Overcoming Weak Expectations

(Invited Paper)

Yevgeniy Dodis
Department of Computer Science
New York University
Email: dodis@cs.nyu.edu

Yu Yu
Institute for Interdisciplinary Information Sciences
Tsinghua University, Beijing, 100084, P.R.China
Email: yuyu@yuyu.hk

***Abstract*—Recently, there has been renewed interest in basing cryptographic primitives on weak secrets, where the only information about the secret is some non-trivial amount of (min-) entropy. From a formal point of view, such results require to upper bound the expectation of some function $f(X)$, where X is a weak source in question. We show an elementary inequality which essentially upper bounds such 'weak expectation' by two terms, the first of which is independent of f, while the second only depends on the 'variance' of f under uniform distribution. Quite remarkably, as relatively simple corollaries of this elementary inequality, we obtain some 'unexpected' results, in several cases noticeably simplifying/improving prior techniques for the same problem. Examples include non-malleable extractors, leakage-resilient symmetric encryption, seed-dependent condensers and improved entropy loss for the leftover hash lemma.**

The full version of this (unrefereed) survey is available here [1].

I. Introduction

Formal cryptographic models take for granted the availability of perfect randomness. However, in reality we may only obtain 'weak' random sources that are far from uniform but only guaranteed with high unpredictability (formalized with min-entropy), such as biometric data [2], [3], physical sources [4], [5], secrets with partial leakage, and group elements from Diffie-Hellman key exchange [6], [7]. We refer to the former as ideal model and the latter as real model.

From a formal point of view, the standard (T, ε)-security (in the ideal model) of a cryptographic application P essentially requires that for any adversary A with resource[1] T, the expectation of $f(U_m)$ is upper bounded by ε, where function $f(r)$ denotes A's advantage conditioned on secret key being r, and U_m denotes uniform distribution over $\{0,1\}^m$. In the real model, keys are sampled from some non-uniform distribution R and thus the resulting security is the expected value of $f(R)$, which we call 'weak expectation'. We would hope that if P is (T, ε)-secure in the ideal setting, then P is also (T', ε') in the real setting by replacing U_m with R of sufficiently high min-entropy, where T' and ε' are not much worse than T and ε respectively.

In this paper, we present an elementary inequality that upper bounds the weak expectation of $f(R)$ by two terms: the first term only depends on the *entropy deficiency* (i.e. the difference between the length of source R and the amount of entropy it has), and the second is essentially the 'variance' of f under uniform distribution U_m. Quite surprisingly, some 'unexpected' results follow as simple corollaries of this inequality, such as non-malleable extractors [8], [9], [10], [11], leakage-resilient symmetric encryptions [12], seed-dependent condensers [13] and improved entropy loss for the leftover hash lemma [14]. We provide a unified proof for these diversified problems and in many cases significantly simply and/or improve known techniques for the same problems.

II. Preliminaries

Notations and Definitions. We use $s \leftarrow S$ to denote sampling an element s according to distribution S. The min-entropy of a random variable X is defined as $\mathbf{H}_\infty(X) \stackrel{\text{def}}{=} -\log(\max_x \Pr[X = x])$. We use $\mathsf{Col}(X)$ to denote the collision probability of X, i.e., $\mathsf{Col}(X) \stackrel{\text{def}}{=} \sum_x \Pr[X = x]^2 \leq 2^{-\mathbf{H}_\infty(X)}$, and collision entropy $\mathbf{H}_2(X) \stackrel{\text{def}}{=} -\log \mathsf{Col}(X) \geq \mathbf{H}_\infty(X)$. We also define average (aka conditional) collision entropy and average min-entropy of a random variable X conditioned on another random variable Z by

$$\mathbf{H}_2(X|Z) \stackrel{\text{def}}{=} -\log\left(\ \mathbb{E}_{z \leftarrow Z}\left[\ \textstyle\sum_x \Pr[X = x|Z = z]^2\ \right]\ \right)$$
$$\mathbf{H}_\infty(X|Z) \stackrel{\text{def}}{=} -\log\left(\ \mathbb{E}_{z \leftarrow Z}\left[\ \max_x \Pr[X = x|Z = z]\ \right]\ \right)$$

respectively, where $\mathbb{E}_{z \leftarrow Z}$ denotes the expected value over $z \leftarrow Z$.

We denote with $\Delta_D(X, Y)$ the advantage of a circuit D in distinguishing the random variables X, Y: $\Delta_D(X, Y) \stackrel{\text{def}}{=} |\ \Pr[D(X) = 1] - \Pr[D(Y) = 1]\ |$. The *statistical distance* between two random variables X, Y, denoted by $\mathsf{SD}(X, Y)$, is defined by

$$\frac{1}{2}\sum_x |\Pr[X = x] - \Pr[Y = x]| = \max_D \quad \Delta_D(X, Y)$$

we write $\mathsf{SD}(X, Y|Z)$ as shorthand for $\mathsf{SD}((X, Z), (Y, Z))$.

Abstract security games. We first define the general type of applications where our technique applies. The security of an application P can be defined via an interactive game between a probabilistic attacker A and a probabilistic challenger $\mathsf{C}(r)$, where A and C jointly compute function f on value r (derived from U_m in the ideal setting and from distribution R in the real setting). The game can have an arbitrary structure, but at the end $\mathsf{C}(r)$ should output a bit, with output 1 indicating that A 'won' the game and 0 otherwise.

[1] We use the word "resource" to include all the efficiency measures we might care about, such as running time, circuit size, number of oracle queries, etc.

For unpredictability games, $f(r)$ is the expected value of $\mathsf{C}(r)$ taken over the internal coins of A and C so that $f(r) \in [0;1]$; and for indistinguishability games, $f(r)$ is the expectation of $\mathsf{C}(r) - 1/2$, and hence $f(r) \in [-1/2; 1/2]$. We will refer to $|\mathbb{E}(f(U_m))|$ as the security in the "ideal model" (against A), and to $|\mathbb{E}(f(R))|$, with $\mathbf{H}_c(R) \geq m - d$ and $c \in \{2, \infty\}$, as the security in the "$(m-d)$-real$_c$ model". Note that a security result in the real$_2$ model is more desirable than (and implies) that in the real$_\infty$ model.

III. Overcoming Weak Expectations

UNPREDICTABILITY APPLICATIONS. For unpredictability applications (with non-negative f), the following inequality implies that the security degrades at most by a factor of 2^d compared with the ideal model (which is stated as Corollary 3.1), where d is the entropy deficiency.

Lemma 3.1: For any (deterministic) real-valued function $f : \{0,1\}^m \to \mathbb{R}^+ \cup \{0\}$ and any random variable R with $\mathbf{H}_\infty(R) \geq m - d$, we have

$$\mathbb{E}[f(R)] \;\leq\; 2^d \cdot \mathbb{E}[f(U_m)] \qquad (1)$$

Proof:

$$\mathbb{E}[f(R)] \;=\; \sum_r \Pr[R = r] \cdot f(r) \;\leq\; 2^d \cdot \sum_r \frac{1}{2^m} \cdot f(r)$$

■

Corollary 3.1: If an unpredictability application P is (T, ε)-square secure in the ideal model, then P is $(T, 2^d \cdot \varepsilon)$-secure in the $(m-d)$-real$_\infty$ model.

The above only applies to all "unpredictability" applications such as one-way functions, MACs and digital signatures.

INDISTINGUISHABILITY APPLICATIONS. Unfortunately, Corollary 3.1 critically depends on the non-negativity of f, and is generally false when f can be negative, which happens for indistinguishability applications. In fact, for certain indistinguishability applications, such as one-time pad, pseudo-random- generators and functions (PRGs and PRFs), there exists R with $d = 1$ such that $\mathbb{E}[f(U_m)]$ is negligible (or even zero!) but $\mathbb{E}[f(R)] = 1/2$ (see [14] for more discussions). Fortunately, below we give another inequality for general f, which will be useful for other indistinguishability applications.

Lemma 3.2: For any (deterministic) real-valued function $f : \{0,1\}^m \to \mathbb{R}$ and any random variable R with $\mathbf{H}_2(R) \geq m - d$, we have

$$|\ \mathbb{E}[f(R)]\ | \;\leq\; \sqrt{2^d} \cdot \sqrt{\mathbb{E}[f(U_m)^2]} \qquad (2)$$

Proof: Denote $p(r) = \Pr[R = r]$, and also recall the Cauchy-Schwartz inequality $|\sum a_i b_i| \leq \sqrt{(\sum a_i^2) \cdot (\sum b_i^2)}$. We have

$$|\ \mathbb{E}[f(R)]\ | = \left| \sum_r p(r) \cdot f(r) \right|$$
$$\leq \sqrt{2^m \cdot \sum_r p(r)^2} \cdot \sqrt{\frac{1}{2^m} \sum_r f(r)^2} = \sqrt{2^d \cdot \mathbb{E}[f(U_m)^2]}$$

■

Lemma 3.2 upper bounds the (squared) weak expectation by the product of 2^d and $\mathbb{E}[f(U_m)^2]$. Intuitively, 2^d gives the security loss due to the entropy deficiency, and $\mathbb{E}[f(U_m)]$ defines the ideal model security of the application in consideration, but notice we only get $\mathbb{E}[f(U_m)^2]$, for which we define the notion of "square security". Lemma 3.2 essentially applies to square secure applications, which we state as Corollary 3.2.

Definition 3.1 (Square Security): An application P is (T, σ)-square secure if for any T-bounded adversary A we have $\mathbb{E}[f(U_m)^2] \leq \sigma$, where $f(r)$ denotes A's advantage conditioned on key being r.

Corollary 3.2 (Square security implies real model security): If P is (T, σ)-square secure, then P is $(T, \sqrt{2^d \cdot \sigma})$-secure in the $(m-d)$-real$_2$ model.

WHAT APPLICATIONS HAVE SQUARE SECURITY? First, all (T, ε)-secure unpredictability applications P are (T, ε)-square secure, since for non-negative f we have $\mathbb{E}[f(U_m)^2] \leq \mathbb{E}[f(U_m)]$. Hence, we immediately get $\sqrt{2^d \cdot \varepsilon}$-security in $(m-d)$-real$_2$ model for such application.[2]

Moving to indistinguishability applications, it is known that PRGs, PRFs, one-time pads cannot have good square security (see [14]). To see why, consider a 1-bit one time pad encryption $c = m \oplus r$, where $m, r, c \in \{0,1\}$ are the message, the key and the ciphertext, respectively, and $\oplus$ is "exclusive OR". Consider also the attacker A who guesses that $m = c$. When the key $r = 0$, A is right and $f(0) = 1 - \frac{1}{2} = \frac{1}{2}$. Similarly, when the key $r = 1$, A is wrong and $f(1) = 0 - \frac{1}{2} = -\frac{1}{2}$. This gives perfect $\varepsilon = \mathbb{E}[f(U_1)] = 0$, but $\sigma = \mathbb{E}[f(U_1)^2] = \frac{1}{4}$.

Fortunately, there are still many interesting indistinguishability objects whose square security is of roughly the same order as their regular security, such as stateless CPA- and CCA-secure (symmetric-key and public-key) encryption schemes, weak pseudo-random functions (weak PRFs), and q-wise independent hash functions. We now discuss some examples.

A. Application to Encryption Schemes and Weak PRFs

We will only show that CPA-secure symmetric-key encryption schemes are square secure, and we prove that using the "double-run" technique from [14]. Other schemes (mentioned above) can be proven similarly by adapting the double-run trick to the actual security game (see [14] for the subtleties).

Lemma 3.3 ([14]): Assume P is a symmetric-key encryption scheme which is 2ε-secure, in the ideal model, against all chosen-plaintext attackers with running time $2t + O(1)$ and making $2q+1$ queries. Then P is ε-square secure against all chosen-plaintext attackers with running time t, and making q queries. Hence, $(T = (2t + O(1), 2q+1), 2\varepsilon)$-security implies $((t, q), \varepsilon)$-security.

DOUBLE-RUN TRICK. We sketch the proof of the above lemma for completeness. It suffices to show that for any r

[2]This bound is weaker than the $2^d\varepsilon$ bound in Corollary 3.1, although it applies whenever $\mathbf{H}_2(R) \geq m - d$ (instead of only when $\mathbf{H}_\infty(R) \geq m - d$). Still, we will find Lemma 3.2 useful even for unpredictability applications when we talk about key derivation functions in Section IV.

and any attacker A with running time t and q queries, there exists another attacker B with running time roughly $2t$ and $2q+1$ queries such that B's advantage is twice the squared advantage of A. The strategy of B is that it first simulates the challenger C (using one query), runs A against the simulated C, and then runs A against the real C. If A wins the game in its first run (against the simulated C), then B returns A's answer in the second run, or otherwise B reverses the answer of A. Thus,

$$\begin{aligned} \Pr[\text{B wins}] &= \Pr[\text{A wins twice}] + \Pr[\text{A loses twice}] \\ &= (\frac{1}{2} \pm \varepsilon)^2 + (\frac{1}{2} \mp \varepsilon)^2 = \frac{1}{2} + 2\varepsilon^2 \end{aligned}$$

■

The following theorem immediately follows from Corollary 3.2 and Lemma 3.3.

Theorem 3.1: Assume P is a $((2t+O(1),2q),2\varepsilon)$-CPA secure symmetric-key encryption scheme in the ideal model. Then P is also $((t,q),\sqrt{2^d \cdot \varepsilon})$-secure in the $(m-d)$-real$_2$ model.

Same argument (as Theorem 3.1) works for all aforementioned square secure applications, such as stateless (public-key and symmetric-key) CPA- and CCA- secure encryption schemes, and weak PRFs, simplifying [12].

MULTI-RUN EXTENSION. In the double-run game we use a test-run to estimate the sign of the advantage (whether it's positive or not), which advises attacker B whether or not to reverse A's answer in the real run. We can generalize this to a multi-run setting: the attacker B test-runs A for some odd $(2i{+}1)$ times, and takes a majority vote before the actual run, which gives B more accurate estimate on the sign of the advantage (using the technique of Brakerski and Goldreich [15]). This applies to all double-run-friendly applications (like the CPA encryption), but we only state it for the case of weak PRF for concreteness, and also because it simplifies [12] a lot.

Corollary 3.3 (Weak PRFs on Weak Keys): For any ε, d and $c \le O(1/\sqrt{2^d \cdot \varepsilon})$, if P is a $(((1+c^4)t,(1+c^4)q),\varepsilon)$-secure weak PRF in the ideal model, then P is also $((t,q),O(\frac{1}{c}\cdot\sqrt{2^d\cdot\varepsilon}))$-secure in the $(m-d)$-real$_2$ model.

B. Application to Alternative LHL and NM-Extractors

We now show that 2-wise and 4-wise independent hash functions give rise to strong and non-malleable extractors respectively. For our convenience, we use the following definition for (q,δ)-wise independence (slightly weaker than the traditional q-wise independence), where one point s is *randomly* chosen and the rest $q-1$ points can be arbitrarily dependent on s (as long as they are distinct from s).

Definition 3.2 ((q,δ)-wise independence): A family $\mathcal{H}$ of functions $\{h_r : \{0,1\}^n \to \{0,1\}^l \mid r \in \{0,1\}^m\}$ is (q,δ)-wise independent, if for $r{\leftarrow}U_m$, $s \leftarrow U_n$, and for $s_1,\cdots,s_{q-1} \in \{0,1\}^n$ that are distinct from and arbitrarily correlated to s, we have

$$\mathsf{SD}(\ h_r(s),\ U_l \mid s, h_r(s_1),\cdots,h_r(s_{q-1})\ ) \le \delta$$

Notice, we can naturally view the above definition as a game between a challenger C and the attacker A, where $(q-1)$ measures the "resources" of A (distinct from s points where he learns the true value of h_r), and δ is the advantage of distinguishing $h_r(s)$ from random. Thus, we can naturally define the (q,σ_q)-square security of $\mathcal{H}$ (with random key $r \leftarrow U_m$) and then use Corollary 3.2 to bound the security of $\mathcal{H}$ in the $(m-d)$-real$_2$ model, when using a weak key R with $\mathbf{H}_2(R) \ge m-d$. In fact, we can successfully apply the double-run trick above to show that if $\mathcal{H}$ is $(2q,\delta)$-wise independent, then its *square* security σ_q as a q-wise (rather than $2q$-wise) independent hash function is at most $\sigma_q \le \delta + \frac{q}{2^n}$, where $\frac{q}{2^n}$ accounts for the probability that the real challenge point (chosen uniformly at random) collides with the q points of the test-run. Applying now Corollary 3.2, we get

Theorem 3.2: If function family $\mathcal{H}$ is $(2q,\delta)$-wise independent, then $\mathcal{H}$ is also (q,ε)-wise independent in the $(m-d)$-real$_2$ model, where $\varepsilon = \sqrt{(\delta + \frac{q}{2^n})\cdot 2^d}$.

ALTERNATIVE LHL. We will first consider the consequences for $q=1$, where the notion of $(1,\varepsilon)$-wise independence in the $k=(m-d)$-real$_2$ model becomes a *randomness extractor*.

Definition 3.3 (Extractors): We say that an efficient function $\mathsf{Ext} : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^l$ is a strong (k,ε)-*extractor*, if for all R (over $\{0,1\}^m$) with $\mathbf{H}_2(R) \ge k$ and for random S (uniform over $\{0,1\}^n$), we get

$$\mathsf{SD}(\ \mathsf{Ext}(R;S)\ ,\ U_l \mid S) \le \varepsilon$$

where coins $S \leftarrow U_n$ is the random seed of Ext. The value $L = k - l$ is called the *entropy loss* of Ext.

Applying Theorem 3.2 to pairwise independent hash functions (i.e., $2q=2$, $\delta=0$, $k=m-d$), we get:

Corollary 3.4 (Alternative LHL): If $\mathcal{H} \stackrel{\text{def}}{=} \{h_r : \{0,1\}^n \to \{0,1\}^l \mid r \in \{0,1\}^m\}$ is pairwise independent, then $\mathsf{Ext}(r;s) \stackrel{\text{def}}{=} h_r(s)$ is a strong $(k,\sqrt{2^{m-k-n}})$ extractor.

To compare this result with the standard LHL [16], the optimal key length m for a family of pairwise independent hash functions from n to l bits is known to be $m=n+l$ (e.g., using Toeplitz matrices or "augmented" inner product discussed below). Plugging this to our bound in ε above, we get the same bound $\varepsilon = \sqrt{2^{l-k}} = 2^{-L/2}$ as the leftover hash lemma, where in both cases l is output size and k is the entropy of the source. Interestingly, standard leftover hashing [16] uses universal $\mathcal{H}$ (see Definition 4.3 below), which is weaker, but sets $\mathsf{Ext}(r;s) = h_s(r)$, swapping the roles of source and seed.[3]

NON-MALLEABLE EXTRACTORS. Next, we consider the case of $q=2$, where the notion of $(2,\varepsilon)$-wise independence in the $k=(m-d)$-real$_2$ model becomes a *non-malleable extractor*.

Definition 3.4 (Non-Malleable Extractors): We say that an efficient function $\mathsf{nmExt} : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^l$ is a (k,ε)-*non-malleable extractor*, if for all R (over $\{0,1\}^m$) with

[3]Curiously, when l divides n, the following "augmented" inner product function $h_r(s)$ is *simultaneously* an optimal pairwise independent hash function when keyed by r, and an optimal universal function when keyed by s: $h_r(s) = r_1{\cdot}s_1{+}\ldots{+}r_p{\cdot}s_p{+}r_{p+1}$, where $p=n/l$, $r=(r_1,\ldots,r_{p+1})$, $s=(s_1,\ldots,s_p)$, and r_i and s_j are interpreted as elements of $GF[2^l]$.

$\mathbf{H}_2(R) \geq k$, for random S (uniform over $\{0,1\}^n$), and for all functions $g : \{0,1\}^n \to \{0,1\}^n$, s.t. $g(s) \neq s$ for all s, we get

$$\mathsf{SD}(\ \mathsf{nmExt}(R;S)\ ,\ U_l\ |\ \ S, \mathsf{nmExt}(R;g(S))\ ) \leq \varepsilon$$

Applying Theorem 3.2 to 4-wise independent hash functions (i.e., $2q = 4$, $\delta = 0$, $k = m - d$), we get:

Corollary 3.5 (Non-Malleable Extractors): If $\mathcal{H} \stackrel{\text{def}}{=} \{h_r : \{0,1\}^n \to \{0,1\}^l \mid r \in \{0,1\}^m\}$ is 4-wise independent, then $\mathsf{nmExt}(r;s) \stackrel{\text{def}}{=} h_r(s)$ is a $(k, \sqrt{2^{m-k-n+1}})$-non-malleable extractor.

For a simple instantiation, let $\mathcal{H}$ be the following (optimal) 4-wise independent hash function with known parameters $n = m/2$ and $l = m/4$ (using BCH codes; see [11]). The key $r \in \{0,1\}^m$ is viewed as a tuple of 4 elements (r_1, r_2, r_3, r_4) in $GF[2^{m/4}] = GF[2^l]$, and a seed $s \in \{0,1\}^n \backslash 0^n$ is viewed as a non-zero point in $GF[2^n]$. Then, the m-bit value of $(s \| s^3)$ is viewed as 4 elements (s_1, s_2, s_3, s_4) in $GF[2^l]$, and the l-bit output of the function is set to $h_r(s) = r_1 \cdot s_1 + \ldots + r_4 \cdot s_4$. Using Corollary 3.5, this simple function is a $(k, \sqrt{2^{m/2-k+1}})$-non-malleable extractor, which improves the construction of [9] and matches the recent results of [11] with a much simplified proof.

IV. Key Derivation Functions

So far we use weak sources directly on ε-square secure objects (and we still get extractors), which requires entropy deficiency $d < \log(1/\varepsilon)$. For low entropy sources where $d \gg \log(1/\varepsilon)$, we need to apply a key derivation function (KDF) that preprocess the source to get some better randomness (by discarding some 'bad' bits), where the setting is mainly characterized by the entropy of the source k and the output size of the KDF m.

Definition 4.1: (k, m)-real$_c$ model (for $c \in \{2, \infty\}$) refers to the key derivation setting where a given KDF h with range $\{0,1\}^m$ is applied to any source X with $\mathbf{H}_c(X) \geq k$ to get a secret key $R = h(X)$ (for some application in question).

Next we propose randomness condensers as generalization of extractors, and justify the use of condensers as key derivation functions. Intuitively, a condenser is a probabilistic function that reduces entropy deficiency.

Definition 4.2 (Condensers): Let $c \in \{2, \infty\}$. We say that an efficient function $\mathsf{Cond} : \{0,1\}^n \times \{0,1\}^v \to \{0,1\}^m$ is a $(\frac{k}{n} \to \frac{m-d}{m})_c$-condenser if for $\mathbf{H}_c(X) \geq k$ and uniformly random S we have $\mathbf{H}_c(\ \mathsf{Cond}(X;S)\ |\ S\ ) \geq m - d$.

Both $\mathbf{H}_\infty$- and $\mathbf{H}_2$- condensers are useful in cryptography. The former connects well with Lemma 3.1, and the latter is more in line with Lemma 3.2. In the sequel, though, we will only use $\mathbf{H}_2$ (and let $c = 2$ hereafter) since it seems to give stronger final bounds (even for unpredictability applications), and applies to more cases (e.g. indistinguishability applications). See [13] for more discussion.

A. Improved Leftover Hash Lemma

We know by the standard leftover hash lemma [16] that universal hash functions are efficient extractors and thus are good KDFs, but the entropy loss L (entropy of the source minus the length of extracted randomness) must be positive. Below we recall the notion of universal hashing [17], and state their condensing properties. We show if they are used as KDFs for all "square-friendly" applications,[4] we improve L (reducing it by half) and make it meaningful even for $L \leq 0$, where entropy deficiency $d \approx -L$.

Definition 4.3 (Universal Hashing): A family of functions $\mathcal{G} \stackrel{\text{def}}{=} \{g_s : \{0,1\}^n \to \{0,1\}^m \mid s \in \{0,1\}^v\}$ is universal, if for any distinct $x_1, x_2 \in \{0,1\}^n$ we have

$$\Pr_{s \leftarrow U_v}[g_s(x_1) = g_s(x_2)] = 2^{-m}$$

Lemma 4.1: Universal hash function family $\mathcal{G} \stackrel{\text{def}}{=} \{g_s : \{0,1\}^n \to \{0,1\}^m \mid s \in \{0,1\}^v\}$ defines a $(\frac{k}{n} \to \frac{m-d}{m})_2$-condenser $\mathsf{Cond}(x;s) \stackrel{\text{def}}{=} g_s(x)$, where $2^d = 1 + 2^{m-k}$.

Proof:

$$\begin{aligned} & \Pr[g_S(X_1) = g_S(X_2)] \\ \leq\ & \Pr[X_1 = X_2] + \Pr[\ g_S(X_1) = g_S(X_2)\ \wedge\ X_1 \neq X_2] \\ \leq\ & 2^{-k} + 2^{-m}\ =\ 2^{-m} \cdot (2^{m-k} + 1)\ =\ 2^{d-m} \end{aligned}$$

■

We use a slightly differently version of Lemma 3.2 (whose proof is very similar as well) for the improved entropy loss results.

Lemma 4.2 ([14]): For any (deterministic) real-valued function $f : \{0,1\}^m \to \mathbb{R}$ and any random variable R with $\mathbf{H}_2(R) \geq m - d$, we have

$$|\ \mathbb{E}[f(R)] - \mathbb{E}[f(U_m)]\ |\ \leq\ \sqrt{2^d - 1} \cdot \sqrt{\mathbb{E}[f(U_m)^2]} \quad (3)$$

Corollary 4.1 (Using Universal Hashing as KDF): If P is (T, ε)-secure and (T, σ)-square secure, then using $R = g_s(X)$ makes P (T, ε')-secure in the (k, m)-real$_2$ model, where $R \in \{0,1\}^m$, $\mathbf{H}_2(X) \geq k$, and $\varepsilon' \leq \varepsilon + \sqrt{\sigma \cdot 2^{m-k}}$.

Reduced entropy loss for leftover hash lemma. Recall that we can have $\sigma \approx \varepsilon$ for many square-secure applications. Let $L = k - m$ denote entropy loss. To achieve $\varepsilon' \approx \varepsilon$ we need to set $L = \log(1/\varepsilon)$, while the standard leftover hash lemma achieved a weaker bound $\varepsilon' \leq \varepsilon + \sqrt{2^{m-k}}$, and required $L = 2\log(1/\varepsilon)$. Moreover, our entropy loss is meaningful even for negative L, in which case entropy deficiency of $R = g_s(X)$ is $d \approx -L$ and $\varepsilon' \approx \sqrt{\varepsilon \cdot 2^{-L}} \approx \sqrt{\varepsilon \cdot 2^d}$.

B. Seed-Dependent Key Derivation

We now generalize the notion of a condenser to the seed-dependent setting, where the adversarial sampler A can depend on the seed S but is computationally bounded. This challenging setting was considered by [18] in the context of seed-dependent *extractors*, where the authors made a pessimistic conclusion that the complexity of the seed-dependent extractor

[4] As observed by [14], we can also compose universal hashing with (square-friendly) weak PRFs to also handle all computational (even "non-square-friendly") applications, such as PRFs and PRGs.

must be larger than that of the sampler A, making this notion not very useful for key derivation in practical applications. In contrast, we show that (strong enough) collision-resistant hash functions (CRHFs) must be seed-dependent *condensers*, and thus can be used as KDFs for all square secure applications, despite having much smaller complexity than the complexity of the sampler A. This partially explains the use of CRHFs as KDFs in practical applications.

Definition 4.4 (Seed-Dependent Condensers): An efficient function $\mathsf{Cond} : \{0,1\}^n \times \{0,1\}^v \to \{0,1\}^m$ is a $(\frac{k}{n} \to \frac{m-d}{m}, t)_2$-seed-dependent condenser if for all probabilistic adversaries A of size t who take a random seed $s \leftarrow U_v$ and output (using more coins) a sample $X \leftarrow \mathsf{A}(s)$ of entropy $\mathbf{H}_2(X|S) \geq k$, we have $\mathbf{H}_2(\ \mathsf{Cond}(X;S) \mid S\ ) \geq m-d$.

Definition 4.5 (CRHF): A family of hash functions $\mathcal{G} \stackrel{\text{def}}{=} \{g_s : \{0,1\}^n \to \{0,1\}^m \mid s \in \{0,1\}^v\}$ is (t,δ) -collision-resistant if for any (non-uniform) attacker B of size t, we have

$$\Pr[g_s(x_1) = g_s(x_2) \ \wedge \ x_1 \neq x_2] \leq \delta$$

where $s \leftarrow U_v$ and $(x_1, x_2) \leftarrow \mathsf{B}(s)$.

Lemma 4.3 (CRHFs are seed-dependent condensers): A family of $(2t, \frac{D(t)}{2^m})$-collision-resistant hash functions $\mathcal{G} \stackrel{\text{def}}{=} \{g_s : \{0,1\}^n \to \{0,1\}^m \mid s \in \{0,1\}^v\}$ defines a seed-dependent $(\frac{k}{n} \to \frac{m-d}{m}, t)_2$-condenser $\mathsf{Cond}(x;s) = g_s(x)$, where $2^d = 2^{m-k} + D(t)$.

Proof:

$$\begin{aligned}
& \Pr[g_S(X_1) = g_S(X_2)] \\
\leq \ & \Pr[X_1 = X_2] + \Pr[\ g_S(X_1) = g_S(X_2) \ \wedge \ X_1 \neq X_2] \\
\leq \ & 2^{-k} + D(t) \cdot 2^{-m} \ = \ 2^{-m} \cdot (2^{m-k} + D(t)) \ = \ 2^{d-m}
\end{aligned}$$

■

In the above, entropy deficiency d is essentially the logarithm of $D(t)$, which is a function on the sampler's complexity t. We note $D(t) = \Omega(t^2)$ due to birthday attacks, and this bound can be achieved in the random oracle model. In general, it is reasonable to assume $D(t) = \text{poly}(t)$ for strong enough CRHFs. Then, using the definition of condensers and Corollary 3.2, we get the following surprising result, which partially explains the prevalent use of CRHFs (which do not appear to have any extraction properties based on their definition) for key derivation:

Corollary 4.2 (Using CRHFs as KDFs): If P is (T,σ)-square secure, $\{g_s\}$ is a family of $(2t, \frac{poly(t)}{2^m})$-CRHFs, and X is a source produced by a sampler $\mathsf{A}(s)$ of complexity at most t and having $\mathbf{H}_2(X|S) \geq k \geq m - O(\log t)$, then using $R = g_s(X)$ makes P (T, ε')-secure, where $\varepsilon' \leq O(\sqrt{\sigma \cdot poly(t)})$.

From an asymptotic point of view, for square-friendly applications (e.g. CPA-secure encryptions, weak PRFs, unpredictability primitives) with negligible ideal ε (and hence negligible $\sigma \approx \varepsilon$), and all source samplers running in polynomial time t (all in the "security parameter"), we get negligible security $\varepsilon' = O(\sqrt{\sigma \cdot poly(t)})$ in the real model.

ACKNOWLEDGMENT

We would like to thank our co-authors in [14], [13] for useful comments and suggestions.

Yevgeniy Dodis was supported by NSF Grants CNS-1065134, CNS-1065288, CNS-1017471, CNS-0831299 and Google Faculty Award. Yu Yu was supported by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61172085, 61061130540, 61073174, 61103221, 11061130539, 61021004 and 61133014.

REFERENCES

[1] Y. Dodis and Y. Yu, "Overcoming weak expectations," 2012, full version of this (unrefereed) survey. Available at http://cs.nyu.edu/~dodis/ps/weak-expe.pdf.
[2] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM Journal on Computing*, vol. 38, no. 1, pp. 97–139, 2008.
[3] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith, "Secure remote authentication using biometric data," in *Advances in Cryptology—EUROCRYPT 2005*, ser. LNCS, R. Cramer, Ed., vol. 3494. Springer-Verlag, 2005, pp. 147–163.
[4] B. Barak, R. Shaltiel, and E. Tromer, "True random number generators secure in a changing environment," in *Proceedings of the 5th Cryptographic Hardware and Embedded Systems*, 2003, pp. 166–180.
[5] B. Barak and S. Halevi, "A model and architecture for pseudo-random generation with applications to /dev/random," in *Proceedings of the 12th ACM Conference on Computer and Communication Security*, 2005, pp. 203–212.
[6] R. Gennaro, H. Krawczyk, and T. Rabin, "Secure hashed diffie-hellman over non-ddh groups," in *Advances in Cryptology—EUROCRYPT 2004*, ser. LNCS, C. Cachin and J. Camenisch, Eds., vol. 3027. Springer-Verlag, 2004, pp. 361–381.
[7] H. Krawczyk, "Cryptographic Extraction and Key Derivation: The HKDF Scheme," in *Advances in Cryptology - CRYPTO 2010*, ser. LNCS, T. Rabin, Ed., vol. 6223. Springer-Verlag, 2010, pp. 631–648.
[8] Y. Dodis and D. Wichs, "Non-malleable extractors and symmetric key cryptography from weak secrets," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, M. Mitzenmacher, Ed. Bethesda, MD, USA: ACM, 2009, pp. 601–610.
[9] Y. Dodis, X. Li, T. D. Wooley, and D. Zuckerman, "Privacy amplification and non-malleable extractors via character sums," in *Proceedings of the 52nd IEEE Symposium on Foundation of Computer Science*, 2011, pp. 668–677.
[10] G. Cohen, R. Raz, and G. Segev, "Non-malleable extractors with short seeds and applications to privacy amplification," in *Proceedings of the 27th Computational Complexity*, 2012, pp. 110–124.
[11] X. Li, "Non-malleable extractors, two-source extractors and privacy amplification," in *Proceedings of the 53rd IEEE Symposium on Foundation of Computer Science*, 2012, pp. xxx–xxx.
[12] K. Pietrzak, "A leakage-resilient mode of operation," in *Advances in Cryptology - EUROCRYPT 2009*, ser. LNCS, A. Joux, Ed., vol. 5479. Springer-Verlag, 2009, pp. 462–482.
[13] Y. Dodis, T. Ristenpart, and S. P. Vadhan, "Randomness condensers for efficiently samplable, seed-dependent sources," in *9th Theory of Cryptography Conference*, 2012, pp. 618–635.
[14] B. Barak, Y. Dodis, H. Krawczyk, O. Pereira, K. Pietrzak, F.-X. Standaert, and Y. Yu, "Leftover hash lemma, revisited," in *CRYPTO*, ser. LNCS, P. Rogaway, Ed. Springer, 2011, pp. 1–20.
[15] Z. Brakerski and O. Goldreich, "From absolute distinguishability to positive distinguishability," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 16, p. 31, 2009.
[16] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby, "Construction of pseudorandom generator from any one-way function," *SIAM Journal on Computing*, vol. 28, no. 4, pp. 1364–1396, 1999.
[17] J. Carter and M. Wegman, "Universal classes of hash functions," *Journal of Computer and System Sciences*, vol. 18, pp. 143–154, 1979.
[18] L. Trevisan and S. Vadhan, "Extracting randomness from samplable distributions," in *41st Annual Symposium on Foundations of Computer Science*. Redondo Beach, California: IEEE, Nov. 2000, pp. 32–42.