Authentication, Enhanced Security and Error Correcting Codes (Extended Abstract)

Yonatan Aumann[1] and Michael O. Rabin[2]

[1] Department of Mathematics and Computer Science, Bar Ilan University, Ramat-Gan, Israel, aumann@cs.biu.ac.il
[2] DEAS, Harvard University, Cambridge, MA, and Institute of Computer Science, The Hebrew University, Jerusalem, Israel, rabin@cs.huji.ac.il

Abstract. In electronic communications and in access to systems, the issue of authentication of the Sender S of a message M, as well as of the message itself, is of paramount importance. Recently S. Goldwasser has raised the additional issue of *Deniable Authentication* where the sender S authenticates the message M to the Receiver's (R) satisfaction, but can later deny his authorship of M even to an Inquisitor INQ who has listened to the exchange between S and R and who gains access to *all* of the the secret information used by S and R. We present two practical schemes for Deniable Authentication of messages M of arbitrary length n. In both schemes the Receiver R is assured with probability greater than $1 - 2^{-k}$, where k is a chosen security parameter, that M originated with the Sender S. Deniability is absolute in the information theoretic sense. The first scheme requires $2.4kn$ XOR operations on bits and one public key encoding and decoding of a short message. The second scheme requires the same number of XOR operations and k multiplications mod N, where N is some fixed product of two large primes. A key new feature of our method is the use of a Shannon-style error correction code. Traditional authentication for a long message M starts by hashing M down to a standard word-size. We *expand* M through error correction. The first Deniable Authentication method is *provably* valid for any encryption scheme with minimal security properties, i.e. this method is generic. The second Deniable Authentication method is provably valid under the usual assumption that factorization is intractable.

Background and New Results

The question of authentication of transmitted messages is of paramount importance. When a Sender S communicates with a receiver R and sends him a message M, it does not suffice for R to authenticate (identify) S in order to know that M has actually originated with S. An Adversary AD can actively tap the line between S and R, and after R has authenticated the sender S, AD can block the Sender's transmission and inject his own message $\tilde{M}$ to R.

There is also an obvious need for *Deniable Authentication (DA)*. In electronic voting schemes DA is a tool for providing freedom from coercion. In negotiations

over the Internet it may be desirable for S to be able to make price offers M to R in a manner that prevents R from showing the offer to another party in order to elicit a better offer. Namely, R cannot prove to the third party that S has made the offer contained in M. It should be noticed that the manner in which the Internet and Electronic Commerce are evolving, call for the widespread use of public-key signatures and for public-key based schemes for establishing shared secret keys.

The usual approach to creating Message Authentication Code (MAC) assumes that S and R share a secret key K. The message M is hashed down to a fixed block size b by use of a hash function $H(K, M)$ which folds the key K into the hashing process. The Sender S then sends $(M, H(K, M))$ to R who verifies the tag $H(K, M)$. Alternatively, S digitally signs $H(M)$, where H is a known hash function, using a public key signature $Sgn_S(H(M))$, and R verifies $Sgn_S(H(M))$.

There are a number of difficulties associated with this approach. To be efficient we need fast hash functions H and fast digital signatures. When it comes to the construction of MAC schemes that are *provably* secure (based on an assumption such as intractability of factoring), one has to use particularly compute intensive hash functions such as the beautiful scheme proposed in [8, 3].

As to deniability of authorship of M, it is obvious that a scheme using digital signatures in a straightforward manner has in consequence also strict *undeniability*, which is the purpose of digital signatures.

As mentioned in the abstract, our schemes are highly efficient, are provably secure, and provide information theoretic deniability. We shall outline our solutions after discussing previous work and background.

Previous Work. Because of the significant practical importance of Message Authentication, there is a very extensive literature on MACs. This literature deals with theoretical as well as with practical issues of authentication. For long messages, hashing down to a short message is the first step. In the papers that aim at creating MACs for actual systems use, there is strong emphasis on rate, i.e. speed, of the hashing process. Let us mention here as representative important examples the papers by Wegman and Carter [14], Bellare, Canetti and Krawczyk [1], Halevi and Krawczyk [9], and Krawczyk [10]. The papers, as well as for example Schneier's book [12], contain a wealth of references to the literature on authentication. The present practical MACs do not require interaction. The message M, with some authenticating tag, is sent by the Sender to the Receiver who verifies the tag. The Deniable Authentication schemes presented here do require, after transmission of the message, a small number of additional message rounds. The additional messages are of size at most $O(k \log n)$, where M is the the length of the message to be authenticated, and k is the security parameter. On the other hand, these schemes do not require pre-shared secret keys for S and R. In this setting interaction seems to be necessary for Deniable Authentication. We feel that the cost of interaction is not onerous.

Canetti et. al. [2] solve a problem closely related to the Deniable Authentication problem, namely the problem of deniable encryption, in a model where

the Inquisitor INQ listens to the transmission between S and and R. In their model the Sender is identified in the sense that the eavesdropper knows that he is listening to a conversation between S and R. The only issue for him is to be able to prove what the contents of that conversation was. The sender S sends an encrypted message $E(M) = C$ to R, where E is the a probabilistic encryption function. INQ, who knows C, can then go to S and/or R and interrogate them as to the value of M. [2] provide deniable encryption in the sense that S or R can produce any other message $\bar{M}$ so that $C = E(\bar{M})$. If one assumes a secret one-time pad of length $|M| = n$ which is shared by R and S, then the problem is trivial. The challenging problem arises in a setting where only public keys and the corresponding private keys held by the participants are used. The [2] solution provides only polynomially secure deniability and the Inquisitor INQ is limited to polynomial computing power. If INQ can compel every participant in the protocol to reveal their private keys then deniability collapses. The protocol is compute intensive.

In a new paper [4], Dwork et al address the deniable authentication of messages as an application of concurrent zero knowledge proofs. They require a timing constraint that they call an (α, β)-assumption on the response time of processes. Their solutions directly apply to messages M shorter than the public keys used and are compute intensive.

New Results. Coming to our solutions, we assume a model in which the Sender S and the Receiver R are connected by an insecure link. The adversaries in the schemes we construct include an Impostor who tries to impersonate S and send to R a message $\bar{M}$ appearing to originate from S. The Impostor can also be a Person In the Middle (PIM), sitting on the link between S and R, intercepting the traffic between them and injecting messages of his own. In essence, the PIM can employ the Sender S as an oracle in his attempt to fool R. Thus general chosen message attacks should also be protected against.

When discussing deniability of authentication, we assume that the communication between S and R is such that listening to the transmission does not identify S. For example, S may use a notebook computer and a modem at a public telephone. We allow an Inquisitor INQ who listens on the line to the exchange between S and R. INQ later comes to S and R and compels them to reveal all the secret data, such as encryption/signature keys, used in the protocol. Even so, INQ cannot prove that the message M was authored by S. It follows that the Receiver R himself cannot prove after the fact to a third party that M was authored by S. Also, the INQ cannot impersonate R to S and elicit from S an authenticated message M to R. This seems to be impossible if INQ has the capabilities of a Person In the Middle , but our schemes do have this property as well.

The central tool in our schemes is the use of an error correction code C. Let us assume messages M comprising n bits. We assume that $C(M) = y_1, y_2, \ldots, y_m$ has the property that if $M \neq \bar{M}$ then the Hamming distance between $C(M)$ and $C(\bar{M})$ is greater than $m/4$, i.e. $C(M)$ and $C(\bar{M})$ differ at more than $m/4$ indices. For our purposes we choose a code C which is very efficient to encode.

We never have a need to decode $C(M)$. Also, in our application S and R need to compute only a fixed number $2.4k$ of (randomly chosen) bits of $C(M)$.

For our first Deniable Authentication scheme we assume a public key encryption function E_S for S (who, of course, possesses the corresponding secret decryption function D_S). The Sender S sends M to R. They then create a random sequence $Y = i_1, \ldots, i_k$ of k different indices between 1 and m. The bits of $C(M)$ at these indexes are computed by S and by R. Sender S then deniably authenticates these bits as well as Y to R. Thus Deniable Authentication of the long message M is reduced to Deniable Authentication of a short message.

For our second Deniable Authentication scheme we assume a publicly available Directory containing certain public keys for each potential Sender. The sender S wants to transmit messages $M = x_1 x_2 \ldots x_n$, where each x_i is a bit.

We again employ the error correction code C which codes M into $C(M) = y_1 y_2 \ldots y_m$, where $m = cn$ (say $m = 5n$) and the Hamming distance between any two code words Y_1 and Y_2 is αm. With $m = 5n$ we ensure $\alpha > 1/4$. The code C is publicly known and is used by every Sender.

The public Directory contains C and a number $N = p \cdot q$ chosen as a product of two large primes, and where the factorization of N is not known to R (and possibly not to S either). Every potential sender S randomly chooses $a_0, a_1, g_0, \ldots, g_m$ in Z_N^*, computes their squares mod N, and publishes those squares $A_0, A_1, G_0, \cdot, G_m$, in the Directory. In the full paper we give a version of our protocol that allows to reduce the size of each Sender's Directory entry from $m + 2$ to $\log_2 m + 2$.

The Sender S sends M to R. To authenticate M as having originated with S, the Receiver R randomly chooses $L = d \cdot k$ (where $d > 1$ depends only on c, i.e. on the code C; for $c = 5$ we have $d = 2.4$) indices $i_1, \ldots, i_L$ between 1 and m (the size of the error correcting coded message $C(M) = y_1 y_2 \ldots y_m$). He then computes $y_{i_1}, \ldots, y_{i_L}$. For the code C that we use, each such computation of a y_{i_j} requires just n XOR operations on bits regardless of c.

The Receiver R then conducts an L-round interaction with S. Roughly speaking, in round j the Receiver R verifies that y_{i_j} is the i_j-th bit in the code word of a message that S has actually sent him. The precise details and the proof of authentication are given in the full paper. Each round requires four multiplications mod N by the sender and by the receiver. If we want a more compact Directory with just $\log_2 m + 2$ words for each Sender, then the above 4 is replaced by $\log_2 m + 3$. However, precomputation by the Sender and by the Receiver (in case R will receive many authenticated messages from S), will again reduce the number of multiplications to 4. Note that the total number of multiplications $2.4 \cdot 4k = 9.6k$ for each participant, and is independent of the message length n.

After this interaction, R knows, with probability of being cheated smaller than 2^{-k} that M has originated with S. This is provable on the assumption that factorization of N is intractable.

We then prove that, provided that S does not conduct more than a fixed number of message-authentications simultaneously, our message authentication is deniable in the strong information-theoretic sense explained in the Abstract.

Under any reasonable timing restrictions on concurrency, such as those in [4], we directly achieve deniability in the unbounded concurrency setting.

The intractability of extracting square roots mod N based on the intractability of factoring N, which lies at the heart of our authentication scheme, was first introduced and used in [11]. Square roots mod N are used for user authentication and for digital signatures in [6] and in [5]. Zero Knowledge Proofs of languages involving squares mod N and of knowledge of square roots mod N are discussed in [7] and in [13].

References

1. M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *Proceedings of Crypto 96*, 1996.
2. R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky. Deniable encryption. In *Proceedings of Crypto 97*, 1997.
3. I. Damgard. Collision free hash functions. In *Eurocrypt '87*, pages 203–216, 1987.
4. C. Dwork, M. Naor, and A. Sahari. Concurrent zero knowledge. In *Proceedings of the 30th STOC*, 1998.
5. U. Feige, A. Fiat, and A. Shamir. Zero knowledge proofs of identity. In *Proceedings of the 19th STOC*, 1987.
6. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problem. In *Proceedings of Crypto '86*, pages 186–194, 1997.
7. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18:186–208, 1989.
8. S. Goldwasser, S. Micali, and R. Rivest. A secure digital signature scheme. *SIAM Journal on Computing*, 17(2):281–308, 1988.
9. S. Halevi and H. Krawczyk. Mmh: Message authentication in software in the gbit/second rates. In *Proceedings of the 4th Workshop on Fast Software Encryption*, 1997.
10. H. Krawczyk. Lfsr-based hashing and authentication. In *Proceedings of Crypto '94*, pages 129–139, 1994.
11. M. O. Rabin. Digitized signatures and public key functions as intractible as factorization. MIT Laboratory for Computer Science Technical Report LCS/TR-212, MIT, 1979.
12. B. Schneier. *Applied Cryptography : Protocols, Algorithms, and Source Code in C.* John Wiley and Sons, 1995.
13. M. Tompa and H. Woll. Random self-reducibility and zero-knowledge interactive proofs of possession of information. In *Proceedings 28th STOC*, pages 472–482, 1987.
14. M.N. Wegman and J.L. Carter. New hash functions and their use in authentication and set equality. *JCSS*, 22:265–279, 1981.