Robust Fuzzy Extractors and Authenticated Key Agreement from Close Secrets

Yevgeniy Dodis[1,⋆], Jonathan Katz[2,⋆⋆], Leonid Reyzin[3,⋆⋆⋆], and Adam Smith[4,†]

[1] New York University. dodis@cs.nyu.edu
[2] University of Maryland. jkatz@cs.umd.edu
[3] Boston University. reyzin@cs.bu.edu
[4] Weizmann Institute of Science. adam.smith@weizmann.ac.il

Abstract. Consider two parties holding correlated random variables W and W', respectively, that are within distance t of each other in some metric space. These parties wish to agree on a uniformly distributed secret key R by sending a single message over an insecure channel controlled by an all-powerful adversary. We consider both the *keyless* case, where the parties share no additional secret information, and the *keyed* case, where the parties share a long-term secret SK that they can use to generate a sequence of session keys $\{R_j\}$ using multiple pairs $\{(W_j, W'_j)\}$. The former has applications to, e.g., biometric authentication, while the latter arises in, e.g., the bounded storage model with errors.

Our results improve upon previous work in several respects:

- The best previous solution for the keyless case with no errors (i.e., $t = 0$) requires the min-entropy of W to exceed $2|W|/3$. We show a solution when the min-entropy of W exceeds the *minimal* threshold $|W|/2$.
- Previous solutions for the keyless case in the presence of errors (i.e., $t > 0$) required random oracles. We give the first constructions (for certain metrics) in the standard model.
- Previous solutions for the keyed case were stateful. We give the first stateless solution.

1 Introduction

A number of works have explored the problem of *secret key agreement based on correlated information* by which two parties holding instances of correlated random variables W and W' communicate and thereby generate a shared, secret (uniformly-random) key SK. Early work [Wyn75,BBR88,Mau93,BBCM95] assumed that the parties could communicate over a *public* but *authenticated* channel or, equivalently, assumed a passive adversary. This assumption was relaxed in later work [Mau97,MW97,Wol98,MW03,RW03], which considered an active adversary who could modify all messages sent between the two parties.

⋆ Supported by NSF grants #0133806, #0311095, and #0515121.
⋆⋆ Supported by NSF grants #0310499, #0310751, and #0447075.
⋆⋆⋆ Supported by NSF grants #0311485 and #0515100.
† Supported by the Louis L. and Anita M. Perlman Fellowship.

The motivation of the above works was primarily to explore the possibility of *information-theoretic* security; however, this is not the only motivation. The problem also arises in the context of using noisy data (such as biometric information) for cryptographic purposes, even if computational security suffices. The problem also arises in the context of the *bounded storage model* (BSM) [Mau92] in the presence of errors [Din05,DS05]. We discuss each of these in turn.

Authentication using noisy data. In the case of authentication using noisy data, the random variables W, W' are *close* (with respect to some metric) but not *identical*. For simplicity, we assume the noisy data represents biometric information though the same techniques apply to more general settings. In this context, two different scenarios have been considered:

"Secure authentication:" Here, a trusted server stores the "actual" biometric data W of a user; periodically, the user obtains a fresh biometric scan W' which is close, but not identical, to W. The server and user then wish to mutually authenticate and agree on a key R.

"Key recovery:" Here, a user (on his own) uses his "actual" biometric data W to generate a random key R along with some public information P, and then stores P on a (possibly untrusted) server. The key R might then be used, say, to encrypt some data for long-term storage. At a later point in time, the user obtains a fresh biometric scan W' along with the value P from the server; together, these values enable recovery of R (and hence enable decryption of the data).

In the second setting the user is, in effect, running a key agreement protocol with *himself* at two points in time, with the (untrusted) server acting as the "communication channel" between these two instances of the user. This second scenario inherently requires a *non-interactive* (i.e., one-message) key agreement protocol since W is no longer available at the later point in time. Note also that any solution for the second scenario is also a solution for the first.

Solutions for achieving secret key agreement using noisy data and an *authenticated* channel are known [BBR88,BBCM95,DORS06,JW99,FJ01,LT03]. However, existing work such as [Mau97,MW97,Wol98,MW03,RW03] does *not* solve the above problem when the parties communicate over an *unauthenticated* channel. Positive results in the unauthenticated setting were known only for two special cases: (1) when $W = W'$ and (2) when W and W' consist of (arbitrarily-many) independent realizations of the same random experiment; i.e., $W = (W^{(1)}, W^{(2)}, \ldots)$ and $W' = (W'^{(1)}, W'^{(2)}, \ldots)$. In the case of biometric data, however, W, W' are not likely to be equal and we cannot in general obtain an unbounded number of samples.

Recently, some partial solutions to the problems considered above have been obtained in the unauthenticated setting. Boyen [Boy04] shows, in the random oracle model, how to achieve *unidirectional* authentication with noisy data, as well as a weak form of security in the "key recovery" setting (essentially, R remains secret but the user can be fooled into using an incorrect key R'). Subsequent work of Boyen, et al. [BDK$^+$05] shows two solutions: the first is non-interactive but relies on random oracles; the second solution can be used for secure authentication but does not apply to the "key recovery" scenario because it requires interaction. This second solution has some other limitations as well: since it relies on a underlying password-based key-exchange protocol, it inherently provides *computational* rather than *information-theoretic* security; furthermore, given the current state-of-the-art for password-based key exchange [BPR00,BMP00,KOY01,GL01,GL03], the resulting protocol is either impractical or else requires additional assumptions such as random oracles/ideal ciphers or public parameters.

THE BOUNDED STORAGE MODEL AND THE "KEYED" CASE. Key agreement using correlated information arises also in the context of the *bounded storage model* (BSM) [Mau92] in the presence of errors [Din05,DS05]. In the BSM, two parties share a long-term secret key SK. In each of an unlimited number of time periods $j = 1, \ldots$, a long random string Z_j is broadcast to the parties (and observed by an adversary); the assumption is that the length of Z_j is more than what the adversary can store. The parties use SK and Z_j to generate a secret session key R_j, with $|R_j| \gg |\mathsf{SK}|$, in each period. This process should achieve "everlasting security" [ADR02], meaning that even if SK is revealed to the adversary in some time period n, all session keys $\{R_j\}_{j<n}$ remain independently and uniformly distributed from the perspective of the adversary.

A typical paradigm for achieving the above is for the parties to sample (using SK) shorter strings W_j and W'_j, respectively, from the random string Z_j in each period. Next, the parties use W_j (resp., W'_j) and SK to generate R_j. In standard treatments of the BSM (e.g., [Mau92,ADR02]), it is assumed that both parties receive identical copies of Z_j and hence $W_j = W'_j$. In the presence of transmission errors in Z_j, however, it is possible for W_j and W'_j to be close but not identical [Din05,DS05]. The parallels to the case of biometric authentication, as discussed earlier, should now be clear. Nevertheless, the problems are incomparable: in the case of the BSM with errors there is a stronger setup assumption (the parties share a long-term key SK), but the security requirements are more stringent.

OUR CONTRIBUTIONS. We focus on the abstract problem of secret key agreement between two parties holding instances w, w' of correlated random variables W, W' that are guaranteed to be close but not necessarily identical. Specifically, we assume that w and w' are within distance t with respect to some underlying metric. Some of our results hold for arbitrary metric spaces, while others assume the Hamming metric in particular.

We consider only *non-interactive* protocols defined by procedures $(\mathsf{Gen}, \mathsf{Rep})$ that operate as follows: the first party, holding w, computes $(R, P) \leftarrow \mathsf{Gen}(w)$ and sends P to the second party; this second party computes $R' \leftarrow \mathsf{Rep}(w', P)$. (If the parties share a long-term key SK then $\mathsf{Gen}, \mathsf{Rep}$ take this as additional input.) The basic requirements, informally, are

Correctness: $R = R'$ whenever w' is within distance t of w.

Security: If the entropy of W is high, R is uniformly distributed even given P.

So far, this gives exactly a *fuzzy extractor* as defined by Dodis et al. [DORS06] (although we additionally allow the possibility of a long-term key). Since we are interested in the case when the parties communicate over an *unauthenticated* channel, however, we actually want to construct *robust* fuzzy extractors [BDK$^+$05] that additionally protect against malicious modification of P. Robustness requires that if the adversary sends any modified value $P' \neq P$, then with high probability the second player will reject (i.e., $\mathsf{Rep}(w', P) = \perp$). *Strong* robustness requires this to hold even if the adversary learns the R (held by the first player). This property is essential in settings where the first party may begin using R before the second party computes R', and is also needed for the "key recovery" scenario discussed earlier (since previous usages of R may leak information about it). *Weak* robustness, still sufficient for some applications, only requires robustness when R is not learned by the adversary.

Letting $H_\infty(X)$ denote the min-entropy of a random variable X, we now describe our results.

The keyless case with no errors. Although our focus is on the case when W, W' are *close*, we obtain improvements also in the case when they are equal (i.e., $t = 0$). Specifically, the best previous non-interactive solution in this setting is due to Maurer and Wolf[5] [MW03] who show that when $H_\infty(W) > 2|W|/3$ it is possible to achieve weak robustness and generate a shared key R of length $H_\infty(W) - 2|W|/3$. On the other hand, results of [DS02] imply that a non-interactive solution is impossible when $H_\infty(W) \leq |W|/2$.

We bridge the gap between known upper- and lower-bounds and show that whenever $H_\infty(W) > |W|/2$ it is possible to achieve weak robustness and generate a shared key R of length $2H_\infty(W) - |W|$. This improves both the required min-entropy of W as well as the length of the resulting key. Moreover, we give the first solution satisfying *strong* robustness which still works as long as $H_\infty(W) > |W|/2$ (but extracts a slightly shorter key).

The keyless case with errors. The only previously-known construction of robust fuzzy extractors [BDK$^+$05] relies on the random oracle model. (This solution is generic and applies to any metric admitting a good error-correcting code.) We (partially) resolve the main open question of [BDK$^+$05] by showing a construction of strongly robust fuzzy extractors *in the standard model*, for the case of the Hamming and set difference metrics. In fact, our solution is also better than the second (interactive) solution of [BDK$^+$05] in the "secure authentication" scenario: their solution achieves computational (rather than information-theoretic) security, and is impractical unless additional assumptions (such as the existence of public parameters) are made.

The keyed case with errors. Recent work focusing on the BSM with errors [Din05,DS05] shows that a constant relative Hamming distance between W_j and W'_j (recall, these are the samples recorded by the two parties) can be tolerated using a non-interactive protocol. The solution of [Din05] is stateful (i.e., SK is updated between each time period) while the second solution [DS05] requires the

[5] The journal version fixes some incorrect claims made in [MW97].

parties to communicate over an authenticated channel. We construct a robust *keyed* fuzzy extractor (for generic metrics), and show that this enables a *stateless* BSM solution (for the Hamming metric) using an *unauthenticated* channel. In doing so, we retain essentially all other parameters of the previous BSM solutions.

2 Definitions

If S is a set, $x \leftarrow S$ means that x is chosen uniformly from S. If X is probability distribution, then $x \leftarrow X$ means that x is chosen according to distribution X. The notation $\Pr_X[x]$ denotes the probability assigned by X to the value x. (We often omit the subscript when the probability distribution is clear from context.) If A is a probabilistic algorithm and x an input, $A(x)$ denotes the random variable $A(x;\omega)$ where random coins ω are sampled uniformly. If X is a random variable, then $A(X)$ is defined in the analogous manner. All logarithms are base 2.

The *min-entropy* of a random variable X is $\mathbf{H}_\infty(X) = -\log(\max_x \Pr_X[x])$. We define the (average) conditional min-entropy of X given Y as $\widetilde{\mathbf{H}}_\infty(X \mid Y) = -\log(\mathbf{E}_{y \leftarrow Y}(2^{-\mathbf{H}_\infty(X|Y=y)}))$. This (non-standard) definition is convenient for cryptographic purposes [DORS06,RW05].

Definition 1. *A family of efficient functions* $\mathcal{H} = \left\{h_i : \{0,1\}^n \to \{0,1\}^\ell\right\}_{i \in I}$ *is* δ-almost universal *if for all distinct* x, x' *we have* $\Pr_{i \leftarrow I}[h_i(x) = h_i(x')] \leq \delta$. *Families with* $\delta = 2^{-\ell}$ *are called* universal. ♢

Let X_1, X_2 be two probability distributions over S. Their *statistical distance* is $\mathbf{SD}\left(X_1, X_2\right) \stackrel{\text{def}}{=} \frac{1}{2}\sum_{s \in S} |\Pr_{X_1}[s] - \Pr_{X_2}[s]|$. If two distributions have statistical distance at most ε, we say they are ε-close, and write $X_1 \approx_\varepsilon X_2$. Note that ε-close distributions cannot be distinguished with advantage better than ε even by a computationally unbounded adversary.

Definition 2. *An efficient probabilistic function* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^\ell$ *is a* strong (m, ε)-extractor *if for all distributions* X *over* $\{0,1\}^n$ *with* $\mathbf{H}_\infty(X) \geq m$ *we have* $\mathbf{SD}\left((I, \mathsf{Ext}(X; I)), (I, U_\ell)\right) \leq \varepsilon$. *The randomness* I *is called the* seed. ♢

Lemma 1 ([BBR88,HILL99]). *Fix* $m, \varepsilon > 0$, *and* $\ell \leq m - 2\log\left(\frac{1}{\varepsilon}\right)$. *If* $\mathcal{H} = \{h_i : \{0,1\}^n \to \{0,1\}^\ell\}_{i \in I}$ *is a* $2^{-\ell}(1+\varepsilon^2)$*-almost universal family then* $\mathcal{H}$ *is a strong* (m,ε)*-extractor.*

One-time message authentication codes (MACs). One-time MACs allow information-theoretic authentication of a message using a key shared in advance.

Definition 3. *A function family* $\left\{\mathsf{MAC}_\mu : \{0,1\}^{\tilde{n}} \to \{0,1\}^v\right\}$ *is a* strongly δ-secure (one-time) MAC *if: (a) for any* x *and* σ, $\Pr_\mu\left[\mathsf{MAC}_\mu(x) = \sigma\right] = 2^{-v}$*; and (b) for any* $x \neq x'$ *and any* σ, σ', $\Pr_\mu\left[\mathsf{MAC}_\mu(x') = \sigma' \mid \mathsf{MAC}_\mu(x) = \sigma\right] \leq \delta$. ♢

The definition above is stronger than usual, since part (b) requires security conditioned on a worst-case choice of σ, rather than taking an average over μ. However, it is convenient because it is satisfied by standard constructions, and also composes nicely with universal hash families:

Lemma 2. *If* $\{\mathsf{MAC}_\mu : \{0,1\}^u \to \{0,1\}^v\}$ *is a strongly* δ*-secure MAC and* $\left\{h_i : \{0,1\}^{\tilde{n}} \to \{0,1\}^u\right\}$ *is* δ'*-almost universal, then* $\mathsf{MAC}'_{\mu,i}(x) = \mathsf{MAC}_\mu(h_i(x))$ *is a strongly* $(\delta + \delta')$*-secure MAC for* $\tilde{n}$*-bit messages.*

Proof. Regarding condition (a), it holds irrespective of the inner hash family $\{h_i\}$, by the condition (a) of the outer MAC. As for condition (b), for any distinct $x, x' \in \{0,1\}^n$ and arbitrary $\sigma, \sigma' \in \{0,1\}^v$ we have:

$$\begin{aligned}
&\Pr_{\mu,i}\left[\mathsf{MAC}_\mu(h_i(x')) = \sigma' \mid \mathsf{MAC}_\mu(h_i(x)) = \sigma\right] \\
&\qquad \le \Pr_{\mu,i}[h_i(x) = h_i(x') \mid \mathsf{MAC}_\mu(h_i(x)) = \sigma] \\
&\qquad\quad + \max_{y \neq y'}\left\{\Pr_\mu\left[\mathsf{MAC}_\mu(y') = \sigma' \mid \mathsf{MAC}_\mu(y) = \sigma\right]\right\} \;\; \le \;\; \delta' + \delta\,,
\end{aligned}$$

where the conditioning on $\mathsf{MAC}_\mu(h_i(x)) = \sigma$ is dropped due to property (a) of the outer MAC. □

Secure Sketches and Fuzzy Extractors. We start by reviewing the definitions of (ordinary) secure sketches and fuzzy extractors from [DORS06]. Let $\mathcal{M}$ be a metric space with distance function dis.

Definition 4. *An* $(m, \tilde{m}, t)$-secure sketch *is a pair of efficient randomized procedures (*$\mathsf{SS}, \mathsf{SRec}$*) s.t.:*

1. *The sketching procedure* SS *on input* $w \in \mathcal{M}$ *returns a bit string* $s \in \{0,1\}^*$*. The recovery procedure* SRec *takes an element* $w' \in \mathcal{M}$ *and* $s \in \{0,1\}^*$*.*
2. Correctness: *If* $\mathsf{dis}(w, w') \le t$ *then* $\mathsf{SRec}(w', \mathsf{SS}(w)) = w$*.*
3. Security: *For any distribution* W *over* $\mathcal{M}$ *with min-entropy* m*, the (average) min-entropy of* W *conditioned on* s *does not decrease very much. Specifically, if* $\mathbf{H}_\infty(W) \ge m$ *then* $\widetilde{\mathbf{H}}_\infty(W \mid \mathsf{SS}(W)) \ge \tilde{m}$*.*

The quantity $m - \tilde{m}$ *is called the* entropy loss *of the secure sketch.* ♢

For the case of the Hamming metric on $\mathcal{M} = \{0,1\}^n$, the following "code-offset" construction [BBR88,Cré97] is well known. The sketch $s = \mathsf{SS}(w)$ consists of the syndrome[6] of w with respect to some (efficiently decodable) $[n, k, 2t+1]$-error-correcting code C. We do not need any details of this construction other than the facts that s is a (deterministic) *linear function* of w and that the entropy loss is at most $|s| = n - k$.

Definition 5. *An* $(m, \ell, t, \varepsilon)$-fuzzy extractor *is a pair of efficient randomized procedures (*$\mathsf{Gen}, \mathsf{Rep}$*) with the following properties:*

1. *The generation procedure* Gen*, on input* $w \in \mathcal{M}$*, outputs an extracted string* $R \in \{0,1\}^\ell$ *and a helper string* $P \in \{0,1\}^*$*. The reproduction procedure* Rep *takes an element* $w' \in \mathcal{M}$ *and a string* $P \in \{0,1\}^*$ *as inputs.*
2. Correctness: *If* $\mathsf{dis}(w, w') \le t$ *and* $(R, P) \leftarrow \mathsf{Gen}(w)$*, then* $\mathsf{Rep}(w', P) = R$*.*
3. Security: *For any distribution* W *over* $\mathcal{M}$ *with min-entropy* m*, the string* R *is close to uniform even conditioned on the value of* P*. Formally, if* $\mathbf{H}_\infty(W) \ge m$ *and* $(R, P) \leftarrow \mathsf{Gen}(W)$*, then we have* $\mathbf{SD}\left((R, P), (U_\ell, P)\right) \le \varepsilon$*.* ♢

Composing an $(m, \tilde{m}, t)$-secure sketch with a strong $(\tilde{m} - \log\left(\frac{1}{\varepsilon}\right), \varepsilon)$-extractor $\{h_i \colon \mathcal{M} \to \{0,1\}^\ell\}$ yields a $(m, \ell, t, 2\varepsilon)$-fuzzy extractor [DORS06]. In that case $P = (\mathsf{SS}(w), i)$ and $R = h_i(w)$.

[6] For a linear code with parity check matrix H, the syndrome of w is $wH^\top$.

2.1 Robust Fuzzy Extractors

Fuzzy extractors, defined above, protect against a *passive* attack in which an adversary observes P, and tries to learn something about the extracted key R. However, the definition says nothing about what happens if an adversary can modify P as it is sent to the user holding w'. That is, there are no guarantees about the output of $\mathsf{Rep}(w', \tilde{P})$ for $\tilde{P} \neq P$.

Boyen *et al.* [BDK$^+$05] propose the notion of *robust* fuzzy extractors which provides strong guarantees against such an attack. Specifically, Rep can output either a key or a special value $\perp$ ("fail"). We require that any value $\tilde{P} \neq P$ produced by the adversary given P causes $\mathsf{Rep}(w', \tilde{P})$ to output $\perp$. Modified versions of the correct public information P can therefore be detected.

We consider two variants of this idea, depending on whether Gen and Rep additionally share a (short) long-term key SK. Boyen *et al.* considered the keyless primitive; this is what we define first. Further below, we adjust the definitions to the case of a shared key.

If W, W' are two (correlated) random variables over a metric space $\mathcal{M}$, we say $\mathsf{dis}(W, W') \leq t$ if the distance between W and W' is at most t with probability one. We call (W, W') a (t, m)-*pair* if $\mathsf{dis}(W, W') \leq t$ and $\mathbf{H}_\infty(W) \geq m$.

Definition 6. *Given algorithms* $(\mathsf{Gen}, \mathsf{Rep})$ *and values* $w, w' \in \mathcal{M}$*, consider the following game involving an adversary* $\mathcal{A}$*: Compute* $(R, P) \leftarrow \mathsf{Gen}(w)$ *and* $\tilde{P} = \mathcal{A}(R, P)$*. The adversary* succeeds *if* $\tilde{P} \neq P$ *and* $\mathsf{Rep}(w', \tilde{P}) \neq \perp$.

$(\mathsf{Gen}, \mathsf{Rep})$ *is a (strong)* $(m, \ell, t, \varepsilon, \delta)$-robust fuzzy extractor *if it is an* $(m, \ell, t, \varepsilon)$-*fuzzy extractor, and for all* (t, m)*-pairs* (W, W') *and all adversaries* $\mathcal{A}$*, the probability of success is at most* δ*. The notion of a* $(m, \ell, t, \varepsilon, \delta)$-weakly-robust fuzzy extractor *is defined similarly, except that* $\mathcal{A}$ *is given only* P *and not* R*.*

See Fig. 1 for an illustration. $\diamondsuit$

Re-using Robust Extractors. The definition of robust extractors composes with itself in some situations. For example, a generalization of the above (used in [BDK$^+$05]) allows the adversary to output $(\tilde{P}_1, \ldots, \tilde{P}_j)$; the adversary succeeds if there exists an i with $\mathsf{Rep}(w', \tilde{P}_i) \neq \perp$. A simple union bound shows that the success probability of an adversary in this case increases at most linearly in j.

Similarly, suppose that two players (Alice and Bob) receive a sequence of correlated pairs of random variables $(W_1, W_1'), (W_2, W_2') \ldots$, such that each pair is at distance at most t and the entropy of W_i conditioned on information from other time periods $\left\{(W_j, W_j')\right\}_{j \neq i}$ is at least m (we call such a sequence (t, m)-*correlated*). Once again, a simple hybrid argument shows that Alice and Bob can agree on (essentially) random and uncorrelated keys $R_1, R_2, \ldots$, by having Alice apply Gen to each W_i and send P_i to Bob. Namely, after j periods the attacker's advantage at distinguishing the vector of unknown keys from random is at most $j\varepsilon$, and her probability of forging a valid message $\tilde{P}_i$ is at most δ in each period.

Keyed Robust Fuzzy Extractors. In some scenarios, such as the bounded storage model, the parties running Gen and Rep can additionally share a short, truly random key to help them extract a (long) session key R from close variables

W and W'. Syntactically, this simply means that Gen and Rep now also take an extra input SK: namely, we have $(R, P) \leftarrow \mathsf{Gen}_{\mathsf{SK}}(w)$, $R' = \mathsf{Rep}_{\mathsf{SK}}(w', P)$, and require that for any SK we have $R = R'$ whenever $\mathsf{dis}(w, w') \leq t$.

The robustness property of keyed fuzzy extractors (Def. 6) does not change with the addition of SK: in particular, the attacker does not get the secret key SK in the unforgeability game of Def. 6. At first glance, this appears to trivialize the problem of constructing keyed robust fuzzy extractors. For example, one might attempt the following transformation. Given an output (R, P) or a regular fuzzy extractor, let SK be a key to an information-theoretic MAC, and simply append to P its own tag $\mathsf{MAC}_{\mathsf{SK}}(P)$ computed using SK. This transformation is not sufficient, however, because keyed fuzzy extractors must satisfy a very strong security (i.e. extraction) condition which we define next.

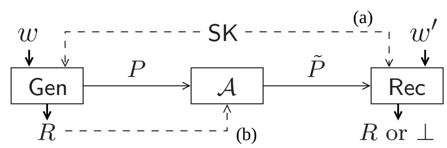

Fig. 1. Robustness of extractors (Def. 6). Dotted lines indicate variations in the definition. (a) *Keyed extractors* take an additional input SK shared by Gen and Rep. (b) For *weak robustness*, the adversary does not have access to the extracted key R.

Definition 7. *A keyed* keyed *$(m, \ell, t, \varepsilon)$-fuzzy extractor ($\mathsf{Gen}, \mathsf{Rep}$) is* secure *if for any distribution W over $\mathcal{M}$ with min-entropy m, the string (SK, R) is close to a pair of fresh uniform random strings, even conditioned on the value of P: if $\mathbf{H}_\infty(W) \geq m$ and $(R, P) \leftarrow \mathsf{Gen}_{\mathsf{SK}}(W)$, then $(\mathsf{SK}, R, P) \approx_\varepsilon (U_{|\mathsf{SK}|}, U_\ell, P)$.*

We say the extractor is strongly secure *if $(\mathsf{SK}, R, P) \approx_\varepsilon (U_{|\mathsf{SK}|}, U_\ell, U_{|P|})$.* $\diamondsuit$

The security condition for keyed extractors ensures that value of SK is essentially independent from the attacker's view. For example, the simplistic transformation above from regular fuzzy extractors leaks the tag of P (which is a known deterministic function of SK) to the attacker $\mathcal{A}$, implying that SK no longer looks random to $\mathcal{A}$ — it therefore does not satisfy Def. 7. This security condition is important for two reasons: first, it means that the session key R remains secure even if the long-term key SK is revealed in the future; second, the long-term key can be re-used (e.g., for future authentication). If Alice and Bob are given a sequence of j correlated pairs (as discussed above), then $\mathcal{A}$ has advantage at most $j\varepsilon$ in distinguishing the vector of unknown session keys from random. Similarly, her probability of forging a valid $\tilde{P}_j$ in the j-th execution of the robustness game (Def. 6) is at most $j\varepsilon + \delta$. The repeated game and its reduction to the one-time definitions presented here are given in detail in the full version of this paper.

Finally, we note that some settings require a more stringent condition called *strong* security in Def. 7. In this case the adversary's view hides both the long-term key SK and the exact distribution of W (since P is distributed identically regardless of W). The bounded storage model, discussed in Section 4, is an example of such a setting.

3 Constructing Robust Fuzzy Extractors

In this section, we describe new constructions of robust fuzzy extractors. In particular, these solve the problem of secret key generation over a completely insecure channel. We begin by analyzing the case of no errors (i.e., $t = 0$) and then consider the more challenging case where errors may occur.

3.1 The Errorless Case ($w = w'$)

Recall the "standard" solution using strong extractors, which works when the adversary is *passive*. In this case, $\mathsf{Gen}(w)$ chooses a random seed i for a strong extractor and sets $R = h_i(w)$ and $P = i$; the recovery procedure $\mathsf{Rep}(w, i)$ simply outputs $R = h_i(w)$. Unfortunately, this solution does not work if the adversary is *active*. In particular, if $i' \neq i$ there is no longer any guarantee on the output $\mathsf{Rep}(w, i')$ (and it is easy to show counterexamples where a malicious i' completely determines $\mathsf{Rep}(w, i')$ even if w is uniform). One idea is to somehow authenticate i using the extracted key R; in general, this does not work either. It turns out, however, that w itself can be used to authenticate i, at least for a particular choice of MAC and a particular strong extractor. Details follow.

Construction. We define the procedures ($\mathsf{Gen}, \mathsf{Rep}$). To compute $\mathsf{Gen}(w)$, parse the n-bit string w as two strings a and b of lengths $n - v$ and v, respectively (the value of v will be determined later). View a as an element of $\mathbb{F}_{2^{n-v}}$ and b as an element of $\mathbb{F}_{2^v}$ so that addition in the field corresponds to exclusive-or of bit strings. Choose a random $i \in \mathbb{F}_{2^{n-v}}$, compute $ia \in \mathbb{F}_{2^{n-v}}$, and let $[ia]_{v+1}^{n-v}$ denote the most significant $n-2v$ bits of ia and $[ia]_1^v$ denote the remaining v bits. View $[ia]_1^v$ as an element of $\mathbb{F}_{2^v}$. Then compute $\sigma = [ia]_1^v + b$, set $P = (i, \sigma)$, and let the extracted key be $R = [ia]_{v+1}^{n-v}$.

$\mathsf{Rep}(w, P')$, where $P' = (i', \sigma')$, proceeds as follows. Parse w as two strings a and b as above. Then verify that $\sigma' = [i'a]_1^v + b$ and output $\perp$ if this is not the case. Otherwise, compute the extracted key $R = [i'a]_{v+1}^{n-v}$.

Theorem 1. *Let $|w| = n$. For an appropriate setting of v, the above construction is an $(m, \ell, 0, \varepsilon, \delta)$-weakly-robust extractor for any $m, \ell, \varepsilon, \delta$ satisfying $\ell \leq 2m - n - 2\max\left\{\log\left(\frac{1}{\delta}\right), 2\log\left(\frac{1}{\varepsilon}\right)\right\}$. It is an $(m, \ell, 0, \varepsilon, \delta)$-robust extractor when $\ell \leq \min\left(\frac{2m-n}{3} - \frac{2}{3}\log\left(\frac{1}{\delta}\right), 2m - n - 4\log\left(\frac{1}{\varepsilon}\right)\right)$.*

The proof is given in the full version. Observe that extraction is possible as long as $\mathbf{H}_\infty(W) \stackrel{\text{def}}{=} m > |W|/2$. Furthermore, in the case of weakly robust extraction (which is the notion of security considered by Maurer and Wolf [MW03]) we extract a key of length roughly $2 \cdot \mathbf{H}_\infty(W) - |W|$.

3.2 Authenticating a Message While Extracting

The above construction uses the input w to authenticate the extractor seed i. It can be extended to additionally authenticate a (bounded-length) message M; i.e., to be simultaneously a robust fuzzy extractor and an information-theoretic one-time MAC. In this case, both Gen and Rep will take an additional input M, and it should be difficult for an adversary to cause Rep to accept a different M. (We are being informal here since this is merely a stepping stone to the results of the following section.) Naturally, this could be done easily by using (a part of) R as a key for a MAC, but this would correspondingly reduce the final number of extracted bits. In contrast, the approach presented here (almost) does not reduce the length of R at all. We adapt a standard technique [WC81] for authenticating long messages using almost-universal hash functions.

Construction. Assume $|M| \le L \cdot (n - v)$, where L is known to all parties in advance. Split M into L chunks $M_0, \ldots, M_{L-1}$, each $n - v$ bits long, and view these as coefficients of a polynomial $M(x) \in \mathbb{F}_{2^{n-v}}[x]$ of degree $L - 1$. Modify the construction of the previous section as follows: to compute $\mathsf{Gen}(w, M)$, parse w as $a\|b$, choose random $i \in \mathbb{F}_{2^{n-v}}$, compute $\sigma = [a^2 M(a) + ia]_1^v + b$, and set $P = (i, \sigma)$. As before, the extracted key is $R = [ia]_{v+1}^{n-v}$.

Given w, M', and $P' = (i', \sigma')$, verify that $|M'| \le L(n - v)$ and that $\sigma' = [a^2 M'(a) + i'a]_1^v + b$. If so, compute $R = [i'a]_{v+1}^{n-v}$.

The property we need is that every distinct pair of tuples $(M, i) \ne (M', i')$ the "difference" polynomial $f(x) = x^2(M(x) - M'(x)) + (i - i')x$ is non-constant and has degree at most $L + 1$. The analysis is deferred to the full version.

3.3 Adding Error-Tolerance

We can now consider settings when the input w' held by the receiver is close, but not identical to, the value w used by the sender. An obvious first attempt, given the scheme just discussed, is to include a secure sketch $s = \mathsf{SS}(w)$ along with (i, σ), and to authenticate s using the message authentication technique discussed previously; s would allow recovery of w from w', and then verification could proceed as before. Unfortunately, this does not quite work: if the adversary modifies s, then a different value $w^* \ne w$ may be recovered; however, the results of the previous section apply only when the receiver uses the same w as used by the sender (and so in particular we can no longer claim that the adversary could not have modified s without being detected). In effect, we have a circularity: the receiver uses w to verify that s was not modified, but the receiver computes w (from w') using a possibly modified s.

We show how to break this circularity using a modification of the message authentication technique used earlier. One key idea is to use the part of w that is "independent of s" (in a way made clear below) to authenticate s.

The second key idea is to exploit algebraic structure in the metric space, and to change the message authentication code so that it remains secure *even when the adversary can influence the key* (sometimes called security against related-key attacks). Specifically, we will assume that the errors are small in the Hamming metric, and that we are given a deterministic, linear secure sketch (for example, a syndrome-based construction). We will later extend the approach to a related *set difference* metric.

Construction for Hamming Errors. Suppose the input w is an n bit string. Our starting point is a *deterministic, linear* secure sketch $s = \mathsf{SS}(w)$ that is k

bits long; let $n' = n-k$. We assume that SS is a surjective, linear function (this is the case for the code-offset construction for the Hamming metric), and so there exists an $k \times n$ matrix S of rank k such that $\mathsf{SS}(w) = Sw$ (see footnote 6. Let $S^\perp$ be an $n' \times n$ matrix such that the $n \times n$ matrix $\left(\frac{S}{S^\perp}\right)$ has full rank. We let $\mathsf{SS}^\perp(w) \stackrel{\text{def}}{=} S^\perp w$. One can view $\mathsf{SS}^\perp(w)$ as the information remaining in w once $\mathsf{SS}(w)$ has been learned by the adversary.

$\mathsf{Gen}(w)$:

1. Set $s = \mathsf{SS}(w)$, $c = \mathsf{SS}^\perp(w)$
   - Parse c as $a \| b$ with $|a| = n' - v$ and $|b| = v$
   - Let $L = 2\lceil k/(2(n'-v))\rceil$. Pad s with 0s to length $L(n'-v)$. Parse s as $s_{L-1} \| s_{L-2} \| \dots \| s_0$, for $s_i \in GF(2^{n'-v})$.
2. Select $i \leftarrow GF(2^{n'-v})$
   - Define $f_{s,i}(x) = x^{L+3} + x^2(s_{L-1}x^{L-1} + s_{L-2}x^{L-2} + \cdots + s_0) + ix$
3. Set $\sigma = [f_{s,i}(a)]_1^v + b$ and output $R = [ia]_{v+1}^{n'-v}$ and $P = (s, i, \sigma)$.

To reproduce R given w' and $P' = (s', i', \sigma')$, first compute $w^* = \mathsf{SRec}(w', s')$; make sure $\mathsf{SS}(w^*) = s'$ and $\mathsf{dis}(w^*, w') \le t$ (if not, output $\perp$). Let $c' = \mathsf{SS}^\perp(w^*)$; parse c' as $a' \| b'$. Compute σ^* as above using s', a', b', i', and check that this matches the value σ' received. If so, output $R = [i'a']_{v+1}^{n'-v}$, else output $\perp$.

The polynomial $f_{s,i}$ defined above differs from the message authentication technique in the previous section only in the leading term x^{L+3} (and the forcing of L to be even). It has the property that for any pair $(s', i') \neq (s, i)$, and for any fixed offset Δ_a, the polynomial $f_{s,i}(x) - f_{s',i'}(x + \Delta_a)$ is a non-constant polynomial of degree at most $L+2$ (this is easy to see for $\Delta_a = 0$; if $\Delta_a \neq 0$, then the leading term is $((L+3) \bmod 2)\Delta_a x^{L+2}$). In our analysis, we use the linearity of the scheme to understand the offset $\Delta_a = a' - a$, and conclude that the adversary succeeds only if she can guess the last v bits of $f_{s,i}(x) - f_{s',i'}(x + \Delta_a)$, which happens with low probability. Note that this definition of $f_{s,i}$ amounts to a message authentication code (MAC) provably secure against a class of related key attacks where the adversary Eve can force the receiver to use a key shifted by an offset of Eve's choice. We obtain:

Theorem 2. *Let $|w| = n$. Assume SS is an $(m, m-k, t)$-secure sketch for the Hamming metric. Then for an appropriate setting of v, the above construction is an $(m, \ell, t, \varepsilon, \delta)$-weakly robust fuzzy extractor for any $m, \varepsilon, \delta, \ell \ge 0$ satisfying $\ell \le 2m - n - k - 2t\log(\frac{en}{t}) - 2\log(\frac{n}{\varepsilon^2\delta}) - O(1)$. It is an $(m, \ell, t, \varepsilon, \delta)$-strongly robust extractor when $\ell \le \frac{1}{3}\big(2m - n - k - 2t\log(\frac{en}{t}) - 2\log(\frac{n}{\varepsilon^2\delta})\big) - O(1)$.*

The proof of this theorem is deferred to the full version. We briefly discuss the parameters in the statement. In the weakly robust case, the bound on ℓ differs in two large terms from the errorless bound $2m - n$ (assuming, say, that $\varepsilon, \delta = 2^{-o(n)}$). First, we lose the length of the sketch k. This is not surprising, since we need to publish the sketch in order to correct errors.[7] The second term $2t(\log \frac{en}{t})$ is also easy to explain, although it appears to be a technicality arising from our analysis. Our analysis essentially starts by giving the attacker the error pattern $\Delta = w' \oplus w$ "for free", which in the worst case can reduce the min-entropy w by the logarithm of volume of the Hamming ball of radius t. This logarithm is at most $t \log \frac{en}{t}$. Our analysis can, in fact, yield a more general result: if $\hat{m} = \widetilde{\mathbf{H}}_\infty(W \mid \Delta)$, then $2(m - t \log \frac{en}{t})$ in the above bounds on ℓ simply gets replaced with $2\hat{m}$. For instance, when knowing the error pattern $w' \oplus w$ does not reduce the entropy of w (say, the errors are independent of w, as in the work of Boyen [Boy04]), then the term $2t \log \frac{en}{t}$ disappears from the bounds.

The analysis gives away Δ since we can then use the linearity of the sketch to conclude that the adversary knows the difference between the original input w and the value w^* that $\mathsf{Rep}(w', \tilde{P})$ reconstructs. This means she knows $\Delta_a = a' - a$, and we can use the properties of $f_{s,i}$ to bound the forgery probability.

Extension to Set Difference Metric. The analysis of the previous section relies heavily on the linearity of the secure sketch used in the protocol and on the structure of the Hamming space. We briefly indicate how it can be extended to the set difference metric. In the *set difference* metric, inputs in W are sets of at most r elements in a large universe $[N] = \{1, ..., N\}$, and the distance between two sets is the size of their symmetric difference. This is geometrically identical to the Hamming metric, since one can represent sets as characteristic vectors in $\{0,1\}^N$. However, the efficiency requirement is much stricter: for set difference, we require that operations take time polynomial in the description length of the inputs, which is only $r \log N$, not N.

In order to extend the analysis of the previous section to handle this different representation of the input, we need a pair of functions $\mathsf{SS}(), \mathsf{SS}^\perp()$ that take sets and output bit strings of length k and $(r \log N) - k$, respectively. A set w of size up to r should be unique given the pair $(\mathsf{SS}(w), \mathsf{SS}^\perp(w))$, and the functions should possess the following linearity property: the addition or removal of a particular element to/from the set should correspond to adding a particular bit vector to the output. The $\mathsf{SS}()$ function of the BCH secure sketch of Dodis et al. [DORS06] (called "PinSketch") is, in fact, linear; it outputs t values of $\log N$ bits each in order to correct up to t errors, thus producing sketches of length $k = t \log N$. For $\mathsf{SS}^\perp()$, we can use the last $r - t$ values computed by PinSketch with error parameter r. Since N is large, $t \log N$ is a good upper bound on the logarithm of the volume of the ball of radius t. We obtain the following statement, proved in the full version:

Corollary 1. *For any $r, m, t, \varepsilon, \delta$, there exists a weakly robust fuzzy extractor for set difference over sets of size up to r in $[N]$ with extracted key length* $\ell = 2m - (r + 3t) \log N - O(\log(\frac{r \log N}{\varepsilon\delta}))$, *and a strongly robust extractor with* $\ell = \frac{1}{3}\left(2m - (r + 3t) \log N\right) - O(\log(\frac{r \log N}{\varepsilon\delta}))$.

[7] In fact, a more naive construction would lose $2k$, since the sketch reduces the min-entropy m by k, and blindly applying the errorless bound $2m - n$ would double this loss. The use of $\mathsf{SS}^\perp$ is precisely what allows us not to lose the value k twice.

4 Keyed Robust Fuzzy Extractors and Their Applications

In this section we show that the addition of a very short secret key SK allows us to achieve considerably better parameters when constructing *keyed* robust fuzzy extractors. The parameters are optimal up to constant factors.

To motivate our construction, let us recall the naive transformation from regular fuzzy extractors to keyed robust fuzzy extractors discussed in Section 2. Suppose we start from the generic construction of a fuzzy extractor: $P = (s, i)$, where $s \leftarrow \mathsf{SS}(w)$, and $R = \mathsf{Ext}(w; i)$ where Ext is a strong extractor. In an attempt to make this construction robust, we set $\mathsf{SK} = \mu$ and $\sigma = \mathsf{MAC}_\mu(s, i)$, and redefine P to also include the tag σ. The problem is that the value σ allows the attacker to distinguish the real key μ from a random key $U_{|\mu|}$, since the attacker knows the authenticated message (s, i). Thus this scheme fails to meet the extraction requirement (Def. 7).

We can change the scheme to avoid this. First, note that Rep must recover the input $w = \mathsf{Rec}(w', s)$ before computing R. Thus, we can add w to the authenticated message without sacrificing the correctness of the scheme: that is, set $\sigma = \mathsf{MAC}_\mu(w, s, i)$. This does not strengthen the robustness property (Def. 6), which was already satisfied by the naive scheme. However, it does help satisfy extraction (Def. 7). In the naive scheme the attacker $\mathcal{A}$ *knows the message* (s, i) *we are authenticating*. In contrast, W has high entropy from $\mathcal{A}$'s point of view, even given $\mathsf{SS}(W)$ and R (for appropriate parameters). Thus, to make the pair (P, R) independent of $\mathsf{SK} = \mu$, it suffices to construct *information-theoretic* MACs whose key μ looks independent from the tag, as long as the authenticated message has high min-entropy. In other words, if we can ensure that the MAC is simultaneously a strong randomness extractor, we can solve our problem.

4.1 Extractor-MACs

Definition 8. *A family of functions* $\left\{\mathsf{MAC}_\mu : \{0,1\}^{\tilde{n}} \to \{0,1\}^v\right\}$ *is a strong* $(\tilde{m}, \varepsilon, \delta)$*-extractor-MAC if it is simultaneously a strongly* δ*-secure one-time MAC (Def. 3) and a* $(\tilde{m}, \varepsilon)$*-strong extractor (Def. 2, where the key* μ *is the seed).* $\diamondsuit$

We can construct extractor-MACS with (essentially optimal) key length $O(\log \tilde{n} + \log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right))$. The idea is to modify the "AU" extractor construction of Srinivasan and Zuckerman [SZ99] so that it is also MAC.

Before giving an optimal construction, note that pairwise independent hash functions are simultaneously (strong) one-time MACs and strong extractors. For example, let $\alpha = (a, b)$, where $a \in \mathbb{F}_{2^u}$, $b \in \mathbb{F}_{2^v}$, and consider the function family $f_\alpha(x) = [ax]_1^v + b$, where $[ax]_1^v$ denotes the truncation of ax to the first v bits. This family is pairwise independent [CW79], and setting $u = \tilde{n}$ and $v = \log\left(\frac{1}{\delta}\right)$ gives an extractor-MAC with key length $u + v = \tilde{n} + \log\left(\frac{1}{\delta}\right)$, which is rather large. However, we can reduce the key length by first reducing the size of the input using almost-universal hashing.

Specifically, let $\{p_\beta\}$ be a $(\delta\varepsilon^2/2)$-almost-universal hash family mapping $\tilde{n}$ bits to u bits, and compose it with a pair-wise independent family $\{f_\alpha\}$ from u to v bits, where $v = \log\left(\frac{1}{\delta}\right)+1$ (i.e., $2^{-v} = \frac{\delta}{2}$). That is, set $\mathsf{MAC}_{\alpha,\beta}(w) = p_\beta(f_\alpha(w))$.

By Lemma 2, $\{MAC_{\alpha,\beta}\}$ is a strong δ-secure MAC, since $\delta\varepsilon^2/2+2^{-v} \le \delta$. Also, composing δ_1- and δ_2-almost universal families yields a $(\delta_1+\delta_2)$-almost-universal family. Thus $\{\mathsf{MAC}_{\alpha,\beta}\}$ is $(1+\varepsilon^2)2^{-v}$-almost-universal. By the left-over hash lemma (Lemma 1), it is a (m,ε)-extractor with $m = v + 2\log\left(\frac{1}{\varepsilon}\right)$.

It remains to set u so that we can construct a convenient almost universal hash family $\{p_\beta\}$. We can use a standard polynomial-based construction, also used in previous sections. The key β is a point in $\mathbb{F}_{2^u}$, and the message x is split into $c = \tilde{n}/u$ pieces $(x_1,\ldots,x_c)$, each of which is viewed as an element of $\mathbb{F}_{2^u}$. Now, set $p_\beta(x_1\ldots x_c) = x_c\beta^{c-1} + \ldots + x_2\beta + x_1$. This family is $c/2^u$-almost universal with key length u. We can set $u = v + \log(\frac{\tilde{n}}{2\varepsilon^2}) = 2\log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right) + \log\tilde{n} - 1$ to make $c/2^u < \delta\varepsilon^2/2$. This gives key length $u+2v$, and we obtain:

Theorem 3. *For any δ,ε and $\tilde{m} \ge \log\left(\frac{1}{\delta}\right)+2\log\left(\frac{1}{\varepsilon}\right)+1$, there exists a $(\tilde{m},\varepsilon,\delta)$-extractor-MAC with key length $\kappa = 2\log\tilde{n} + 3\log\left(\frac{1}{\delta}\right) + 4\log\left(\frac{1}{\varepsilon}\right)$ and tag length $v = \log\left(\frac{1}{\delta}\right) + 1$.*

4.2 Building Keyed Robust Fuzzy Extractors

We now apply the extractor-MACs to build keyed robust fuzzy extractors for $\mathcal{M}$ (which we assumed for simplicity is $\{0,1\}^n$). We start with a generic construction and set the parameters below.

Assume $(\mathsf{SS},\mathsf{SRec})$ is a $(m,\tilde{m},t)$-secure sketch with sketch length k, Ext is a strong $(\tilde{m}-\log\left(\frac{1}{\varepsilon}\right),\varepsilon)$-extractor having a seed i of length d and an output of length ℓ, and MAC is a $(\tilde{m}-\ell-\log\left(\frac{1}{\varepsilon}\right),\varepsilon,\delta)$-extractor-MAC from $\tilde{n} = n+k+d$ bits to v bits having a key μ of length κ. We now define a keyed robust fuzzy extractor with secret key $\mathsf{SK} = \mu$:

- $\mathsf{Gen}_\mu(w)$: compute sketch $s \leftarrow \mathsf{SS}(w)$, sample i at random, set key $R = \mathsf{Ext}(w;i)$, tag $\sigma = \mathsf{MAC}_\mu(w,s,i)$, $P = (s,i,\sigma)$ and output (R,P).
- $\mathsf{Rep}_\mu(w',(s',i',\sigma'))$: Let $\bar{w} = \mathsf{SRec}(w',s')$. If $\mathsf{MAC}_\mu(\bar{w},s',i') = \sigma'$, then $R = \mathsf{Ext}(\bar{w};i)$; else $R = \perp$.

Theorem 4. *The above construction is a $(m,\ell,t,4\varepsilon,\delta)$-robust keyed fuzzy extractor, which uses a secret key SK of length κ and outputs public information P of length $k+d+v$.*

The proof of this Theorem is given in the full version. We remark that if Lemma 1 (resp. Theorem 3) is used to instantiate the extractor Ext (resp. extractor-MAC MAC) above, then a $(\tilde{m},\varepsilon)$-extractor (resp. $(\tilde{m}-\ell,\varepsilon,\delta)$-extractor-MAC) is sufficient for Theorem 4 to hold. We also note that a variant of this construction (whose security is proven analogously) would move the extractor seed i into the secret key SK. Namely, set $\mathsf{SK} = (\mu,i)$, $\sigma = \mathsf{MAC}_\mu(w,S)$ and $P = (S,\sigma)$. The main advantage of this variant is that the scheme becomes non-interactive in the case of no errors (i.e., $t=0$). However, in order to keep the length of SK low one must use considerably more complicated strong extractors than those given in Lemma 1.

The Price of Authentication. We compare the parameters of Theorem 4 to the original (non-robust, non-keyed) constructions of [DORS06]. First, note that the choice of a sketch and strong extractor can be done in the same manner as for non-robust fuzzy extractors. For concreteness, assume we use almost-universal hash functions as extractors, and let us now apply Theorem 3 to choose the extractor-MAC. Then the secret key SK is the just the MAC key μ, which has length $\kappa = 2\log\tilde{n} + 3\log\left(\frac{1}{\delta}\right) + 4\log\left(\frac{1}{\varepsilon}\right)$. This is $2\log n + 3\log\left(\frac{1}{\varepsilon}\right) + 4\log\left(\frac{1}{\varepsilon}\right) + O(1)$ when $d, k = O(n)$. Second, recall from Theorem 3 that the extractor-MAC is a good extractor as long as its min-entropy threshold $\tilde{m} - \ell$ is at least $v + 2\log\left(\frac{1}{\varepsilon}\right) = 1 + \log\left(\frac{1}{\delta}\right) + 2\log\left(\frac{1}{\varepsilon}\right)$. We get security as long as $\ell \leq \tilde{m} - 2\log\left(\frac{1}{\varepsilon}\right) - (\log\left(\frac{1}{\delta}\right) + 1)$. Compared with non-robust extractors, which required $\ell \leq \tilde{m} - 2\log\left(\frac{1}{\varepsilon}\right)$ [DORS06], the keyed, robust construction loses at most $\log\left(\frac{1}{\delta}\right) + 1$ bits in the possible length of the extracted key. Finally, the length of the public information P increases by the (short) tag length $v = \log\left(\frac{1}{\delta}\right) + 1$. Overall, the parameters remain similar to the corresponding "non-robust" case.

4.3 Application to the Bounded Storage Model with Errors

We briefly recall the key elements of the bounded storage model (BSM) with errors [Din05,DS05], concentrating only on the *stateless variant* of [DS05]. Our discussion will be specific to *Hamming errors*.

In the bounded storage model, the parties (say, Alice and Bob) have a long-term secret key sk, and at each period j have access to two noisy versions X_j and X'_j of a very long random string Z_j (of length N). Both the honest parties and the attacker $\mathcal{A}$ are limited in storage to fewer than N bits. More specifically, $\mathcal{A}$ can look at Z_j and store any γN bits of information about Z_j, for $\gamma < 1$ (so that on average Z_j has entropy about $(1-\gamma)N$ from her point of view). The honest parties are also limited in their storage, but they can use their shared key to gain an advantage. For example, in "sample-and-extract" protocols [Vad04], one part of the shared key consists of a key sam for an *oblivious sampler* [BR94,Vad04]. Roughly, sam specifies n secret physical bits of X_j/X'_j which Alice and Bob will read, obtaining n-bit substrings W_j and W'_j. The properties of the sampler ensure that (a) with high probability W_j and W'_j are still close (i.e., within Hamming distance t from each other); and (b) with high probability, $\mathcal{A}$ still has some uncertainty (min-entropy $m \approx (1-\gamma)n$) about W_j and W'_j.

This setup is quite similar to the setting of keyed robust fuzzy extractors, and provides a natural application for them. Alice and Bob can use part of the shared secret as a key SK for the robust fuzzy extractor; she can then run $\mathsf{Gen}_{\mathsf{SK}}(W_j)$ to obtain (P_j, R_j) and send P_j to Bob. Bob can run $\mathsf{Rep}_{\mathsf{SK}}$ to (hopefully) get either the key R_j or $\perp$, an indication that Alice's message was modified in transmission.

However, there is a subtle difference between the setting of the keyed robust extractors induced by the BSM and the setting we considered so far, which already causes difficulties even in the case of authenticated channels [DS05]. In our model, discussed in Section 2, the (t, m)-correlated pairs (W_j, W'_j) were arbitrary but fixed *a priori*. In contrast, in the BSM $\mathcal{A}$ can adaptively choose her storage function at each period, based on what was seen so far, and therefore affect the specific (still high-entropy) conditional distribution of each sampled W_j. In particular, if the public values P_j seen by $\mathcal{A}$ could reveal something about the long-term key $sk = (sam, \mathsf{SK})$, then Eve can affect the conditional distribution of the subsequent W_j in a manner dependent on sk, making it unsound to reuse sk in the future.

On a positive side, our definition of keyed robust extractors (Def. 7) was strong enough to ensure that the public value P is statistically independent from the key SK (meaning it is safe to reuse SK). On a negative side, it still allowed the value P to depend on the *distribution of* W (which, in turn, depends on the sampling key sam), making it insufficient for reusing the sampling key sam. And this is precisely why for this application of keyed robust extractors we will need the enhanced notion of *strongly secure* keyed robust extractors mentioned in Def. 7. Namely, the public value P not only hides the secret key SK, but even *the distribution* of W: $(\mathsf{SK}, R, P) \approx_\varepsilon (U_{|\mathsf{SK}|}, U_\ell, U_{|P|})$. A similar argument to the authenticated case of [DS05] shows that strongly secure keyed robust extractors are sufficient to solve the unauthenticated setting. Thus, we turn to constructions of such robust extractors.

Strongly Secure Keyed Robust Extractors. Examining the keyed construction in Theorem 4, we see that the only place where the value $P = (S, I, \sigma)$ depends on the distribution of W is when computing the sketch $S \leftarrow \mathsf{SS}(W)$. Indeed, the seed I is chosen at random, and the value $\sigma = \mathsf{MAC}_\mu(W, S, I)$ looks random by the properties of the extractor-MAC. Thus, to solve our problem we only need to build an $(m, \tilde{m}, t)$-secure sketch SS such that $\mathsf{SS}(W)$ is statistically close to uniform whenever W has enough min-entropy: $\mathsf{SS}(W) \approx_\varepsilon U_{|\mathsf{SS}(W)|}$ (notice, such sketches can no longer be deterministic). Luckily, such (probabilistic) sketches, called $(m, \tilde{m}, t, \varepsilon)$*-extractor-sketches* (or "entropically-secure" sketches) were studied by Dodis and Smith [DS05], since they were already needed to solve the noisy BSM problem even in the authenticated channel case. In particular, [DS05] built extractor-sketches for the binary Hamming metric whose parameters were only a constant factor worse than those of regular sketches.

Theorem 5 ([DS05]). *For any min-entropy $m = \Omega(n)$, there exists efficient $(m, \tilde{m}, t, \varepsilon)$-extractor-sketches for the Hamming metric over $\{0,1\}^n$, where $\tilde{m}$, t and $\log\left(\frac{1}{\varepsilon}\right)$ are all $\Omega(n)$, and the length of the sketch is $O(n)$.*

Returning to the construction of strongly secure, keyed robust extractors, we observe that Theorem 4 indeed yields such extractors (with error 5ε instead of 4ε) if one uses $(m, \tilde{m}, t, \varepsilon)$-extractor-sketches in place of regular $(m, \tilde{m}, t)$-sketches. Combining this observation with Theorem 5, we obtain:

Theorem 6. *For any min-entropy $m = \Omega(n)$, there exists efficient $(m, \ell, t, \varepsilon, \delta)$-strongly secure, robust, keyed fuzzy extractor for the Hamming metric over $\{0,1\}^n$, which uses a secret key of length $O(\log n + \log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right))$, tolerates $t = \Omega(n)$ errors, extracts $\ell = \Omega(n)$ bits, and has public information P of length $O(n)$.*

Application to the BSM. As we stated, after Alice and Bob use the shared sampling key to obtain close n-bit strings W and W', respectively, they will use a strongly secure, keyed, robust fuzzy extractor $(\mathsf{Gen}_{\mathsf{SK}}, \mathsf{Rep}_{\mathsf{SK}})$ to agree on a session key R over an unauthenticated channel. To get a specific construction, we can use Theorem 6 above. In doing so, we see that the only difference between the resulting scheme and the solution of Dodis and Smith [DS05] (for the authenticated channel) is that Alice and Bob additionally share a (short) extractor-MAC key SK, and also append a (short) extractor-MAC of (W, S, I) to the public information (S, I) that Alice sends to Bob. Therefore, our construction retains the nearly optimal parameters of [DS05], while also adding authentication.

More specifically, assume N, ε, δ are given. Since the number of read bits n can be chosen by Alice and Bob, it is convenient to specify the required number of extracted bits ℓ, and choose n afterwards. Then we obtain a stateless protocol in the BSM model with Hamming errors satisfying: (1) key reuse (stateless) and everlasting security; (2) having shared secret key sk of size $O(\log N + \log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right))$; (3) having forgery probability at most δ against active attacker; (4) having Alice and Bob read $n = O(\ell)$ random bits W from the source and extract ℓ bits R which are ε-close to uniform; (5) having Alice and Bob tolerate linear fraction of errors (i.e., $t = \Omega(n)$); and (6) having Alice send a single $O(\ell)$-bit message to Bob. All these parameters are optimal up to a constant factor.

Acknowledgments. We would like to thank Hoeteck Wee for his collaboration at the early stages of this work.

References

[ADR02] Y. Aumann, Y. Ding, and M. Rabin. Everlasting security in the bounded storage model. *IEEE Trans. on Information Theory*, 48(6):1668–1680, 2002.

[BBCM95] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer. Generalized privacy amplification. *IEEE Trans. on Information Theory*, 41(6), 1995.

[BBR88] C. Bennett, G. Brassard, and J. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[BDK+05] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith. Secure remote authentication using biometric data. In *EUROCRYPT 2005*, Springer.

[BMP00] V. Boyko, P. MacKenzie, and S. Patel. Provably-secure password-authenticated key exchange using Diffie-Hellman. In *EUROCRYPT 2000*.

[Boy04] Xavier Boyen. Reusable cryptographic fuzzy extractors. In *11th ACM Conference on Computer and Communication Security*. ACM, 2004.

[BPR00] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In *EUROCRYPT 2000*.

[BR94] M. Bellare and J. Rompel. Randomness-efficient oblivious sampling. In *35th Annual Symposium on Foundations of Computer Science*. IEEE, 1994.

[Cré97] Claude Crépeau. Efficient cryptographic protocols based on noisy channels. In *Advances in Cryptology—EUROCRYPT 97*, volume 1233 of *LNCS*.

[CW79] J.L. Carter and M.N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18:143–154, 1979.

[Din05] Yan Zong Ding. Error correction in the bounded storage model. In *2nd Theory of Cryptography Conference — TCC 2005*, volume 3378 of *LNCS*.

[DORS06] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. Technical Report 2003/235, Cryptology ePrint archive, `http://eprint.iacr.org`, 2006. Previous version appears in *EUROCRYPT 2004*.

[DS02] Y. Dodis and J. Spencer. On the (non-)universality of the one-time pad. In *43rd Annual Symposium on Foundations of Computer Science*. IEEE, 2002.

[DS05] Y. Dodis and A. Smith. Correcting errors without leaking partial information. In 37th Annual ACM Symposium on Theory of Computing, 2005.

[FJ01] N. Frykholm and A. Juels. Error-tolerant password recovery. In *Eighth ACM Conference on Computer and Communication Security*. ACM, 2001.

[GL01] O. Goldreich and Y. Lindell. Session-key generation using human passwords only. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 408–432, Springer.

[GL03] R. Gennaro and Y. Lindell. A framework for password-based authenticated key exchange. In *EUROCRYPT 2003*, volume 2656 of *LNCS*.

[HILL99] J. Håstad, R. Impagliazzo, L.A. Levin, and M. Luby. Construction of pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[JW99] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Sixth ACM Conference on Computer and Communication Security*, pages 28–36, 1999.

[KOY01] J. Katz, R. Ostrovsky, and M. Yung. Efficient password-authenticated key exchange using human-memorable passwords. In *EUROCRYPT 2001*.

[LT03] J.-P. M. G. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *AVBPA*, 2003.

[Mau92] Ueli Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology*, 5(1):53–66, 1992.

[Mau93] Ueli Maurer. Secret key agreement by public discussion from common information. *IEEE Transactions on Information Theory*, 39(3):733–742, 1993.

[Mau97] Ueli Maurer. Information-theoretically secure secret-key agreement by NOT authenticated public discussion. In *EUROCRYPT 97*, pp. 209–225.

[MW97] U. Maurer and S. Wolf. Privacy amplification secure against active adversaries. In *Advances in Cryptology—CRYPTO '97*, pages 307–321.

[MW03] U. Maurer and S. Wolf. Secret-key agreement over unauthenticated public channels — Part III: Privacy amplification. *IEEE Transactions on Information Theory*, 49(4):839–851, 2003.

[RW03] R. Renner and S. Wolf. Unconditional authenticity and privacy from an arbitrarily weak secret. In *Advances in Cryptology—CRYPTO 2003*.

[RW05] R. Renner and S. Wolf. Simple and tight bounds for information reconciliation and privacy amplification. In *ASIACRYPT 2005*, Springer.

[SZ99] A. Srinivasan and D. Zuckerman. Computing with very weak random sources. *SIAM Journal on Computing*, 28(4):1433–1459, 1999.

[Vad04] S. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *Journal of Cryptology*, 17(1), 2004.

[WC81] M.N. Wegman and J.L. Carter. New hash functions and their use in authentication and set equality. *J. Computer and System Sciences*, 22, 1981.

[Wol98] S. Wolf. Strong security against active attacks in information-theoretic secret-key agreement. In *ASIACRYPT '98*, volume 1514 of *LNCS*.

[Wyn75] A.D. Wyner. The wire-tap channel. *Bell System Technical Journal*, 54(8):1355–1387, 1975.