

A Fuzzy Vault Scheme

Ari Juels
 RSA Laboratories
 Bedford, MA, USA
 e-mail: ajuels@rsasecurity.com

Madhu Sudan
 MIT LCS
 Cambridge, MA, USA
 e-mail: madhu@mit.edu

Abstract — We describe a simple and novel cryptographic construction that we call a *fuzzy vault*. Alice may place a secret value κ in a fuzzy vault and “lock” it using an unordered set A of elements from some public universe U . If Bob tries to “unlock” the vault using an unordered set B , he obtains κ only if B is close to A , i.e., only if A and B overlap substantially.

I. WHAT IS A FUZZY VAULT?

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. One approach she might take is to compile a set A of her favorite movies and publish a ciphertext C_A representing an encryption of her telephone number tel_A under this set (here, key) A . 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. A drawback to this approach is its 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 may still not learn tel_A . It seems likely in this case, though, that Alice would still like Bob to obtain her telephone number, as their tastes are quite similar.

In this work, 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 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, like that in [2, 3]. 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 .

II. OUR FUZZY VAULT CONSTRUCTION IN BRIEF

Observe that due to the requirement for fuzziness, simple schemes based on secret sharing [5] do not achieve the desired solution here. Another idea that does not work is that of imposing a common ordering on the sets A and B and using a fuzzy vault proposal that lacks order invariance, e.g., [3]. This is because a small difference between sets can produce large differences in an ordered element-by-element comparison. If Alice and Bob’s lists of favorite movies include all Oscar winners, except that Alice does not like *Antonia’s Line*, a movie-by-movie comparison in alphabetical order will yield almost no matches, while in fact A and B overlap considerably.

This said, let us briefly sketch the intuition behind our fuzzy vault 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. (These notions are made more precise in the full paper.)

The hardness of our scheme is based on the *polynomial reconstruction* problem, a special case of the Reed-Solomon list decoding problem [1] employed in some constructions, such as [4]. An important difference between our scheme and previous ones is our achievable range of parameter choices. While previous schemes have relied on computational hardness assumptions, we are able to select parameters providing information-theoretic security guarantees for the same problems.

The full paper, with definitions, theorems, and proofs, is available at www.ari-juels.com or theory.lcs.mit.edu/~madhu.

ACKNOWLEDGMENTS

Thanks to Daniel Bleichenbacher, Markus Jakobsson, and Burt Kaliski for their helpful comments.

REFERENCES

- [1] D. Bleichenbacher and P. Nuyuen. Noisy polynomial interpolation and noisy chinese remaindering. In B. Preneel, editor, *Eurocrypt '00*, pages 53–69, 2000.
- [2] G.I. Davida, Y. Frankel, and B.J. Matt. On enabling secure applications through off-line biometric identification. In *IEEE Symposium on Privacy and Security*, 1998.
- [3] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In G. Tsudik, editor, *Sixth ACM Conference on Computer and Communications Security*, pages 28–36. ACM Press, 1999.
- [4] F. Monrose, M. K. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. In G. Tsudik, editor, *Sixth ACM CCS*, pages 73–82. ACM Press, 1999.
- [5] A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.