P \in \{0,1\}^*$.
2. Correctness: If $dis(w, w') \leq t$ and $(R, P) \leftarrow Gen(w)$, then $Rep(w', P) = R$.
3. Security: For all m-sources W over $\mathcal{M}$, the string R is nearly uniform even given P; that is, if $\widetilde{H}_\infty(W|E) \geq m$, then $(R, P, E) \approx_\varepsilon (U_\ell, P, E)$.

Fuzzy extractor - notes

- *Entropy loss* of a secure sketch (resp. fuzzy extractor) is $m - \widetilde{m}$ (resp. $m - \ell$).
- the nearly-uniform random bits output by a fuzzy extractor can be used in a variety of cryptographic contexts that require uniform random bits (e.g., for secret keys).
- The slight nonuniformity of the bits may decrease security, but by no more than their distance ε from uniform.
- By choosing ε sufficiently small (e.g., 2^{-100}) one can make the reduction in security irrelevant.
- If more than ℓ random bits are needed, then pseudorandom bits can be obtained by inputting R to a pseudorandom generator.

Secure Sketches Imply Fuzzy Extractors

Given a secure sketch, we can always construct a fuzzy extractor that generates a key of length almost $\widetilde{m}$ by composing the sketch with a good (standard) strong extractor. The following lemma is stated for universal hash functions:

Lemma

Suppose we compose an $(m, \widetilde{m}, t)$-secure sketch (SS, Rec) for a space $\mathcal{M}$ and a universal hash function $Ext : \mathcal{M} \to \{0,1\}^*$ as follows: In Gen, choose a random i and let $P = (SS(w), i)$ and $R = Ext(w; i)$; let $Rep(w', (s, i)) = Ext(Rec(w', s), i)$. The result is an $(m, \ell, t, \varepsilon)$-fuzzy extractor with $\ell = \widetilde{m} + 2 - 2\log(1/\varepsilon)$.

Secure Sketches Imply Fuzzy Extractors (Fig. from [1])

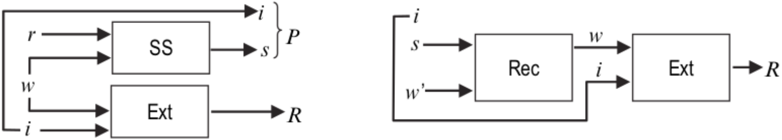


Figure: Illustration of the previous lemma.

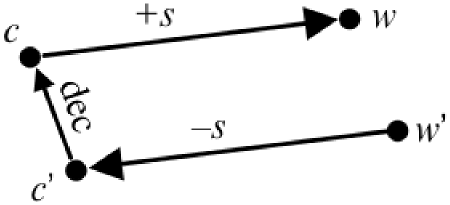
Construction of secure sketch for Hamming distance

- Constructions of secure sketches are based on error-correcting codes.
- To obtain a secure sketch for correcting Hamming errors over $\mathbb{F}^n$ ($\mathbb{F}$ is a field), we start with a $[n, k, 2t+1]$ error-correcting (linear) code C.
- The idea is to use C to correct errors in w, even though w may not be in C, by shifting the code so that a codeword matches up with w and storing the shift as the sketch.

Construction of secure sketch for Hamming distance

Definition

Construction 1 (Code-offset construction). On input w, select a uniformly random codeword $c \in C$, and set $SS(w)$ to be the shift needed to get from c to w: $SS(w) = w - c$. To compute $Rec(w', s)$, subtract the shift s from w' to get $c' = w' - s$, decode c' to get c (note that since $dis_{\mathrm{Ham}}(w', w) \leq t$ then $dis_{\mathrm{Ham}}(c', c) \leq t$), and compute w by shifting back to get $w = c + s$.



Construction of fuzzy extractor for Hamming distance

Theorem

For any m*, given an* $[n, k, 2t+1]$ *error-correcting code, Construction 1 is an* $(m, m-(n-k)\log|\mathbb{F}|, t)$*-secure sketch for the Hamming distance over* $\mathbb{F}^n$*. Combined with Lemma "Secure Sketches Imply Fuzzy Extractors", this construction give, for any* ε*, an* $(m, m-(n-k)\log|\mathbb{F}|+2-2\log(1/\varepsilon), t, \varepsilon)$ *fuzzy extractor for the same metric.*

Construction of fuzzy extractor for Hamming distance

- The trade-off between the error tolerance and the entropy loss depends on the choice of error-correcting code.
- For large alphabets ($\mathbb{F}$ is a field of size $\geq n$), one can use Reed-Solomon codes to get the optimal entropy loss of $2t \log |\mathbb{F}|$.
- No secure sketch construction can have a better trade-off between error tolerance and entropy loss than *Construction 1* (there are more constructions, see [1])), as searching for better secure sketches for the Hamming distance is equivalent to searching for better error-correcting codes.

Source

[1] Tuyls, P., Škoric, B., & Kevenaar, T. (Eds.). (2007). *Security with noisy data: on private biometrics, secure key storage and anti-counterfeiting*. Springer Science & Business Media.