CHAPTER 7

A Survey of Information Authentication*

G. J. SIMMONS
Sandia National Laboratories
Albuquerque, New Mexico 87185

*This work was performed at Sandia National Laboratories and supported by the U.S. Department of Energy under contract no. DE-AC04-76DP00789.

Abstract—**In both commercial and private transactions, authentication of information (messages) is of vital concern to all of the participants. For example, the party accepting a check usually insists on corroborating identification of the issuer—authentication of the originator, or as we shall say throughout this chapter, the transmitter—and the party issuing the check not only fills in the face amount in numerals, but also writes out the amount in script, and may even go so far as to emboss that part of the check to make it more difficult for anyone to subsequently alter the face amount appearing on an instrument bearing his valid signature, that is, a primitive means of providing for the later authentication of the communication or message. Although this example illustrates the two main concerns of the participants in the authentication of information, namely, the verification that the communication was originated by the purported transmitter and that it hasn't subsequently been substituted for or altered, it fails to illustrate perhaps the most important feature in the current use of authentication. The information conveyed on the check is inextricably linked to a physical instrument, the check itself, for which there exist legally accepted protocols to establish the authenticity of the signature and the integrity of what the issuer wrote in the event of a later dispute as to whether the check is valid or the signature genuine, independent of the information content (date, amount, etc.) recorded there. The contemporary concern in authentication, though, is with situations in which the exchange involves only information, that is, in which there is no physical instrument that can later be used to corroborate the authenticity of either the transmitter's identity or of the communication.**

In deference to the origins of the problem of authentication in a communications context, we shall refer to the authenticated information as the message* and, as mentioned earlier, to the originator (of a message) as the transmitter. The message, devoid of any meaningful physical embodiment, is presented for authentication by a means that we shall call the authentication channel. This channel is by definition insecure, that is, all communications that pass through it are public and may even be intercepted and replaced or altered before being relayed on to the intended receiver. In the simplest possible authentication scheme the party receiving the message (the receiver) is also the one wishing to verify its authenticity; although, as we shall see, there are circumstances in which this is not the case. Authentication, however, is much broader than this communications-based terminology would suggest. The information to be authenticated may indeed be a message in a communications channel, but it can equally well be data in a computer file or resident software in a computer; it can be quite literally a fingerprint in the application of the authentication channel to the verification of the identity of an individual [14,22] or figuratively a "fingerprint" in the verification of the identity of a physical object such as a document or a tamper-sensing container [11]. In the broadest sense, authentication is concerned with establishing the integrity of information purely on the basis of the internal structure of the information itself, irrespective of the source of that information.

1 INTRODUCTION

Since information authentication often depends on complex cryptographic protocols and algorithms that cause the process of authentication itself to appear complex, it is useful to first discuss the general principles that underlie all authentication schemes before discussing the (unavoidably) complicated real authentication schemes. In the simplest terms possible, *authentication* is nothing more nor less than the determination by the authorized receiver(s), and perhaps the arbiter(s), that a particular message was most probably sent by the authorized transmitter under the existing authentication protocol and that it hasn't subsequently been altered or substituted for. Implicit in this statement is the fact that in any particular authentication protocol the receiver will accept as authentic only a fraction out of the total number of possible messages and that the transmitter will only use some subset (perhaps all) of this fraction to communicate with the authorized receiver(s). It should also be obvious that an opponent should not, in all probability, be able to select a message that the receiver will accept as authentic, otherwise he could impersonate the transmitter and/or substitute fraudulent messages of his choice for legitimate ones. The conditions determining the set of messages the receiver will accept and which of these the transmitter may use are what specifies a particular authentication scheme. As we will see, this commonly involves some form of encryption/decryption operation that the transmitter and receiver can do since they each know a secret cryptographic key, but that outsiders who do not know the key (probably) cannot. The following example, however, illustrates the essential features of message authentication without having to appeal to cryptography.

At the end of the last century and early in this one commercial codes were in widespread use to provide economy in telegraphic charges by encoding common (often

---

*This choice of terminology is not as straightforward as it might seem, and will be fully justified later.

very long) phrases into five-letter groups which were treated, and charged for, by the cable companies as single words. One of the best known of these was the Acme Commodity and Phrase Code [13] which consisted "of one-hundred thousand five-letter code [words] ciphers with at least two-letter difference between each and every [code] word. No transposition of any two adjoining letters will make another [code] word." Since the Acme Code is long since out of use and presumably forgotten, for our example we will assume that the transmitter and receivers have a copy and have agreed that the transmitter will only use and the receiver will only accept five-letter groups appearing in the Acme Code and that an opponent is ignorant of this code. Thus OGHAU would be accepted by the receiver as authentic and interpreted to mean "some are and some are not" while none of the 129 "nearby" five-letter groups AGHAU, . . . , OGHAS, GOHAU . . . OGAUA would be accepted since they do not appear in the code. Incidentally, it is interesting to note (from our vantage point of 1991) that the Acme Code was genuinely a precursor of modern error detecting and correcting codes, since a systematic procedure to correct "mutilations" was given [30]. If a codeword was received that did not appear in the code, it was assumed that this was the result of a mutilation (error) in transmission. The procedure constructed the five codewords that did appear in the code and differed in only one letter from the received codeword, and thus were the most likely codewords to have been sent. Context was then used to select one out of these five codewords, that is, a maximum likelihood detector in current terminology. For the purposes of our example, however, if an opponent knows nothing of the rule used by A. C. Meisenbach, the Acme Code designer, to construct acceptable (read authentic) codewords, but only knows that messages consist of five-letter groups, his probability of "choosing" an acceptable (to the receiver) codeword and hence his probability of deceiving the receiver would be

$$P_d = \frac{10^5}{26^5} \approx 0.0084$$

Although this example illustrates the essential notions involved in the authentication of information, that is, the definition of a restricted set of messages that the receiver will accept (as authentic) and of a subset (not necessarily a proper subset) of these messages that the transmitter will use, commercial codes (or similar fixed rules specifying the set of acceptable messages) have no useful authentication capability since the collection of acceptable messages would be either known to the opponent a priori or else quickly exposed by continued use.

It is at this point that cryptography commonly enters into authentication since it provides an easy to implement way for the transmitter and receiver to define the subset of messages that they will use and accept, respectively, dependent on the secret cryptographic key(s) known only to them that will consequently be opaque to an opponent who does not know the key(s).

For example, in a common U.S. military authentication protocol the transmitter and receiver each have matching sealed authenticators (note the use of the term "authenticator"), actually a short random sequence of symbols produced and distributed by the National Security Agency. They must also each have a common cryptographic keying variable (key) which must be protected to ensure its secrecy and integrity. The sealed authenticator packets are constructed so as to provide a positive indication (tattle-tale) if they are opened. Each communicant is responsible for the protection of

his sealed authenticator and is administratively constrained from opening it until it is used to authenticate a message. Because of the sensitivity of these authenticators, that is, as will be apparent in a moment anyone having access to one could authenticate a fraudulent message, they must normally be handled under two-man control in the same way that the associated cryptographic keying variables must be handled which greatly complicates their generation, distribution, and control, and more importantly often limits the level of authentication that can be achieved in an exposed situation.

To authenticate a message, the transmitter opens his sealed packet, appends the enclosed authentication suffix to the message he wishes to authenticate, and then encrypts the resulting extended message with cipher or text feedback, using the cryptographic key he shares with the intended receiver, so that the effect of the appended authenticator is spread throughout the resulting cipher. In principle, the extended message could be block encrypted, but the size of the blocks involved rules this out in all practical applications. This encryption is done using a secret key (in a single-key cryptographic system) that the transmitter shares with the intended receiver(s). The resulting cipher is then transmitted as the authenticated message. The receiver, on receiving and decrypting the cipher using his copy of the secret key, opens his matching sealed authenticator and accepts the message as genuine if and only if the cipher decrypts to a string of symbols with the proper authenticating suffix, and otherwise rejects it as unauthentic. In this example, the subset of messages (ciphers) that the transmitter will use and that the receiver will accept are precisely those that decrypt to have the authenticating suffix. If the crypto algorithm is secure, an opponent who doesn't know the secret key(s) being used by the transmitter and receiver can do no better than to randomly choose a cipher in the hope that it will be accepted by the receiver. If there are r bits of information in the authenticator, an opponent (if he cannot break the "sealing" encryption algorithm) would have only a 2^{-r} probability of choosing (guessing) a cipher that would decrypt into a message ending with the unknown (to him) authentication suffix, and hence that would be accepted as authentic by the receiver.

This example illustrates an essential feature in all authentication schemes, namely, that authentication depends on the presence of redundant information, either introduced deliberately as in this example, or else inherently present in the structure of the message, that will be recognizable to the receiver. This results in the transmitter and receiver restricting their use to only a fraction of the total number of communications possible, that is, to those messages containing this redundant information; any others would be rejected by the receiver as unauthentic since the transmitter would not have sent them. As used in this example, and in widespread cryptographic use, the term "authenticator" denotes the redundant information appended to the message that is to be authenticated, which is functionally independent of the information content of the message itself, to give an extended message that is then encrypted, etc. as described above. The object actually communicated by the transmitter to the receiver through the communications channel in this case is a "cipher." This use of the term "cipher" is in accordance with the accepted conventions of cryptography, since both the content of the original message and the authenticating redundant information must be concealed (kept secret) in the cipher, otherwise the appended authenticator, if it were revealed, could be used to authenticate an arbitrary fraudulent message.

On the other hand, an equally common use of the term authenticator, with a quite different meaning, occurs in connection with the authentication of electronic funds transfers in the Federal Reserve System. By directive of the Secretary of the Treasury

[3], all such transfers must be authenticated using a procedure that de facto depends on the Data Encryption Standard (DES) single-key cryptographic algorithm. The protocol includes precise format requirements, etc.; however, the essential feature for our purposes is that an authenticator is generated using a DES mode of operation known as cipher block chaining. The information to be authenticated is first broken into blocks of 64 bits each. The first block is added bitwise modulo two (exclusive-or) to a 64-bit initial vector, which is changed daily and kept secret by the member banks, and the sum encrypted using a secret DES key (known to both the transmitter and the receiver). The resulting 64-bit cipher is then exclusive-or'ed with the second block of text and the result encrypted to give a second 64-bit cipher, etc. This procedure is iterated until all blocks of the text have been processed. The final 64-bit cipher is clearly a function of the secret key, the initial vector, and of every bit of the text, irrespective of its length. This cipher, called a message authenticating code (MAC), is appended to the information being authenticated to form an extended message. The resulting extended message itself is normally sent in the clear, that is, unencrypted, although it may be superencrypted if privacy is desired, but this operation is independent of the authentication function. The authenticator (MAC) can be easily verified by anyone in possession of the secret key and initial vector by simply repeating the procedure used by the transmitter to generate it in the first place. An outsider, however, cannot generate an acceptable authenticator to accompany a fraudulent message, nor can he separate an authenticator from a legitimate message to use with an altered or forged message since the probability of it being acceptable in either case is the same as his chance of "guessing" an acceptable authenticator, that is, 1 in 2^{64}. In this application, which is a classic example of an appended authenticator, the authenticator is a complex function of the information that it authenticates. The subset of acceptable extended messages in this case consists of those text-MAC pairs that pass the test of the MAC being related to the text by the secret DES key. Since this makes up only 2^{-64} of all possible extended messages, the probability of an opponent being able to "guess" an acceptable message is less than his chance of "guessing" the secret DES key: 1 in 2^{56}. This simplified discussion of the authentication security provided by a MAC is misleading (and in some instances untrue). The reader should refer to the chapter on Digital Signatures in this volume for a much more complete discussion of the security provided by appended authenticators.

In both of these uses of the term, the term "authenticator" denotes additional information communicated by the transmitter to enable the receiver to satisfy himself that the message should be accepted (as authentic). In the first case, the redundant information was appended to an unrelated message, and could therefore, if directly accessible to an opponent, be stripped off of one message and used to authenticate any other message. To prevent this, the resulting extended message was secured in a (block or feedback) cipher in which each bit of the cipher was a function not only of a secret encryption key but also of all of the bits of the extended message. In the second case, the redundant information was already by virtue of the generating procedure, a function of the secret key and initial vector as well as all of the bits in the information being authenticated, and hence, with high probability, inseparable from the original text in the sense that it has no better chance of being accepted as the authenticator for some other text than would any other randomly chosen 64-bit sequence.

This is a convenient point to comment on the terminology we will use in the balance of this chapter in discussing authentication. The preceding two examples indicate the confusing state of affairs as to the use of the term "authenticator." At least in

both of these cases the term authenticator referred to the redundant information on which authentication depends, but in general even this isn't true.

It might at first seem a natural choice to call the actual sequence of symbols communicated by the transmitter to the receiver the authenticator paralleling the use of the term "cipher" in cryptography. Unfortunately, this leads to an immediate, and irresolvable, logical difficulty. In the Federal Reserve example, the transmitted sequence consisted of the original message concatenated with the 64-bit "authenticator" suffix. If the entire sequence consisting of both the information symbols and appended authenticator were to also be called an authenticator, there would be an unavoidable confusion between this authenticator and the appended authenticator it contains. A similar logical difficulty is clearly inherent in the other scheme as well. Consequently, the widespread use of the term authenticator, although not precise, has already preempted the most natural term to designate the sequence of symbols actually communicated by the transmitter to the receiver.

In the military authentication protocol, the sequence transmitted could be accurately described as a cipher. However, there is the same logical difficulty in using the term cipher to describe the total sequence of symbols communicated in the Federal Reserve protocol as there was in using the term authenticator in the military protocol. The appended authenticator in the Federal Reserve protocol can be properly described as a cipher, since it is produced by a block chaining encryption of the original message. However, if one described the entire sequence consisting of the concatenation of the original sequence of symbols and the appended authenticator (cipher) as a "cipher," there is both the problem of the confusion produced by the two uses of the term cipher, as well as the fact that the information being authenticated is not concealed in the inclusive cipher (extended message), contrary to the commonly accepted meaning of that term. Hence cipher is an even less satisfactory term than authenticator, for similar reasons.

The point of the preceding discussion was to make it clear that there is no completely satisfactory terminology for the primary object in the theory authentication, namely, the sequence of symbols actually transmitted through the authentication channel, and that each of the terms already in use for comparable objects in allied subjects is logically unacceptable. Nor does any constructed new word seem sufficiently natural to be persuasive. We have therefore settled on the cumbersome term "authenticated message," or where no confusion is likely, simply "message," as denoting the transmitted sequence of symbols. This conflicts with the equally natural use of the term "message" to denote the information that is to be authenticated. To avoid this, we tolerate the rather artificial device that the information conveyed by a message is the state of a hypothetical source. This convention has a precedent in error detecting and correcting codes where one speaks of a "message source" and/or "source encoding." The difference is that we are compelled to refer to the information that is being authenticated as a "state of the source," etc. In this convention, for example, the message in an appended authenticator scheme is of the form

$$m_{i,j} = s_i : f(\mathbf{e}_j, s_i)$$

where s_i is the ith state of the source, $\mathbf{e}_j$ is the jth encoding rule, and f is the authenticating function, so that $f(\mathbf{e}_j, s_i)$ is the authenticator that is to be appended to s_i when encoding rule $\mathbf{e}_j$ is being used.

The examples in the preceding paragraphs illustrate very clearly an essential dichotomy in all authentication schemes, namely, the division according to whether authentication is achieved with or without secrecy for the information being authenticated. In many applications secrecy isn't an important consideration, so it doesn't much matter whether the authentication is with or without secrecy. There are, however, applications in which authentication is essential but in which secrecy cannot be tolerated. One such application is the authentication of data taken to verify compliance with a comprehensive nuclear weapons test ban described in the chapter "How to Insure That Data Acquired to Verify Treaty Compliance Are Trustworthy" in this volume. We will return to this classification of authentication schemes according to whether authentication is achieved with or without secrecy later.

There is one final item of terminology that is important to understand for many contemporary applications of authentication. We pointed out above that the sealed authenticators used in the military authentication protocol had to be generated, distributed, and protected in precisely the same way that the cryptographic keying variables had to be. Furthermore, all of the schemes that depend on single-key cryptoalgorithms to define the acceptable sets of (authentic) messages, that is, that depend on the encryption operation to spread the collection of acceptable messages in what appears to an opponent to be a random and uniform manner in the set of all possible messages, requires that both the transmitter and receiver protect their copy of the cryptographic key to the same level of security demanded of the authentication scheme. One of the principal benefits of two-key or public key cryptography is that the decryption key does not have to be kept secret to have confidence in the authenticity of authenticated messages. This is true because it is by definition, or assumption, computationally infeasible to calculate the encryption key given the decryption key. It is, of course, essential that the integrity, against substitution or alteration, of the decryption key be insured during generation, distribution, and storage, at the same level of security demanded of the authentication scheme. If there were a lapse in the protection of the decryption key so that an opponent could substitute a fraudulent decryption key corresponding to an encryption key that he knows for the genuine key, he could then deceive the receiver by undetectably authenticating any fraudulent message he might wish. It might seem that if the integrity must be protected for the two-key cryptoalgorithm based schemes, then why not simply protect the secrecy also and use the much simpler and faster single-key cryptoalgorithms. One answer is that protecting secrecy often involves splitting up the information, that is, two-man control, or using trusted couriers to deliver sealed tattletale containers, etc. or repositories that require more than one combination lock be opened to gain access to the information. If only the integrity of the keying variable must be insured, then the key could be distributed by what is often called the "Merkle channel" (after Ralph Merkle who proposed it), that is, communicated over so many different public channels that it is improbable that any opponent could usurp them all: telephone, telex, radio, television, mail, etc. After the key is received, it only need be protected from substitution or alteration, for example, by keeping it in a regular single combination safe, or even more simply, by several responsible persons keeping a copy, and comparing these copies when needed. The descriptive term that has come to be accepted for this type of authentication in which the information that is used to verify the authenticity of a message doesn't have to be kept secret is "desensitized authentication," since the keying variables are no longer sensitive information.

2 THE THREAT(S)

In the discussion of authentication thus far, we have treated the problem as though it were strictly a matter of protecting the transmitter/receiver (the insiders) against deception by an opponent (the outsider). In other words, the communication takes place between mutually trusting and trustworthy parties over a channel that is assumed to be under the surveillance, and perhaps the control of a knowledgeable and sophisticated opponent who is capable of carrying out complex calculations and then either originating messages of his own devising or else of intercepting and modifying messages from the legitimate transmitter. In military and diplomatic schemes, this is almost always assumed to be the case, or at least the problem of insider cheating is considered to be an acceptable risk. In the commercial world almost the reverse is true, that is, the originator and the receiver (the insiders) to a communication are even more likely to be the ones trying to cheat each other than are outsiders.

It is perhaps useful in understanding insider deceits to think of the receiver as a stockbroker and the transmitter as one of the broker's customers. In this setting it is easy to believe that a customer might wish to disavow an order that he actually issued if it later turns out that the decision was a bad one that cost him money. Similarly, the broker, who is managing the customer's account, might very well wish to execute an order of his own devising when he received no such instructions from the customer, or even to execute orders contrary to the customer's instructions, to generate commissions for himself or in his judgment to make better investments. In either case, the function of authentication would be, in the event of a dispute between the broker and the customer as to whether the broker had faithfully carried out the customer's instructions, to make it possible for an impartial third party to decide who was, in all probability, liable. Although it won't be possible to discuss solutions to all of the possible authentication threats here, several will be illustrated in the applications that are discussed. Because the subject is so convoluted, it is desirable to describe each of the various deceptions as clearly as possible.

In the most general model of authentication there are four essential participants; of these four participants, the "insiders" are the transmitter (the authorized originator for messages), the legitimate receiver(s), and, depending on the particular authentication scheme being used, perhaps the arbiter(s). Whether the arbiter is an insider or an outsider depends on whether he is in possession of any privileged information, that is, information not available to one or more of the other participants. We mention in passing that for the unconditionally secure authentication codes that permit arbitration described here and in [25,26] the arbiter is necessarily an insider, while for the authentication channel that permits arbitration based on public key cryptographic techniques [21], that is, for computationally secure authentication with arbitration, the arbiter is an outsider. The chapter "Digital Signatures" by Mitchell, Piper, and Wild in this volume gives a comprehensive treatment of digital signature schemes in which the arbiter has no privileged information. The fourth participant, the opponent, is always an outsider who is assumed to have no privileged information, but who is assumed to be knowledgeable of the general authentication scheme being used by the transmitter and receiver (an extension of Kerckhoffs' criteria in cryptology to authentication) and to be capable of sophisticated eavesdropping, computation, and message alterations. Given this general setting, there are (at least) eight types of cheating (attempted deceptions) that can occur.

The opponent can send a fraudulent message to the receiver in hopes of having it be accepted as an authentic communication from the transmitter before the legitimate transmitter sends any message at all. This type of attempted deception we will denote by I_0. He can also wait to observe ℓ authentic messages, $\ell \geq 1$, sent by the legitimate transmitter before he attempts to deceive the receiver; after which he can either substitute some other message in the stead of the last message intercepted or else forward it unchanged and then attempt to get some message of his choosing accepted as authentic by the receiver before the legitimate transmitter sends another authentic message. We denote this type of cheating by the notation I_ℓ and S_ℓ, $\ell \geq 1$. The cases I_0 and I_ℓ and S_ℓ, $\ell \geq 1$, are sufficiently different that we will describe them separately.

1. I_0. The opponent, based only on his knowledge of the general authentication scheme being used by the transmitter and receiver, can send a fraudulent message to the receiver when in fact no message has yet been sent by the transmitter. The probability of his succeeding in deceiving the receiver in this case is simply the value of the two-person game whose representation is the incidence matrix of the authenticating rules—mapping source information into messages—in the general authentication scheme [1,23,32]. The calculation of this probability is computationally easy even for large authentication schemes. I_0 is commonly referred to as (the opponent) impersonating the transmitter.

2. I_ℓ and S_ℓ. The opponent can wait to observe $\ell - 1$ legitimate messages from the transmitter which he allows to pass to the receiver without tampering with them. When he intercepts the ℓth message there are two courses of action available to him: He can either substitute some other message of his own devising in its stead or else he can forward it without modification to the receiver. He could then, based on what he has learned from the ℓ observations he has made of legitimate messages, send a message of his own choosing to the receiver before the legitimate transmitter sends the next authentic message, that is, he can attempt to impersonate the legitimate transmitter. The first type of deception is an ℓth order substitution attack, S_ℓ, where S_1 is commonly referred to as simply substitution, while the second type of deception is an ℓth order impersonation denoted by I_ℓ. The opponent's strategy in either of these cases, $\ell \geq 1$, is defined by conditional probabilities, that is, his decision as to which message to substitute or of the message he should use to maximize his chances of successfully impersonating the legitimate transmitter will be affected by the legitimate messages he has observed (and also by whether he knows or doesn't know the information being conveyed by the observed messages [1,20,23,32]). Already for the case of S_1 this problem is computationally difficult (even for modest-sized authentication schemes) unless very stringent conditions are imposed on the regularity and symmetry of the associated designs.

The present author has restricted attention in both his earlier work on authentication codes [20, 23–26] and in this chapter to deceits I_0 and S_1, that is, transmitter impersonation and/or message substitution. The reason for this is that for opponent deceits I_1 and I_ℓ and S_ℓ, $\ell > 1$, an ad hoc rule must be introduced to prevent the opponent from simply substituting a legitimate message already observed prior to the ℓth communication and hence known to be acceptable to the receiver, for the ℓth message—if they are different. Various other authors [4,12,18,23] have considered authentication codes for the cases $\ell > 1$. In all cases, the opponent wins if the receiver accepts the fraudulent message as being an authentic communication from the transmitter *and,* if $\ell > 0$, ends up being misinformed as to the transmitter's communicated information in consequence.

Insider cheating involves a participant who knows some piece of information about the authentication scheme not known to all of the other participants: the transmitter, receiver, or in some instances as noted above, the arbiter(s). Protection against transmitter or receiver cheating presupposes that there is an arbiter who will arbitrate disputes between them, that is, who will assign liability to the party most likely to be responsible. This arbiter, for the scheme described in this chapter to work, must be assumed to be unconditionally trustworthy. In other words, we assume that the arbiter will not misuse his privileged position to deceive either the transmitter or receiver. We will not consider the case of insider-arbiter cheating since the authentication codes described in this chapter provide no protection against this type of deceit. In the most general setting, though, arbiter cheating is a fourth type of deception that needs to be protected against in addition to the three that are considered here. Brickell and Stinson [2] have constructed authentication schemes capable of detecting (in probability) dishonest arbiters.

The receiver can cheat if he can successfully attribute a message of his own devising to the transmitter, that is, a message not sent by the transmitter. The adverb "successfully" means that when the transmitter later claims (correctly) that he didn't send the message in question, that the arbiter will rule against him. The receiver can wait to attribute a fraudulent message to the transmitter until after he has received ℓ, $\ell \geq 0$, legitimate messages. We denote this second form of cheating by the notation R_ℓ, $\ell \geq 0$.

3. R_ℓ. The receiver, using both the public knowledge of the general authentication scheme and his privileged information, claims to have received from the transmitter in an authentic message fraudulent information of his own devising. He is successful if and only if the arbiter later certifies the fraudulent message as being one that the transmitter could have sent under the existing protocol. If he attempts to cheat before any legitimate message has been sent, this is an R_0 deception. If he waits until after he has received ℓ legitimate messages, $\ell \geq 1$, from the transmitter, and then—using this additional information—attempts to cheat in the same way as before, this is said to be an R_ℓ deception.

The category R_ℓ, $\ell > 1$, will not be considered here for the same reason that the categories I_ℓ, $\ell \geq 1$, and S_ℓ, $\ell > 1$, were excluded from consideration for opponent deceptions.

4. T. The transmitter can attempt to cheat the receiver by sending a message that the receiver may accept and then claiming that he didn't send it, that is, by disavowing a legitimate message. He will be successful in this deceit if and only if when the receiver later claims to have been cheated the arbiter rules that the message is not one that the transmitter would have sent under the existing protocol and, in consequence, the receiver is held liable.

There is an exact parallel between the deceptions available to an opponent and to an arbiter since the object of either of them in attempting a deception is to cause the receiver to either accept a fraudulent or an altered message and to be misinformed thereby. If the arbiter is not privy to any privileged information, which, for example, is the case in the use of the Rivest–Shamir–Adleman (RSA) digital signature scheme described in the next section, then the arbiter's capabilities are the same as those of an opponent; that is, for I_ℓ- and S_ℓ-type deceptions, $\ell \geq 1$. If, however, the arbiter is an insider, with perhaps information not known to either the transmitter or receiver, then he will have a class of deceptions unique to him, which we denote by A_{I_ℓ} or A_{S_ℓ}.

5. A_{I_0}. The arbiter, using both the public knowledge of the general authentication scheme and his privileged information, chooses a fraudulent message that he sends to

the receiver when in fact no message has been sent by the transmitter. If this message is accepted the receiver will certainly be deceived (as to the transmitter's intent).

6. A_{I_ℓ} and A_{S_ℓ}. The arbiter can wait to intercept ℓ legitimate messages, $\ell \geq 1$, sent by the authorized transmitter and then using the additional information acquired from observing which messages have been used by the transmitter, either substitute in the stead of the ℓth message a fraudulent message that has a maximum chance of being accepted by the receiver or else forward the ℓth message unmodified and attempt to get a fraudulent message accepted as authentic before the legitimate transmitter sends the $(\ell + 1)$st authentic message. In this case, he (the arbiter) wins if and only if the receiver ends up being misinformed of the transmitter's intended communication.

This is an appropriate point to point out the second natural dichotomy of authentication schemes—depending on whether they permit arbitration of transmitter/receiver disputes as to the authenticity of messages or not.

In all of the authentication schemes discussed thus far, since the transmitter and receiver must both know the same secret (from the opponent) information (either the key in a simple key cryptographic algorithm, or a sealed authenticator and a cryptographic key or an initial vector and a cryptographic key, etc.) they can each do anything the other can do. In particular, because of this duality, the receiver cannot "prove" to a third party that a message he claims to have received from the transmitter was indeed sent by the transmitter, since he (the receiver) has the capability to utter an indetectable forgery, that is, the transmitter can disavow a message that he actually sent. Similarly, the receiver can claim to have received a message when none was sent, that is, to falsely attribute a message to the transmitter, who cannot prove that he didn't send the message since he could have. In the classic military authentication scheme this is an acceptable situation, since a superior commander doesn't worry that a subordinate will attribute an order to him that he didn't issue and the subordinate doesn't worry that his superior will disavow an order that he did send. There is, in fact, some rudimentary protection against this sort of cheating provided by the sealed authenticators the military uses since if either party can produce his unopened authenticator, it is prima facie evidence that he doesn't know its contents and hence could not have authenticated a message using its contents. This doesn't protect the transmitter from the receiver's stripping the authenticator from the legitimate message and attaching it to a fraudulent one—but again, in the military setting, this is not regarded as a serious problem. In many situations, and in almost all commercial and business applications, the primary concern is with insider cheating, for example, the person withdrawing cash from an ATM may not be the account holder or the amount shown on a properly signed and valid check may be altered to a larger figure, etc. If the authentication scheme permits arbitration, the arbiter's sole function is to certify on demand whether a particular message presented to him is authentic or not, that is, whether it is a message that the transmitter could have sent under the established protocol. He can never say that the transmitter did send the message—although the probability that it could have come from a source other than the authorized transmitter can be made as small as one likes—only that he could have under the established protocol.

3 A NATURAL CLASSIFICATION OF AUTHENTICATION SCHEMES

We have already pointed out two classifications for authentication schemes depending on whether authentication is achieved with or without secrecy for the information being authenticated, and whether it is possible to arbitrate a dispute between the transmitter and receiver as to the authenticity of a message. We now consider the most

important classification of all arising from the source of the security of the authentication scheme itself.

Authentication schemes are classified as being computationally secure, provably secure, or unconditionally secure, terms that sound very much alike. In fact, they describe quite different bases for confidence in the integrity of the authentication.

A scheme is said to be *computationally secure* if the security depends on a would-be cheater carrying out some computation that in principle is possible but in which all of the known methods of attack require an infeasible amount of computation. A good example of a computationally secure authentication scheme is a widely used variation of the Federal Reserve electronic funds transfer (EFT) protocol described above, in which the only difference is that there is no daily initialization vector. The protocol for generating the appended authenticator in this case is described in the work by Meyer and Matyas, *Cryptography: A New Dimension in Computer Data Security* [15].

> The suggested technique for MAC generation is as follows: The first 64 bits of that portion of the transaction to be protected are block-encrypted using DES and the secret key. Then the next 64 transaction bits to be thus protected are exclusive-or'ed (modulo 2 added) with the just produced cipher. The result is then block encrypted using the same key, producing a new 64 bits of cipher. This procedure is continued until all critical transaction fields have been included. (The final data block will likely be less than 64 bits, so it is padded with zeros to make a full 64 bits prior to being exclusive-or'ed with the just-produced cipher.) Some subset of the final cipher, at least six decimal digits or five hexadecimal digits, serves as the MAC.

In this case an opponent, given the plaintext (state of the source) and matching MAC has all of the information needed to solve for the unknown DES key. DES keys are 8 bytes long, each byte consisting of 7 bits of actual key with the eighth bit being an unused parity check bit over the other 7 bits. With virtual certainty (≈ 0.996) for a 64-bit MAC there is only one key that will generate the observed MAC from the plaintext, therefore an opponent would only have to test at most the 2^{56} possible keys (with probability 0.996) to find the key used by the transmitter to generate the MAC and hence to be able to authenticate an arbitrary message of his choosing. This is the brute force cryptanalytic attack on the DES discussed by Diffie and Hellman [5] which remains at this time computationally infeasible to carry out. In other words, the security of the MAC generated by the protocol described in [15] is equivalent to the difficulty of carrying out a well-defined, but at present infeasible, computation. There is an important, but subtle, difference between the Federal Reserve EFT protocol and the one described here that has to do with the uncertainty introduced by the initialization vector. The net result is that the EFT protocol essentially requires cryptanalysis in depth requiring impractically many matched plaintext-MAC pairs to achieve the same degree of confidence in recovering the unknown DES key as for the simplified protocol. Both schemes are, however, only computationally secure.

An authentication scheme is said to be *provably secure* if it can be shown that breaking it implies that some other, presumed hard, problem, such as factoring suitable chosen large composite integers or extracting discrete logarithms in a finite field $GF(q)$ where q has been carefully chosen, etc. could be solved with comparable effort. We illustrate this concept using a provably secure authentication scheme equivalent in security to the difficulty of factoring.

The RSA cryptoalgorithm [17] is the most widely known and used of the two-key (originally public key) algorithms. This algorithm depends for its security on the difficulty of factoring suitably chosen large composite integers. A bare bones description of the algorithm* is the following. Two primes p and q, sufficiently large that their product (modulus) $n = pq$ is infeasible (impossible?) to factor, are randomly chosen, subject to some side conditions that are not relevant to the present discussion. An exponent, e, is randomly chosen such that

$$(e, (p - 1)(q - 1)) = 1$$

that is, so that e has no factor in common with $(p - 1)$ or $(q - 1)$ other than 1, and the multiplicative inverse exponent, d, calculated:

$$ed = 1 \pmod{(p - 1)(q - 1)}$$

Given e this is easy to do, requiring only $0(\log(n))$ work, using the Euclidean algorithm if one knows the factorization of n, and as hard as factoring n otherwise.

To establish an authentication channel using the RSA cryptoalgorithm, the pair $(d; n)$ is made public and the pair $(e; n)$ kept secret. It should be remarked that if one knows both e and d, n is easy to factor. Messages are integers, $m < n$, where the message is assumed to contain redundant authenticating information known to both the transmitter and receiver, and the opponent; say that all authentic messages must end with fifty 0's in the least significant bit positions. The RSA algorithm is a block encryption algorithm, defined by

$$c \equiv m^e \pmod{n}$$

where m is the extended message (including the appended authenticator $00 \ldots 0$) and c is the least positive residue of m^e modulo n. The receiver will accept as authentic any cipher, $\bar{c}$ that when decrypted using the key $(d; n)$

$$\bar{m} \equiv (\bar{c})^d \pmod{n}$$

yields a text, $\bar{m}$ with the suffix of fifty 0's. The probability that a randomly chosen cipher $\bar{c}$ will decrypt to a text $\bar{m}$ of this form is $2^{-50} \approx 8.9 \times 10^{-16}$ which is the confidence one would have in the authenticity of a message for this example. Clearly one could arrange for a higher or lower degree of confidence by varying the amount of prearranged redundant information in the authentic messages; limited above, of course, by the difficulty of factoring n, since if this is possible the secret e could be calculated from the public d and any message authenticated. This authentication scheme might at first appear to satisfy the definition of a provably secure system, however, this isn't the case since the implication is in the wrong direction; that is, the integrity of the authentication is no greater than factoring is difficult. What we would like to be able to say is that deception using this authentication scheme is at least as hard as factoring. In spite of an enormous amount of work on this and related questions of security dependent on

---

*The reader is referred to J. L. Massey's chapter "Contemporary Cryptology: An Introduction" or to the chapter on "Public Key Cryptography" by J. Nechvatal for a more complete discussion of the RSA cryptoalgorithm.

factoring, it is not known whether the decryption of almost all ciphers for arbitrary exponents, e, in the RSA encryption scheme is as hard as factoring. However, Rabin [16] has shown that if one uses the encryption function

$$c \equiv m(m + b) \pmod{n}$$

where b is an integer, $0 \leq b < n$, which is effectively the same as encrypting using the exponent $e = 2$ in the RSA algorithm, then decryption is not simply a consequence of being able to factor n but is actually equivalent. It should be pointed out that if the encryption exponent, e, is chosen to be 2 in an RSA scheme, that there is no corresponding decryption exponent, d, since there can be no solution to the congruence

$$2d \equiv 1 \pmod{(p - 1)(q - 1)}$$

and that consequently the Rabin scheme is slightly different from the RSA scheme. From the standpoint of a secrecy channel, Rabin's scheme, however, has the serious shortcoming that the authorized user is almost always left with an ambiguity among four potential messages. This is true since a quadratic residue, r, modulo $n = pq$, that is, a residue that is the square of some other residue, has four square roots if $(r, n) = 1$. For example, $2^2 \equiv 7^2 \equiv 8^2 \equiv 13^2 \equiv 4 \pmod{15}$. Williams has extended Rabin's work by showing that if the primes p and q are chosen so that $p \equiv 3 \bmod 8$ and $q \equiv 7 \bmod 8$, then there is an easy procedure for identifying one of the four roots to a quadratic equation as the distinguished root, that is, the intended message. The price that is exacted for this resolution of the ambiguity, besides the insignificant restriction on the primes p and q, is that the modulus, n, must be roughly four times larger than the largest message. The precise reasons for this are not important to the present discussion, the interested reader is referred to [34] for the full details. Since the Rabin–Williams scheme is a secrecy channel, not an authentication channel, it doesn't directly satisfy our needs. The reason why it is only a secrecy channel is easy to see; in the Rabin–Williams scheme the public users know the rules for transforming a message into a form suitable for encryption and that squaring (or in the most general form raising the transformed message to a publicly known even exponent) is the public encryption operation. The receiver, knowing the factorization of the modulus n, can recover the transformed message which he can then unambiguously interpret by reversing the initial transformation.

Williams does describe a digital signature scheme based on his and Rabin's technique that requires each user to have two of the private channels: one for signing and one for sealing messages. The basic idea is that the transmitter first carries out an operation that only he can do, that is, extracting a square root with respect to one of his moduli and then performs an operation that only the receiver can undo, that is, squaring with respect to the receiver's public encryption modulus. This cipher, along with a plaintext message identifying the transmitter, is sent to the receiver. The receiver, since he knows who the purported transmitter is, can carry out the inverse operations in reverse order to recover a message, which, if it contains the expected redundant (authenticating) information, could only have come from someone knowing the factorization of the identified transmitter's signing modulus—ipso facto the transmitter.

To see roughly how this scheme would work, assume that the source can take any one of a thousand equally likely states, labeled digitally 000,001, . . . , 999 and that the redundant (appended authenticator) information consists of three terminal 0's, that

is, only ciphers that decrypt to texts of the form *xyz* 000 will be accepted as authentic. In the Rabin–Williams scheme, not only must $p \equiv 3 \bmod 8$ and $q \equiv 7 \bmod 8$ but it must also be true that $2m + 1$ is a quadratic residue with respect to $n_t = p_t q_t$ and that $4(2m+1) < n_t$, where m is the message. In the present example $n_t > 7992004$. Therefore, let the transmitter's signing modulus be defined by the primes $p_t = 2027$ and $q_t = 3943$ and $n_t = 7992461$. The receiver's sealing (public encryption) modulus must be greater than $8n_t + 1$ in order to be certain that the cascaded encryption and decryption operations will not interfere with each other, that is, $n_r > 63939689$. Let $p_r = 7907$ and $q_r = 8087$ so that $n_r = 63943909$. Using these parameters, which were selected to make nearly optimal use of the Rabin–Williams channel bandwidth, any message of the form *xyz* 000 can be authenticated with a probability of successful deception by an opponent of only 10^{-3}. The reason for describing this example in such detail was to highlight a common cost in authentication; ≈ 26 bits must be transmitted to communicate the superencrypted cipher (residue with respect to $n_r = 63943909$). This cipher when processed by the receiver conveys roughly 10 bits of information (one message out of 1000 equally likely triples) and provides roughly 10 bits of authentication (10^{-3} probability of deception) at the expense of an extra 6 bits having to be transmitted that cannot be used to either convey useful information or to confer additional security. The important point, though, is that authentication using the Rabin–Williams algorithm has been proven to be as secure as factoring is difficult.

The distinction between computationally secure schemes and provably secure schemes is a subtle one since in both cases the security is dependent on the computational difficulty of solving some hard problem. Perhaps the difference is only that in the first case one has reason to believe that the security is as great as a hard problem is difficult to solve, while in the other, one knows that it is at least that great. One important difference, though, is that in the case of a provably secure scheme, a proof must be given showing that subverting the security (for almost all cases) is equivalent to solving the hard problem. Because of this common dependence on computational infeasibility to achieve security, the present author combined computationally secure and provably secure authentication schemes into a single class of computationally secure schemes in his taxonomy for authentication schemes [27].

Figure 1 shows the natural taxonomy for authentication schemes based on the categories identified here; security, secrecy and arbitration. The divisions are:

- Unconditionally secure—computationally secure
- With secrecy—without secrecy
- With arbitration—without arbitration

In Fig. 1 for the four computationally secure categories, examples that have been discussed here and that are representative of each category are indicated. For the unconditionally secure categories which have yet to be discussed the references cited indicate where the first schemes illustrative of a category appeared in the literature. No one lays claim to the "brute force" solution for unconditionally secure authentication codes that provide for both secrecy and arbitration. Given the authentication codes for the other categories, it is obvious how to replicate a code sufficiently many times in a Cartesian product construction to achieve the desired result, however, the resulting constructions are so extravagantly wasteful of key (the information secretly

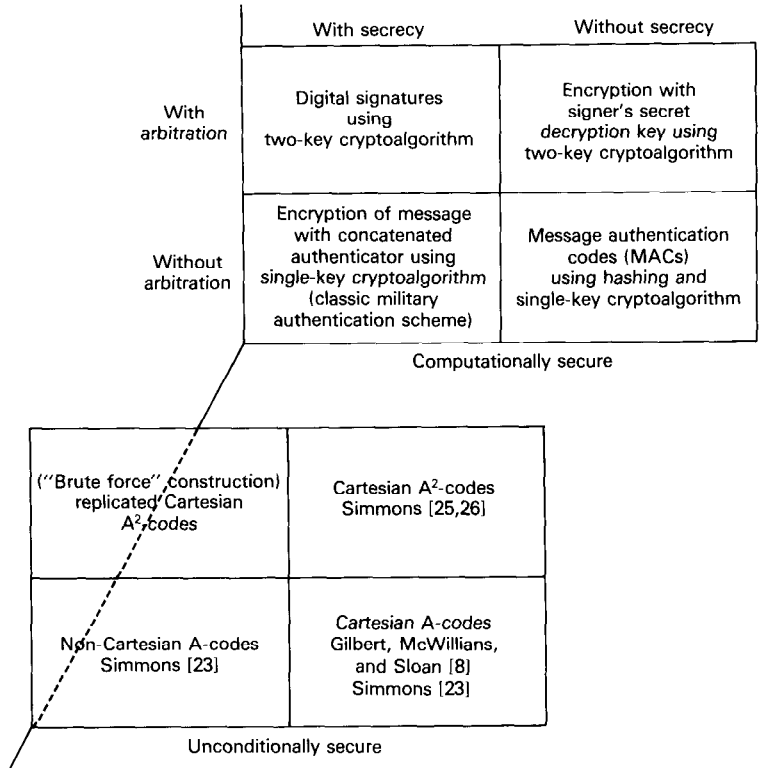

Figure 1 A natural taxonomy for authentication schemes.

exchanged between the insiders) that they are of no practical interest. The construction of efficient authentication codes for this category of authentication schemes is still an open problem.

In computationally and provably secure authentication schemes, the sets of acceptable messages are often determined (virtually constructed) by either appending a cryptographically related MAC to the information being authenticated, or else by first appending an unrelated authenticator and then cryptographically "sealing" the resulting extended message using either a single-key or a two-key cryptoalgorithm. Each choice of a key defines one subset of acceptable messages. In the case of authentication schemes based on single-key cryptography this is unavoidable since the only operation that the insiders (who know the key) can do, that outsiders cannot, is to encrypt or decrypt information using the secret key. In the case of two-key cryptographic techniques though, or especially in the case of a pure authentication channel, this need not be true. This is because an authentication channel can differ significantly from a secrecy channel, since in the one case it is only necessary that the receiver be able to verify that the authentication operation has been correctly carried out to establish that the communication is authentic, while in the other case he must be able to "invert" the operation to actually recover the information concealed in the cipher.

A well-known example of the latter type, that is, of using a public key encryption algorithm to define the set of acceptable messages by concealing them in ciphers is the

digital signature scheme defined by Rivest, Shamir, and Adleman [17]. In this case, the information being authenticated is concealed by the authentication operation and revealed as an essential part of the process of verifying its authenticity. It is, of course, essential that only a predesignated fraction of the messages will be accepted as authentic: for example, those ending in a preagreed-on suffix. On the other hand, the digital signature scheme of El Gamal [6] is essentially an appended authenticator (MAC) to the message which need not be, and in fact can't be, decrypted by the receiver in the process of verifying that the message is authentic. Both of these schemes are computationally secure as are most of the other digital signature schemes based on two-key cryptographic techniques.

There are, however, a few provably secure authentication (digital signature) schemes [10,16,29,34] based on public key cryptographic algorithms. The digital signature scheme of Goldwasser, Micali, and Rivest [10] in addition passes a very strong security requirement; namely, it is secure against adaptive chosen message attacks. This is an appropriate point to remark that a digital signature is more than just a computationally secure or provably secure authentication with arbitration scheme. Anyone (having access to entirely public information) can verify the authenticity of a signature—not just the predesignated arbiter(s). It is an important point and should be clearly stated that the price paid to achieve unconditional security (in all presently known realizations) is to restrict the ability to authenticate messages to insiders, that is, to parties possessing some information not known to all of the other participants.

An authentication scheme is said to be unconditionally secure if the security is independent of the computing power or time an opponent can bring to bear, or equivalently if the opponent can do no better than to randomly choose a message on the chance that it may be acceptable to the receiver, irrespective of whether he attempts an I_0 or an I_1 or S_1 deception. Since the only known unconditionally secure authentication schemes are of the form of an explicit description of the various encoding (from states of the source into messages) rules, we will normally refer to such schemes as authentication codes. The notion of unconditionally secure authentication codes (similar in many respects to unconditionally secure encryption using one-time keys) is originally due to Simmons [23–24] although the subject now has a burgeoning literature. Because authentication codes are not nearly so well known as either computationally secure or provably secure schemes, we must develop the essential concepts for unconditionally secure schemes to illustrate one.

As the present author has pointed out elsewhere [23], unconditionally secure authentication codes are in a strict sense a mathematical dual to error detecting and correcting codes. In both cases, redundant information is introduced into the sequence of symbols that are transmitted, resulting in only a fraction out of the set of all possible sequences being available for use by the transmitter. In the one case, if the receiver receives a sequence that would not have been used by the transmitter, a fixed rule (usually a maximum likelihood detector) is invoked to decide which sequence was most likely transmitted, while in the other the receipt of a sequence that would not have been used by the transmitter under the agreed-on protocol is interpreted to mean that he didn't send it, or that if he did someone else has subsequently altered it, with the result that the message is to be rejected as unauthentic in either event.

Coding theory is concerned with schemes (codes) that introduce a redundancy in such a way that the most likely alterations to the encoded messages are in some sense close to the codeword they derive from. The receiver can then use a maximum likeli-

hood detector to decide which (acceptable) message he should infer as having most likely been transmitted from the (possibly altered) codeword that was received. In other words, the object in coding theory is to cluster the most likely alterations of an acceptable codeword as closely as possible (in an appropriate metric) to the codeword itself, and disjoint from the corresponding clusters about other acceptable codewords to make unambiguous decoding possible.

Authentication codes are also concerned with schemes that introduce redundancy in the message, but in such a way that for any message the transmitter may send, the substitute or altered messages that the opponent may introduce using his optimal strategy are spread in what appears to be a random manner, that is, as uniformly as possible (again with respect to an appropriate metric) over the set of all possible messages. The theory of authentication codes is concerned with devising and analyzing schemes (codes) to achieve this "spreading." It is in this sense that coding theory and authentication theory are dual theories: one is concerned with clustering the most likely alterations of acceptable messages (codewords) as closely about the original codeword as possible and the other with spreading the optimal (to the opponent) alterations as uniformly as possible over the set of all messages.

Just as in error detecting and correcting codes, in either authentication codes or in constructive procedures for the authentication of digital information, for any fixed encoding rule, there is also a subset of acceptable, that is, authentic, sequences (those containing the specified redundant information) and a nonempty collection of sequences that do not contain the redundant information and hence that would be rejected as unauthentic. The difference is that whereas in coding theory there is but one encoding rule corresponding to the fixed code, in authentication codes there are many encoding rules from which the transmitter/receiver can choose the particular (secret) rule they will employ.

We illustrate these concepts with the smallest possible example. The source in our example is a fair coin toss observed by the transmitter, whose outcome, denoted by H or T, is the information (1 bit) the transmitter wishes to communicate to the receiver. The opponent wishes to deceive the receiver into either believing a legitimate coin toss has occurred when it hasn't, an I_0 deception, or else in misinforming him as to the outcome of the legitimate toss, an S_1 deception. An unconditionally secure authentication code to provide protection (1 bit) against both of these deceptions is the following:

$$\mathbf{A} = \begin{bmatrix} H- \\ -H \end{bmatrix} \otimes \begin{bmatrix} T- \\ -T \end{bmatrix} = \begin{array}{c|cccc} & m_1 & m_2 & m_3 & m_4 \\ \hline \mathbf{e}_1 & H & — & T & — \\ \mathbf{e}_2 & H & — & — & T \\ \mathbf{e}_3 & — & H & T & — \\ \mathbf{e}_4 & — & H & — & T \end{array} \tag{1}$$

There are four encoding rules, e_i, each of which encode the source states (H and T) into two out of the four possible messages, m_j; that is, 2 bits must be communicated through the channel to specify a message. To use this code to authenticate a communication of the outcome of a coin toss, the transmitter and receiver would choose (in secret from the opponent) one of the encoding rules with the uniform probability distribution on the $\mathbf{e}_i$ (their optimal authentication strategy) in advance of their need to authenticate a communication. The opponent knows the code (see Eq. (1)) and their strategy for choosing an $\mathbf{e}_i$. If he chooses to impersonate the transmitter and send an

unauthentic message when no message has yet been sent by the transmitter, it should be obvious that irrespective of which message, m_j, he chooses, the probability that it will correspond to an encoding of a source state under encoding rule $\mathbf{e}_i$, and hence that it will be accepted by the receiver as an authentic message, is $\frac{1}{2}$. Similarly, if the opponent waits to observe a legitimate communication by the transmitter, his uncertainty about the encoding rule being used will drop from one of four equally likely possibilities to one of two. However, his probability of choosing an acceptable (to the receiver) substitute message will still be $\frac{1}{2}$. For example, if the opponent observes m_1 he knows that the transmitter and receiver are using either encoding rule $\mathbf{e}_1$ or $\mathbf{e}_2$.

In the first case m_3 would be an acceptable message to the receiver while m_4 would be rejected as unauthentic, while in the second case, exactly the opposite would be true. Hence, the opponent's probability of deceiving the receiver is $\frac{1}{2}$ irrespective of whether he impersonates the transmitter, an I_0 deception, or substitutes (modifies) legitimate messages, an S_1 deception. Another way of expressing this is that the opponent's a priori and a posteriori probabilities of success are both the same as the probability that a randomly chosen message will be acceptable to the receiver.

It should be pointed out that the code in Eq. (1) can be viewed as an appended authenticator scheme; functionally the same as the provably secure EFT scheme described above. Since messages m_1 and m_2 communicate only source state H, and messages m_3 and m_4 only source state T, the messages in this code are of the form shown above.

$$m_{i,j} = s_i : f(\mathbf{e}_j, s_j) \tag{2}$$

where

	i	
j	H	T
$\mathbf{e}_1$	0	0
$\mathbf{e}_2$	0	1
$\mathbf{e}_3$	1	0
$\mathbf{e}_4$	1	1

$f(\mathbf{e}_j, s_i)$ (3)

Eq. (1) can be rewritten with the messages m_i in the form $s_i : f(\mathbf{e}_j, s_i)$ that makes clear that this is an appended authenticator scheme

	$H:0$	$H:1$	$T:0$	$T:1$
$\mathbf{e}_1$	H	—	T	—
$\mathbf{A} = \mathbf{e}_2$	H	—	—	T
$\mathbf{e}_3$	—	H	T	—
$\mathbf{e}_4$	—	H	—	T

By the arguments already given, there is 1 bit of uncertainty as to authenticator that should accompany a source state in a message, even if the message conveying the other source state and matching acceptable authenticator has been observed. Irrespective of the computing power an opponent may have, he cannot reduce this uncertainty as to the acceptable appended authenticator for either an S_0 or I_1 deception.

This small example (1-bit source) can easily be extended to give an arbitrarily high degree of confidence against deception: $p_d = 1/n$, n an integer. Replace **A** in Eq. (1) by the Cartesian product of the $n \times n$ matrices

$$\mathbf{A} = \begin{bmatrix} H & \cdots & \\ & H & \\ \vdots & & \vdots \\ & \cdots & H \end{bmatrix} \otimes \begin{bmatrix} T & \cdots & \\ & T & \\ \vdots & & \vdots \\ & \cdots & T \end{bmatrix} \tag{4}$$

This forms an authentication code with n^2 rows (encoding rules) and $2n$ columns (messages) in which the optimal strategy for the transmitter/receiver is still to choose an encoding rule $\mathbf{e}_i$ with the uniform probability distribution. The opponent's chances of success will be $1/n$ for either an I_0 or S_1 deception by the same arguments used before. The channel requires $\log_2(2n) = 1 + \log_2 n$ bits be communicated to convey the state of the source (1 bit) and to secure $1/n$ confidence in the authenticity of the received message ($\log n$ bits). The code is therefore said to be perfect in the sense that every bit of information communicated in a message is used to either convey the state of the source or else to confound the opponent. All of these codes are unconditionally secure.

Given the simple nature of the construction in the preceding paragraph, it might appear at first that there would be no difficulty in constructing unconditionally secure authentication codes for arbitrary sources and for any desired level of security. If the source can assume any one of k states, and if the desired level of security is $1/n$, n an integer, then the obvious generalization of Eq. (4) would be the k-fold repeated Cartesian product of $n \times n$ matrices

$$\mathbf{A} = \begin{bmatrix} s_1 & \cdots & \\ & s_1 & \\ \vdots & & \vdots \\ & \cdots & s_1 \end{bmatrix} \otimes \begin{bmatrix} s_2 & \cdots & \\ & s_2 & \\ \vdots & & \vdots \\ & \cdots & s_2 \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} s_k & \cdots & \\ & s_k & \\ \vdots & & \vdots \\ & \cdots & s_k \end{bmatrix} \tag{5}$$

which yields a code with n^k rows (encoding rules) and nk columns (messages). This unconditionally secure authentication code is also perfect in the sense defined above since the number of bits required to specify a particular message out of nk equally likely messages is

$$\log_2(nk) = \log_2 k + \log_2 n$$

which is precisely the amount of information needed to convey the state of the source ($\log_2 k$ bits if all states are equally likely) and to present the opponent with an equivocation of $\log n$ bits as to the appended authenticator that the receiver will accept. This code is not very efficient, however, in another important respect which we will discuss below.

In the military sealed-authenticator authentication scheme described above, the information being authenticated (state of the source) was concealed in the cipher, that is, it was secret. On the other hand, in the Federal Reserve EFT authentication protocol, the information being authenticated was transmitted in the clear, that is, it was unencrypted or without secrecy. This distinction of whether the authentication is made with or without secrecy for the underlying information is an essential division of all authentication schemes [27], and can be of enormous importance for some applications such as treaty verification where secrecy is unacceptable. For the moment we merely

remark that the Cartesian product constructions just described, since they lead to only appended authenticator schemes, are necessarily authentication without secrecy schemes. It is possible, however, to construct perfect authentication with secrecy codes corresponding to the authentication without secrecy schemes already exhibited. An authentication with secrecy code corresponding to the authentication without secrecy code shown in Eq. (1) is:

$$\mathbf{A} = \begin{array}{c|cccc} & m_1 & m_2 & m_3 & m_4 \\ \hline \mathbf{e}_1 & H & — & T & — \\ \mathbf{e}_2 & — & H & — & T \\ \mathbf{e}_3 & T & — & — & H \\ \mathbf{e}_4 & — & T & H & — \end{array} \tag{6}$$

The source is the same as before; the channel still requires 2 bits to specify a particular message and the optimal strategies for the transmitter/receiver and for the opponent remain the same. The difference is that no matter which message the opponent may observe, he has 1 bit of uncertainty remaining as to the encoding rule the transmitter/receiver are using and thus 1 bit of uncertainty as to the source state. But there was only 1 bit of a priori uncertainty as to the source state before an observation was made; hence, this is a perfect authentication with secrecy scheme. It should be noted that if the opponent does somehow learn the state of the source and also observes the message that the transmitter uses to communicate this state to the receiver, he can then unambiguously identify the encoding rule they are using and hence substitute an acceptable message to misinform the receiver of the state of the source, an S_1 deception, with certainty of success.

There is a simple example that demonstrates in a striking way the difference between computationally secure authentication schemes and unconditionally secure schemes. To present it here, we need to anticipate a result that will be derived later in Section 4. This is a bound on the security of any authentication channel, whether computationally secure or unconditionally secure, first derived by Gilbert, McWilliams, and Sloan [8] (Eq. (28)) in a slightly different setting than used here, which says that the probability, P_d, of an opponent succeeding in either an I_0 or an S_1 deception is bounded below by

$$P_d \geq \frac{1}{\sqrt{|\mathscr{E}|}}$$

(see Eq. 28) where $|\mathscr{E}|$ is the total number of encoding rules in the authentication scheme, that is, the number of rows in the encoding matrix. In 1985, the present author reported [24] an apparently paradoxical situation in which this bound appeared to be violated in a computationally secure authentication scheme.

Let the source be a fair coin toss as before and let the sets of acceptable (authentic) messages be constructed by encrypting with a secret DES key the state of the source concatenated with a redundant authenticating suffix of sixty-three 1's. In other words, the only ciphers that the receiver will accept are those that decrypt to one of the two texts

$$\overbrace{011\ldots1}^{63} \quad \text{or} \quad \overbrace{111\ldots1}^{63}$$

communicating, say, tails or heads, respectively. As is well known, there are 2^{56} DES keys, and hence in the scheme just described 2^{56} encoding rules. Consequently, $|\mathscr{E}| = 2^{56}$, and the bound (Eq. (10)) says that even if the transmitter/receiver choose among the 2^{56} encoding rules optimally, they cannot limit the opponent's probability of successfully deceiving the receiver into accepting an unauthentic message to less than

$$P_d \geq \frac{1}{\sqrt{|\mathscr{E}|}} = \frac{1}{2^{28}} \approx 3.7 \times 10^{-9} \tag{7}$$

or roughly 4 parts in a billion.

In practice, the opponent's chances of success are dramatically less than Eq. (7) would suggest. There are 2^{64} possible ciphers (messages), only 2 of which are acceptable for any particular choice of a key (authentication encoding rule). Therefore, if the opponent merely selects a cipher at random and attempts to impersonate the transmitter, his probability of success is 2^{-63} or approximately 1 chance in 10^{19}. The question is, Can he do better? As far as an I_0 deception is concerned, the answer is essentially no, even if he has unlimited computing power. For each choice of an encoding rule, there are two (out of 2^{64}) ciphers that will be acceptable as authentic. Assuming that the mapping of the sequences 11 . . . 1 and 01 . . . 1 into 64-bit cipher sequences under DES keys is a random process, this says that the total expected number of acceptable ciphers (over all 2^{56} keys) is $\approx 2^{56.9888}$, that is, ε close to 2^{57}. Even if the opponent could feasibly carry out the enormous amount of computation that would be required to permit him to restrict himself to choosing a cipher from among this collection, his chances of having a fraudulent message be accepted by the receiver would still only be $\approx 2^{-56}$ or roughly 1 chance in 10^{17} which is what we meant when we said that the answer was essentially no since 10^{-17} isn't much different than 10^{-19} while both differ enormously from the bound (see Eq. (7)) of $\approx 10^{-9}$. The opponent could not do better, nor worse (in attempting to impersonate the transmitter) even if he possessed infinite computing power than choose a cipher randomly, with a probability distribution weighted to reflect the number of times each cipher occurs, from among the $\approx 2^{57}$ potentially acceptable ciphers.

However, the channel bound $p_d \geq |\mathscr{E}|^{-\frac{1}{2}}$ (31) applies to all authentication schemes, hence the apparent contradiction must arise in connection with the opponent's substitution, S_1, strategy. If the opponent waits to observe a legitimate message (cipher), can the information acquired by virtue of this observation be put to practical use to improve his chances of deceiving the receiver? Even if he doesn't know the state of the source, he knows that the cipher is the result of encrypting one of the two 64-bit sequences 111 . . . 1 or 011 . . . 1 with one of the 2^{56} DES keys. He also knows that with a probability of essentially 1 (≈ 0.996), there is only one key that maps the observed cipher into either of these two sequences, hence, he is faced with a classic "meet in the middle" cryptanalysis of DES as discussed by Diffie and Hellman [5].

Clearly if he succeeds in identifying the DES key, that is, the encoding rule being employed by the transmitter receiver, he can encrypt the other binary string and be certain of having it be accepted, and hence be certain of deceiving the receiver. The point, though, is that for him to make use of his observation of a message he must be able to determine the DES key the transmitter and receiver(s) are using, that is, he must be able to cryptanalyze DES. If he can do this, the expected probability of deceiving the receiver is e close to 1, the small deviation being attributable to the exceedingly

small chance that two (or more) DES keys might have encoded source states into the same message (cipher). But even if this is the case (improbable though it may be), his probability of success in an S_1 deception is simply the probability that he selects the same DES key as the receiver and transmitter are using out of a set of two (or few) possible choices. Thus, we have the paradoxical result that the practical system in this example is some eight or nine orders of magnitude more secure than the theoretical limit simply because it is computationally infeasible for the opponent to carry out in practice what he should be able to do in principle. In this respect, practical message authentication is closely akin to practical cryptography where security is equated to the computational infeasibility of inverting from arbitrarily much known matching cipher text and plaintext pairs to solve for the unknown key, even though in principle there may be more than enough information available to insure a unique solution.

4 HOW INSECURE CAN UNCONDITIONALLY SECURE AUTHENTICATION BE?

We have already adopted a model for unconditionally secure authentication codes in which the encoding rules, $\mathbf{e}_i$, are organized in an array, $\mathbf{A}$, specifying the available mappings of the source states into the set of all possible messages. The rows of $\mathbf{A}$ are indexed by encoding rules and the columns by messages. The entry in $a(\mathbf{e}_i, m_j)$ is the element of $\mathcal{S}$ encoded by rule $\mathbf{e}_i$ into message m_j if such a source mapping exists under $\mathbf{e}_i$ and 0 otherwise. Every element of $\mathcal{E}$ appears in each row of $\mathbf{A}$ at least once and perhaps several times since the transmitter must be able to communicate an arbitrary state of the source to the receiver. Clearly, if an encoding rule uses all of the messages, that is, every message is acceptable to the receivers as an authentic communication, then an opponent is certain of success in an S_0 deception whenever that encoding rule is used. Therefore, in any strategy that holds the opponent to less than a certainty of success (in deceiving the receiver) any such encoding rule will not be used, and can therefore with no loss of generality be assumed to not occur in $\mathbf{A}$. Put another way, every encoding rule will have at least one unused message. Similarly, no message can occur in all encoding rules, otherwise the opponent could impersonate using that message with certainty of success. We now define another $|\mathcal{E}| \times |\mathcal{M}|$ matrix $\mathbf{X}$, in which

$$\chi(\mathbf{e}_i, m_j) = \begin{cases} 1 & \text{if } a(\mathbf{e}_i, m_j) \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases}$$

For example, for the authentication scheme in Eq. (1):

$$\mathbf{A} = \begin{array}{c|cccc} & m_1 & m_2 & m_3 & m_4 \\ \hline & s_1 & — & s_2 & — \\ & s_1 & — & — & s_2 \\ & — & s_1 & s_2 & — \\ & — & s_1 & — & s_2 \end{array} \quad \text{and} \quad \mathbf{X} = \begin{vmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{vmatrix}$$

It is now easy to see the relationship of the impersonation "game," the I_0 deception, to the matrix $\mathbf{X}$. If m_j is an acceptable (authentic) message to the receiver when encoding rule $\mathbf{e}_i$ has been agreed to by the transmitter and receiver then $\chi(\mathbf{e}_i, m_j) = 1$

and the opponent has a probability of success of $p = 1$ if he communicates m_j to the receiver. Conversely, whenever $\chi(\mathbf{e}_i, m_j) = 0$ he is certain the message will be rejected. It is certainly plausible, and in fact rigorously true, that the opponent's probability of success in impersonating the transmitter is the value, v_I, of the zero-sum game whose payoff matrix is $\mathbf{X}$. It is possible to define a companion payoff matrix $\mathbf{Y}$ for the substitution game, although it is considerably more complex since the probabilities are now all conditioned based on the observed legitimate message, so that the size of the matrix Y grows as a power of $|\mathcal{M}|$. The value of this game, v_s, is the probability that the opponent will be successful in deceiving the receiver through intercepting a message sent by the transmitter and substituting one of his own devising, that is, the probability of an S_1 deception being successful. Given an authentication code, the transmitter/receiver have the freedom to choose among the encoding rules and if some state(s) of the source can be encoded into more than one message under some of the encoding rules, a choice of which messages to use, that is, a splitting strategy. The opponent, on the other hand, can choose between impersonation and substitution with whatever probability distribution he wishes and then choose according to his optimal strategy which fraudulent message he will communicate to the receiver, either with no conditioning if he is impersonating the transmitter or else conditioned on the message he observed if he is substituting messages. Not surprisingly there exist authentication codes in which the optimal strategy for the opponent is either pure impersonation, pure substitution, immaterial mixes of the two, or most interestingly, essential mixing of both as well as examples in which splitting is essential in the transmitter/receiver's optimal strategies. The point of these remarks is that it has been shown [1,23,32] that an opponent's overall probability of success in deceiving the receiver, P_d, is simply the value of a game, G, whose payoff matrix is the concatenation of $\mathbf{X}$ and $\mathbf{Y}$, and hence that

$$P_d = v_G \geq \max(v_I, v_s)$$

The notation we will use is:

Name	Set	Element	Variable	Entropy
Source	$\mathcal{S}$	s_i	S	$H(S)$
Message space	$\mathcal{M}$	m_j	M	$H(M)$
Encoding rules	$\mathcal{E}$	e_k	E	$H(E)$
Splitting strategies		$\pi(m_j \mid s_1 \mathbf{e}_k)$	Π	$H(M \mid ES)$
Impersonation strategy	Q	q_j	Q	$H(Q)$

where

$P(X = x)$ is the probability that the random variable X takes the value x, as, for example, $P(M = m)$, $P(S = s)$, or $P(E = e)$

$|\mathbf{e}_i| = \Sigma_{m \in \mathcal{M}} \chi(\mathbf{e}_i, m)$ is the number of nonzero entries in the $\mathbf{e}_i$ row of either $\mathbf{A}$ or $\mathbf{X}$

$|m_j| = \Sigma_{\mathbf{e} \in \mathcal{E}} \chi(e, m_j)$is the number of nonzero entries in the m_j column of either $\mathbf{A}$ or $\mathbf{X}$

It is now an easy matter to state and prove several easy lower bounds for just how insecure an unconditionally secure authentication code can be. All of these results appear in [23], but are repeated here to make this chapter self-contained. The notation P_d was introduced earlier, and denotes the probability that an opponent will be successful in an I_0 or an S_1 deception.

Theorem 1.

$$P_d \geq \frac{\min_{\mathscr{E}} |\mathbf{e}_i|}{|\mathcal{M}|} \tag{10}$$

Comment: As has already been noted, the opponent has available as part of his strategy the choice of whether to impersonate the transmitter or to substitute messages, that is, whether to attempt a type I_0 or S_1 deception. Therefore, the value of the concatenated game is at least as large as the value of either game alone (Eq (9)). What is actually proven in [23] is that for the impersonation game:

$$v_I \geq \frac{\min_{\mathscr{E}} |\mathbf{e}_i|}{|\mathcal{M}|} \tag{11}$$

Corollary: Since $\min_{\mathscr{E}} |e| \geq |\mathcal{S}|$

$$P_d = v_G \geq \frac{|\mathcal{S}|}{|\mathcal{M}|} \tag{12}$$

Theorem 2. Given an authentication scheme for which

$$v_G = \frac{\min_{\mathscr{E}} |\mathbf{e}_i|}{|\mathcal{M}|} \tag{13}$$

in every optimal strategy, E, for the transmitter/receiver $P(E = \mathbf{e}) = 0$ for any encoding rule for which $|\mathbf{e}| > \min_{\mathscr{E}} |\mathbf{e}_i|$. In other words, the transmitter/receiver will not use any encoding rule in which $|\mathbf{e}| > \min_{\mathscr{E}} |\mathbf{e}|$ if equality Eq. (13) holds.

Corollary: If for an authentication scheme

$$v_G = \frac{|\mathcal{S}|}{|\mathcal{M}|}$$

which by Theorem 1 can only happen if $\min|\mathbf{e}| = |\mathcal{S}|$, then every optimal strategy for the transmitter/receiver, E, has $P(E = \mathbf{e}_j) = 0$ for any encoding rule $\mathbf{e}_j$ for which $|e_j| > |\mathcal{S}|$.

Another way of stating the conclusion of the corollary is that if $v_G = |\mathcal{S}|/|\mathcal{M}|$ no splitting occurs in any encoding rule occuring in an optimal strategy! It is worth remarking that

$$v_G = \frac{\min_{\mathscr{E}} |\mathbf{e}|}{|\mathcal{M}|}$$

does not imply that splitting does not occur in any of the encoding rules that occur in $\mathscr{E}$. What is true, by Theorem 2, is that in this case all of the encoding rules that occur (with positive probability) in an optimal strategy use the same number of messages.

Several other channel capacity theorems of similar flavor can be proven; however, we now turn to the primary object in this section of formulating lower bounds for the insecurity of unconditionally secure authentication schemes in terms of the entropies of the primary variables: S, E, M, Q, etc.

A trivial bound can be given in terms of $H(E)$. Since $H(E)$ is the total equivocation that the opponent has as to which encoding rule is being used by the transmitter/receiver, and since he could deceive the receiver with certainty if he only knew the rule they had chosen, we have

$$\log P_d = \log v_G \geq - H(E) \tag{14}$$

Eq. (14) isn't a particularly useful result since as we shall see later there is a much stronger bound in terms of $H(E)$. The bound of the following theorem is the main result in the theory of unconditionally secure authentication codes.

Theorem 3. (Authentication Channel Capacity)

$$\log P_d \geq H(MES) - H(E) - H(M) \tag{15}$$

Comment: The proof of Theorem 3 is similar in style to many channel capacity proofs in information theory in which multiple summations of the entropies for discrete events are manipulated, usually by interchanging the orders of summation to reexpress and regroup terms in a form where Jensen's inequality* can be invoked. Such proofs seem to be unavoidably tedious and long. The reader is referred to [23] where the (tedious and long) proof of Theorem 3 is given. Massey gives an elegant (and short) proof in his "Contemporary Cryptology: An Introduction" in this volume of a bound on the probability of an impersonation deception being successful. He uses a slightly different notation than is used here, but the reader should have no difficulty in converting results from either chapter into the form used in the other.

A variety of useful equivalent expressions can be derived from Eq. (15) using simple identities from information theory, for the cases of authentication either with or without secrecy. We illustrate the technique in Theorem 4 for the case of authentication with secrecy; that is, the opponent does not know the state of the source observed by the transmitter. This, of course, only matters when the opponent elects to substitute messages rather than to impersonate the transmitter.

---

*If $q(x)$ is a convex function on an interval (a, b) and $x_1, x_2, \ldots, x_n$ arbitrary real numbers $a < x_i < b$, and if $w_1, w_2, \ldots, w_n$ are positive numbers with $\Sigma w_i = 1$, then

$$g\left[\sum_{i=1}^{n} w_i x_i\right] \leq \sum_{i=1}^{n} w_i g(x_i)$$

Theorem 4. $H(MES) - H(E) - H(M)$ is equivalent to any of the following eight entropy expressions:

Equivalent Form	Equation
$H(M \mid ES) + H(S) - H(M)$	(16)
$H(E \mid MS) - H(E) + H(MS) - H(M)$	(17)
or	
$H(E \mid MS) - H(E) + H(S \mid M)$	(18)
$H(E \mid M) - H(E)$	(19)
or	
$H(M \mid E) - H(M)$	(20)
$H(ME \mid S) + H(S) - H(E) - H(M)$	(21)
$H(MS \mid E) - H(M)$	(22)
$H(ES \mid M) - H(E)$	(23)

Using the results of Theorem 4 it is possible to derive some (generally) weaker but enlightening lower bounds for the insecurity of unconditionally secure authentication codes. We first note that the total effective equivocation to the opponent playing the substitution game but without knowledge of the source state, that is, authentication with secrecy is no greater than $H(E \mid M)$ and as remarked above, the opponent's total effective equivocation if he knows the source state, that is, authentication without secrecy, is at most $H(E \mid MS)$.

Theorem 5. For authentication with secrecy

$$\log v_G \geq -\frac{1}{2} H(E) \tag{24}$$

while for authentication without secrecy

$$\log v_G \geq -\frac{1}{2}\{H(E) - H(MS) + H(M)\} = -\frac{1}{2}\{H(E) - H(S \mid M)\} \tag{25}$$

The results contained in Theorem 5 are so important to the theory of authentication that it is perhaps worthwhile to indicate how they are obtained. For authentication with secrecy

$$\log v_G \geq \min\{\log v_I, -H(E \mid M)\} \tag{26}$$

while for authentication without secrecy

$$\log v_G \geq \min\{\log v_I, -H(E \mid MS)\} \tag{27}$$

In either Eq. (26) or Eq. (27) the bounds derived in Theorems 3 and 4 on the value of the impersonation game can be substituted, since the opponent's impersonation strategy is independent of whether he plays substitution with or without secrecy. Replacing the minimum on the right-hand side of the inequality by the average of the two bracketed terms either weakens the inequality if the terms are not identical or leaves it unaffected if they are. Therefore for authentication with secrecy, replacing v_I with the bound in Eq. (19) in Eq. (26) we get

$$\log v_G \geq \frac{1}{2}\{H(E \mid M) - H(E) - H(E \mid M)\} = -\frac{1}{2}H(E)$$

and similarly by replacing v_I with the bounds in Eq. (17) or Eq. (18) in Eq. (27) we get

$$\log v_G \geq \frac{1}{2}\{H(E \mid (MS) - H(E) + H(MS) - H(M) - H(\mathrm{E} \mid (MS)\}$$

$$= -\frac{1}{2}\{H(E) - H(MS) + H(M)\}$$

or

$$\log v_G \geq \frac{1}{2}\{H(E \mid MS) - H(E) + H(S \mid M) - H(\mathrm{E} \mid MS)\}$$

$$= \frac{1}{2}\{H(E) - H(S \mid M)\}$$

Corollary

$$P_d = v_G \geq \frac{1}{\sqrt{|\mathcal{E}|}} \tag{28}$$

As was mentioned above, the bound in Eq. (28) was first derived by Gilbert, McWilliams, and Sloan [8] under slightly more restrictive conditions and proven directly in the same generality used here by Brickell and Simmons in [1]. The more useful, and informative lower bounds on the insecurity of unconditionally secure authentication schemes are Eqs. (24) and (25) for authentication with and without secrecy, respectively.

The inescapable conclusion that must be drawn from the theoretical results just given is that a large number of encoding rules must be available in any unconditionally secure authentication code—on the order of $1/P_d^2$ at least—to realize a security of P_d and that these encoding rules must also have a well-defined structural interdependence to insure that the conditional entropy conditions be met to make this level of security available at all. As we have seen from the example at the end of the previous section, neither of these comments applies to computationally secure (and hence also to provably secure) authentication schemes.

5 THE PRACTICE OF AUTHENTICATION

We have already described two of the more straightforward applications of authentication to the problem of enabling a receiver on a communications channel to verify that in all probability a message received over the channel was originated by the legitimate (authorized) transmitter and that it hasn't been tampered with subsequently. In the one case an authenticator, known to both the transmitter and to the receiver, was appended to the information to be authenticated and then encrypted to produce a cipher which served to provide both secrecy and authentication. In the other case the information

was partitioned into blocks which were then block-chain-encrypted to produce a MAC that was appended to the text. In this case only authentication was achieved, unless the extended message was then superencrypted to yield a cipher that would provide secrecy.

The most important point about these two schemes, though, is that if in either case a single-key cryptoalgorithm is used for the encryption and decryption functions, then the transmitter and receiver must be mutually trusting and trustworthy. This is unavoidable since for a single-key cryptoalgorithm the transmitter and receiver must both possess the same cryptographic keying variable and hence each be able to interchangeably do anything the other can. The more challenging case is when the transmitter and receiver cannot be assumed to trust each other, or even the more extreme case of when they must both be assumed to be untrustworthy and deceitful, that is, that if either can get away with cheating, they will cheat. The problem in either of these cases is further complicated by what action the transmitter or receiver will take if they conclude the other has cheated. If the action is unilateral, that is, if no third party need be convinced that cheating has occurred, the demands on the authentication scheme are entirely different, and much simpler, than if the authentication must be logically compelling to a third party or arbiter.

In a companion chapter to this one in this volume entitled "How to Insure That Data Acquired to Verify Treaty Compliance Are Trustworthy," a detailed account is given of an authentication scheme serving mutually distrusting and deceitful parties in which deceits must be logically demonstrable to third parties who themselves may wish to deceive either the transmitter or receiver, host and monitor, respectively, in the terminology of treaty verification schemes. This application of authentication is, to this author's knowledge, the most intricate and convoluted, and hence the most interesting, that has been made to date.

Rather than repeat, even in abbreviated form, the details of that application here, we describe instead a common business/commercial application of authentication with similar requirements, but with a quite different solution. Much more efficient, and hence applicable, techniques are described in the chapters "Digital Signatures" by Mitchell, Piper, and Wild and "Smart Card: A Standarized Security Device Dedicated to Public Cryptology" by Guillou, Quisquater, and Ugon, however the simplicity of the protocol described here recommends it as an illustrative example for an important area of application.

Consider the needs of the various participants in a credit card transaction at the point of sale. The merchant (or automated teller machine [ATM], etc.) will give up something of real value, that is, an item of merchandise, money, services, etc. if the transaction is completed in exchange for a record (information) evidencing the credit due from the customer. The merchant must be able to satisfy himself as to the customer's identity, the validity of his claimed account, and perhaps of the level of credit in that account as well as the integrity of the record he holds establishing the credit due to him. In other words, the merchant must suspect that the customer isn't who he claims to be, that the credit card (credential) presented is either a forgery or else that even if it is genuine that it doesn't belong to that customer, and that the customer will later disavow having made the purchase or if he does admit that a purchase was made, will aver that it was for a lesser amount or was made at a later date, etc.

The customer, on the other hand, needs to be able to satisfy himself that the record of the transaction is accurate, that is, it is for the proper amount and shows the

correct date, etc.; it can only be presented for collection once; it can't be altered to increase the customer's liability; and the merchant can't later, as a result of any number of transactions with the customer, impersonate him to fraudulently make purchases, withdrawals, etc. on his (the customer's) account. In other words, the customer must suspect that the merchant will attempt to alter the record of the transaction or submit it for collection more than once, or to alter the dates of transactions, etc. He must also suspect that the merchant, or his agents, will accumulate records of his past transactions with the object of impersonating him to other merchants to collect goods, services, or monies in his (the customer's) name.

The bottom line is that a normal credit transaction provides the classic paradigm for an information exchange between mutually distrustful and untrustworthy participants. Since the resolution of a dispute over the validity of a claimed transaction will necessarily involve third parties, that is, a bank, a court, or an arbiter, etc.; a satisfactory solution should address all of the concerns of all of the participants and produce a record that can be logically evaluated to assign liability to the party that should, in all probability, be held liable.

We discuss how authentication of information can be used to satisfy these needs in several stages. First, we show how a customer can be identified in a way that can later be verified (as to the correctness of the identification) by third parties who were not parties to the transaction in real time. There are only two ways an individual can be identified, irrespective of whether the identification is made by a manned or an automated facility; these two ways depend on whether intrinsic or extrinsic identifying information is used.

Intrinsic information are the physical identifiers of the individual; fingerprints, voiceprint, retinal prints, signature and/or signature dynamics, hand geometry, physical appearance (usually as compared to a photograph), height, weight, distinguishing marks, etc. Intrinsic identifiers are inextricably linked to the person they identify, so that to successfully impersonate someone else, an imposter must succeed in mimicking their identifiers.

Extrinsic information is something that the right individual is known to know or be able to do, but which an imposter probably won't know or be able to do; for example, computer passwords, telephone credit card numbers, Swiss bank account numbers, personal identification numbers (PINs), signs or countersigns for sentries, etc. are all examples of extrinsic identifiers. Extrinisic identifiers are not linked to the person being identified, hence it is only necessary for a would-be imposter to learn what the identifiers are to undetectably impersonate the legitimate owner. Consequently, there is the logical problem that this type of identifying information, once exhibited by the owner to prove his identity, is potentially compromised and could be used by anyone, in particular the merchant or his agents in the present discussion, to undetectably impersonate the legitimate owner.

To have an acceptably secure identification scheme based on extrinsic identifiers, one needs to devise a protocol that will allow an individual to "prove" that he knows the secret piece(s) of information, whose possession is equated with his identity, without revealing anything about the information itself which could aid a would-be cheater to impersonate him. Several investigators have proposed identification schemes to accomplish this [7, 9, 19, 28] that depend on interactive-proof schemes, often referred to as zero-knowledge proofs, in which the individual responds to a series of queries in a way that the legitimate user could, but which an imposter (probably) could not.

Later we will describe a simple identity verification scheme that uses a public authentication channel to validate the public part of a private authentication channel belonging to the individual who wishes to prove his identity. He can then prove that he is who he claims to be by being able to generate authenticated messages in the private authentication channel which is possible (in probability) only if he knows the secret part of that private channel. The public and the private channels can be completely independent and can even be based on different authentication algorithms, or they can both be of the same type. This scheme also provides certified receipts for transactions whose legitimacy can later be verified by impartial arbiters who were not involved in the transaction itself.

First, however, we describe an identity verification scheme that uses an authentication channel to validate intrinsic identifiers for the customer to the merchant. Since we are considering point-of-sale protocols, we assume that the identification must be made on the basis of information supplied to the merchant by the customer. For practical reasons, we impose the additional restriction that very little communication to remote locations be required in advance of, and none at the time that the customer's identity is to be verified. Finally, there is the practical constraint, arising from enormous numbers of merchants with whom a customer may unexpectedly wish to do business as well as the number of persons and the turnover in the staff having access to the customer identification equipment in the store, that while the merchant's information must be protected from alteration or substitution, it is generally not possible to guarantee its secrecy. In other words, a pure authentication channel distinct and separate from a secrecy channel is what is involved.

In either case, that is, for either intrinsic or extrinsic identifying information, it is assumed that there exists some party or facility that is unconditionally trusted by both the customer(s) and the merchant(s); the issuer of validated (signed) identification credentials. This could be, depending on the application, a government agency, a credit card center or financial institution, a military command center, a centralized computer facility, etc. The trusted issuer first establishes a public, that is, desensitized, authentication channel to which he retains the (secret) authenticating function.

For simplicity, throughout the discussion in this section we will use the well-known authentication channel based on the RSA cryptoalgorithm for both the public (issuer) and the private (customer) channels, although authentication channels based on any other algorithm would serve equally well. As described in the body of the chapter, the issuer would choose a pair of primes p and q by the same standards used to compute a good RSA modulus, that is, so that it is computationally infeasible for anyone to factor n, and then calculate a matching pair of encryption/decryption exponents, e and d, and publish n and d as the public key. The issuer would keep e (and equivalently the factors p and q) secret; in fact, the security of the system against fraudulent claims of validated identity is no better than the quality of protection given to e by the issuer.

The solution to the problem of identifying the customer is now so simple as to be almost anticlimatic. The central facility, the issuer, is entrusted to first establish the identity and accuracy of the associated information for each potential customer to whatever degree of certainty is deemed necessary and then to generate an authenticated ID record that would be given to the customer as his identifying credential. This record will comprise intrinsic information, that is, personal attributes (photograph, fingerprints, hand geometry, voiceprint, retinal prints, signature, etc.) encrypted using the encrypt key of a two-key system, modulus n and exponent e in our example, along with

extrinsic identifiers such as name, social security number, etc. User confidence is totally dependent on the issuer keeping the encrypt key secret, that is, maintaining the security of the authentication channel.

Depending on the application, this may require a two-man (or more) rule for access, or k of n shared key reconstructions, at the issuer's facility so that an improbably high level of collusion would have to occur for subversion to be possible. The decrypt key would be delivered as an authenticated, but not necessarily secret, message to all of the merchants, ATMs, etc. who would thereafter have to protect the integrity but not the privacy of the key. When, at some later time, a customer appears at a point of sale with a claimed identity, he would present the cipher record (credential) in his possession and permit his individual attributes to be reread by the merchant's point-of-sale equipment.

Using the decrypt key, the merchant would first decrypt the ID cipher and verify the authenticity of the cipher by the presence of the expected redundant information. He would next check for a suitable agreement between the individual attributes just measured and the decrypted information (intrinsic attributes) contained in the authenticated message. If an acceptable match is achieved, the identity of the customer will have been confirmed since the cipher could only have been generated using the secret encrypt key held by the enrollment station that was responsible for verifying the identity of the customer in the first place and the supposition is that an imposter could not adequately mimic the attributes of someone else to be accepted as them. The only advance communication required between the site and central facility is the authenticated (but not necessarily secret) exchange of the decrypt key to set up the public authentication channel.

The other channel of communication is the public one of the user bringing his own ID cipher to the site. Since authentication is possible in some two-key cryptosystems, that is, those in which there is a one-to-one mapping between plaintext and ciphertext spaces as in the RSA cryptoalgorithm, where the sender's key is known to be secure, the site can be certain (to the same level of confidence as the two-key cryptosystem is cryptosecure) that ID records it has received are authentic, that is, that they were issued and signed by the central issuing authority. Then, to the degree that the information in the ID records can identify an individual, the merchant can be confident of the identification. No communication with the central facility is required at the time that the individual is identified, and more importantly, no files of identifying information for possible customers need be transmitted to and stored at the merchant's place of business.

It is worth noting that the first reported application of public key cryptography techniques (fielded by the Sandia National Laboratories in 1978), made use of the authentication channel based on the RSA cryptoalgorithm, exactly as described here, to create trusted credentials that users could carry with them and present at the time they requested access, in this case to the very sensitive Zero Power Plutonium Reactor at Idaho Falls, Idaho [14,22]. The authenticated information in the credential included physical descriptors for the individual being identified as well as the details of the nature, type, duration, etc. of the access authorized. The object of this scheme was to make it possible for each user to carry with him what would have effectively been his entry in a trusted directory (a trusted credential in this case) at the remote site had a directory of potential users been feasible to compile, update, and protect at the remote site. The user-supplied information could be authenticated by the verifier at the site but

would be of no assistance to anyone wishing to produce a fraudulent credential. In this particular application, the identification information was intrinsic to the user (hand geometry, body weight, etc.), however, in other applications [11] the same basic technique has been used with extrinsic information in a manner similar to the protocol to be described here.

The use of an authentication channel to validate to public receivers intrinsic identifying information "signed" by a trusted issuer has an enormous range of applications beyond the simple customer identification protocol just described. For example, every physical object at some level of fineness in detail is unique, that is, everything has a "fingerprint." Two pieces of paper of the same dimensions cut from the same roll may appear identical, but at a sufficient magnification, the cotton fibers lie in random and unreproducible three-dimensional patterns. The pattern at any selected location can be thought of as a fingerprint for that piece of paper. The important points are that every piece of paper has such a fingerprint by virtue of its existence, but that given a particular fingerprint it is impossible to produce another piece of paper with the same fingerprint. Unfortunately, it is not feasible to "read" the fiber layup of a piece of paper in any practical way to fingerprint a document. Any specific fine structure that can be placed deliberately in genuine objects, such as detailed engraving, etc., can also be cloned or placed in counterfeit objects. If, however, an object (a piece of paper, a physical container, a very large-scale integration [VLSI] chip, etc.) can be caused to have *a unique random fingerprint that is feasible to read but infeasible to clone*, that is, to reproduce, then we can use the protocol described above to insure that the object cannot be counterfeited or substituted for.

The issuing authority responsible for either producing the genuine articles, or for signing genuine articles for authenticity, reads and digitizes the unique intrinsic information (fingerprint) for each object, and then computes the matching authenticator using the secret encryption key for a public authentication channel. This authenticator is attached to the object, perhaps simply printed on it. Later, anyone can verify whether an article is genuine or not by rereading the fingerprint from the object and matching it against the authenticated fingerprint, either openly contained in the message if an authentication without secrecy channel was used, or else obtained by decrypting the authenticated cipher with the public key for the authentication channel. The physical fingerprint may have been degraded by wear and tear, improper location or alignment in reading, etc., however, the digital description of the fingerprint as it was originally read by the issuing authority will be recovered exactly.

The match between the two need not be perfect, and in all practical schemes will be far from perfect, however, if there is sufficient information content to the fingerprint large discrepancies can be tolerated with a very high confidence that counterfeit items will be detected and that genuine ones will not be rejected. The point is that a pure authentication channel makes it possible to transfer confidence in the identity or integrity of any piece of information from the issuing authority to publicly exposed remote locations. There are only two ways that undetectable counterfeits could be created in such schemes: Either the fingerprint of a genuine article would have to be cloned to match the genuine authenticator computed by the issuing authority or else the authentication channel would have to be cryptanalyzed to make it possible to "sign" the fingerprints read from bogus articles.

Returning to the point-of-sale problem, identifying the customer and verifying his account and credit rating, etc. satisfy some of the merchant's concerns, but not all of

them. He also needs a record of the transaction that can be validated by impartial third parties in the event of a dispute; in other words, he needs the equivalent of a legal signature. If the transaction is occurring at a manned location, once the customer has been identified an actual signature will suffice. This is the way credit card transactions are handled presently. If, however, the transaction is either remote (conducted from a terminal or over a telephone, for example) or at an unmanned site such as an ATM or ticket vendor, etc. then the protocol must provide means for the customer to "sign" the charge slip, if significant liabilities are to be acceptable.

Conversely, the customer must end up with a copy of the record of the transaction that he can produce later to verify the amount, the date, the items purchased, the taxes paid, etc. for his own protection. All of these objectives can be satisfied by making a slight modification to the protocol with which credentials are created and employed and an addition to the information authenticated by the central issuing authority in the customer's credential. It should be pointed out that although the identity verification scheme just described has been successfully applied to a variety of very sensitive access control problems, such as the Zero Power Plutonium Reactor mentioned above or to the nuclear reactor fuel rod reprocessing plant at Savannah River, Georgia, etc., the extension of these techniques to provide verifiable and unforgeable transaction receipts has not yet been fielded. We include a description in this chapter, even though these techniques are not yet strictly "practice," simply because they complement what is practice in a very logical way and illustrate the underlying principles clearly. The scheme described here is a simplified version of a general identification scheme proposed by Simmons [29] in 1989.

To make verifiable receipts possible, the central issuing authority in addition to setting up a public authentication channel as before, would also choose a polyrandom hashing function f that maps arbitrary strings of symbols to the range $[0, n)$. This function may well have unlimited life, and at the very worst, would be changed only after very long periods of use. By polyrandom we mean that f cannot be distinguished from a truly random function by any polynomially bounded computation. Many strong, single-key, cryptographic functions, such as the DES when used with a publicly known key in a block-chain encryption mode, appear to adequately approximate this condition. f is also made public by the issuer.

At the time the credential is created for customer i, in addition to checking his identity and verifying the correctness of all of the other information and intrinsic identifiers, etc., the issuer would require the customer to provide to him the public part of a private (i.e., belonging to the customer) authentication channel. By assumption (here) this would be a suitable RSA modulus, n_i, and a decryption exponent d_i; the matching encryption exponent e_i (and equivalently the factorization of n_i) the customer would, of course, keep secret. This string of information, I_i, must also include redundant information, such as message format, fixed fields of symbols common to all identifiers, etc. The issuer calculates

$$s_i \equiv m_i^d \pmod{n}$$

where

$$m_i = f(I_i)$$

and gives the credential (I_i, s_i) to customer i. No part of this credential need be kept secret. However, the customer must keep secret his private encryption exponent, e_i,

corresponding to d_i. His security against impersonation is dependent on his protecting e_i, since his proof of identity in the scheme is equated to knowing e_i.

The public part of the issuer's authentication channel is the modulus n and decryption exponent d, the hashing function f, and a knowledge of the redundant information present in all of the I_i, which must be sufficient to prevent a forward search cryptanalytic attack [8] on the function f. This is a type of cryptanalytic attack on a two-key secrecy channel that has no counterpart in single-key cryptography that exploits the fact that the encryption key is publicly known to allow a would-be eavesdropper to precompute a table of likely plaintext/ciphertext pairs. Even though he cannot decrypt ciphers not appearing in the resulting table, he can replace any cipher that does appear with its matching plaintext. In the present case, this means that anyone wishing to fraudulently validate an identity could calculate $s_j^d = m_j$ for randomly chosen signatures s_j in the hopes of obtaining a hit with $f(I)$ for some usable I. By making I contain sufficient redundant information, the probability of success of this sort of attack can be made as small as desired.

When customer i wishes to prove his identity to vendor j, say to gain access to a restricted facility or to log onto a computer or to withdraw money from an ATM etc., he initiates the exchange by identifying himself to j by presenting his identification credential, (I_i, s_i) concatenated with a string of symbols, u_i, that describes or identifies the transaction i is requesting; u_i could be the date, the amount of the desired withdrawal, etc. j replies with his identification credential (I_j, s_j) concatenated with a string of symbols that describe the transaction from his standpoint; terminal ID, transaction number, confirmation of withdrawal amount, etc. Both i and j form the concatenation of u_i and u_j, $u = u_i, u_j$, and calculate the function $f(u)$ of the resulting string,

$$z = f(u)$$

In addition, j also calculates $f(I_i) = m_i$ and s_i^d. j accepts the credential (I_i, s_i) as valid if and only if

$$f(I_i) = s_i^d \pmod{n}$$

i can carry out a similar calculation to verify that the credential (I_j, s_j) was indeed issued by the issuing authority, however, this is unnecessary in the case being considered here, since i will get immediate delivery of goods or services from the merchant in exchange for a promise (on his part) to pay later. At this point in the protocol, j is confident that the customer identified in I_i can authenticate messages using the private authentication channel described therein, in other words, that customer i knows e_i matching the exponent d_i authenticated by the issuer. If the customer is who he claims to be, he can calculate

$$t_i = z^{e_i} \pmod{n}$$

using his private exponent e_i, which he communicates to j.

The merchant j calculates

$$t_j = z^{e_j} \pmod{n_i}$$

and sends t_j to the customer. Note that in both cases z is being used effectively as a one-time key, indeterminate to both i and j because of the polyrandom nature of f, to

permit i to give to j an encrypted function of z in a form that will permit j to satisfy himself that i had to know e_i without providing any information whatsoever about e_i. j knows the identity claimed by i from I_i, which he accepts as valid if and only if the following identity is satisfied:

$$t_i^{d_i} = z \pmod{n_i} \tag{29}$$

If the person seeking to be recognized as customer i really is who he claims to be, that is, if he knows e_i, then in order for him to be able to impersonate i, that is, to cause Eq. (32) to be satisfied, he would have to be able to find a number x such that

$$x^{d_i} = z \pmod{n_i} \tag{30}$$

n_i or d_i are values signed by the issuer in I_i with only the authorized customer knowing e_i or equivalently the factorization of n_i. z is a pseudorandom number jointly determined by i and by j. Solving Eq. (29) without knowing e_i is equivalent to breaking the RSA cryptoalgorithm from ciphertext alone.

j keeps the 4-tuple $((I_i, s_i) : u, t_i)$ as his certified receipt for the transaction while the customer keeps the 4-tuple $((I_j, s_j) : u, t_j)$. Anyone, using only publicly available information, that is, n, d, and f, can later verify that the merchant's 4-tuple satisfies Eq. (29) which validates the transaction description and verifies that it was signed, that is, endorsed, by customer i. The customer's 4-tuple can be validated in a similar manner.

6 CONCLUSIONS

The problem of how to make it possible to decide whether a piece of information is what it purports to be or not is a much broader and more difficult question than this survey suggests. However, the method of solution remains the same, even in the most general case; namely, that one or more of the participants in an information-based system (which may include physically secure devices) perform operations on the information, the correctness of which can be verified by others, but that some of the other participants (probably) can't duplicate. What makes the problem so complex is the seemingly unlimited number of motives and methods for cheating in information-based systems that involve manipulating the information as a means.

Since this survey is a chapter in a volume on cryptology, we have emphasized those aspects of authentication that use cryptographic transformations (encryption and decryption) as the operation that persons in the know can do and others (probably) cannot. As was pointed out in the discussion of unconditionally secure authentication codes, though, there are alternative authentication operations available. The more difficult questions concerning the integrity (authenticity) of information in systems in which the participants may join forces and pool their privileged information to conspire to defraud others, lead into the area of provably secure protocols, which incidentally draws very heavily on cryptographic techniques for their realization; such questions were considered to be beyond the scope of this chapter.

Probably the most important development in authentication in recent years, growing out of the discovery of public key cryptography, is the recognition that the authentication channel can be separated from the secrecy channel. This made desensitized

authentication possible, which removed the necessity that the originator (of the authenticated information) and the receiver trust each other unconditionally. This, in turn, was a first step in the evolution of authentication schemes and protocols in which no one need trust anyone (specifically), but only need trust that some number of the participants (but not specific ones) will perform in a trustworthy manner to be able to trust the integrity of the information itself. When viewed in this setting, cryptography is seen to be a much wider, and more vital, subject than in its classic setting of merely concealing information from outsiders (secrecy) or of preventing outsiders from spoofing the insiders (authentication). It is hoped that this survey has conveyed some notion of the breadth of this subject.

References

[1] E. F. Brickell, "A few results in message authentication," presented at 15th Southeastern Conf. on Combinatorics, Graph Theory and Computing, (Baton Rouge, LA), March 5–8, 1984, in *Congressus Numerantium*; vol. 43, pp. 141–154, Dec. 1984.

[2] E. F. Brickell and D. R. Stinson, "Authentication codes with multiple arbiters," extended abstract in *Lecture Notes in Computer Science 330; Advances in Cryptology: Proc. Eurocrypt '88,* C. G. Günther, Ed., Davos, Switzerland, May 25–27, 1988, pp. 51–55. Berlin: Springer-Verlag, 1988.

[3] Department of the Treasury, "Electronic Funds and Securities Transfer Policy—Message Authentication," Directive signed by Donald T. Regan, Secretary of the Treasury, Aug. 16, 1984.

[4] M. De Soete, "Some constructions for authentication—secrecy codes," in *Lecture Notes in Computer Science 330; Advances in Cryptology: Proc. Eurocrypt '88,* C. G. Günther, Ed., Davos, Switzerland, May 25–27, 1988, pp. 57–75. Berlin: Springer-Verlag, 1988.

[5] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[6] T. El Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inform. Theory*, vol. IT-31, no. 4, pp. 469–472, July 1985.

[7] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Lecture Notes in Computer Science 263; Advances in Cryptology: Proc. Crypto '86,* A. M. Odlyzko, Ed., Santa Barbara, CA, Aug. 11–15, 1986, pp. 186–194. Berlin: Springer-Verlag, 1987.

[8] E. N. Gilbert, F. J. MacWilliams, and N. J. A. Sloane, "Codes which detect deception," *Bell Syst. Tech. J.*, vol. 53, no. 3, pp. 405–424, March 1974.

[9] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but the validity of the assertion and the methodology of cryptographic protocol design," in *Proc. 27th Annu. Symp. Foundations Comput. Sci.*, Toronto, Canada, Oct. 27–29, 1986, pp. 174–187. Los Angeles, CA: IEEE Computer Society, 1987.

[10] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Jour. Computing*, vol. 17, no. 2, pp. 281–308, April 1988.

[11] C. L. Henderson and A. M. Fine, "Motion, intrusion and tamper detection for surveillance and containment," *Sandia National Laboratories Rept. SAND79-0792*, March 1980; also published by the International Safeguards Project Office for the International Atomic Energy Agency (IAEA) as *ISPO Report No. 91*, 1980.

[12] J. L. Massey, "Cryptography—a selective survey," presented at International Tirrenia Workshop on Digital Communications, Tirrenia, Italy, Sept. 1–6, 1985, in *Alta Frequenza*, vol. 55, no. 1, pp. 4–11, Jan.–Feb. 1986; also published in *Digital Communications*, E. Biglieri and G. Prati, Eds., pp. 3–25. Amsterdam: Elsevier, 1986.

[13] A. C. Meisenbach, *Acme Commodity and Phrase Code*, Acme Code Co., San Francisco, CA, 1923.

[14] P. D. Merillat, "Secure stand-alone positive personnel identity verification system (SSA-PPIV)," *Sandia National Laboratories Tech. Rept. SAND79-0070*, March 1979.

[15] C. H. Meyer and S. M. Matyas, *Cryptography: A New Dimension in Computer Data Security*, New York: Wiley, 1982.

[16] M. O. Rabin, "Digitized signatures and public-key functions as intractable as factorization," *Massachusetts Institute of Technology Laboratory for Computer Science, Tech. Rept. LCS/TR-212*, 1979.

[17] R. A. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. Assn. Comput. Mach.*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[18] P. Schoebi, "Perfect authentication systems for data sources with arbitrary statistics," presented at Eurocrypt '86, Linköping, Sweden, May 20–22, 1986.

[19] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Lecture Notes in Computer Science 196; Advances in Cryptology: Proc. Crypto '84*, G. R. Blakley and D. Chaum, Eds., Santa Barbara, CA, Aug. 19–22, 1984, pp. 47–53. Berlin: Springer-Verlag, 1985.

[20] G. J. Simmons, "Message authentication without secrecy," in *Secure Communications and Asymmetric Cryptosystems* (AAAS Selected Symposia Series), G. J. Simmons, Ed., pp. 105–139. Boulder, CO: Westview Press, 1982.

[21] G. J. Simmons, "Verification of treaty compliance—revisted," in *Proc. IEEE Computer Soc. 1983 Symp. on Security and Privacy*, G. R. Blakley and D. Denning, Eds., Oakland, CA, April 25–27, 1983, pp. 61–66. Los Angeles: IEEE Computer Society Press, 1983.

[22] G. J. Simmons, "A system for verifying user identity and authorization at the point-of-sale or access," *Cryptologia*, vol. 8, no. 1, pp. 1–21, Jan. 1984.

[23] G. J. Simmons, "Authentication theory/coding theory," in *Lecture Notes in Computer Science 196; Advances in Cryptology: Proc. Crypto '84*, G. R. Blakley and D. Chaum, Eds., Santa Barbara, CA, Aug. 19–22, 1984, pp. 411–431. Berlin: Springer-Verlag, 1985.

[24] G. J. Simmons, "The practice of authentication," in *Lecture Notes in Computer Science 219; Advances in Cryptology: Proc. Eurocrypt '85*, F. Pichler, Ed., Linz, Austria, April 1985, pp. 261–272. Berlin: Springer-Verlag, 1986.

[25] G. J. Simmons, "Authentication codes that permit arbitration," presented at 18th Southeastern Conf. on Combinatorics, Graph Theory and Computing, Boca Raton, FL, Feb. 23–27, 1987, in *Congressus Numerantium*, vol. 59, pp. 275–290, March 1988.

[26] G. J. Simmons, "Message authentication with arbitration of transmitter/receiver disputes," in *Lecture Notes in Computer Science 304; Advances in Cryptology: Proc. Eurocrypt '87,* D. Chaum and W. L. Price, Eds., Amsterdam, The Netherlands, April 13–15, 1987, pp. 151–165. Berlin: Springer-Verlag, 1988.

[27] G. J. Simmons, "A natural taxonomy for digital information authentication schemes," in *Lecture Notes in Computer Science 293; Advances in Cryptology: Proceedings of Crypto '87,* C. Pomerance, Ed., Santa Barbara, CA, Aug. 16–20, 1987, pp. 269–288. Berlin: Springer-Verlag, 1988.

[28] G. J. Simmons, "An impersonation-proof identity verification scheme," in *Lecture Notes in Computer Science 293; Advances in Cryptology: Proceedings of Crypto '87,* C. Pomerance, Ed., Santa Barbara, CA, Aug. 16–20, 1987, pp. 211–215. Berlin: Springer-Verlag, 1988.

[29] G. J. Simmons, "A protocol to provide verifiable proof of identity and unforgeable certified receipts," *IEEE Jour. Selected Areas Commun. Special Issue on Secure Communications*, vol. 7, no. 4, pp. 435–447, May 1989.

[30] G. J. Simmons, "How good is the Acme code?" *Supplementary volume to the Proceedings of the 4th Joint Swedish-Soviet International Workshop on Information Theory*, Visby, Sweden, Aug. 27–Sept. 1, 1989, pp. 24–30, 1989.

[31] G. J. Simmons, "A Cartesian product construction for authentication codes that permit arbitration," *Jour. Cryptology*, vol. 2, no. 2, pp. 77–104, 1990.

[32] D. R. Stinson, "Some constructions and bounds for authentication codes," in *Lecture Notes in Computer Science 263; Advances in Cryptology: Proc. Crypto '86,* A. M. Odlyzko, Ed., Santa Barbara, CA, Aug. 11–15, 1986, pp. 418–425. Berlin: Springer-Verlag, 1987.

[33] D. R. Stinson, "A construction for authentication secrecy codes from certain combinatorial designs," in *Lecture Notes in Computer Science 293; Advances in Cryptology: Proceedings of Crypto '87,* C. Pomerance, Ed., Santa Barbara, CA, Aug. 16–20, 1987, pp. 355–366, Berlin: Springer-Verlag, 1988. Also appears in *J. Cryptology,* vol. 1, no. 2, pp. 119–127, 1988.

[34] H. C. Williams, "A modification of the RSA public-key encryption procedure," *IEEE Trans. Inform. Theory*, vol. IT-26, no. 6, pp. 726–729, Nov. 1980.