A Fuzzy Vault Scheme

Ari Juels[1] and Madhu Sudan[2]

[1] RSA Laboratories
Bedford, MA 01730, USA
E-mail: ajuels@rsasecurity.com
[2] MIT Laboratory for Computer Science
200 Technology Square, Cambridge, MA 02139
E-mail: madhu@mit.edu

Abstract. We describe a simple and novel cryptographic construction that we refer to as a *fuzzy vault*. A player Alice may place a secret value κ in a fuzzy vault and "lock" it using a set A of elements from some public universe U. If Bob tries to "unlock" the vault using a set B of similar length, he obtains κ only if B is close to A, i.e., only if A and B overlap substantially. In constrast to previous constructions of this flavor, ours possesses the useful feature of *order invariance*, meaning that the ordering of A and B is immaterial to the functioning of the vault. As we show, our scheme enjoys provable security against a computationally unbounded attacker.

Keywords: biometrics, error-correcting codes, fuzzy commitment, fuzzy vault

1 Introduction

Alice is a movie lover. She is looking to find someone who shares her taste in movies, but does not want to reveal information about her preferences indiscriminately to other people. One approach she might take is to compile a set A of her favorite movies and publish it in a concealed form. For instance, Alice might post to a Web newsgroup a ciphertext C_A representing an encryption of her telephone number tel_A under the set (here, key) A. In this case, if another person, say Bob, comes along with a set B of his own favorites that is identical to A, then he can decrypt C_A and obtain Alice's phone number. If Bob tries to decrypt C_A with a set different than Alice's, he will fail to obtain her telephone number. A drawback to this approach is its exactitude, or lack of error-tolerance. If Bob's interests are very similar to Alice's, e.g., if he likes two or three films that Alice doesn't, then he will not learn tel_A. It seems very likely in this case, though, that Alice would still like Bob to obtain her telephone number, as their tastes are quite similar.

In this paper, we introduce the notion of a *fuzzy vault*. This is a cryptographic construction whereby Alice can *lock* her telephone number tel_A using the set A, yielding a *vault* denoted by V_A. If Bob tries to *unlock* the vault V_A using his own set B, he will succeed provided that B overlaps largely with A. On the

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

other hand, anyone who tries to unlock V_A with a set of favorite movies differing substantially from Alice's will fail, helping to ensure that Alice's set of favorites remains private. Thus, a fuzzy vault may be thought of as a form of error-tolerant encryption operation where keys consist of sets. Our fuzzy vault proposal has two important features that distinguish it over similar, prior work. First, the sets A and B may be arbitrarily ordered, i.e., true sets rather than sequences. Second, in contrast to previous work, we are able to prove information-theoretic security bounds over some natural non-uniform distributions on the set A.

Error-tolerant cryptographic algorithms are useful in many circumstances in which security depends on human factors, and thus exactitude represents a drawback. We offer just a few examples here, all of which might benefit from use of our fuzzy vault scheme:

1. **Privacy-protected matching:** As an extension of our movie lover's example above, we might consider a business scenario. Bisco Corp. is looking to sell routers possessing a set $A = \{a_1, a_2, \ldots, a_k\}$ of specifications. It might publish a fuzzy vault V_A with its identity κ, locked under A. If Disco Corp. is looking for routers with a set B of similar specifications, then it will be able to open the vault. Anyone who tries to unlock the vault with a dissimilar set will not learn κ. (We address this idea in detail later in the paper, and decribe an important security enhancement using on-line throttling mechanisms.)
2. **Personal entropy systems:** Proposed by Ellison *et al.* [10], this is a system that enables users to recover passwords by answering a series of questions. In recognition of the unreliability of human memory, the system permits users to answer some of these questions incorrectly. A serious vulnerability in this system is exposed in [4], who show more broadly that the underlying hardness assumption is weak. Our fuzzy vault scheme offers an alternative implementation that is provably secure in an information-theoretic sense and that may involve use of sets, and not just fixed-order answers.
3. **Biometrics:** Alice authenticates to her server using fingerprint information. Her system administrator wishes to store her fingerprint on the server or, more precisely, a set A of features characterizing her fingerprint. (Such sets are known as biometric *templates*.) If an attacker breaks into the server, however, Alice does not want her template A compromised. An additional complication is that biometric systems are error-prone: When Alice tries to authenticate herself, her fingerprint reader is likely to produce a template A' that is similar to, but not identical to A (with bit errors and random permutation and erasure of elements). Alice might store a PIN locked in a fuzzy vault on a set A of features describing her fingerprint, thereby achieving both error-tolerance and privacy. Note that order-invariance is critical here. It is usually not possible to impose an order effectively on biometric features because of the problem of erasures. For this reason, previous schemes like that of Juels and Wattenberg [15] described below are unlikely to work well for this problem.

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

1.1 Previous work

A somewhat less naïve approach to a fuzzy vault construction than straightforward encryption might be achieved through use of Shamir secret sharing techniques [23]. Alice partitions her secret value κ into shares $s_1, s_2, \ldots, s_n$, and encrypt these shares respectively under each of the elements $a_1, a_2, \ldots, a_n$ in her set A. This would yield a set of ciphertexts $e_1, e_2, \ldots, e_n$. Given use of a (t, n)-secret sharing scheme, Bob would only need to decrypt t shares successfully in order to unlock Alice's secret κ. The problem with this approach is twofold. First, suppose that Bob's set B consists of elements $b_1, b_2, \ldots, b_n$. Because A and B are unordered sets, Bob has no way of knowing which of the ciphertexts e_i to try to decrypt with a given set element b_j. Even if Bob tries all n^2 possible decryption operations, there is a second problem: He still does not know which decryptions were successful. Straightforward mechanisms to reveal this information to Bob leak substantial information about A. Indeed, this may be regarded as the source of the weakness in, e.g., the Ellison *et al.* construction. It is also possible for Bob to try to deduce by means of a brute-force search which elements of B do not overlap with those of A. This strategy is inflexible and likely to be prohibitively slow in many practical scenarios, as the computational requirements grow exponentially in the size of $|A \bigcap B|$.

Another idea that does not work well is that of imposing a common ordering on the sets A and B and then using a fuzzy vault construction or similar technique that does not have the property of order invariance, e.g., [15]. This appeoach fails because a small difference between sets can produce large differences in an ordered element-by-element comparison. Suppose, for example, that A and B again represent respective lists of Alice and Bob's favorite movies. If Alice and Bob's favorites include all Oscar winners, except that Alice does not like *Antonia's Line*, then a movie-by-movie comparison of these lists in alphabetical order will yield almost no matches, while in fact A and B overlap considerably. This problem also applies to attempts to impose ordering on features in biometric systems.

To overcome these difficulties, we invoke *error-correcting codes* as the basis for our construction. Given the strong technical and historical affinities between error-correcting codes and cryptographic codes, it is not surprising that error-correcting codes appear in many areas of cryptography, such as quantum cryptography [2, 6], public-key cryptography (via the well known McEliece cryptosystem) [18], identification schemes [26], digital signature schemes [1], and cryptanalytic techniques [13], just to name a few examples. We do not explore this extensive branch of the literature here. We note, however, that Reed-Solomon codes, the most popular form of error-correcting code and the one we focus on here, may be viewed as a general, error-tolerant form of Shamir secret sharing.

The starting point for our fuzzy vault construction is the *fuzzy commitment* scheme of Juels and Wattenberg [15], which is also based on the use of error-correcting codes. This is a cryptographic primitive whereby a user commits to a secret value κ under a key x. The user may decommit using any key x' that is "close" to x under some suitable metric, such as Hamming distance. An at-

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

tacker without any knowledge of x, however, cannot feasibly decommit κ. One application of fuzzy commitment, as suggested by the authors, is to securing biometric systems, as described above. An enrolled fingerprint image (known as a *template*), for example, might be viewed as a key x. The user tries to authenticate using another, slightly different image of the same finger, which we may denote by x'. Authentication is successful if and only if x' is "close" to x.

As the fuzzy commitment scheme in [15] is antecedent to our own, it is worth briefly sketching the details. Let $\mathcal{F}$ be a field, and C be the set of codewords for some error-correcting code; assume that codewords lie in $\mathcal{F}^n$. To commit to a value $x \in \mathcal{F}^n$, the user selects a codeword c uniformly at random from C and computes an offset of the form $\delta = c - x \in \mathcal{F}^n$, i.e., the difference over individual field elements. The commitment then consists of the pair (δ, y), where $y = h(c)$ for some suitable one-way function h. To decommit using key x', the user computes $\delta + x'$ and, if possible, decodes to the nearest codeword c'. The decommitment is successful iff $h(c') = y$.

The construction in [15] has the advantageous features of conceptual simplicity and the ability to make use of any underlying error-correcting code. Moreover, provided that x is drawn uniformly at random from $\mathcal{F}^n$, the scheme enjoys rigorously proveable security linear in the cardinality of C. Suppose that the attacker gains no information about c or x from y, as would be the case under a random oracle assumption on h given sufficiently large security parameters. It is easy to see then that the task of the attacker is to guess c uniformly over C. A similar, less resilient antecedent scheme is proposed in [7, 8], while another system with similar goals but no rigorously provable security characteristics is proposed in [24, 25].

Note that if the hashed value $h(c)$ is removed from the Juels and Wattenberg scheme, i.e., if we no longer think of it as a commitment scheme, then we obtain a kind of fuzzy vault in which the vault itself is equal to δ. If x is uniformly distributed, then the scheme enjoys easily provable information-theoretic security, i.e., security against a computationally unbounded attacker (also proportional to the cardinality of C). Like our own fuzzy vault construction, this one can also be applied to any of the three practical scenarios described above, i.e., privacy-protected matching, personal entropy systems, or biometrics.

As a fuzzy vault variant, though, the scheme of Juels and Wattenberg has two shortcomings. First, while it tolerates some number of errors in the information symbols in x, it does not tolerate substantial re-ordering of these symbols. Given that translation and rotation errors are common in, e.g., biometric systems, it is reasonable to expect that the image x' may consist of a permutation of symbols in x. The property of *order-invariance* is thus likely to be desirable in a fuzzy commitment scheme. A second shortcoming of [15] is the difficulty of proving rigorous results about security over non-uniform distributions. Our proposed scheme addresses these two shortcomings, and may be thought of as an order-invariant version of the Juels-Wattenberg scheme.

The present work has appeared previously in the form of a one-page abstract [**?**].

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

1.2 Our scheme

Like the scheme of Juels and Wattenberg, ours is conceptually simple, and can be implemented using any underlying error-correcting code (although we focus on Reed-Solomon codes in our exposition here). While possessing the advantages of order-invariance and easier analysis on non-uniform distributions, our scheme does have a couple of drawbacks that are important to note from the outset. First, it typically has substantially greater – though still quite practical – memory requirements than the Juels-Wattenberg scheme. Second, it is somewhat less flexible in terms of available parameter choices at a given security level, as we shall see.

Let us briefly sketch the intuition behind our scheme. Suppose that Alice aims to lock a secret κ under set A. She selects a polynomial p in a single variable x such that p encodes κ in some way (e.g., has an embedding of κ in its coefficients). Treating the elements of A as distinct x-coordinate values, she computes evaluations of p on the elements of A. We may think of Alice as projecting the elements of A onto points lying on the polynomial p. Alice then creates a number of random *chaff* points that do not lie on p, i.e., points that constitute random noise. The entire collection of points, both those that lie on p and the chaff points, together constitute a commitment of p (that is, κ). Call this collection of points R. The set A may be viewed as identifying those points in R that lie on p, and thus specifying p (and κ). As random noise, the chaff points have the effect of concealing p from an attacker. They provide the security of the scheme.

Suppose now that Bob wishes to unlock κ by means of a set B. If B overlaps substantially with A, then B identifies many points in R that lie on p, so that Bob is able to recover a set of points that is largely correct, but perhaps contains a small amount of noise. Using error correction, he is able to reconstruct p exactly, and thereby κ. If B does not overlap substantially with A, then it is infeasible for Bob to learn κ, because of the presence of many chaff points. (If B overlaps "somewhat", then he may still be able to recover κ. The gap between feasible recovery and infeasible is fairly small, however, as we discuss below.) We present details and analysis in the body of the paper.

The hardness of our scheme is based on the *polynomial reconstruction* problem, a special case of the Reed-Solomon list decoding problem [4]. Other schemes making use of this problem include, for example, the scheme proposed by Monrose, Reiter, and Wetzel for hardening passwords using keystroke data [19]. An important difference between our scheme and previous ones of this flavor is our range of parameter choices. The [19] scheme bases its security on the computational hardness of small polynomial reconstruction instances, while we select parameters enabling us to achieve information theoretic security guarantees for the same problem.

Organization

We sketch protocol and security definitions for our scheme in section 2. In section 3, we present protocol details for our fuzzy vault scheme. We offer security

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.