An earlier version of this paper appears in: Advances in Cryptology—EUROCRYPT 2005, Lecture Notes in Computer Science, Springer-Verlag, 2005. This is a revised version, available online at: `http://www.cs.stanford.edu/~xb/eurocrypt05b/`

Secure Remote Authentication Using Biometric Data

Xavier Boyen* Yevgeniy Dodis† Jonathan Katz‡ Rafail Ostrovsky§ Adam Smith¶

Abstract

Biometric data offer a potential source of high-entropy, secret information that can be used in cryptographic protocols provided two issues are addressed: (1) biometric data are not uniformly distributed; and (2) they are not exactly reproducible. Recent work, most notably that of Dodis, Reyzin, and Smith, has shown how these obstacles may be overcome by allowing some auxiliary public information to be reliably sent from a server to the human user. Subsequent work of Boyen has shown how to extend these techniques, in the random oracle model, to enable unidirectional authentication from the user to the server without the assumption of a reliable communication channel.

We show two efficient techniques enabling the use of biometric data to achieve *mutual* authentication or authenticated key exchange over a completely insecure (i.e., adversarially controlled) channel. In addition to achieving stronger security guarantees than the work of Boyen, we improve upon his solution in a number of other respects: we tolerate a broader class of errors and, in one case, improve upon the parameters of his solution and give a proof of security in the standard model.

1 Using Biometric Data for Secure Authentication

Biometric data, as a potential source of high-entropy, secret information, have been suggested as a way to enable strong, cryptographically-secure authentication of human users without requiring them to remember or store traditional cryptographic keys. Before such data can be used in existing cryptographic protocols, however, two issues must be addressed: first, biometric data are *not uniformly distributed* and hence do not offer provable security guarantees if used directly as, say, a key for a pseudorandom function. While the problem of non-uniformity can be addressed using a hash function, viewed either as a random oracle [2] or a strong extractor [20], a second and more difficult problem is that biometric data are *not exactly reproducible*, as two biometric scans of the

*Voltage, Inc. `xb@boyen.org`.

†New York University. Supported by NSF CAREER award #0133806 and Trusted Computing grant #0311095. `dodis@cs.nyu.edu`.

‡University of Maryland. Supported by NSF CAREER award #0447075 and Trusted Computing grants #0310751 and #0310499. `jkatz@cs.umd.edu`.

§UCLA. Supported in part by a gift from Teradata, an Intel equipment grant, an OKAWA research award, and an NSF Cybertrust grant. `rafail@cs.ucla.edu`.

¶Weizmann Institute. `asmith@csail.mit.edu`.

same feature are rarely identical. Thus, traditional protocols will not even guarantee correctness when the parties use a shared secret derived from biometric data.

Much work has focused on addressing these problems in an effort to develop secure techniques for biometric authentication [8, 15, 19, 14, 22, 21]. Most recently, Dodis, Reyzin, and Smith [9] showed how to use biometric data to securely derive cryptographic keys which could then be used, in particular, for the purposes of authentication. They introduce two primitives (see Section 2 for formal definitions): a *secure sketch* which allows recovery of a shared secret given a "close" approximation thereof, and a *fuzzy extractor* which extracts a uniformly distributed string s from this shared secret in an error-tolerant manner. Both primitives work by constructing a "public" string pub which is stored by the server and transmitted to the user; loosely speaking, pub encodes the information needed for error-tolerant reconstruction of the secret and subsequent extraction. The primitives are designed to be "secure" even when an adversary learns the value of pub (by, say, eavesdropping on the channel between the server and the user).

Unfortunately, although these primitives suffice to obtain security in the presence of an eavesdropping adversary, the work of Dodis *et al.* does not address the issue of malicious modification of pub. As a consequence, their work does not provide a method for secure authentication in the presence of an *active* adversary who may modify the messages sent between the server and the user. Indeed, depending on the specific sketch or fuzzy extractor being utilized, an adversary who maliciously alters the public string sent to a user may be able to learn that user's biometric data in its entirety. A "solution" is for the user to store pub himself rather than obtain it from the server, or to authenticate pub using a certificate chain or a MAC, but this defeats the purpose of using biometric data in the first place: namely, to avoid the need for the user to store *any* additional cryptographic information (even if that information need not be kept secret).

Boyen [5], *inter alia*, partially addresses potential adversarial modification of pub (although his work focuses primarily on the orthogonal issue of re-using biometric data with multiple servers, which we do not explicitly address here). The main drawback of his technique in our context is that it provides only *unidirectional* authentication from the user to the server. Indeed, Boyen's approach cannot be used to achieve authentication of the server to the user since his definition of "insider security" (cf. [5, Section 5.2]) does not preclude an adversary from knowing the (incorrect) value s' of the shared secret recovered by the user if the adversary forwards a modified pub' to this user; once the adversary knows s', then from the viewpoint of the user the adversary can do anything the server could do and hence authentication of the server to the user is impossible. The lack of mutual authentication implies that — when communicating over an insecure network — the user and server cannot securely establish a shared session key with which to encrypt and authenticate future messages: the user may unwittingly share a key with an adversary who can then decrypt any data sent by that user as well as authenticate arbitrary data.

1.1 Our Contributions

In this paper, we provide the first full solution to the problem of secure remote authentication using biometric[1] data: in particular, we show how to achieve mutual authentication and/or authenticated key exchange over a completely insecure channel. We offer two constructions. The first one may be viewed as a generic solution which protects against modification of the public value pub in any

[1]Of course, our techniques are applicable to *any* scenario which relies on secret data that, like biometric data, are non-uniform and/or not exactly reproducible.

context in which secure sketches or fuzzy extractors are used; thus, this solution may be viewed as a drop-in replacement that "compiles" any protocol which is secure when pub is assumed to be transmitted reliably into one which is secure even when pub might be tampered with. (We do not formalize this notion of "compilation," but rather view it as an intuitive way to understand our results.) Our second construction is specific to the settings of remote authentication and key exchange, where it offers some advantages as compared to the first solution.

Compared with the work of Boyen [5], our constructions enjoy the following additional advantages (i.e., besides achieving mutual authentication rather than unidirectional authentication):

- Both our solutions tolerate a stronger class of errors. In particular, Boyen's work only allows *data-independent* errors, whereas our analysis handles *arbitrary* (but bounded) errors. We remark that small yet data-dependent errors seem natural in the context of biometric data.
- Our second solution is proven secure in the standard model.
- Our solutions can achieve improved bounds on the entropy loss, on the order of 128 bits of entropy for practical choices of the parameters. This point is particularly important since the entropy of certain biometric features is roughly this order of magnitude (e.g., 175–250 bits for an iris scan [8, 13]).

Organization. We review some basic definitions as well as the sketches/fuzzy extractors of Dodis *et al.* [9] in Section 2. In Section 3 we introduce the notion of *robust* sketches/fuzzy extractors which are resilient to modification of the public value. In that section, we also show applications of robust fuzzy extractors to the problem of mutual authentication. In Section 4, we describe our second solution which is specific to the problem of using biometric data for authentication and offers some advantages with respect to the first construction.

2 Definitions

All logarithms are base 2. We let U_ℓ denote the uniform distribution over ℓ-bit strings. A *metric space* $(\mathcal{M}, d)$ is a set $\mathcal{M}$ equipped with a symmetric distance function $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+ \cup \{0\}$ satisfying the triangle inequality and such that $d(x, y) = 0 \Leftrightarrow x = y$; when d is unimportant, we will sometimes call $\mathcal{M}$ itself a metric space. (All metric spaces $\mathcal{M}$ considered in this work are finite, and the distances integer-valued.) For our application, we assume that the format of the biometric data is such that it forms a metric space under some appropriate distance function. We will not need to specify any particular metric space in our work, as our results build in a generic way on earlier sketch and fuzzy extractor constructions over any such space (e.g., those constructed in [9] for a variety of metrics).

A *(finite) probability space* (Ω, P) is a finite set Ω and a function $P : \Omega \to [0, 1]$ such that $\sum_{\omega \in \Omega} P(\omega) = 1$. A *random variable* W defined over the probability space (Ω, P) and taking values in a set $\mathcal{M}$ is a function $W : \Omega \to \mathcal{M}$. For such a random variable, we let $w \leftarrow W$ refer to the experiment in which $r \in \Omega$ is chosen according to P, and then w is assigned the value $W(r)$. If (Ω, P) is a probability space over which two random variables W and W' are defined, taking values in a metric space $\mathcal{M}$ with associated distance function d, then we say that $d(W, W') \leq t$ if for all $r \in \Omega$ it holds that $d(W(r), W'(r)) \leq t$.

Given a metric space $(\mathcal{M}, d)$ and a point $x \in \mathcal{M}$ we define

$$\mathsf{Vol}_t^{\mathcal{M}}(x) \stackrel{\text{def}}{=} \left|\{x' \in \mathcal{M} \mid d(x,x') \leq t\}\right|, \qquad \mathsf{Vol}_t^{\mathcal{M}} \stackrel{\text{def}}{=} \max_{x\in\mathcal{M}}\{\mathsf{Vol}_t^{\mathcal{M}}(x)\}.$$

The former is the number of points in a "ball" of radius t centered at x; the latter is the maximum number of points in any ball of radius t.

For random variables A, B, the *min-entropy* of A is given by

$$H_\infty(A) \stackrel{\text{def}}{=} -\log\left(\max_a \Pr[A=a]\right)$$

and, following [9], we define the *average min-entropy of A given B* as

$$\bar{H}_\infty(A|B) \stackrel{\text{def}}{=} -\log\left(\mathrm{Exp}_{b\leftarrow B}[2^{-H_\infty(A|B=b)}]\right).$$

The *statistical difference* between random variables A and B taking values in the same set $\mathcal{M}$ is defined as $\mathbf{SD}(A,B) \stackrel{\text{def}}{=} \frac{1}{2}\sum_{x\in\mathcal{M}} |\Pr[A=x] - \Pr[B=x]|$.

2.1 Secure Sketches and Fuzzy Extractors

We review the definitions from [9] using slightly different terminology. Recall from the introduction that a secure sketch provides a way to recover a shared secret w from any value w' which is a "close" approximation of w. More formally:

Definition 1 An (m, m', t)-*secure sketch* over a metric space $(\mathcal{M}, d)$ comprises a *sketching procedure* $\mathsf{SS} : \mathcal{M} \to \{0,1\}^*$ and a *recovery procedure* Rec, where:

(Security) For all random variables W taking values in $\mathcal{M}$ such that $H_\infty(W) \geq m$, we have $\bar{H}_\infty(W \mid \mathsf{SS}(W)) \geq m'$.

(Error tolerance) For all $w, w' \in \mathcal{M}$ with $d(w,w') \leq t$, it holds that $\mathsf{Rec}(w', \mathsf{SS}(w)) = w$. $\diamondsuit$

While secure sketches address the issue of error correction, they do not address the issue of the possible non-uniformity of W. Fuzzy extractors, defined next, correct for this.

Definition 2 An $(m, \ell, t, \varepsilon)$-*fuzzy extractor* over a metric space $(\mathcal{M}, d)$ comprises a (randomized) *extraction algorithm* $\mathsf{Ext} : \mathcal{M} \to \{0,1\}^\ell \times \{0,1\}^*$ and a *recovery procedure* Rec such that:

(Security) For all random variables W taking values in $\mathcal{M}$ and satisfying $H_\infty(W) \geq m$, if $\langle R, \mathsf{pub}\rangle \leftarrow \mathsf{Ext}(W)$ then $\mathbf{SD}(\langle R, \mathsf{pub}\rangle, \langle U_\ell, \mathsf{pub}\rangle) \leq \varepsilon$.

(Error tolerance) For all pairs of points $w, w' \in \mathcal{M}$ with $d(w,w') \leq t$, if $\langle R, \mathsf{pub}\rangle \leftarrow \mathsf{Ext}(w)$ then it is the case that $\mathsf{Rec}(w', \mathsf{pub}) = R$. $\diamondsuit$

As shown in [9, Lemma 3.1], it is easy to construct a fuzzy extractor over a metric space $(\mathcal{M}, d)$ given any secure sketch defined over the same space, by applying a strong extractor [20] using a random "key" which is included as part of pub. Starting with an (m, m', t)-secure sketch and with an appropriate choice of extractor, this yields an $(m, m' - 2\log(\frac{1}{\varepsilon}), t, \varepsilon)$-fuzzy extractor.

2.2 Modeling Error in Biometric Applications

As error correction is a key motivation for our work, it is necessary to develop some formal model of the types of errors that may occur. In prior work by Boyen [5], the error in various biometric readings was assumed to be under adversarial control with the restriction that the adversary could only specify data-*independent* errors (e.g., constant shifts). It is not clear that this is a realistic model in practice: one certainly expects, say, portions of the biometric data where "features" are present to be more susceptible to error.

Here, we consider a more general error model where the errors may be data-*dependent* and hence correlated not only with each other but also with the biometric secret itself. Furthermore, as we are ultimately interested in modeling "nature" — as manifested in the physical processes that cause fluctuations in the biometric measurements — we do not even require that the errors be efficiently computable (although we will impose this requirement in Section 4). The only restriction we make is that the errors be "small" and, in particular, less than the desired error-correction bound; since the error-correction bound in any real-world application should be selected to ensure correctness with high probability, this restriction seems reasonable. Formally:

Definition 3 A *t-bounded distortion ensemble* $\mathcal{W} = \{W_i\}_{i=0,\ldots}$ is a sequence of random variables $W_i : \Omega \to \mathcal{M}$ such that for all i we have $d(W_0, W_i) \leq t$. $\diamondsuit$

For our purposes, W_0 represents the biometric reading obtained when a user initially registers with a server, and W_i represents the biometric reading on the i^{th} authentication attempt by this user. Note that, regardless of the protocol used, an adversary can always impersonate the server if the adversary can guess W_i for some $i > 0$. The following lemmas bound the probability of this occurrence. First, we show that the min-entropy of each W_i is, at worst, $\log(\mathsf{Vol}_t^{\mathcal{M}})$ bits less than that of W_0. Moreover, we show that W_i is no easier to guess than W_0 assuming $\mathsf{SS}(W_0)$ is available.

Lemma 1 *Let W_0, W_i be random variables taking values in $\mathcal{M}$ and satisfying $d(W_0, W_i) \leq t$, and let B be an arbitrary random variable. Then*

$$\bar{H}_\infty(W_i \mid B) \geq \bar{H}_\infty(W_0 \mid B) - \log \mathsf{Vol}_t^{\mathcal{M}}.$$

Proof Fix $x \in \mathcal{M}$ and any outcome $B = b$. Since $d(W_0, W_i) \leq t$, we have $\Pr[W_i = x \mid B = b] \leq \sum_{x' | d(x,x') \leq t} \Pr[W_0 = x' \mid B = b] \leq \mathsf{Vol}_t^{\mathcal{M}} \cdot 2^{-H_\infty(W_0 | B = b)}$, which means that $H_\infty(W_i \mid B = b) \geq H_\infty(W_0 \mid B = b) - \log \mathsf{Vol}_t^{\mathcal{M}}$. Since this inequality holds for every b, the lemma follows. $\blacksquare$

We can prove better bounds on the "entropy loss" of W_i if a sketch of W_0 is already available. The intuition is that in this case a correct guess for W_i implies a correct guess of W_0.

Lemma 2 *Let W_0, W_i be random variables taking values in $\mathcal{M}$ and satisfying $d(W_0, W_i) \leq t$, and let B be an arbitrary random variable. Let $(\mathsf{SS}, \mathsf{Rec})$ be a $(\star, \star, t)$-secure sketch. Then*

$$\bar{H}_\infty(W_i \mid \mathsf{SS}(W_0), B) \geq \bar{H}_\infty(W_0 \mid \mathsf{SS}(W_0), B).$$

Proof Since $d(W_0, W_i) \leq t$, we have $\mathsf{Rec}(W_i, \mathsf{SS}(W_0)) = W_0$ which means that for any x, b, pub:

$$\Pr[W_0 = \mathsf{Rec}(x, \mathsf{pub}) \mid \mathsf{SS}(W_0) = \mathsf{pub}, B = b] \geq \Pr[W_i = x \mid \mathsf{SS}(W_0) = \mathsf{pub}, B = b].$$

Since this holds for all x, b, and pub, the lemma follows. $\blacksquare$

The analogue of Lemma 2 for fuzzy extractors holds as well (with $\mathsf{SS}(W_0)$ replaced by pub).

3 Robust Sketches and Fuzzy Extractors

Recall that a secure sketch, informally speaking, takes a secret w and returns a value $\mathsf{pub} \leftarrow \mathsf{SS}(W)$ which allows the recovery of w given any "close" approximation w' of w; a fuzzy extractor allows recovery of an "almost uniform" string using w' and pub. When pub is transmitted to a user over an insecure network, however, an adversary might modify pub in transit and, in general, no security guarantees are provided in this case by "ordinary" sketches and fuzzy extractors. In this section, we define the notion of *robust* sketches and fuzzy extractors that protect against this sort of attack in a very strong way: with high probability, the user will detect any modification of pub and can thus immediately abort in this case. We then show: (1) a construction of a robust sketch in the random oracle model, starting from any secure sketch satisfying a certain technical property; and (2) a conversion from any robust sketch to a robust fuzzy extractor, again in the random oracle model. We conclude this section by showing the immediate application of robust fuzzy extractors to the problem of mutual authentication/key exchange.

We first define a technical property for secure sketches:

Definition 4 An (m, m', t)-secure sketch $(\mathsf{SS}, \mathsf{Rec})$ is said to be *well-formed* if it satisfies the conditions of Definition 1 with the following modifications: (1) Rec may now return either an element in $\mathcal{M}$ or the distinguished symbol $\perp \notin \mathcal{M}$; and (2) for all $w' \in \mathcal{M}$ and arbitrary pub', if $\mathsf{Rec}(w', \mathsf{pub}') \neq \perp$ then $d(w', \mathsf{Rec}(w', \mathsf{pub}')) \leq t$. $\diamondsuit$

It is straightforward to transform any secure sketch $(\mathsf{SS}, \mathsf{Rec})$ into a well-formed secure sketch $(\mathsf{SS}, \mathsf{Rec}')$: Rec' runs Rec and then verifies that its output w is within distance t of the input w'. If yes, it outputs w; otherwise, it outputs $\perp$.

We now define the notion of a *robust* sketch:

Definition 5 Given algorithms $(\mathsf{SS}, \mathsf{Rec})$ and random variables $\mathcal{W} = \{W_0, W_1, \ldots, W_n\}$ over metric space $(\mathcal{M}, d)$, consider the following game between an adversary $\mathcal{A}$ and a challenger: Let w_0 (resp., w_i) be the value assumed by W_0 (resp., W_i). The challenger computes $\mathsf{pub} \leftarrow \mathsf{SS}(w_0)$ and gives pub to $\mathcal{A}$. Next, $\mathcal{A}$ outputs $(\mathsf{pub}_1, \ldots, \mathsf{pub}_n)$ with $\mathsf{pub}_i \neq \mathsf{pub}$ for all i. If there exists an i with $\mathsf{Rec}(w_i, \mathsf{pub}_i) \neq \perp$ we say the adversary *succeeds* and this event is denoted by Succ.

We say $(\mathsf{SS}, \mathsf{Rec})$ is an (m, m'', t, n, δ)*-robust sketch* over $(\mathcal{M}, d)$ if it is a well-formed (m, m'', t)-secure sketch, and for all t-bounded distortion ensembles $\mathcal{W}$ with $H_\infty(W_0) \geq m$ and all adversaries $\mathcal{A}$ we have $\Pr[\mathsf{Succ}] \leq \delta$. $\diamondsuit$

A simpler definition would be to consider only random variables $\{W_0, W_1\}$ and to have $\mathcal{A}$ only output a single value $\mathsf{pub}_1 \neq \mathsf{pub}$. A standard hybrid argument would then imply the above definition with ε increased by a multiplicative factor of n. We have chosen to work with the more general definition above as it potentially allows for a tighter concrete security analysis. Also, although the above definition allows all-powerful adversaries, we will consider adversaries whose queries to a random oracle are bounded (but are otherwise computationally unbounded).

Remark: The proceedings version of this work considered a slightly different definition in which m'' was the average min-entropy of W_0 *conditioned on the adversary's view* View *over the course of the experiment* (rather than simply conditioned on $SS(W_0)$). Given that $\Pr[\mathsf{Succ}] \leq \delta$, one can lower bound $\bar{H}_\infty(W_0 \mid \mathsf{View})$ in terms of $m'' = \bar{H}_\infty(W_0 \mid SS(W_0))$; however, as it turns out, for the application to mutual authentication in Section 3.3 the present definition is all that is needed.

3.1 Constructing a Generic Robust Sketch

Let $H : \{0,1\}^* \to \{0,1\}^k$ be a hash function that will be modeled as a random oracle. We construct a robust sketch $(\mathsf{SS}, \mathsf{Rec})$ from any well-formed secure sketch $(\mathsf{SS}^*, \mathsf{Rec}^*)$ as follows:

$\underline{\mathsf{SS}(w)}$	$\underline{\mathsf{Rec}(w, \mathsf{pub} = \langle \mathsf{pub}^*, h\rangle)}$
$\mathsf{pub}^* \leftarrow \mathsf{SS}^*(w)$	$w' = \mathsf{Rec}^*(w, \mathsf{pub}^*)$
$h = H(w, \mathsf{pub}^*)$	if $w' = \perp$ output $\perp$
return $\mathsf{pub} = \langle \mathsf{pub}^*, h\rangle$	if $H(w', \mathsf{pub}^*) \neq h$ output $\perp$
	otherwise, output w'

Theorem 1 *If $(\mathsf{SS}^*, \mathsf{Rec}^*)$ is a well-formed (m, m', t)-secure sketch over metric space $(\mathcal{M}, d)$ and $H : \{0,1\}^* \to \{0,1\}^k$ is a random oracle, then $(\mathsf{SS}, \mathsf{Rec})$ is an (m, m'', t, n, δ)-robust sketch over $(\mathcal{M}, d)$ for any adversary making at most q_H queries to H, where*

$$\begin{aligned} \delta &= (q_H^2 + n) \cdot 2^{-k} + (3q_H + 2n \cdot \mathsf{Vol}_t^{\mathcal{M}}) \cdot 2^{-m'}, \\ m'' &= m' - \log(3q_H + 2)\,. \end{aligned}$$

When $k \geq m' + \log q_H$, the above simplifies to

$$\delta \;\leq\; (4q_H + 2n \cdot \mathsf{Vol}_t^{\mathcal{M}}) \cdot 2^{-m'}.$$

Proof (Sketch) It is easy to see that $(\mathsf{SS}, \mathsf{Rec})$ is a well-formed $(m, \star, t)$-secure sketch. We first bound the success probability of any adversary in the game of Definition 5, and then compute the value m'' such that $(\mathsf{SS}, \mathsf{Rec})$ is an (m, m'', t)-secure sketch. Let $\mathsf{pub} = \langle \mathsf{pub}^*, h\rangle$ denote the value output by SS in an execution of the game described in Definition 5. Note that if $\mathsf{pub}_i = \langle \mathsf{pub}_i^*, h_i\rangle$ with $\mathsf{pub}_i^* = \mathsf{pub}^*$, then $\mathsf{Rec}(w_i, \mathsf{pub}_i) = \perp$ since $h_i \neq h$; thus we will assume $\mathsf{pub}_i^* \neq \mathsf{pub}^*$ for all i.

Fix a t-bounded distortion ensemble $\{W_0, W_1, \ldots, W_n\}$ with $H_\infty(W_0) \geq m$. For any output $\mathsf{pub}_i = \langle \mathsf{pub}_i^*, h_i\rangle$ of $\mathcal{A}$, define the random variable $W_i' \stackrel{\text{def}}{=} \mathsf{Rec}^*(W_i, \mathsf{pub}_i^*)$. In order not to complicate notation, we define

$$H_\infty(W_i') \stackrel{\text{def}}{=} -\log\left(\max_{x\in\mathcal{M}} \Pr[W_i' = x]\right);$$

i.e., we ignore the probability that $W_i' = \perp$ since $\mathcal{A}$ does not succeed in this case. $\bar{H}_\infty(W_i' \mid X)$ for a random variable X is defined similarly. Let w_0, w_i, and w_i' denote the values taken by the random variables W_0, W_i, and W_i', respectively.

We classify the random oracle queries of $\mathcal{A}$ into two types: *type 1* queries are those of the form $H(\cdot, \mathsf{pub}^*)$, and *type 2* queries are all the others. Informally, type 1 queries represent attempts by $\mathcal{A}$ to learn the value of w_0; in particular, if $\mathcal{A}$ finds w such that $H(w, \mathsf{pub}^*) = h$ then it is "likely" that $w_0 = w$. Type 2 queries represent attempts by $\mathcal{A}$ to determine an appropriate value for some h_i; i.e., if $\mathcal{A}$ "guesses" that $w_i' = w$ for a particular choice of pub_i^* then a "winning" strategy is for $\mathcal{A}$ to obtain $h_i = H(w, \mathsf{pub}_i^*)$ and output $\mathsf{pub}_i = \langle \mathsf{pub}_i^*, h_i\rangle$.

Without loss of generality, we assume that $\mathcal{A}$ makes all its type 1 queries first, followed by all its type 2 queries, and then outputs $(\mathsf{pub}_1, \ldots, \mathsf{pub}_n)$. The validity of the assumption on the ordering of the type 1 and type 2 queries follows essentially from the analysis that follows.

Let $\mathcal{Q}_1$ (resp., $\mathcal{Q}_2$) be a random variable denoting the sequence of type 1 (resp., type 2) queries made by $\mathcal{A}$ and the corresponding responses, and let q_1 (resp., q_2) denote the value assumed by Q_1

(resp., Q_2). For some fixed value of pub, define $\gamma_{\mathsf{pub}} \stackrel{\text{def}}{=} H_\infty(W_0|\mathsf{pub})$. Notice, since $(\mathsf{SS}^*, \mathsf{Rec}^*)$ is an (m, m', t)-secure sketch, we have $\mathrm{Exp}_{\mathsf{pub}}[2^{-\gamma_{\mathsf{pub}}}] \leq 2^{-m'}$. Now, define $\gamma'_{\mathsf{pub},q_1} \stackrel{\text{def}}{=} H_\infty(W_0 \mid \mathsf{pub}, q_1)$, and let us call the value q_1 "bad" if $\gamma'_{\mathsf{pub},q_1} \leq \gamma_{\mathsf{pub}} - 1$. We consider two cases: If $2^{\gamma_{\mathsf{pub}}} \leq 2q_H$ we will not have any guarantees, but using Markov's inequality we have $\Pr[2^{\gamma_{\mathsf{pub}}} \leq 2q_H] = \Pr[2^{-\gamma_{\mathsf{pub}}} \geq 2^{-m'} \cdot (2^{m'}/2q_H)] \leq 2q_H \cdot 2^{-m'}$. Otherwise, if $2^{\gamma_{\mathsf{pub}}} > 2q_H$, we observe that the type 1 queries of $\mathcal{A}$ may be viewed as guesses of w_0. In fact, it is easy to see that we only improve the success probability of $\mathcal{A}$ if in response to a type 1 query of the form $H(w, \mathsf{pub}^*)$ we simply tell $\mathcal{A}$ whether $w_0 = w$ or not.[2] It is immediate that $\mathcal{A}$ learns the correct value of w_0 with probability at most $q_H \cdot 2^{-\gamma_{\mathsf{pub}}}$. Moreover, when this does *not* happen, $\mathcal{A}$ has eliminated at most $q_H \leq 2^{\gamma_{\mathsf{pub}}}/2$ (out of at least $2^{\gamma_{\mathsf{pub}}}$) possibilities for w_0, which means that $\gamma'_{\mathsf{pub},q_1} \geq \gamma_{\mathsf{pub}} - 1$, or in other words that q_1 is "good". Therefore, the probability that q_1 is "bad" in this second case is at most $q_H \cdot 2^{-\gamma_{\mathsf{pub}}}$.

Combining the above two arguments, we see that

$$\begin{aligned}\mathrm{Exp}_{\mathsf{pub}}[\Pr[q_1 \text{ bad}]] &\leq \Pr_{\mathsf{pub}}[2^{\gamma_{\mathsf{pub}}} \leq 2q_H] + \mathrm{Exp}_{\mathsf{pub}}[q_H \cdot 2^{-\gamma_{\mathsf{pub}}}] \\ &\leq 2q_H \cdot 2^{-m'} + q_H \cdot 2^{-m'} \;=\; 3q_H \cdot 2^{-m'}. \end{aligned} \tag{1}$$

Next, define $\gamma''_{\mathsf{pub},q_1} \stackrel{\text{def}}{=} \min_i(H_\infty(W'_i \mid \mathsf{pub}, q_1))$. Since $\{W_0, W_1, \ldots\}$ is a t-bounded distortion ensemble we have $d(W_0, W_i) \leq t$. Furthermore, since $(\mathsf{SS}^*, \mathsf{Rec}^*)$ is well-formed, $\{W_i, W'_i\}$ is also a t-bounded distortion ensemble[3] regardless of pub^*_i, which means $d(W_i, W'_i) \leq t$. Applying Lemma 2 on $\{W_0, W_i\}$ (noticing that pub contains pub^*), followed by Lemma 1 on $\{W_i, W'_i\}$, we have

$$\gamma''_{\mathsf{pub},q_1} \;\geq\; \min_i(H_\infty(W_i \mid \mathsf{pub}, q_1)) - \log \mathsf{Vol}_t^{\mathcal{M}} \;\geq\; \gamma'_{\mathsf{pub},q_1} - \log \mathsf{Vol}_t^{\mathcal{M}}. \tag{2}$$

We now consider the type 2 queries made by $\mathcal{A}$. Clearly, the answers to these queries do not affect the conditional min-entropies of W'_i (since these queries do not include pub^*), so the best probability for the attacker to predict any of the W'_i is still given by $2^{-\gamma''_{\mathsf{pub},q_1}}$, for fixed pub and q_1. Assume for a moment that there are no collisions in the outputs of any of the adversary's random oracle queries, and consider the adversary's i^{th} query $\langle \mathsf{pub}^*_i, h_i \rangle$ to the challenger. The probability that this query is "successful" is at most the probability that $\mathcal{A}$ asked a type 2 query of the form $H(w'_i, \cdot)$ for the correct w'_i plus the probability that such a query was not asked, yet $\mathcal{A}$ nevertheless managed to predict the value $H(w'_i, \mathsf{pub}^*_i)$. Clearly, the second case happens with probability at most 2^{-k}. As for the first case, for any h_i there is at most one w for which $H(w, \cdot) = h_i$, since, by assumption, there are no collisions in these type 2 queries. Thus, the adversary succeeds on its i^{th} query if this w is equal to the correct value w'_i. By what we just argued, the probability that this occurs is at most $2^{-\gamma''_{\mathsf{pub},q_1}}$, irrespective of pub^*_i. Therefore, assuming no collisions in type 2 queries, the success probability of $\mathcal{A}$ in any one of its n parallel queries is at most $n \cdot (2^{-\gamma''_{\mathsf{pub},q_1}} + 2^{-k})$. Furthermore, by the birthday bound the probability of a collision is at most $q_H^2/2^k$. Therefore, conditioned on pub and q_1 and for the corresponding value of $\gamma''_{\mathsf{pub},q_1}$, we find that $\Pr[\mathsf{Succ} \mid \mathsf{pub}, q_1] \leq n \cdot 2^{-\gamma''_{\mathsf{pub},q_1}} + (q_H^2 + n) \cdot 2^{-k}$.

[2] This has no effect when $H(w, \mathsf{pub}^*) \neq h$ as then $\mathcal{A}$ learns anyway that $w \neq w_0$. The modification has a small (but positive) effect on the success probability of $\mathcal{A}$ when $H(w, \mathsf{pub}^*) = h$ since this fact by itself does not definitively guarantee that $w = w_0$.

[3] This ignores the case when $W'_i = \perp$; see the definition of $H_\infty(W'_i)$ given earlier.

To conclude, the adversary's overall probability of success is thus bounded by the expectation, over pub and q_1, of this previous quantity; that is:

$$\begin{aligned}\Pr[\mathsf{Succ}] &= \mathrm{Exp}_{\mathsf{pub},q_1}\left[\Pr[\mathsf{Succ} \mid \mathsf{pub}, q_1]\right] \\ &\leq (q_H^2+n)\cdot 2^{-k} \\ &\quad + \mathrm{Exp}_{\mathsf{pub}}\left[\Pr_{q_1\leftarrow\mathcal{Q}_1}[q_1 \text{ bad} \mid \mathsf{pub}] + \sum_{q_1 \text{ good}} n\cdot 2^{-\gamma''_{\mathsf{pub},q_1}}\cdot \Pr[Q_1 = q_1 \mid \mathsf{pub}]\right].\end{aligned}$$

Using Equation 2, we see that $2^{-\gamma''_{\mathsf{pub},q_1}} \leq \mathsf{Vol}_t^{\mathcal{M}}\cdot 2^{-\gamma'_{\mathsf{pub},q_1}}$. Moreover, for good q_1 we have $\gamma'_{\mathsf{pub},q_1} \geq \gamma_{\mathsf{pub}} - 1$, which means that $2^{-\gamma''_{\mathsf{pub},q_1}} \leq 2\cdot\mathsf{Vol}_t^{\mathcal{M}}\cdot 2^{-\gamma_{\mathsf{pub}}}$. Finally, using Equation 1, we have $\mathrm{Exp}_{\mathsf{pub}}[\Pr[q_1 \text{ bad} \mid \mathsf{pub}]] \leq 3q_H\cdot 2^{-m'}$. Combining all these, we successively derive:

$$\begin{aligned}\Pr[\mathsf{Succ}] &\leq (q_H^2+n)\cdot 2^{-k} + 3q_H\cdot 2^{-m'} \\ &\quad + \mathrm{Exp}_{\mathsf{pub}}\left[2n\cdot\mathsf{Vol}_t^{\mathcal{M}}\cdot 2^{-\gamma_{\mathsf{pub}}}\cdot \Pr_{q_1\leftarrow\mathcal{Q}_1}[q_1 \text{ good}]\right] \\ &\leq (q_H^2+n)\cdot 2^{-k} + 3q_H\cdot 2^{-m'} + 2n\cdot\mathsf{Vol}_t^{\mathcal{M}}\cdot\mathrm{Exp}_{\mathsf{pub}}\left[2^{-\gamma_{\mathsf{pub}}}\right] \\ &\leq (q_H^2+n)\cdot 2^{-k} + (3q_H + 2n\cdot\mathsf{Vol}_t^{\mathcal{M}})\cdot 2^{-m'} \;=\; \delta.\end{aligned}$$

As for the claimed value of m'', let γ_{pub} be as before. Again, we assume that for each type-1 query of $\mathcal{A}$ we simply tell $\mathcal{A}$ whether its "guess" for w_0 was correct or not. (Note that type-2 queries are no longer relevant.) Arguing as before, we have:

$$\begin{aligned}\mathrm{Exp}_{\mathsf{pub},q_1}\left[2^{-\gamma'_{\mathsf{pub},q_1}}\right] &\leq \mathrm{Exp}_{\mathsf{pub}}\left[\Pr[q_1 \text{ bad}] + 2\cdot 2^{-\gamma_{\mathsf{pub}}}\right] \\ &\leq 3q_H\cdot 2^{-m'} + 2^{-m'+1} \;=\; (3q_H+2)\cdot 2^{-m'},\end{aligned}$$

as desired. ■

We remark that the above proof uses only a non-programmable random oracle.

The bound on δ that we derive in the above proof has an intuitive interpretation. The sub-expression $\left(q_H + n\cdot\mathsf{Vol}_t^{\mathcal{M}}\right)\cdot 2^{-m'}$ that appears (up to constant factors due to the analysis) can be viewed as the probability that the adversary "gets information" about the point w_0. The contribution $q_H\cdot 2^{-m'}$ is due to the type 1 oracle queries where, for each of at most q_H queries, the adversary "hits" the correct value of w_0 with probability $2^{-m'}$. Then, each of the adversary's n challenges cover no more than $\mathsf{Vol}_t^{\mathcal{M}}$ candidates for w_0, since each such query eliminates at most one value for w'_i (unless collisions in type 2 queries occur), which in turn eliminates up to $\mathsf{Vol}_t^{\mathcal{M}}$ candidates for w_i, each of which can only eliminate one candidate $\mathsf{Rec}(w_i, \mathsf{pub}^*)$ for w_0. Besides the above, the other contributions to δ are due to the probability of collisions in the random oracle, plus a small term to account for the possibility that the adversary can guess the output of the random oracle at an unqueried point.

In practice, k will be large enough so that $\max(q_H, n\cdot\mathsf{Vol}_t^{\mathcal{M}})$ is the dominant factor determining the amount of the additional "loss" incurred as compared to regular "non-robust" sketches.

3.2 From Robust Sketches to Robust Fuzzy Extractors

We now define the notion of a robust fuzzy extractor:

Definition 6 Given algorithms $(\mathsf{Ext}, \mathsf{Rec})$ and random variables $\mathcal{W} = \{W_0,\, W_1,\, \ldots,\, W_n\}$ over a metric space $(\mathcal{M}, d)$, consider the following game between an adversary $\mathcal{A}$ and a challenger: Let w_0 (resp., w_i) be the value assumed by W_0 (resp., W_i). The challenger computes $(R, \mathsf{pub}) \leftarrow \mathsf{Ext}(w_0)$ and gives (R, pub) to $\mathcal{A}$. Next, $\mathcal{A}$ outputs $(\mathsf{pub}_1, \ldots, \mathsf{pub}_n)$ with $\mathsf{pub}_i \neq \mathsf{pub}$ for all i. If there exists an i with $\mathsf{Rec}(w_i, \mathsf{pub}_i) \neq \perp$ we say the adversary *succeeds* and this event is denoted by Succ.

We say $(\mathsf{Ext}, \mathsf{Rec})$ is an $(m, \ell, t, \varepsilon, n, \delta)$*-robust fuzzy extractor* over $(\mathcal{M}, d)$ if it is an $(m, \ell, t, \varepsilon)$-fuzzy extractor, and for all t-bounded distortion ensembles $\mathcal{W}$ with $H_\infty(W_0) \geq m$ and all adversaries $\mathcal{A}$ we have $\Pr[\mathsf{Succ}] \leq \delta$. $\diamondsuit$

Remark: The proceedings version of this work considered a weaker definition in which $\mathcal{A}$ was not given R; however, that definition is not the "right" one to use for our intended application to key exchange. (The current definition also differs from the one given in the proceedings version in that, as in the case of robust sketches, we only condition on pub when requiring that R be statistically indistinguishable from uniform, rather than conditioning on the adversary's entire view. See the remark following Definition 5.)

An easy transformation from any robust sketch to a robust fuzzy extractor is to simply apply an independent random oracle G to the recovered value w. (A proof is omitted, but follows ideas similar to those used in the proof of Theorem 1.) This is essentially the idea used in [9, Lemma 3.1], but using a random oracle instead of pairwise-independent hashing. We remark that naïve use of pairwise-independent hashing as in [9, Lemma 3.1] will *not* work (in general) for at least two reasons: (1) we need to also take into account adversarial modification of the hash function included as part of pub, and (2) we need to take into account the additional entropy loss due to the fact that $\mathcal{A}$ is given the extracted value R. Both these problems essentially "go away" in the random oracle model.

3.3 Application to Secure Authentication

The application of a robust fuzzy extractor to achieve mutual authentication or authenticated key exchange over an insecure channel is immediate. For concreteness, let Π be a protocol that achieves key exchange with explicit (mutual) authentication based on uniformly-distributed symmetric keys of length ℓ (we are assuming definitions for 2-party key exchange along the lines of [3, 1]). Now, given any $(m, \ell, t, \varepsilon, n, \delta)$-robust fuzzy extractor $(\mathsf{Ext}, \mathsf{Rec})$ and any source W_0 with $H_\infty(W_0) \geq m$, consider the protocol Π' constructed as follows:

Initialization. The user samples w_0 according to W_0 (i.e., scans his biometric data) and computes $(R, \mathsf{pub}) \leftarrow \mathsf{Ext}(w_0)$. The user registers (R, pub) at the server.

Protocol execution. The i^{th} time the user wants to run the protocol, the user will sample w_i according to some distribution W_i (i.e., the user re-scans his biometric data). The server sends pub to the user, who then computes $\hat{R} = \mathsf{Ext}(w_i, \mathsf{pub})$. If $\hat{R} = \perp$, the user immediately aborts. Otherwise, the server and user execute protocol Π, with the server and the user respectively using the keys R and $\hat{R}$.

Assume that $\mathcal{W} = \{W_0, W_1, \ldots\}$ is a t-bounded distortion ensemble. Correctness of the above protocol is easily seen to hold: if the user obtains the correct value of pub from the server then, because $d(w_0, w_i) \leq t$, the user will recover $\hat{R} = R$ and thus both user and server will end up using the same key R in the underlying protocol Π.

The security of Π' with respect to an active adversary who may control all messages sent between the user and the server follows from the following observations:

- If the adversary forwards $\mathsf{pub}' \neq \mathsf{pub}$ to at most n different user-instances, these instances will all abort immediately (without running Π) except with probability at most δ. We stress that here we crucially rely on the fact that *the adversary in the game of Definition 6 is given* R, for the following subtle reason: executions of the adversary with server-instances, as well as with user-instances when the adversary forwards the correct value of pub, may reveal information about R (at least in an information-theoretic sense) since R is then used in an execution of Π.

- Assume that all user-instances to which the adversary forwards $\mathsf{pub}' \neq \mathsf{pub}$ are aborted immediately (i.e., without actually running $\mathsf{Rec}(w_i, \mathsf{pub}')$). By what we have said above, this can affect the adversary's overall advantage in attacking Π' by at most δ. Furthermore, in this hybrid game the adversary learns no information about w_0 other than what is revealed by pub.

  The remaining instances (i.e., server-instances, or user-instances to which the adversary forwards the correct value of pub) are simply running Π using a key R which is within statistical difference ε from a uniformly distributed ℓ-bit key. Security of Π thus implies security of these instances.

In terms of concrete security (informally), let ε_Π denote the maximum success probability of an adversary attacking Π and executing at most n sessions with the user and n' sessions with the server (where, to take a concrete example, success here means that the adversary violates mutual authentication by causing an instance to accept without a matching partner [3]). Assuming an $(m, \ell, t, \varepsilon, n, \delta)$-robust fuzzy extractor is used, the success probability of any adversary attacking Π' (using similar resources) is at most $\delta + \varepsilon + \varepsilon_\Pi$.

4 Improved Solution Tailored for Mutual Authentication

As discussed in the introduction, the robust sketches and fuzzy extractors described in the previous section provide a general mechanism for dealing with adversarial modification of the public value pub. In particular, taking any protocol based on the secure sketches or fuzzy extractors of [9] which is secure when the public value is assumed *not* to be tampered with, and plugging in a *robust* sketch or fuzzy extractor, yields a protocol secure against an adversary who may either modify the contents of the server — as in the case where the server itself is malicious — or else modify the value of pub when it is sent to the user.

For specific problems of interest, however, it remains important to explore solutions which might improve upon the general-purpose solution described above. In this section, we show that for the case of mutual authentication and/or authenticated key exchange an improved solution is indeed possible. As compared to the generic solution based on robust fuzzy extractors (cf. Section 3.3), the solution described here has the advantages that: (1) it is provably secure in the standard model; and (2) it can achieve improved bounds on the "effective entropy loss". We provide an overview of our solution now.

Given the proof of Theorem 1, the intuition behind our current solution is actually quite straightforward. As in that proof, let $\mathcal{W} = \{W_0, \ldots\}$ be a sequence of random variables where W_0 represents the initial recorded value of the user's biometric data and W_i denotes the i^{th} scanned value of the biometric data. Given a well-formed secure sketch $(\mathsf{SS}^*, \mathsf{Rec}^*)$ and a value $\mathsf{pub}_i^* \neq \mathsf{pub}^* = \mathsf{SS}^*(W_0)$ chosen by the adversary, let $W_i' \stackrel{\text{def}}{=} \mathsf{Rec}(W_i, \mathsf{pub}_i^*)$ and define the min-entropy of W_i' as in the proof of Theorem 1. At a high level, Theorem 1 follows from the observations that: (1) the average min-entropy of W_i' is "high" for *any* value pub_i^*; and (2) since the adversary succeeds only if it can also output a value $h_i = H(W_i', \mathsf{pub}_i^*)$, where H is a random oracle, the adversary is essentially unable to succeed with probability better than $2^{-H_\infty(W_i')}$ in the i^{th} iteration. Crucial to the proof also is the fact that, except with "small" probability, the value $h = H(W_0, \mathsf{pub}^*)$ does not reduce the entropy of W_0 "very much" (again using the fact that H is a random oracle).

The above suggests that another way to ensure that the adversary does not succeed with probability better than $2^{-H_\infty(W_i')}$ in any given iteration would be to have the user run an "equality test" using its recovered value W_i'. If this equality test is "secure" (in some appropriate sense we have not yet defined) then the adversary will effectively be reduced to simply guessing the value of W_i', and hence its success probability in that iteration will be as claimed. Since we have already noted that the average min-entropy of W_i' is "high" when any well-formed secure sketch is used (regardless of the value pub_i^* chosen by the adversary), this will be sufficient to ensure security of the protocol overall.

Thinking about what notion of security this "equality test" should satisfy, one realizes that it must be secure for arbitrary distributions on the user's secret value, and not just uniform ones. Also, the protocol must ensure that each interaction by the adversary corresponds to a guess of (at most) one possible value for W_i'. Finally, since the protocol is meant to be run over an insecure network, it must be "non-malleable" in some sense so that the adversary cannot execute a man-in-the-middle attack when the user and server are both executing the protocol. Finally, the adversary should not gain any information about the user's true secret W_0 (at least in a computational sense) after passively eavesdropping on multiple executions of the protocol. With the problem laid out in this way, it becomes clear that one possibility is to use a password-only authenticated key exchange (PAK) protocol [4, 1, 6] as the underlying "equality test".

Although the above intuition is appealing, we remark that a number of subtleties arise when trying to apply this idea to obtain a provably secure solution. In particular, we will require the PAK protocol to satisfy a slightly stronger definition of security than that usually considered for PAK (cf. [1, 6, 12]); informally, the PAK protocol should remain "secure" even when: (1) the adversary can dynamically add clients to the system, with (unique) identities chosen by the adversary; (2) the adversary can specify *non-uniform* and *dependent* password distributions for these clients; and (3) the adversary can specify such distributions *adaptively* at the time the client is added to the system. Luckily, it is not difficult to verify that at least some existing protocols (e.g., [1, 17, 18, 11, 16]) satisfy a definition of this sort.[4] (Interestingly, the recent definition of [7] seems to imply the above properties.) Due to lack of space, the formal definition of security required for our application is deferred to the full version.

4.1 A Direct Construction

With the above in mind, we now describe our construction. Let Π be a PAK protocol and let $(\mathsf{SS}, \mathsf{Rec})$ be a well-formed secure sketch. Construct a modified protocol Π' as follows:

[4]In fact, it is already stated explicitly in [17, 11] that the given protocols remain secure even under conditions 1 and 2, and it is not hard to see that they remain secure under condition 3 as well.

Initialization. User U samples w_0 according to W_0 (i.e., takes a scan of his biometric data) and computes $\mathsf{pub} \leftarrow \mathsf{SS}(w_0)$. The user registers (w_0, pub) at the server S.

Protocol execution (server). The server sends pub to the user. It then executes protocol Π using the following parameters: it sets its own "identity" (within Π) to be $S\|\mathsf{pub}$, its "partner identity" to be $\mathsf{pid} = U\|\mathsf{pub}$, and the "password" to be w_0.

Protocol execution (user). The i^{th} time the user executes the protocol, the user first samples w_i according to distribution W_i (i.e., the user re-scans his biometric data). The user also obtains a value pub' in the initial message it receives, and computes $w' = \mathsf{Rec}(w_i, \mathsf{pub}')$. If $w' = \perp$ then the user simply aborts. Otherwise, the user executes protocol Π, setting its own "identity" to $U\|\mathsf{pub}'$, its "partner identity" to $S\|\mathsf{pub}'$, and using the "password" w'.

It is easy to see that correctness holds, since if the user and the server interact without any interference from the adversary then: (1) the identity used by the server is equal to the partner ID of the user; (2) the identity of the user is the same as the partner ID of the server; and (3) the passwords w_0 and w' are identical. Before discussing the security of this protocol, we need to introduce a slight restriction of the notion of a t-bounded distortion ensemble in which the various random variables in the ensemble are (efficiently) computable:

Definition 7 Let $(\mathcal{M}, d)$ be a metric space. An *explicitly computable t-bounded distortion ensemble* is a sequence of boolean circuits $\mathcal{W} = \{W_0, \ldots\}$ and a parameter ℓ such that, for all i, the circuit W_i computes a function from $\{0,1\}^\ell$ to $\mathcal{M}$ and, furthermore, for all $r \in \{0,1\}^\ell$ we have $d(W_0(r), W_i(r)) \leq t$. $\diamondsuit$

In our application, $\mathcal{W}$ will be output by a PPT adversary, ensuring both that the ensemble contains only a polynomial number of circuits and that each such circuit is of polynomial size (and hence may be evaluated efficiently). We remark that it is *not* necessary for our proof that it be possible to efficiently verify whether a given $\mathcal{W}$ satisfies the "t-bounded" property or whether the min-entropy of W_0 is as claimed, although the security guarantee stated below only holds if $\mathcal{W}$ does indeed satisfy these properties.[5] With the above in mind, we now state the security achieved by our protocol:

Theorem 2 *Let Π be a secure PAK protocol (with respect to the definition sketched earlier) and let $\mathcal{A}$ be a* PPT *adversary. If $(\mathsf{SS}, \mathsf{Rec})$ is a well-formed (m, m', t)-secure sketch over a metric space $(\mathcal{M}, d)$, and $\mathcal{W} = \{W_0, \ldots\}$ is an explicitly-computable t-bounded distortion ensemble (output adaptively by $\mathcal{A}$) with $H_\infty(W_0) \geq m$, then the success probability of $\mathcal{A}$ in attacking protocol Π' is at most $q_s \cdot 2^{-m''} + \mathsf{negl}(\kappa)$, where q_s represents the number of sessions in which the adversary attempts to impersonate one of the parties, and $m'' = m' - \log \mathsf{Vol}_t^{\mathcal{M}}$.*

Due to space limitations, the proof is deferred to the full version.

Specific instantiations. As noted earlier, a number of PAK protocols satisfying the required definition of security are known. If one is content to work in the random oracle model then the protocol of [1] may be used (note that this still represents an improvement over the solution based on robust fuzzy extractors since the "effective key size" will be larger, as we discuss in the next

[5] As to whether the adversary can be "trusted" to output a $\mathcal{W}$ satisfying these properties, recall that $\mathcal{W}$ anyway is meant to model naturally-occurring errors. Clearly, if a real-world adversary has the ability to, e.g., introduce arbitrarily-large errors then only weaker security guarantees can be expected to hold.

paragraph). To obtain a solution in the standard model which is only slightly less efficient, the PAK protocols of [17, 11, 16] could be used.[6] Note that although these protocols were designed for use with "short" passwords, they can be easily modified to handle "large" passwords without much loss of efficiency; we discuss this further in the full version.

4.2 Comparing Our Two Solutions

It is somewhat difficult to compare the security offered by our two solutions (i.e., the one based on robust fuzzy extractors and the one described in this section) since an exact comparison depends on a number of assumptions and design decisions. As we already observed, the main advantage of the solution described in this section is that it does not rely on random oracles. On the other hand, the solution based on robust fuzzy extractors is simpler and more efficient.

The solution presented in this section does not require any randomness extraction, and it therefore "saves" $2\log\delta^{-1}$ bits of entropy as compared with solutions that apply standard randomness extractors to the recovered biometric data. Since a likely value in practice is $\delta \leq 2^{-64}$, this results in a potential savings of at least 128 bits of entropy. When the entropy of the original biometric data is "large", however, we notice that (1) as mentioned already in the previous section, we may use a random oracle as our randomness extractor and thereby avoid the loss of $2\log\delta^{-1}$ bits of entropy; and (2) our two approaches can be combined, and one can use a PAK protocol with any *robust* sketch. If this is done then additional extraction is not required, and so we again avoid losing $2\log\delta^{-1}$ bits of entropy.

On the other hand, the solution of the present section offers a clear advantage when the entropy of the original biometric data is "small". Although in this case the adversary can succeed by an exhaustive, on-line "dictionary" attack, the security of our second solution implies that this is the *best* an adversary can do. In contrast, our solution based on robust sketches would not be appropriate in this case since the adversary could determine the user's secret biometric data using *off-line* queries to the random oracle (cf. the factor proportional to $q_H \cdot 2^{-m'}$ in Theorem 1).

References

[1] M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated Key Exchange Secure Against Dictionary Attacks. *Adv. in Cryptology — Eurocrypt 2000*, LNCS vol. 1807, Springer-Verlag, pp. 139–155, 2000.

[2] M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. ACM CCS 1993, ACM Press, 1993.

[3] M. Bellare and P. Rogaway. Entity Authentication and Key Distribution. *Adv. in Cryptology — Crypto 1993*, LNCS vol. 773, Springer-Verlag, pp. 232–249, 1993.

[4] S. Bellovin and M. Merritt. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. *IEEE Symposium on Research in Security and Privacy*, IEEE, pp. 72–84, 1992.

[6] Although these protocols require public parameters, such parameters can be "hard coded" into the implementation of the protocol and are fixed for all users of the system; thus, users are not required to remember or store these values. The difference is akin to the difference between PAK protocols in a "hybrid" PKI model (where clients store their server's public key) and PAK protocols (including [17, 11, 16]) in which clients need only remember a short password.

[5] X. Boyen. Reusable Cryptographic Fuzzy Extractors. ACM CCS 2004, ACM Press, pp. 82–91, 2004.

[6] V. Boyko, P. MacKenzie, and S. Patel. Provably-Secure Password-Authenticated Key Exchange Using Diffie-Hellman. *Adv. in Cryptology — Eurocrypt 2000*, LNCS vol. 1807, Springer-Verlag, pp. 156–171, 2000.

[7] R. Canetti, S. Halevi, J. Katz, Y. Lindell, and P. MacKenzie. Universally Composable Password-Based Key Exchange. Eurocrypt 2005 (these proceedings).

[8] G. Davida, Y. Frankel, and B. Matt. On Enabling Secure Applications Through Off-Line Biometric Identification. IEEE Security and Privacy '98.

[9] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *Adv. in Cryptology — Eurocrypt 2004*, LNCS vol. 3027, Springer-Verlag, pp. 523–540, 2004.

[10] N. Frykholm and A. Juels. Error-Tolerant Password Recovery. ACM CCS 2001.

[11] R. Gennaro and Y. Lindell. A Framework for Password-Based Authenticated Key Exchange. *Adv. in Cryptology — Eurocrypt 2003*, LNCS vol. 2656, Springer-Verlag, pp. 524–543, 2003.

[12] O. Goldreich and Y. Lindell. Session-Key Generation Using Human Passwords Only. *Adv. in Cryptology — Crypto 2001*, LNCS vol. 2139, Springer-Verlag, pp. 408–432, 2001.

[13] A. Juels. Fuzzy Commitment. Slides from a presentation at DIMACS, 2004. Available at `http://dimacs.rutgers.edu/Workshops/Practice/slides/juels.ppt`

[14] A. Juels and M. Sudan. A Fuzzy Vault Scheme. IEEE Intl. Symp. on Info. Theory, 2002.

[15] A. Juels and M. Wattenberg. A Fuzzy Commitment Scheme. ACM CCS 1999, ACM Press, 1999.

[16] J. Katz, P. MacKenzie, G. Taban, and V. Gligor. Two-Server Password-Only Authenticated Key Exchange. Manuscript, Jan. 2005.

[17] J. Katz, R. Ostrovsky, and M. Yung. Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords. *Adv. in Cryptology — Eurocrypt 2001*, LNCS vol. 2045, Springer-Verlag, pp. 475–494, 2001.

[18] J. Katz, R. Ostrovsky, and M. Yung. Forward Secrecy in Password-Only Key-Exchange Protocols. *Security in Communication Networks: SCN 2002*, LNCS vol. 2576, Springer-Verlag, pp. 29–44, 2002.

[19] F. Monrose, M. Reiter, and S. Wetzel. Password Hardening Based on Keystroke Dynamics. ACM CCS 1999, ACM Press, 1999.

[20] N. Nisan and A. Ta-Shma. Extracting Randomness: A Survey and New Constructions. J. Computer and System Sciences 58(1): 148–173, 1999.

[21] P. Tuyls and J. Goseling. Capacity and Examples of Template-Protecting Biometric Authentication Systems. Biometric Authentication Workshop, 2004.

[22] E. Verbitskiy, P. Tuyls, D. Denteneer, and J.-P. Linnartz. Reliable Biometric Authentication with Privacy Protection. 24th Benelux Symp. on Info. Theory, 2003.