Τσακαλίδης Γ. Ευάγγελος
Μεταπτυχιακός Φοιτητής

tsakal@uom.gr

Μάθημα : Τεχνολογίες Τηλεπικοινωνιών & Δικτύων
Καθηγητής : Ανδρέας Σ. Πομπόρτσης
Εργασία : 5η
Θέμα : «Ασφάλεια Επικοινωνιακών Συστημάτων : Αλγόριθμοι Κρυπτογράφισης - Ψηφιακή Υπογραφή»
(Βιβλιογραφική Αναζήτηση στο www)

Εισαγωγή

Τι είναι η κρυπτογραφία;

Η *κρυπτογραφία*, για τους περισσότερους ανθρώπους, αφορά στη μη δημοσιοποίηση των επικοινωνιών. Στην πραγματικότητα όμως η προστασία των ευαίσθητων επικοινωνιών είναι το σημείο στο οποίο με έμφαση εφαρμόζεται η κρυπτογραφία, κατά το μεγαλύτερο μέρος της ιστορικής της εξέλιξης. Ωστόσο, όπως θα δούμε είναι μόνο ένα μέρος της σημερινής κρυπτογραφίας.

Απόκρυψη είναι ο μετασχηματισμός των δεδομένων σε μια μη αναγνώσιμη μορφή. Σκοπός της είναι να εξασφαλίσει μυστικότητα, με το να κρατά τις πληροφορίες κρυμμένες από οποιονδήποτε, τον οποίον δεν αφορούν, ακόμη και από όποιον μπορεί να δει τα κρυμμένα δεδομένα. *Αποκάλυψη / Αποκωδικοποίηση* είναι το αντίθετο της απόκρυψης : είναι ο μετασχηματισμός των κρυμμένων / κωδικοποιημένων δεδομένων σε κάποια κατανοητή μορφή.

Η Απόκρυψη και η Αποκάλυψη απαιτούν την χρήση κάποιων μυστικών πληροφοριών, οι οποίες συνήθως αναφέρονται ως *κλειδί*. Ανάλογα με τον μηχανισμό κρυπτογράφησης, που χρησιμοποιείται, το ίδιο κλειδί μπορεί να χρησιμοποιηθεί για απόκρυψη και αποκάλυψη, ενώ σε άλλους μηχανισμούς τα κλειδιά που χρησιμοποιούνται για τα στάδια της απόκρυψης και αποκάλυψης είναι διαφορετικά.

Η σημερινή κρυπτογραφία είναι κάτι παραπάνω από απόρρητη τήρηση εγγράφων, κάτι παραπάνω από απόκρυψη και αποκάλυψη. Η *αυθεντικότητα* είναι τόσο βασικό μέρος της ζωής μας όσο και η μυστικότητα. Χρησιμοποιούμε την αυθεντικότητα καθημερινά στη ζωή μας, για παράδειγμα όταν υπογράφουμε σε κάποια έγγραφα ή δίνουμε εντολή για πληρωμή λογαριασμών χωρίς την παρουσία μας. Εφ' όσον κινούμαστε σε έναν κόσμο, που οι αποφάσεις μας και οι συμφωνίες μας διαβιβάζονται ηλεκτρονικά, προκύπτει η ανάγκη της πιστής αντιπροσώπευσής μας σ' αυτές τις διαδικασίες.

Η κρυπτογραφία παρέχει μηχανισμούς για τέτοιες διαδικασίες. Η *ψηφιακή υπογραφή* συνάπτει ένα έγγραφο στον επεξεργαστή ενός ιδιαίτερου κλειδιού, ενώ η *ψηφιακή χρονοσήμανση* επισυνάπτει σε ένα έγγραφο τον ακριβή χρόνο της δημιουργίας του. Αυτοί οι κρυπτογραφικοί μηχανισμοί μπορούν να χρησιμοποιηθούν επίσης και για τον έλεγχο της πρόσβασης σε κοινής χρήσης οδηγούς δίσκων, σε εγκαταστάσεις υψηλής ασφάλειας, ή σε τηλεοπτικά κανάλια που χρεώνουν τον θεατή όποτε αυτός τα βλέπει.

Αλλά το πεδίο της κρυπτογραφίας περιέχει πολλά περισσότερα, όταν εμείς περιλαμβάνουμε μερικά από τα πράγματα, που αυτή μας επιτρέπει να κάνουμε. Με λίγα

μόνο βασικά εργαλεία είναι δυνατόν να αναπτύξουμε περίπλοκα σχήματα και πρωτοκολλά, τα οποία μας επιτρέπουν να πληρώσουμε, χρησιμοποιώντας *ηλεκτρονικό χρήμα*, ή να χειριστούμε από κοινού με άλλα άτομα μιας ομάδας μια απόρρητη πληροφορία.

Ενώ η σύγχρονη κρυπτογραφία εξελίσσεται με εξαιρετικούς ρυθμούς και εφαρμόζεται από το κοινό χωρίς σημαντικές δυσκολίες, στηρίζεται ουσιαστικά σε μαθηματικά προβλήματα, που είναι δύσκολο να επιλυθούν. Το πρόβλημα μπορεί να είναι δύσκολο επειδή η επίλυσή του απαιτεί κάποια απόρρητη γνώση, όπως είναι η αποκωδικοποίηση ενός αποκεκρυμμένου μηνύματος, ή η υπογραφή ενός ψηφιακού εγγράφου, ή το πρόβλημα μπορεί να είναι σκληρό επειδή είναι εξαιρετικά δύσκολη η συμπλήρωση, σαν να αναζητάς ένα μήνυμα που παράγει μια δεδομένη τιμή που προέκυψε από μία διαδικασία κατακερματισμού.

Έτσι ενώ το πεδίο της κρυπτογραφίας είναι εξαιρετικά προωθημένο, οι διαχωριστικές γραμμές, μεταξύ του τι είναι κρυπτογραφία και τι όχι, είναι συγκεχυμένες. Η κρυπτογραφία σήμερα πρέπει να εστιαστεί στην μελέτη των τεχνικών και των εφαρμογών, οι οποίες εξαρτώνται από δύσκολα προβλήματα. Η *κρυπτανάλυσις* επιχειρεί να σπάσει κρυπτογραφικούς μηχανισμούς και η *κρυπτολογία* είναι η συνδυασμένη άσκηση της κρυπτογραφίας και της κρυπταναναλύσεως.

Σημείωση : Το κείμενο που ακολουθεί δομήθηκε από μέρη κειμένου, που περιέχονται σε απαντήσεις που δόθηκαν σε υποβληθείσες ερωτήσεις (Answers to Frequently Asked Question About Today's Cryptography) προς : **RSA LABORATORIES / CRYPTOGRAPHIC RESEARCH AND CONSULTATION.**

Η επιλογή έγινε σχεδόν αναγκαστικά, αφού οι οποιεσδήποτε δικτυακές αναφορές (Πανεπιστημίων κυρίως) στο θέμα, περιείχαν το ίδιο ακριβώς υλικό και μάλιστα σε προγενέστερες εκδόσεις του. Διατήρησα την γλώσσα του πρωτοτύπου, αφού η μεγάλη έκτασή του δεν επέτρεπε την μετάφραση αλλά κυρίως την κατανόηση και ερμηνεία του στον χρόνο που διέθετα.

Εκτιμώ πως κάλυψα το θέμα από την πλευρά των σύγχρονων μόνο μεθόδων κρυπτογράφησης, οι οποίες απαιτούν και ένα σημαντικό βαθμό αναζήτησης και στο μαθηματικό υπόβαθρο, πάνω στο οποίο στηρίζονται.

Από τον ίδιο χώρο στο δίκτυο προέρχονται το γλωσσάριο και η βιβλιογραφία, που παρατίθενται στο τέλος της εργασίας.

Παραθέτω επίσης έναν πίνακα με σχετικούς προς το θέμα τίτλους και τις διευθύνσεις του δικτύου, όπου μπορεί κανείς να βρεί επί πλέον πληροφορίες.

What is Cryptography?

Cryptography, to most people, is concerned with keeping communications private. Indeed, the protection of sensitive communications has been the emphasis of cryptography throughout much of its history [Kah67]. As we will see, however, this is only one part of today's cryptography.

Encryption is the transformation of data into some unreadable form. Its purpose is to ensure privacy by keeping the information hidden from anyone for whom it is not intended, even those who can see the encrypted data. Decryption is the reverse of encryption ; it is the transformation of encrypted data back into some intelligible form.

Encryption and decryption require the use of some secret information, usually referred to as a key. Depending on the encryption mechanism used, the same key might be used for both encryption and decryption, while for other mechanisms, the keys used for encryption and decryption might be different.

But today's cryptography is more than secret writing, more than encryption and decryption. Authentication is as fundamental a part of our lives as privacy. We use authentication though out our everyday life, for instance when we sign our name to some document. As we move to a world where our decisions and agreements are communicated electronically, we need to replicate these procedures.

Cryptography provides mechanisms for such procedures. A digital signature binds a document to the possessor of a particular key, while a digital timestamp binds a document to its creation at a particular time. These cryptographic mechanisms can be used to control access to a shared disk drive, a high security installation or to a pay-per-view TV channel.

But the field of cryptography contains even more when we include some of the things cryptography enables us to do. With just a few basic tools it is possible to build elaborate schemes and protocols which allow us to pay using electronic money, to prove we know certain information without revealing the information itself, and to share a secret quantity in such a way that no fewer than three from a pool of five people (for instance) can reconstruct the secret.

While modern cryptography is growing increasingly diverse, cryptography is fundamentally based on problems that are difficult to solve. A problem may be difficult because its solution requires some secret knowledge, such as decrypting an encrypted message or signing some digital document, or the problem may be hard because it is

intrinsically difficult to complete, such as finding a message which produces a given hash value.

So as the field of cryptography has advanced, the dividing lines for what is and what is not cryptography have become blurred. Cryptography today might be summed up as the study of techniques and applications that depend on the existence of difficult problems. A cryptanalyst attempts to compromise cryptographic mechanisms, and cryptology (from the Greek *kryptos logos*, meaning "hidden word") is the discipline of cryptography and cryptanalysis combined.

What is Public-Key Cryptography?

Traditional cryptography is based on the sender and receiver of a message knowing and using the same secret key: the sender uses the secret key to encrypt the message, and the receiver uses the same secret key to decrypt the message. This method is known as *secret-key* or *symmetric cryptography*. The main problem is getting the sender and receiver to agree on the secret key without anyone else finding out. If they are in separate physical locations, they must trust a courier, or a phone system, or some other transmission medium to prevent the disclosure of the secret key being communicated. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all messages encrypted or authenticated using that key. The generation, transmission and storage of keys is called key management; all cryptosystems must deal with key management issues. Because all keys in a secret-key cryptosystem must remain secret, secret-key cryptography often has difficulty providing secure key management, especially in open systems with a large number of users.

The concept of *public-key cryptography* was introduced in 1976 by Whitfield Diffie and Martin Hellman [DH76] in order to solve the key management problem. In their concept, each person gets a pair of keys, one called the *public key* and the other called the *private key*. Each person's public key is published while the private key is kept secret. The need for the sender and receiver to share secret information is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. No longer is it necessary to trust some communications channel to be secure against eavesdropping or betrayal. The only requirement is that public keys are associated with their users in a trusted (authenticated) manner (for instance, in a trusted directory). Anyone can send a confidential message by just using public information, but the message can only be decrypted with a private key, which is in the sole possession of the intended recipient. Furthermore, public-key cryptography can be used not only for privacy (*encryption*), but also for authentication (*digital signatures*).

Encryption

When Alice wishes to send a secret message to Bob, she looks up Bob's public key in a directory, uses it to encrypt the message and sends it off. Bob then uses his private key to decrypt the message and read it. No one listening in can decrypt the message. Anyone can send an encrypted message to Bob but only Bob can read it. Clearly, one requirement is that no one can figure out the private key from the corresponding public key.

Digital Signatures

To sign a message, Alice does a computation involving both her private key and the message itself; the output is called the digital signature and is attached to the message, which is then sent. Bob, to verify the signature, does some computation involving the message, the purported signature, and Alice's public key. If the result properly holds in a simple mathematical relation, the signature is verified as being genuine; otherwise, the signature may be fraudulent or the message might have been altered.

A good history of public-key cryptography is given by Diffie [Dif88].

What is RSA?

RSA is a public-key cryptosystem for both encryption and authentication; it was invented in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman [RSA78]. It works as follows: take two large primes, p and q, and find their product $n = pq$; n is called the modulus. Choose a number, e, less than n and relatively prime to $(p-1)(q-1)$, which means that e and $(p-1)(q-1)$ have no common factors except 1. Find another number d such that $(ed - 1)$ is divisible by $(p-1)(q-1)$. The values e and d are called the public and private exponents, respectively. The public key is the pair (n,e); the private key is (n,d). The factors p and q maybe kept with the private key, or destroyed.

It is difficult (presumably) to obtain the private key d from the public key (n,e). If one could factor n into p and q, however, then one could obtain the private key d. Thus the security of RSA is related to the assumption that factoring is difficult. An easy factoring method or some other feasible attack would "break" RSA.

Here is how RSA can be used for privacy and authentication (in practice, the actual use is slightly different

RSA privacy encryption: Suppose Alice wants to send a message m to Bob. Alice creates the ciphertext c by exponentiating: $c = m^e \bmod n$, where e and n are Bob's public key. She sends c to Bob. To decrypt, Bob also exponentiates: $m = c^d \bmod n$; the relationship between e and d ensures that Bob correctly recovers m. Since only Bob knows d, only Bob can decrypt.

RSA authentication : Suppose Alice wants to send a message m to Bob in such away that Bob is assured that the message is authentic and is from Alice. Alice creates a digital signature s by exponentiating: $s = m^d \bmod n$, where d and n are Alice's private key. She sends m and s to Bob. To verify the signature, Bob exponentiates and checks that the message m is recovered: $m = s^e \bmod n$, where e and n are Alice's public key.

Thus encryption and authentication take place without any sharing of private keys: each person uses only other people's public keys and his or her own private key. Anyone can send an encrypted message or verify a signed message, using only public keys, but only someone in possession of the correct private key can decrypt or sign a message.

How is RSA used for Encryption in Practice?

RSA is combined with a secret-key cryptosystem, such as DES, to encrypt a message by means of an RSA digital envelope. Suppose Alice wishes to send an encrypted message to Bob. She first encrypts the message with DES, using a randomly chosen DES key. Then she looks up Bob's public key and uses it to encrypt the DES key. The DES-encrypted message and the RSA-encrypted DES key together form the RSA digital envelope and are sent to Bob. Upon receiving the digital envelope, Bob decrypts the DES key with his private key, then uses the DES key to decrypt to message itself. This combines the high speed of DES with the key-management convenience of RSA.

What is Diffie-Hellman?

The Diffie-Hellman key agreement protocol (also called exponential key agreement) was developed by Diffie and Hellman [DH76] in 1976 and published in the ground-breaking paper "New Directions in Cryptography." The protocol allows two users to exchange a secret key over an insecure medium without any prior secrets. The protocol has two system parameters p and g. They are both public and may be used by all the users in a system. Parameter p is a prime number and parameter g (usually called a generator) is an integer less than p, which is capable of generating every element from 1 to $p-1$ when multiplied by itself a certain number of times, modulo the prime p. Suppose that Alice and Bob want to agree on a shared secret key using the Diffie-Hellman key agreement protocol. They proceed as follows: First, Alice generates a random private value a and Bob generates a random private value b. Then they derive their public values using parameters p and g and their private values. Alice's public value is $g^a \bmod p$ and Bob's public value is $g^b \bmod p$. They then exchange their public values. Finally, Alice computes $k_{ab} = (g^b)^a \bmod p$, and Bob computes $k_{ba} = (g^a)^b \bmod p$. Since $k_{ab} = k_{ba} = k$, Alice and Bob now have a shared secret key k. The protocol depends on the discrete logarithm problem for its security. It assumes that it is computationally infeasible to calculate the shared secret key $k=g^{ab} \bmod p$ given the two public values $g^a \bmod p$ and $g^b \bmod p$ when the prime p is sufficiently large. Maurer [Mau94] has shown that breaking the Diffie-Hellman protocol was equivalent to computing discrete logarithms under certain assumptions. The Diffie-Hellman key exchange is vulnerable to a *middleperson attack*. In this attack, an opponent, Carol, intercepts Alice's public value and sends her own public value to Bob. When Bob transmits his public value, Carol substitutes it with her own and sends it to Alice. Carol and Alice thus agree on one shared key and Carol and Bob agree on another shared key. After this exchange, Carol simply decrypts any messages sent out by Alice or Bob, and then reads and possibly modifies them before re-encrypting with the appropriate key and transmitting them to the correct party. This vulnerability is due to the fact that Diffie-Hellman key exchange does not authenticate the participants. Possible solutions include the use of digital signatures and other protocol variants.

What are DSA and DSS?

The Digital Signature Algorithm (DSA) was published by the National Institute of Standards and Technology (NIST) in the Digital Signature Standard (DSS), which is a part of the U.S. government's Capstone project. DSS was selected by NIST, in cooperation with the NSA, to be the digital authentication standard of the U.S. government. The standard was issued on May 19, 1994. DSA is based on the discrete logarithm problem and derives from cryptosystems proposed by Schnorr [Sch90] and ElGamal. It is for authentication only. For a detailed description of DSA, see [NIS94b] or [NIS92]. In DSA, signature generation is faster than signature verification, whereas in RSA, signature verification is faster than signature generation (if the public and private exponents, respectively, are chosen for this property, which is the usual case). NIST claims that it is an advantage of DSA that signing is faster, but many people in cryptography think that it is better for verification to be the faster operation. Naccache *et al.* [NMR94] have developed some techniques to improve the efficiency of DSA, both for signing and verification. DSA has been criticized by the computer industry since its announcement. Criticism has focused on a few main issues: it lacks key exchange capability; the underlying cryptosystem is too recent and has been subject to too little scrutiny for users to be confident of its strength;

verification of signatures with DSA is too slow; the existence of a second authentication standard will cause hardship to computer hardware and software vendors, who have already standardized on RSA; and the process by which NIST chose DSA was too secretive and arbitrary, with too much influence wielded by NSA. Other criticisms were addressed by NIST by modifying the original proposal. A more detailed discussion of the various criticisms can be found in [NIS92], and a detailed response by NIST can be found in [SB93].

What is Secret-Key Cryptography?

Secret-key cryptography is the technology in which encryption and decryption involve the same key, a secret key. Pairs of users share a secret key, keeping the key to themselves. Data encrypted with a secret key can be decrypted only with the same secret key.

A secret-key algorithm is an algorithm for encrypting or decrypting data with a secret key. A secret key is typically used to encrypt the content of a message; in such an application, the key is called a content-encryption key and the secret-key algorithm is called a content-encryption algorithm.

A password-based encryption algorithm is a secret-key algorithm in which the key is derived from a user-supplied password.

The Data Encryption Standard (DES) is the standard federal secret-key algorithm, described in FIPS PUB 46–1. Cipher-Block Chaining (CBC) is a mode of DES, described in FIPS PUB 81.

What is DES?

DES is the Data Encryption Standard, an encryption block cipher defined and endorsed by the U.S. government in 1977 as an official standard; the details can be found in the latest official FIPS (Federal Information Processing Standards) publication concerning DES [NIS93b]. It was originally developed at IBM. DES has been extensively studied since its publication and is the most well-known and widely used cryptosystem in the world. DES is a symmetric cryptosystem. When used for communication, both sender and receiver must know the same secret key, which is used both to encrypt and decrypt the message. DES can also be used for single-user encryption, such as to store files on a hard disk in encrypted form. In a multi-user environment, secure key distribution may be difficult; public-key cryptography provides an ideal solution to this problem.

DES has a 64-bit block size and uses a 56-bit key during encryption. It is a 16-round Feistel cipher and was originally designed for implementation in hardware.

NIST has recertified DES as an official U.S. government encryption standard every five years; DES was last recertified in 1993, by default. NIST has indicated, however, that it may not recertify DES again.

What is Triple-DES?

For some time it has been common practice to protect and transport a key for DES encryption with triple-DES. This means that the plaintext is, in effect, encrypted three times. There are, of course, a variety of ways of doing this; we will explore these ways

below. See next topic (**What is Multiple Encryption?**) for a discussion of multiple encryption in general. A number of modes of triple-encryption have been proposed:

- DES-EEE3: Three DES encryptions with three different keys.
- DES-EDE3: Three DES operations in the sequence *encrypt-decrypt-encrypt* with three different keys.
- DES-EEE2 and DES-EDE2: Same the previous formats except that the first and third operations use the same key. Attacks on two-key triple-DES have been proposed by Merkle and Hellman [MH81] and Van Oorschot and Wiener [VW91], but the data requirements of these attacks make them impractical. Further information on triple-DES can be obtained from various sources [Bih95][KR96].

The use of double and triple encryption does not always provide the additional security that might be expected. Preneel [Pre94] provides the following comparisons in the security of various versions of multiple-DES and it can be seen that the most secure form of multiple encryption is triple-DES with three *distinct* keys.

# of Encryptions	# of Keys	Computation	Storage	Type of Attack
single	1	2^{56}	-	known plaintext
single	1	2^{38}	2^{38}	chosen plaintext
single	1	-	2^{56}	chosen plaintext
double	2	2^{112}	-	known plaintext
double	2	2^{56}	2^{56}	known plaintext
double	2	-	2^{112}	chosen plaintext
triple	2	2^{112}	-	known plaintext
triple	2	2^{56}	2^{56}	2^{56} chosen plaintext
triple	2	2^{120-t}	2^{t}	2^{t} known plaintext
triple	2	-	2^{56}	chosen plaintext
triple	3	2^{112}	2^{56}	known plaintext
triple	3	2^{56}	2^{112}	chosen plaintext

Comparison of Different Forms of DES Multiple Encryption

What is Multiple Encryption?

Intuitively, we might expect that by encrypting a message twice with some block cipher (either with the same key or by using two different keys), then we would expect the resultant encryption to be stronger in all but some exceptional circumstances. By using three encryptions, we would expect to achieve a yet greater level of security.

While there are some more complicated issues to consider, this is pretty much the case. Triple-DES has been used for a considerable time as a more secure cipher for protecting the keys used with single-DES. However, there are some surprising results when

we consider exactly how much additional protection is provided by using double and triple encryption.

For instance, the use of double encryption does not provide the expected increase in security [MH81] when compared with the increased implementation requirements, and it cannot be recommended as a good alternative. Instead, triple-encryption is the point at which multiple encryption gives substantial improvements in security. For a more detailed consideration of the situation with DES; for more information on multiple encryption in general see a survey article by Kaliski and Robshaw [KR96].

What is DESX?

DESX, another variant of DES, is supported by RSA Data Security's toolkits. The only difference between DES and DESX is that the input plaintext is XORed with 64 bits of key material before encryption with DES and the output is XORed with 64 bits of either related or unrelated key material. The security of DESX against differential and linear attack is equivalent to that of DES with independent subkeys, while the security against exhaustive search is greatly increased.

What is RC2?

RC2 is a variable key-size block cipher designed by Rivest for RSA Data Security. "RC" stands for "Ron's Code" or "Rivest's Cipher." It is faster than DES and is designed as a "drop-in" replacement for DES. It can be made more secure or less secure than DES against exhaustive key search by using appropriate key sizes. It has a block size of 64 bits and is about two to three times faster than DES in software. The algorithm is confidential and proprietary to RSA Data Security. RC2 can be used in the same modes as DES.

An agreement between the Software Publishers Association (SPA) and the United States government gives RC2 and RC4 special status by means of which the export approval process is simpler and quicker than the usual cryptographic export process. However, to qualify for quick export approval a product must limit the RC2 and RC4 key sizes to 40 bits; 56 bits is allowed for foreign subsidiaries and overseas offices of United States companies. An additional string (40 to 88 bits long) called a *salt* can be used to thwart attackers who try to precompute a large look-up table of possible encryptions. The salt is appended to the encryption key, and this lengthened key is used to encrypt the message; the salt is then sent, unencrypted, with the message. RC2 and RC4 have been widely used by developers who want to export their products; DES is almost never approved for export.

What is RC4?

RC4 is a stream cipher designed by Rivest for RSA Data Security. It is a variable key-size stream cipher with byte-oriented operations. The algorithm is based on the use of a random permutation and analysis shows that the period of the cipher is overwhelmingly likely to be greater than 10 100 . Eight to sixteen machine operations are required per output byte, and the cipher can be expected to run very quickly in software. While the algorithm is confidential and proprietary to RSA Data Security, Inc., it has been scrutinized under conditions of non-disclosure by independent analysts and it is considered secure. The

RC4 stream cipher has a special status by which export from the U.S. can often be facilitated.

What is RC5?

RC5 [Riv95] is a fast block cipher designed by Rivest for RSA Data Security. It is a parameterized algorithm with a variable block size, a variable key size, and a variable number of rounds. The block size can be 32, 64, or 128 bits long. The number of rounds can range from 0 to 255. The key can range from 0 bits to 2048 bits in size. Such built-in variability provides flexibility in levels of security and efficiency.

There are three routines in RC5: *key expansion, encryption*, and *decryption*. In the key-expansion routine, the user-provided secret key is expanded to fill a key table whose size depends on the number of rounds. The key table is then used in both encryption and decryption. The encryption routine consists of three primitive operations: addition, bitwise XOR, and rotation. The exceptional simplicity of RC5 makes it easy to implement and analyze. Indeed, like RSA, RC5 can be written on the "back of the envelope" (except for key expansion).

The security of RC5 is provided by the heavy use of data-dependent rotations and the mixture of different operations. In particular, the use of data-dependent rotations helps defeat differential and linear cryptanalysis, and Kaliski and Yin [KY95] found that RC5 with a block size of 64 bits and 12 or more rounds provides good security against differential and linear cryptanalysis.

RSA Data Security is in the process of patent application for RC5.

What are the Advantages and Disadvantages of Public-Key Cryptography Compared with Secret-Key Cryptography?

The primary advantage of public-key cryptography is increased security and convenience. Private keys never need to transmitted or revealed to anyone. In a secret-key system, by contrast, the secret keys must be transmitted (either manually or through a communication channel), and there may be a chance that an enemy can discover the secret keys during their transmission.

Another major advantage of public-key systems is that they can provide a method for digital signatures. Authentication via secret-key systems requires the sharing of some secret and sometimes requires trust of a third party as well. As a result, a sender can repudiate a previously authenticated message by claiming that the shared secret was somehow compromised by one of the parties sharing the secret.

For example, the Kerberos secret-key authentication system involves a central database that keeps copies of the secret keys of all users; an attack on the database would allow widespread forgery. Public-key authentication, on the other hand, prevents this type of repudiation; each user has sole responsibility for protecting his or her private key. This property of public-key authentication is often called non-repudiation.

A disadvantage of using public-key cryptography for encryption is speed; there are popular secret-key encryption methods that are significantly faster than any currently available public-key encryption method. Nevertheless, public-key cryptography can be used with secret-key cryptography to get the best of both worlds. For encryption, the best solution is to combine public- and secret-key systems in order to get both the security advantages of public-key systems and the speed advantages of secret-key systems. The public-key system can be used to encrypt a secret key which is used to encrypt the bulk of a file or message. Such a protocol is called a digital envelope, which is explained in more detail in the topic of RSA (How is RSA used for Encryption in Practice?).

Public-key cryptography may be vulnerable to impersonation, however, even if users' private keys are not available. A successful attack on a certification authority will allow an adversary to impersonate whomever the adversary chooses to by using a public-key certificate from the compromised authority to bind a key of the adversary's choice to the name of another user.

In some situations, public-key cryptography is not necessary and secret-key cryptography alone is sufficient. This includes environments where secure secret-key agreement can take place, for example by users meeting in private. It also includes environments where a single authority knows and manages all the keys (e.g., a closed banking system). Since the authority knows everyone's keys already, there is not much advantage for some to be "public" and others "private." Also, public-key cryptography is usually not necessary in a single-user environment. For example, if you want to keep your personal files encrypted, you can do so with any secret-key encryption algorithm using, say, your personal password as the secret key. In general, public-key cryptography is best suited for an open multi-user environment.

Public-key cryptography is not meant to replace secret-key cryptography, but rather to supplement it, to make it more secure. The first use of public-key techniques was for secure key exchange in an otherwise secret-key system [DH76]; this is still one of its primary functions. Secret-key cryptography remains extremely important and is the subject of ongoing study and research. Some secret-key cryptosystems are discussed in the sections on Block Ciphers and Stream Ciphers.

Other Public-Key Encryption and Signature Algorithms

What is the ElGamal Cryptosystem?

The ElGamal system [Elg85] is a public-key cryptosystem based on the discrete logarithm problem. It consists of both encryption and signature algorithms. The encryption algorithm is similar in nature to the Diffie-Hellman key agreement protocol.

The system parameters consist of a prime p and an integer g, whose powers modulo p generate a large number of elements, as in Diffie-Hellman. Alice has a private key a and a public key y, where $y = g^a \pmod p$. Suppose Bob wishes to send a message m to Alice. Bob first generates a random number k less than p. He then computes

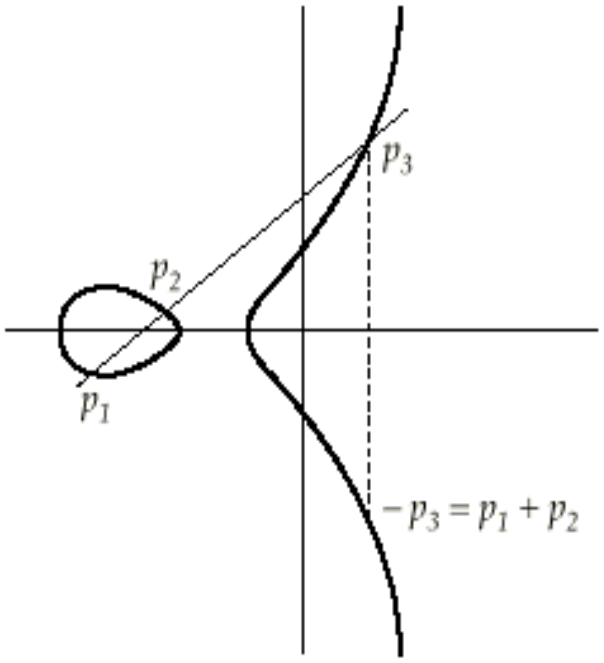
$$y_1 = g^k \pmod p \qquad \text{and} \qquad y_2 = m \oplus y^k ,$$

where $\oplus$ denotes the bit-wise XOR. Bob sends (y_1 , y_2) to Alice. Upon receiving the ciphertext, Alice computes

$$m = (y_1^{\,a} \bmod p) \oplus y_2 .$$

The ElGamal signature algorithm is similar to the encryption algorithm in that the public key and private key have the same form; however, encryption is not the same as signature verification, nor is decryption the same as signature creation as in RSA. DSA is based in part on the ElGamal signature algorithm.

Analysis based on the best available algorithms for both factoring and discrete logarithms shows that RSA and ElGamal have similar security for equivalent key lengths. The main disadvantage of ElGamal is the need for randomness, and its slower speed (especially for signing). Another potential disadvantage of the ElGamal system is that message expansion by a factor of two takes place during encryption. However, such message expansion is negligible if the cryptosystem is used only for exchange of secret keys.



Elliptic Curve Addition

What are Elliptic Curves?

Elliptic curves are mathematical constructions from number theory and algebraic geometry, which in recent years have found numerous applications in cryptography. An elliptic curve can be defined over any field (e.g., real, rational,

complex). However, elliptic curves used in cryptography are mainly defined over finite fields. An elliptic curve consists of elements (x, y) satisfying the equation

$$y^2 = x^3 + ax + b$$

together with a single element denoted O called the "point at infinity," which can be visualized as the point at the top and bottom of every vertical line. Addition of two points on a elliptic curve is defined according to a set of simple rules (e.g., point *p1* plus point *p2* is equal to point *-p3* in Figure). The addition operation in an elliptic curve is the counterpart to modular multiplication in common public-key cryptosystems, and multiple addition is the counterpart to modular exponentiation. Elliptic curves are covered in more recent texts on cryptography, including a informative text by Koblitz [Kob94].

What are Elliptic Curve Cryptosystems?

Elliptic curve cryptosystems [Mil86][Kob87] are analogs of public-key cyptosystems such as RSA and ElGamal, in which modular multiplication is replaced by the elliptic curve addition operation.

The curves used in elliptic curve analogs of discrete logarithm cryptosystems are normally of the form

$$y^2 = x^3 + ax + b \pmod{p},$$

where p is prime. The problem tapped by the discrete logarithm analogs in elliptic curves is the elliptic curve logarithm problem, defined as follows: given a point G on an elliptic curve with order r (number of points on the curve) and another point Y on the curve, find a unique x (0 <= x <= r - 1) such that Y = xG, i.e., Y is the xth multiple of G. Until recently, the best attacks on elliptic curve logarithm problems were the general methods applicable to any group. The methods have a running time of about a constant times the square root of r on average, which is much slower than specialized attacks on certain types of groups. The lack of specialized attacks means that shorter key sizes for elliptic cryptosystems give the same security as larger keys in cryptosystems that are based on discrete logarithm problem . However, for certain elliptic curves, Menezes, Okamoto, and Vanstone [MOV90] have been able to reduce the elliptic logarithm problem to a discrete logarithm problem. It is possible that algorithm development in this area will change the security of elliptic curve discrete logarithm cryptosystems to be equivalent to that of general discrete logarithm cryptosystems; this is an open research problem.

Elliptic curve analogs of RSA have been proposed, and they are based on the difficulty of factoring, just as RSA is. The elliptic curve analogs do not seem to offer any significant advantage over RSA, as the underlying problem is the same and the key sizes are similar for equivalent levels of security. Two of their purported advantages; resistance to "low-exponent" attacks, and to signature forgery against a chosen message attack; have recently been shown not to hold (see [KO95] and [Kal95]).

See [Men93] for more information on elliptic curve cryptosystems.

What are Knapsack Cryptosystems?

The Merkle-Hellman knapsack cryptosystem [MH78] is a public-key cryptosystem that was first published in 1978. It is commonly referred to as the *knapsack cryptosystem*. It

is based on the subset sum problem in combinatorics. The problem involves selecting a number of objects with given weights from a large set such that the sum of the weights is equal to a pre-specified weight. This is considered to be a difficult problem to solve in general, but certain special cases of the problem are relatively easy to solve, which serve as the "trapdoor" of the system. The-single iteration knapsack cryptosystem introduced in 1978 was broken by Shamir [Sha84]. Merkle then published the multiple-iteration knapsack problem which was broken by Brickell [Bri85]. Merkle offered a $100 reward for anybody able to crack the single iteration knapsack and a $1000 reward for anybody able to crack the multiple iteration cipher from his own pocket. When they were cracked, he promptly paid up.

The Chor-Rivest knapsack cryptosystem was first published in 1984, followed by a revised version in 1988 [CR88]. It is the only knapsack-like cryptosystem that does not use modular multiplication. It was also the only knapsack-like cryptosystem that was secure for any extended period of time. Unfortunately, Schnorr and Hφrner [SH95] developed an attack on the Chor-Rivest cryptosystem using improved lattice reduction which reduced to hours the amount of time needed to crack the cryptosystem for certain parameter values (though not for those recommended by Chor and Rivest). They also showed how the attack can be extended to attack Damgɛrd's knapsack hash function [Dam90].

Special Digital Signature Schemes

What are Special Signature Schemes?

Since the time Diffie and Hellman introduced the concept of digital signatures, many signature schemes have been proposed in cryptographic literature. These schemes can be categorized as either conventional digital signature schemes, RSA, DSA) or special signature schemes depending on their security features.

In a *conventional signature scheme* (the original model defined by Diffie and Hellman), we generally assume the following situation:

- The signer knows the contents of the message that he has signed.
- Anyone who knows the public key of the signer can verify the correctness of the signature at any time without any consent or input from the signer. (Digital signature schemes with this property are called *self-authenticating signature schemes*.)
- The security of the signature schemes (i.e., hard to forge, non-repudiation, is based on certain complexity-theoretic assumptions.

In some situations, it may be better to relax some of these assumptions, and/or add certain special security features. For example, when Alice asks Bob to sign a certain message, she may not want him to know the contents of the message. In the past decade, a variety of special signature schemes have been developed to fit security needs in different applications.

There are more examples of such special schemes :

What is a Blind Signature Scheme?

Blind signature schemes, first introduced by Chaum [Cha83][Cha85], allow a person to get a message signed by another party without revealing any information about the message to the other party.

Chaum demonstrated the implementation of this concept using RSA signatures as follows: Suppose Alice has a message m that she wishes to have signed by Bob, and she does not want Bob to learn anything about m. Let (n,e) be Bob's public key and (n,d) be his private key. Alice generates a random value r such that $gcd(r, n) = 1$ and sends

$$m' = r^e m \bmod n$$

to Bob. The value m' is "blinded" by the random value r, and hence Bob can derive no useful information from it. Bob returns the signed value,

$$s' = (m')^d = (r^e m)^d \bmod n$$

to Alice. Since $s' = r^m d \bmod n$, Alice can obtain the true signature s of m by computing

$$s = s' r^{-1} \bmod n.$$

Now Alice's message has a signature she could not have obtained on her own. This signature scheme is secure provided that factoring and root extraction remain difficult. However, regardless of the status of these problems the signature scheme is unconditionally "blind" since *r* is random. The random *r* does not allow the signer to learn about the message even if the signer can solve the underlying hard problems.

There are potential problems if Alice can give an arbitrary message to be signed, since this effectively enables her to mount a chosen message attack. One way of thwarting this kind of attack is described in [CFN88].

Blind signatures have numerous uses including timestamping, anonymous access control, and digital cash. Thus it is not surprising there are now numerous variations on the blind signature theme. Further work on blind signatures has been carried out in recent years [FY94][SPC95].

What is a Designated Confirmer Signature?

A designated confirmer signature [Cha94] strikes a balance between self-authenticating digital signatures and zero-knowledge proofs. While the former allows anybody to verify a signature, the latter can only convince one recipient at a time of the authenticity of a given document, and only through interaction with the signer. A *designated confirmer signature* allows certain designated parties to confirm the authenticity of a document without the need for the signer's input. At the same time, without the aid of either the signer or the designated parties, it is not possible to verify the authenticity of a given document. Chaum developed implementations of designated confirmer signatures with one or more confirmers using RSA digital signatures.

What is a Fail-stop Signature Scheme?

A *fail-stop signature scheme* is a type of signature devised by van Heyst and Pederson [VP92] to protect against the possibility that an enemy may be able to forge a person's signature. It is a variation of the one-time signature scheme, in which only a single message can be signed and protected by a given key at a time. The scheme is based on the discrete logarithm problem. In particular, if an enemy can forge a signature, then the actual signer can prove that forgery has taken place by demonstrating the solution of a supposedly hard problem. Thus the forger's ability to solve that problem is transferred to the actual signer. (The term "fail-stop" refers to the fact that a signer can detect and stop failures, i.e., forgeries. Note that if the enemy obtains an actual copy of the signer's private key, forgery cannot be detected. What the scheme detects are forgeries based on cryptanalysis.)

What is a Group Signature?

A *group signature*, introduced by Chaum and van Heijst [CV91], allows any member of a group to digitally sign a document in a manner such that a verifier can confirm that it came from the group, but does not know which individual in the group signed the document. The protocol allows for the identity of the signer to be discovered, in case of disputes, by a designated group authority who has some auxiliary information.

Unfortunately, each time a member of the group signs a document, a new key pair has to be generated for the signer. The generation of new key pairs causes the length of both the group members' secret keys and the designated authority's auxiliary information to grow. This tends to cause the scheme to become unwieldy when used by a group to sign numerous messages or when used for an extended period of time.

Some improvements [CP94][CP95] have been made in the efficiency of this scheme.

What is a One-time Signature Scheme?

A *one-time signature scheme* allows the signature of only a single message using a given piece of private (and public) information. One advantage of such a scheme is that it is generally quite fast. However, the scheme tends to be unwieldy when used to authenticate multiple messages because additional data needs to be generated to both sign and verify each new message. By contrast, with conventional signature schemes like RSA, the same key pair can be used to authenticate multiple documents. There is a relatively efficient implementation of one-time-like signatures by Merkle called the Merkle Tree Signature Scheme, which does not require new key pairs for each message.

What is an Undeniable Signature Scheme?

Undeniable signature scheme, devised by Chaum and van Antwerpen [CV90][CV92], are non-self-authenticating signature schemes, where signatures can only be verified with the signer's consent. However, if a signature is only verifiable with the aid of a signer, a dishonest signer may refuse to authenticate a genuine document.

Undeniable signatures solve this problem by adding a new component called the *disavowal protocol* in addition to the normal components of signature and verification. The scheme is implemented using public-key cryptography based on the discrete logarithm problem. The signature part of the scheme is similar to other discrete logarithm signature schemes. Verification is carried out by a challenge-response protocol where the verifier, Alice, sends a challenge to the signer, Bob, and views the answer to verify the signature. The disavowal process is similar: Alice sends a challenge and Bob's response shows that a signature is not his. If Bob does not take part, it may be assumed that the document is authentic. The probability that a dishonest signer is able to successfully mislead the verifier in either verification or disavowal is $1/p$ where p is the prime number in the signer's private key. If we consider the average 768-bit private key, there is only a minuscule probability that the signer will be able to repudiate a document he has signed.

Cryptographic Hashing Algorithms

What is a Hash Function?

A hash function *H* is a transformation that takes a variable-size input *m* and returns a fixed-size string, which is called the hash value *h* (that is, *h = H(m)*). Hash functions with just this property have a variety of general computational uses, but when employed in cryptography the hash functions are usually chosen to have some additional properties.

The basic requirements for a cryptographic hash function are:

- the input can be of any length,
- the output has a fixed length,
- *H(x)* is relatively easy to compute for any given *x*,
- *H(x)* is one-way,
- *H(x)* is collision-free.

A hash function *H* is said to be *one-way* if it is hard to invert, where "hard to invert" means that given a hash value *h*, it is computationally infeasible to find some input *x* such that *H(x) = h*.

If, given a message *x,* it is computationally infeasible to find a message *y i x* such that *H(x) = H(y)* then *H* is said to be a *weakly collision-free* hash function.

A *strongly collision-free* hash function *H* is one for which it is computationally infeasible to find any two messages *x* and *y* such that *H(x) = H(y).*

The hash value represents concisely the longer message or document from which it was computed; one can think of a message digest as a "digital fingerprint" of the larger document. Examples of well-known hash functions are MD2, MD5 and SHA.

What are MD2, MD4 and MD5?

MD2 [Kal92], MD4 [Riv91b] [Riv92b], and MD5 [Riv92c] are message-digest algorithms developed by Rivest. They are meant for digital signature applications where a large message has to be "compressed" in a secure manner before being signed with the private key. All three algorithms take a message of arbitrary length and produce a 128- bit message digest. While the structures of these algorithms are somewhat similar, the design of MD2 is quite different from that of MD4 and MD5 and MD2 was optimized for 8-bit machines, whereas MD4 and MD5 were aimed at 32-bit machines. Description and source code for the three algorithms can be found as Internet RFCs 1319 - 1321 [Kal92] [Riv92b][Riv92c].

MD2 was developed by Rivest in 1989. The message is first padded so that its length in bytes is divisible by 16. A 16-byte checksum is then appended to the message, and the hash value is computed on this resulting message. Rogier and Chauvaud have found that collisions for MD2 can be constructed if the calculation of the checksum is omitted [RC95]. This is the only cryptanalytic result known for MD2.

MD4 was developed by Rivest in 1990. The message is padded to ensure that its length in bits plus 448 is divisible 512. A 64-bit binary representation of the original length of the message is then concatenated to the message. The message is processed in 512-bit blocks in the Damgɛrd/Merkle iterative structure, and each block is processed in three distinct rounds. Attacks on versions of MD4 with either the first or the last rounds missing were developed very quickly by Den Boer and Bosselaers [DB92] and others. Dobbertin [Dob95] has shown how collisions for the full version of MD4 can be found in under a minute on a typical PC. Clearly, MD4 should now be considered broken. MD5 was developed by Rivest in 1991. It is basically MD4 with "safety-belts" and while it is slightly slower than MD4, it is more secure. The algorithm consists of four distinct rounds, which have a slightly different design from that of MD4. Message-digest size, as well as padding requirements, remains the same. Den Boer and Bosselaers [DB94] have found pseudo-collisions for MD5, but there are no other known cryptanalytic results.

Van Oorschot and Wiener [VW94] have considered a brute-force search for collisions in hash functions, and they estimate that a collision search machine designed specifically for MD5 (costing $10 million in 1994) could find a collision for MD5 in 24 days on average. The general techniques can be applied to other hash functions.

More details on MD2, MD4, and MD5 can be found in [Pre93] and [Rob95c].

What is the Secure Hash Algorithm (SHA and SHA-1)?

The Secure Hash Algorithm (SHA), the algorithm specified in the Secure Hash Standard (SHS), was developed by NIST and published as a federal information processing standard (FIPS PUB 180) [NIS93a]. SHA-1 [NIS94c] was a revision to SHA that was published in 1994. The revision corrected an unpublished flaw in SHA. Its design is very similar to the MD4 family of hash functions developed by Rivest.

The algorithm takes a message of less than *2* 64 bits in length and produces a 160-bit message digest. The algorithm is slightly slower than MD5, but the larger message digest makes it more secure against brute-force collision and inversion attacks. SHA is part of the Capstone project. For further information on SHA, see [Pre93] and [Rob95c].

Cryptanalysis

What is Differential Cryptanalysis?

Differential cryptanalysis is a type of attack that can be mounted on iterative block ciphers. These techniques were first introduced by Murphy [Mur90] in an attack on FEAL-4, but they were later improved and perfected by Biham and Shamir [BS91a][BS93b] who used them to attack DES. Differential cryptanalysis is basically a chosen plaintext attack and relies on an analysis of the evolution of the differences between two related plaintexts as they are encrypted under the same key. By careful analysis of the available data, probabilities can be assigned to each of the possible keys and eventually the most probable key is identified as the correct one.

Differential cryptanalysis has been used against a great many ciphers with varying degrees of success. In attacks against DES, its effectiveness is limited by what was very careful design of the S-boxes during the design of DES in the mid-1970s [Cop92].

Studies on protecting ciphers against differential cryptanalysis have been conducted by Nyberg and Knudsen [NK95] as well as Lai, Massey and Murphy [LMM92].

Differential cryptanalysis has also been useful in attacking other cryptographic algorithms such as hash functions.

What is Linear Cryptanalysis?

Linear cryptanalysis was first devised by Matsui and Yamagishi [MY92] in an attack on FEAL. It was extended by Matsui [Mat93] to attack DES. Linear cryptanalysis is a known plaintext attack and uses a linear approximation to describe the behavior of the block cipher. Given sufficient pairs of plaintext and corresponding ciphertext, bits of information about the key can be obtained and increased amounts of data will usually give a higher probability of success.

There have been a variety of enhancements and improvements to the basic attack. Langford and Hellman [LH94] introduced an attack called *differential-linear cryptanalysis* which combines elements of differential cryptanalysis with those of linear cryptanalysis. Also, Kaliski and Robshaw [KR94] showed that a linear cryptanalytic attack using multiple approximations might allow for a reduction in the amount of data required for a successful attack. Other issues such as protecting ciphers against linear cryptanalysis have been considered by Nyberg [Nyb95], Knudsen [Knu93], and O'Conner [Oco95].

Other Products

What is PGP?

Pretty Good Privacy (PGP) is a software package originally developed by Phil Zimmerman that provides cryptographic routines for e-mail and file storage applications. Zimmerman took existing cryptosystems and cryptographic protocols and developed a freeware program that can run on multiple platforms. It provides message encryption, digital signatures, data compression, and e-mail compatibility.

The algorithms used for message encryption are RSA for key transport and IDEA for bulk encryption of messages. Digital signatures are achieved by the use of RSA for signing and MD5 for computing the message digest. The freeware program ZIP is used to compress messages for transmission and storage. E-mail compatibility is achieved by the use of Radix-64 conversion.

MIT PGP versions 2.6 and later are legal freeware for non-commercial use based on RSAREF. Viacrypt PGP versions 2.7 and later are legal commercial versions of the same software. PGP is bound by Federal export laws due to the use of the RSA public key cryptosystem.

What is RIPEM?

RIPEM is a program developed by Mark Riordan and enhanced by Jeff Thompson that enables secure Internet e-mail; it provides both encryption and digital signatures, using RSA and DES routines from RSAREF. RIPEM is compatible with PKCS #7 and PKCS #10 in support of S/MIME and other PKCS-based messaging. RIPEM implements certificates, certification hierarchies and CRLs. RIPEM is also PEM-compatible and provides a convenient application programming interface which lets e-mail handlers link to RIPEM's message-handling functions. RIPEM is available free for non-commercial use in the U.S. and Canada.

To get RIPEM, read <ftp://ripem.msu.edu/pub/crypt/ripem/GETTING_ACCESS>.

Digital Signature Law

What is the Law Concerning Digital Signatures?

Just as traditional handwritten (holographic) signatures link people to the content of their agreements in a legally recognized manner, digital signatures can provide similar (but not identical) functions for electronic commerce and other purposes. Perhaps most importantly, digital signatures contribute to non-repudiation — a security service that is increasingly appreciated within the legal and business communities to provide important benefits.

The legal status of digital signatures for many, diverse applications has meaningfully advanced during the past few years. Even undigitally signed messages and records, such as those utilizing traditional electronic data interchange (EDI) or simple e-mail, have gained considerable legal recognition. The lack of litigation is, arguably, testament to the practical use and legal effectiveness of digital practices. The following developments support this assessment.

In 1989, electronic funds transfer laws, such as Article 4A of the Uniform Commercial Code and later the United Nations Commission on International Trade Law's (UNCITRAL's) Model Law on International Credit Transfers, adopted authentication procedures rather than traditional signatures as the basis for verifying transactions and apportioning liability. In 1990, the U.S. Department of Justice issued its Guidelines on the Admissibility of Electronically Filed Federal Records as Evidence, which emphasized the reliability and trustworthiness of computer-based data for evidentiary purposes. In the comptroller general of the United States issued a decision entitled "Use Of Electronic Data Interchange Technology to Create Valid Obligations" that authorized EDI for government contractual obligations "using properly secured EDI systems" and considered the permissible uses of digital signatures. The comptroller general's decision is only one effort, albeit an especially important one, to resolve information security and signature issues.

In 1992, the House of Delegates of the American Bar Association (ABA) went on record supporting government action, to "encourage the use of appropriate and properly implemented security techniques, procedures and practices to assure the authenticity and integrity of information in electronic form." It also recognized that "information in electronic form, where appropriate, may be considered to satisfy legal requirements regarding a written signature to the same extent as information on paper or in other conventional forms when appropriate security techniques, practices, and procedures have been adopted." In 1994, the first comprehensive legal study of digital signature infrastructure was published, *Federal Certification Authority Liability and Policy*, under the auspices of the U.S. government. The study urged the government to forge ahead with implementations and recognized that liability and other legal concerns could be appropriately controlled.

One of the longest and most notorious legal efforts (concerning signatures) has been to reform statutes of frauds, which require traditional writings and signatures to make certain transactions enforceable. The ongoing revision process of Article 2 of the Uniform Commercial Code (addressing commercial sales law) now contemplates the statutes' revision or elimination.

And, of course, 1995 commenced the adoption or consideration of digital signature legislation in various U.S. states. The first Digital Signature Act became law in Utah in May 1995, followed shortly thereafter by California, and other states are contemplating various forms of digital signature legislation. Such legislative efforts generally seek to make digital signatures at least as legally effective as traditional handwritten signatures (for certain purposes). Most recently, draft *Digital Signature Guidelines* developed by the Information Security Committee, Section of Science and Technology, American Bar Association, have been released for comment — the *Guidelines* place digital signatures *at least* on a par with holographic signatures.

Although further law reform is both inevitable and necessary, these developments present a very encouraging picture — indeed one that supports wide-scale adoption of digital signatures by business and government and their corresponding recognition in the law.

Δικτυακές Αναφορές - Κρυπτογραφία

α/α	ΘΕΜΑ	ΔΙΕΥΘΥΝΣΗ
1	RSA Data Security	http://www.rsa.com/
2	Frequently Asked Questions About Today's Cryptography	http://www.rsa.com/rsalabs/faq/
3	Glossary	http://www.rsa.com/rsalabs/faq/html/glossary.html
4	Ronald L. Rivest	http://theory.lcs.mit.edu/~rivest/
5	Adi Shamir	http://www.wisdom.weizmann.ac.il/people/generic?shamir
6	Leonard M. Adleman	http://hto-e.usc.edu/people/Adleman.html
7	Ronald L. Rivest / Cryptography and Security	http://theory.lcs.mit.edu/~rivest/crypto-security.html
8	AN OVERVIEW OF COMPUTER SECURITY	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_1.html
9	Computer Crime	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_10.html
10	Privacy	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_11.html
11	Privacy / Electronic Communications Act	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_12.html
12	Why Is Computer Security Difficult?	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_13.html
13	Objectives of Computer Security	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_14.html
14	Issues for Concern / Most Security Problems Are People Related / Hardware Security / Software Security	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_15.html
15	Issues for Concern / Attacks on Data / Violations to Data Secrecy / Violations to Data Integrity	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_16.html
16	Issues for Concern / Network Security	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_17.html
17	People - the number one problem	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_18.html
18	Methods of defense / Encryption / Hardware Controls / Policies / Software Controls	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_19.html
19	Methods of defense / Overlapping of Controls / Periodic Review	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_20.html
20	Is There A Threat?	• http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_21.html • http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_22.html

21	Early Computer Security Efforts	• http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_23.html • http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_24.html • http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap01_25.html
22	Public Key Cryptography and Protocols	• http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_1.html • http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_6.html
23	Basics of Public-Key Cryptography	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_7.html
24	Basics of Public-Key Cryptography / Services Provided by Cryptosystems (Secrecy - Authenticity - Integrity - Nonrepudiation)	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_8.html
25	Basics of Public-Key Cryptography / Two major applications for public-key systems (Distribution of secret keys - Digital signatures)	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_9.html
26	Basics of Public-Key Cryptography / Merkle-Hellman Knapsacks	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_10.html
27	Basics of Public-Key Cryptography / Superincreasing (Simple) Knapsacks	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_11.html
28	Basics of Public-Key Cryptography / Review of Modular Arithmetic	• http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_12.html • http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_13.html
29	Basics of Public-Key Cryptography / Transforming a Knapsack sequence	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_14html
30	Basics of Public-Key Cryptography / Example Using the Merkle-Hellman Knapsack	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_15.html
31	Basics of Public-Key Cryptography / Ravist-Shamir-Adelman (RSA) Encryption	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_16.html
32	Basics of Public-Key Cryptography / Summary of RSA Encryption	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_17.html
33	Basics of Public-Key Cryptography / The Digital Signature Algorithm.	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_18.html
34	One-Way Hash Functions / Hash Functions and Message Digests / Properties of Hash Functions	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_19.html

35	One-Way Hash Functions / Usage of Hash Functions.	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_20.html
36	One-Way Hash Functions / MD4/MD5 and MD2 / General Description of MD5	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_21.html
37	One-Way Hash Functions / Secure Hash Algorithm (SHA)	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_22.html
38	Protocols / The Purpose of Protocols	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_23.html
39	Protocols / Protocol	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_24.html
40	Protocols / Using RSA To Support Secrecy	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_25.html
41	Protocols / Using RSA To Support Authentication	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_26.html
42	Protocols / Using RSA for Secrecy, Authenticity and Integrity	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_27.html
43	Protocols / Nonrepudiation	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_28.html
44	Protocols / Proof of Delivery	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_29.html
45	Key Management / Conventional system key management	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_30.html
46	Key Management / Public-Key System Key Management	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_31.html
47	Key Management / A Protocol for Exchange of Public Keys	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_32.html
48	Key Management / Use of Certificates	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_33.html
49	Key Management / A phone-book approach to certificates	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_34.html
50	Key Management / Decentralized Management	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_35.html
51	Key Management / Authentication Protocols	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_36.html
52	Key Management / A one-way authentication protocol	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_37.html
53	The CLIPPER Chip	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_38.html
54	Clipper / The Great Debate - Key Escrowing	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_39.html
55	More on Protocols / Anonymous Key Distribution	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_40.html
56	More on Protocols / Secret Sharing Algorithms	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_41html
57	More on Protocols / Secret Sharing Algorithms / Example	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_42tml
58	More on Protocols / Blind signatures	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_43.html
59	More on Protocols / Secure Elections	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_44.html

60	More on Protocols / Simplistic Secure Voting Protocols	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_45.html
61	More on Protocols / Voting with Blind Signatures	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_46.html
62	More on Protocols / Digital Cash	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_47.html
63	More on Protocols / S/Key - One-Time Password System	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_48.html
64	More on Protocols / How one-time passwords are generated	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_49.html
65	BASICS OF CONVENTIONAL KEY CRYPTOGRAPHY	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_1.html
66	Basics of Conventional Key Cryptography	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_6.html
67	Basic Encryption and Decryption / Definitions / Cryptanalyst's chore / Breakable encryption algorithm / Cryptanalyst's Tools	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_7.html
68	Basic Encryption and Decryption / Encryption Algorithms / Cipher Key Systems	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_8.html
69	Basic Encryption and Decryption / Representation of characters	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_9.html
70	Monoalphabetic Ciphers / The Caesar Cipher / Cryptanalysis of Caesar Cipher	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_10.html
71	Monoalphabetic Ciphers / Permute Using a Key / Multiplicative Modulus Permutation	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_11.html
72	Monoalphabetic Ciphers / Advantages of monoalphabetic ciphers / Frequency Distributions	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_12.html
73	Monoalphabetic Ciphers / Are Monoalphabetic Ciphers Secure?	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_13.html
74	Monoalphabetic Ciphers / Meaningful Observations	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_14.html
75	Polyalphabetic Ciphers / The Reason / Advantages of Polyalphabetic Substitutions	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_15.html
76	Polyalphabetic Ciphers / Vigenere Tableaux	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_16.html
77	Polyalphabetic Ciphers / One Method of Using the Vigenere Tableaux	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_17.html
78	Polyalphabetic Ciphers / An Example	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_18.html
79	Polyalphabetic Ciphers / Cryptanalysis of Polyalphabetic Substitutions / The Kasiski Method for Repeated Patterns	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_19.html
80	Polyalphabetic Ciphers / Steps to Kasiski Method	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_20.html

81	Index of Coincidence (IC)	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_21.html
82	Polyalphabetic Ciphers / The Perfect Substitution Cipher / One Time Pad / Problems	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_22.html
83	Polyalphabetic Ciphers / The Vernam Cipher / Method / The Binary Vernam Cipher	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_23.html
84	Polyalphabetic Ciphers / Cracking Random Number Generators	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_24.html
85	Polyalphabetic Ciphers / Transpositions (Permutations) / Method / Advantages / Disadvantages	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_25.html
86	Polyalphabetic Ciphers / Pattern Analysis / The Problem	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_26.html
87	Polyalphabetic Ciphers / Double Transpositions / Cryptanalysis	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_27.html
88	Polyalphabetic Ciphers / Stream and Block Ciphers / Stream Ciphers (Substitution) / Block Ciphers (Transposition)	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_28.html
89	The Data Encryption Standard (DES) / History of DES / Overview of DES / Four Modes of Operation	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_29.html
90	ECB Mode (Native DES) / Overview / Basic Building Blocks	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_30.html
91	ECB Mode (Native DES) / Permutation Box (P-Box) / Exclusive-OR operation	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_31.html
92	ECB Mode (Native DES) / Substitution Box (S-Box)	• http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_32.html • http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_33.html
93	ECB Mode (Native DES) / A Single Cycle of the DES	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_34.html
94	ECB Mode (Native DES) / A Flow Diagram	• http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_35.html • http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_36.html • http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_37.html • http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_38.html
95	ECB Mode (Native DES) / Weakness of ECB Mode / ECB error propagation	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_39.html
96	CBC Mode / CBC Mode Error Propagation	http://www.cs.nps.navy.mil/curricula/t

		racks/security/notes/chap04_40.html
97	Feedback Modes / OFB Mode / CFB Mode	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_41.html
98	The Data Encryption Standard (DES) / Criticisms of the DES / Weaknesses of the DES	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_42.html
99	International Data Encryption Algorithm (IDEA) / Overview / General Description / Speed of IDEA	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_43.html
100	Skipjack Algorithm / Overview / General Description / Speed of IDEA	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap04_44.html
101	Ray Kopsa's Shortcut to Cryptography	http://www.subject.com/crypto/crypto.html
102	What is a one-way hash function?	http://stasi.bradley.edu/privacy/sci.crypt_FAQ_7.html#OneWayHash
103	Public Key Cryptography	http://stasi.bradley.edu/privacy/public_key_crypto.html
104	How electronic encryption works and how it will change your business	http://www.iinet.net.au/~heath/crypto.html/
105	Web pages of interest to cryptography researchers	http://www.swcp.com/~mccurley/cryptographers/cryptographers.html
106	Applied Cryptography / Second Edition	http://www.counterpane.com/applied.html
107	Cryptography Theory and Practice	http://bibd.unl.edu/~stinson/CTAP.html
108	F-Secure Cryptography Products	http://www.datafellows.com/f-secure/
109	OECD ADOPTS GUIDELINES FOR CRYPTOGRAPHY POLICY	http://www.oecd.org/news_and_events/release/nw97-24a.htm
110	Η κρυπτογραφία είναι ένα από τα ισχυρότερα όπλα του πολίτη απέναντι στην κρατική αυθαιρεσία που χώνει τη μύτη της στις ιδιωτικές του υποθέσεις. Στις ΗΠΑ έχουν ήδη αντιληφθεί τη δύναμή της και στρατεύουν περίεργα νομικίστικα τεχνάσματα για να σταματήσουν τη διάδοσή της...	http://knet.compulink.gr/articles/crypto.htm
111	Ο πόλεμος της κρυπτογραφίας συνεχίζεται... Η αμερικανική εταιρία RSA ανακοίνωσε την κατασκευή ενός chip με κλειδί 1024 bits. Θα το κατασκευάζει όμως στην Ιαπωνία γαι να μπορεί και να το εξάγει. Οι πονοκέφαλοι των μυστικών υπηρεσιών και της κυβέρνησης των ΗΠΑ αυξάνονται...	http://knet.compulink.gr/articles/crypto2.htm
111	Μια δικαστική απόφαση στις ΗΠΑ φέρνει τα πάνω κάτω για την κρυπτογραφία και την κυβέρνηση Κλίντον σε πολύ δύσκολη θέση. Η απόφαση της δικαστού Marylin	http://knet.compulink.gr/articles/crypto3.htm

	Patel είναι ιστορική: Ο κώδικας ενός προγράμματος είναι έκφραση και προστατεύεται από το Σύνταγμα ...	
112	Το Pretty Good Privacy εξασφαλίζει το απόρρητο των επικοινωνιών όσον αφορά το ηλεκτρονικό ταχυδρομείο. Η κυβέρνηση Κλίντον δεν μπόρεσε να το σταματήσει. Ο "Ναυτίλος" είναι ένας δεύτερος και μεγαλύτερος πονοκέφαλος για τις μυστικές υπηρεσίες όλου του κόσμου.	http://knet.compulink.gr/articles/nautil.htm
113	Η "απειλή" της κρυπτογραφίας...	http://knet.compulink.gr/articles/arti123/arti7.htm
114	Ηλεκτρονικό Εμπόριο - Ασφάλεια των συναλλαγών	http://www.compuweb.gr/tech-update/ecommerce-security.asp
115	Πως λειτουργεί η κρυπτογράφηση με την μέθοδο του δημόσιου και του ιδιωτικού κλειδιού (public-key cryptography)	http://www.eexi.gr/928/interbiz/answers/publkey.html

Ψηφιακή Υπογραφή

α/α	ΘΕΜΑ	ΔΙΕΥΘΥΝΣΗ
1	Basics of Public-Key Cryptography / Two major applications for public-key systems (Distribution of secret keys - Digital signatures)	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_9.html
2	Basics of Public-Key Cryptography / The Digital Signature Algorithm.	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_18.html
3	FAQ for sci.crypt, part 7: Digital Signatures and Hash Functions	http://stasi.bradley.edu/privacy/sci.crypt_FAQ_7.html#WhatDiff
4	More on Protocols / Blind signatures	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_43.html
5	More on Protocols / Voting with Blind Signatures	http://www.cs.nps.navy.mil/curricula/tracks/security/notes/chap05_46.html
6	Utah Digital Signature Program	http://www.commerce.state.ut.us/web/commerce/digsig/dsmain.htm
7	Utah Digital Signature Act(1996)	http://www.commerce.state.ut.us/web/commerce/digsig/act.htm
8	Creating trust in electronic commerce	http://www.digsigtrust.com/
9	State of Utah Licensed Certification Authorities	http://www.digsigtrust.com/crl/utahdcmrc.html
10	Overview of the Electronic Authentication Act	http://www.wa.gov/sec/ea/overview.htm
11	Frequently asked questions about digital signatures	http://www.wa.gov/sec/ea/dsfaq.htm
12	Public-Key Infrastructure (X.509) (pkix)	http://www.ietf.cnri.reston.va.us/html.charters/pkix-charter.html
13	Digital signatures and the Electronic Authentication Act	http://www.wa.gov/sec/ea/dsdesc.htm
14	Washington State's Electronic Authentication Act (EAA)	http://www.wa.gov/sec/ea.htm
15	Certification Authorities Licensed by the Utah Digital Signature Program	http://www.commerce.state.ut.us/web/commerce/digsig/licensed.htm
16	Τι είναι η ηλεκτρονική υπογραφή (digital signature)	http://www.eexi.gr/928/interbiz/answers/digsig.html

Βιβλιογραφικές Αναφορές / Papers

1. [ACG84] W. Alexi, B. Chor, O. Goldreich, and C.P. Schnorr. RSA and Rabin functions: Certain parts are as hard as the whole. *SIAM Journal of Computing*, October 1984.
2. [Adl94] L.M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266: 021–1024, November 1994.
3. [Adl95] L.M. Adleman. On constructing a molecular computer, University of Southern California, draft, January 1995.
4. [Adl96] L.M. Adleman. Statement, Cryptographer's Expert Panel, RSA Data Security Conference, San Francisco, CA, January 17, 1996.
5. [AGL95] D. Atkins, M. Graff, A.K. Lenstra and P.C. Leyland. The magic words are squeamish ossifrage. In *Advances in Cryptology — Asiacrypt '94*, pages 263–Springer-Verlag, 1995.
6. [ANS83] American National Standards Institute. *American National Standard X3.106:Data Encryption Algorithm, Modes of Operations,* 1983.
7. [ANS85] American National Standards Institute. *American National Standard X9.17:Financial Institution Key Management (Wholesale)*, 1985.
8. [ANS86a] American National Standards Institute. *American National Standard X9.9:Financial Institution Message Authentication (Wholesale)*, 1986.
9. [ANS86b] American National Standards Institute. *American National Standard X9.19:Financial Institution Retail Message Authentication*, 1986.
10. [ANS93a] American National Standards Institute. *Draft: American National Standard X9.30-199X: Public-Key Cryptography Using Irreversible Algorithms for the Financial Services Industry: Part 1: The Digital Signature Algorithm (DSA)*. American Bankers Association, March 1993.
11. [ANS93b] American National Standards Institute. *American National Standard X9.31-Public Key Cryptography Using Reversible Algorithms for the Financial Services Industry: Part 1: The RSA Signature Algorithm*, March 1993.
12. [ANS93c] American National Standards Institute. *American National Standard X9.31-Public Key Cryptography Using Reversible Algorithms for the Financial Services Industry: Part 2: The MDC-2 Hash Algorithm*, June 1993.
13. [ANS94a] American National Standards Institute. *Accredited Standards Committee X9 Working Draft: American National Standard X9.42-1993: Public Key Cryptography for the Financial Services Industry: Management of Symmetric Algorithm Keys Using Diffie-Hellman*, American Bankers Association, September 21, 1994.
14. [ANS94b] American National Standards Institute. *Accredited Standards Committee X9 Working Draft: American National Standard X9.44: Public Key Cryptography Using Reversible Algorithms for the Financial Services Industry: Transport of - Symmetric Algorithm Keys Using RSA*, American Bankers Association, September 21, 1994.
15. [ANS95] American National Standards Institute. *Accredited Standards Committee X9 Working Draft: American National Standard X9.57: Certificate Management*, American Bankers Association, 1995.
16. [Atk95a] R. Atkinson. *RFC 1825: Security Architecture for the Internet Protocol*. Naval Research Laboratory, August 1995.
17. [Atk95b] R. Atkinson. *RFC 1826: IP Authentication Header*. Naval Research Laboratory, August 1995.
18. [Atk95c] R. Atkinson. *RFC 1827: IP Encapsulating Security Payload (ESP)*. Naval Research Laboratory, August 1995.

19.[Bam82] J. Bamford. *The Puzzle Palace.* Houghton Mifflin, Boston, 1982. [Bar92] J.P. Barlow. Decrypting the puzzle palace. *Communications of the ACM*, : 25–31, July 1992.
20.[BBB92] C. Bennett, F. Bessette, G. Brassard, L. Savail, and J. Smolin. Experimental quantum cryptography. *Journal of Cryptology*, 5(1): 3–28, 1992.
21.[BBC88] P. Beauchemin, G. Brassard, C. Crepeau, C. Goutier, and C. Pomerance. The generation of random numbers that are probably prime. *Journal of Cryptology*, 1: 53–64, 1988.
22.[BBL95] D. Bleichenbacher, W. Bosma, and A. Lenstra. Some remarks on Lucas-based cryptosystems. In *Advances in Cryptology – Crypto '95*, pages 386–396, Springer-Verlag, 1995.
23.[BBS86] L. Blum, M. Blum, and M. Shub. A simple unpredicatable random number generator. *SIAM Journal on Computing*, 15: 364–383, 1986.
24.[BD93b] J. Brandt and I. Damgard. On generation of probable primes by incremental search. In *Advances in Cryptology — Crypto '92*, pages 358–370, Springer-Verlag, 1993.
25.[BDB92] M.V.D. Burmester, Y.G. Desmedt, and T. Beth. Efficient zero-knowledge identification schemes for smart cards. *Computer Journal*, 35: 21–29, 1992.
26.[BDK93] E.F. Brickell, D.E. Denning, S.T. Kent, D.P. Maher, and W. Tuchman. S*kipjack Review, Interim Report: The Skipjack Algorithm*. July 28, 1993.
27.[Bea95] D. Beaver. Factoring: The DNA solution. In *Advances in Cryptology — Asiacrypt '94*, pages 419–423, Springer-Verlag, 1995.
28.[Ben82] P. Benioff. Quantum mechanical Hamiltonian models of Turing machines. *Journal of Statistical Physics*, 29(3): 515–546, 1982.
29.[BG85] M. Blum and S. Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In *Advances in Cryptology — Crypto '84*, pages 289–299, Springer-Verlag, 1985.
30.[BGH95] M. Bellare, J.A. Garay, R. Hauser, A. Herzberg, H. Krawczyk, M. Steiner, Tsudik, and M. Waidner. *iKP* - A Family of Secure Electronic Payment Protocols. Usenix Electronic Commerce Workshop, July 1995.
31.[BHS93] D. Bayer, S. Haber, and W.S. Stornetta. Improving the efficiency and reliability of digital timestamping. In *Proceedings Sequences II: Methods in Communication, Security, and Computer Science*, pages 329–334, Springer-Verlag,
32.[Bih95] E. Biham. Cryptanalysis of Multiple Modes of Operation. In *Advances in Cryptology — Asiacrypt '94*, pages 278–292, Springer-Verlag, 1995.
33.[BKR94] M. Bellare, J. Killian and P. Rogaway. The security of cipher block chaining. In *Advances in Cryptology — Crypto '94*, pages 341–358, Springer-Verlag,
34.[Bla79] G.R. Blakley. Safeguarding cryptographic keys. *AFIPS Conference Proceedings*, 48: 313–317, 1979.
35.[BLP94] J.P. Buhler, H.W. Lenstra, and C. Pomerance. The development of the number field sieve. Volume 1554 of *Lecture Notes in Computer Science,* Springer-Verlag, 1994.
36.[BLS88] J. Brillhart, D.H. Lehmer, J.L. Selfridge, B. Tuckerman, and S.S. Wagstaff Jr. *Factorizations of b n ± 1, b* = 2,3,5,6,7,10,11,12 *up to High Powers*. Volume 22 of *Contemporary Mathematics*, American Mathematical Society, 2nd edition.
37.[BLS95] J. Benaloh, B. Lampson, D. Simon, T. Spies, and B. Yee. *The Private Communication Technology Protocol*. Version 1.00, Microsoft Corporation, Redmond, WA, October 1995. <http://www.microsoft.com/windows/ ie/PCT.htm/>
38.[BLZ94] J. Buchmann, J. Loho, and J. Zayer. An implementation of the general number field sieve. In *Advances in Cryptology — Crypto '93*, pages 159–166, Springer-Verlag, 1994.
39.[BM84] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4): 850–1984.

40.[BO88] E.F. Brickell and A.M. Odlyzko. Cryptanalysis: A survey of recent results. *Proceedings of the IEEE*, 76: 578–593, 1988.
41.[BR94] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *Advances in Cryptology — Eurocrypt '94*, pages 92–111, Springer-Verlag, 1994.
42.[Bra88] G. Brassard. *Modern Cryptology*. Volume 325 of *Lecture Notes in Computer Science*, Springer-Verlag, 1988.
43.[Bra93] G. Brassard. Cryptography column — Quantum cryptography: A bibliography. *Sigact News*, 24(3): 16–20, 1993.
44.[Bra95a] G. Brassard. The computer in the 21st Century. *Scientific American*. March
45.[Bra95b] G. Brassard. The impending demise of RSA? *CryptoBytes,* 1(1): 1–4, Spring
46.[Bra95c] G. Brassard. A quantum jump in computer science. *Current Trends in Computer Science*, LNCS 1000, Springer-Verlag, 1995.
47.[Bre89] D.M. Bressoud. *Factorization and Primality Testing*. Springer-Verlag, 1989.
48.[Bri85] E.F. Brickell. Breaking iterated knapsacks. In *Advances in Cryptology — Crypto '84*, pages 342–358, Springer-Verlag, 1985.
49.[BS91a] E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. In *Advances in Cryptology – Crypto '90*, pages 2–21, Springer-Verlag,
50.[BS91b] E. Biham and A. Shamir. Differential cryptanalysis of FEAL and N-Hash. In *Advances in Cryptology – Eurocrypt '91*, pages 156–171, Springer-Verlag.
51.[BS93a] E. Biham and A. Shamir. Differential cryptanalysis of the full 16-round DES. In *Advances in Cryptology — Crypto '92*, pages 487–496, Springer-Verlag,
52.[BS93b] E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer-Verlag, 1993.
53.[CCI88a] CCITT. Recommendation X.400: *Message Handling System and Service Overview.* 1988.
54.[CCI88b] CCITT. Recommendation X.500: *The Directory — Overview of Concepts, Models and Services.* 1988.
55.[CCI88c] CCITT. Recommendation X.509: *The Directory — Authentication Framework*.
56.[CCI91] CCITT. Recommendation X.435: *Message Handling Systems: EDI Messaging System*. 1991.
57.[CFG95] S. Crocker, N. Freed, J. Galvin, and S. Murphy. *RFC 1848: MIME Object Security Services.* CyberCash, Inc., Innosoft International, Inc., and Trusted Information Systems, October 1995.
58.[CFN88] D. Chaum, A. Fiat and M. Naor. Untraceable electronic cash. In *Advances in Cryptology — Crypto '88,* pages 319–327, Springer-Verlag, 1988.
59.[Cha83] D. Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology — Crypto '82,* pages 199–203, Springer-Verlag, 1983.
60.[Cha85] D. Chaum. Security without identification: transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10): 1030–1044, October
61.[Cha94] D. Chaum. Designated confirmer signatures. In *Advances in Cryptology — Eurocrypt '94*, pages 86–91, Springer-Verlag, 1994.
62.[CKM94] D. Coppersmith, H. Krawczyz and Y. Mansour. The shrinking generator. In *Advances in Cryptology — Crypto '93*, pages 22–38, Springer-Verlag, 1994.
63.[CLR90] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, 1990.
64.[Cop92] D. Coppersmith. The data encryption standard and its strength against attacks. *IBM Research Report* RC 18613 (81421), T. J. Watson research center, December 1992.
65.[COS86] D. Coppersmith, A.M. Odlyzko, and R. Schroeppel. Discrete logarithms. In *GF(p). Algorithmica*, 1: 1–15, 1986.

66.[CP94] L. Chen and T.P. Pederson. New group signature schemes. In *Advances in Cryptology — Eurocrypt '94*, pages 171–181, Springer-Verlag, 1994.
67.[CP95] L. Chen and T.P. Pedersen. On the efficiency of group signatures: providing information-theoretic anonymity. In *Advances in Cryptology — Eurocrypt '95*, pages 39–49, Springer-Verlag, 1995.
68.[CR88] B. Chor and R.L. Rivest. A knapsack-type public-key cryptosystem based on arithmetic in finite fields. *IEEE Transactions on Information Theory*, 34(5): 1988.
69.[CV90] D. Chaum and H. van Antwerpen. Undeniable signatures. In *Advances in Cryptology — Crypto '89*, pages 212–216, Springer-Verlag, 1990.
70.[CV91] D. Chaum and E. van Heijst. Group signatures. In *Advances in Cryptology — Eurocrypt '91*, pages 257–265, Springer-Verlag, 1991.
71.[CV92] D. Chaum and H. van Antwerpen. Cryptographically strong undeniable signatures, unconditionally secure for the signer. In *Advances in Cryptology — Crypto '91*, pages 470–484, Springer-Verlag, 1992.
72.[CW93] K.W. Campbell and M.J. Wiener. DES is not a group. In *Advances in Cryptology — Crypto '92*, pages 512–520, Springer-Verlag, 1993.
73.[Dam90] I. Damgɛrd. A design principle for hash functions. In *Advances in Cryptology — Crypto '89*, pages 416–427, Springer-Verlag, 1990.
74.[Dav82] G. Davida. *Chosen signature cryptanalysis of the RSA public key cryptosystem*. Technical Report TR-CS-82-2, Department of EECS, University of Wisconsin, Milwaukee, 1982.
75.[DB92] B. den Boer and A. Bosselaers. An attack on the last two rounds of MD4. In *Advances in Cryptology — Crypto '91*, pages 194–203, Springer-Verlag, 1992.
76.[DB94] B. den Boer and A. Bosselaers. Collisions for the compression function of MD5. In *Advances in Cryptology — Eurocrypt '93*, pages 293-304, Springer-Verlag,
77.[DB95] D.E. Denning and D.K. Branstad. *A taxonomy for key escrow encryption systems*. January, 1995.
78.[DBP96] H. Dobbertin, A. Bosselaers, and B. Preneel. RIPEMD-160: A strengthened version of RIPEMD. To appear in *3rd Workshop on Fast Software Encryption,*
79.[Den93] D.E. Denning. The Clipper encryption system. *American Scientist*, 81(4): July-August 1993.
80.[Den95] D.E. Denning. The Case for "Clipper." *Technology Review*, pages 48-55, July
81.[Des95] Y. Desmedt. Securing traceability of ciphertexts–Towards a secure software key escrow system. In *Advances in Cryptology — Eurocrypt '95*, pages 147– Springer-Verlag, 1995.
82.[Deu92] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society*, London, A439: 553–558, 1992.
83.[DGV94] J. Daemen, R. Govaerts, and J. Vandewalle. Weak keys for IDEA. In *Advances in Cryptology — Crypto '93*, pages 224–231, Springer-Verlag, 1994.
84.[DH76] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22: 644–654, 1976.
85.[DH77] W. Diffie and M.E. Hellman. Exhaustive cryptanalysis of the NBS Data Encryption Standard. *Computer*, 10: 74–84, 1977.
86.[Dif88] W. Diffie. The first ten years of public-key cryptography. *Proceedings of the IEEE*, 76: 560–577, 1988.
87.[DIP94] D. Davies, R. Ihaka, and P. Fenstermacher. Cryptographic randomness from air turbulence in disk drives. In *Advances in Cryptology — Crypto '94*, pages 114–120, Springer-Verlag, 1994.
88.[Div95] D.P. DiVincenzo. Two-bit gates are universal for quantum computation. *Physical Review A*, 51: 1015–1022, 1995.

89.[DL95] B. Dodson and A.K. Lenstra. NFS with four large primes: An explosive experiment. In *Advances in Cryptology – Crypto '95,* pages 372–385, Springer-Verlag, 1995.
90.[DO86] Y. Desmedt and A.M. Odlyzko. A chosen text attack on the RSA cryptosystem and some discrete logarithm schemes. In *Advances in Cryptology — Crypto '85*, pages 516–522, Springer-Verlag, 1986.
91.[Dob95] H. Dobbertin. Alf Swindles Ann. *CryptoBytes*, 1(3): 5, 1995.
92.[DP83] D.W. Davies and G.I. Parkin. The average cycle size of the key stream in output feedback encipherment. In *Advances in Cryptology: Proceedings of Crypto '82*, pages 97–98, Plenum Press, 1983.
93.[DRB95] P. Domokos, M.J. Raimond, M. Brune, and S. Haroche. A simple cavity-QED two-bit universal quantum logic gate: principle and expected performances. *Physical Review A*. To appear.
94.[DVW92] W. Diffie, P.C. van Oorschot, and M.J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2: 107–125,
95.[ECS94] D. Eastlake, 3rd, S. Crocker, and J. Schiller. *RFC 1750: Randomness Recommendations for Security*. DEC, Cybercash, and MIT, December 1994. [Elg85] T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31: 469–472, [Elg95] T. ElGamal. *Commerce on the Internet*. Version 1.00, Netscape
96.Communications Corporation, Mountain View, CA, July 14, 1995. <http://www.netscape.com/ newsref/std/credit.html>
97.[Fei73] H. Feistel. Cryptography and Computer Privacy, *Scientific American*, May 1973.
98.[Fey82] R.P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6): 467–488, 1982.
99.[Fey86] R.P. Feynman. Quantum mechanical computers. *Optic News*, February Reprinted in *Foundations of Physics*, 16(6): 507–531, 1986.
100.[FFS88] U. Feige, A. Fiat and A. Shamir. Zero-knowledge proofs of identity. *Journal of Cryptography*, 1: 66–94, 1988.
101.[For94] W. Ford. *Computer Communications Security — Principles, Standard Protocols and Techniques*, Prentice-Hall, New Jersey, 1994.
102.[FR95] P. Fahn and M.J.B. Robshaw. *Results from the RSA Factoring Challenge*. Technical Report TR-501, version 1.3, RSA Laboratories, January 1995.
103.[FS87] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology — Crypto '86*, pages 186–194, Springer-Verlag, 1987.
104.[FY94] M. Franklin and M. Yung. Blind Weak Signature and its Applications: Putting Non-Cryptographic Secure Computation to Work. In *Advances in Cryptology — Eurocrypt '94*, pages 67–76, Springer-Verlag, 1994.
105.[Gan95] R. Ganesan. Yaksha: Augmenting Kerberos with public key cryptography. In *Proceedings of the 1995 Internet Society Symposium on Network and Distributed Systems Security*, pages 132–143, IEEE Press, 1995.
106.[GC89] D. Gollman and W.G. Chambers. Clock-controlled shift registers: a review. *IEEE Journal on Selected Areas in Communications*, 7(4): 525–533, May 1989.
107.[Gib93] J.K. Gibson. Severely denting the Babidulin version of the McElience public key cryptosystem. In *Prepoceedings of the 4th IMA Conference on Cryptography and Coding*, 1993.
108.[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28: 270–299, 1984.

109.[GM93] D.M. Gordon and K.S. McCurley. Massively parallel computation of discrete logarithms. In *Advances in Cryptology — Crypto '92*, pages 312–323, Springer-Verlag, 1993.
110.[GMR86] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen message attack. *SIAM Journal on Computing,* 17(2): March 1988.
111.[Gor93] D.M. Gordon. Discrete logarithms in *GF(p)* using the number field sieve. *SIAM Journal of Computing*, 6(1): 124–138, February 1993.
112.[GPT91] E.M. Gabidulin, A.V. Paramonov, and O.V. Tretjakov. Ideals over a non-commutative ring and their application in cryptology. In *Advances in Cryptology – Eurocrypt '91*, pages 482–489, Springer-Verlag, 1991.
113.[GQ88] L.C. Guillou and J.J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In *Advances in Cryptology — Eurocrypt '88*, pages 123–128, Springer-Verlag,
114.[Has88] J. Hastad. Solving simultaneous modular equations of low degree. *SIAM Journal of Computing*, 17: 336–241, 1988.
115.[Hel80] M.E. Hellman. A cryptanalytic time-memory trade off. *IEEE Transactions on Information Theory*, IT-26: 401–406, 1980.
116.[Hic95] K.E.B. Hickman. *The SSL Protocol*. December 1995. <http://www.netscape.com/newsref/ std/ssl.html>
117.[HKM95] C. Harpes, G.G. Kramer, and J.L. Massey. A generalization of linear cryptanalysis and the applicability of Matsui's piling-up lemma. In *Advances in Cryptology — Eurocrypt '95*, pages 24–38, Springer-Verlag, 1995.
118.[HS91] S. Haber and W.S. Stornetta. How to timestamp a digital document. *Journal of Cryptology*, 3(2): 99–111, 1991.
119.[IBM95] IBM, Netscape, GTE, CyberCash, and MasterCard. *Secure Electronic Payment Protocol (SEPP).* Draft, Version 1.2, November 3, 1995. <http:// www.mastercard.com/Sepp/sepptoc.html>
120.[IEE95] IEEE Working Group P1363. Working Draft: IEEE 1363: Standard for RSA, Diffie-Hellman and Related Public-Key Cryptography. In preparation,
121.[ISO87] ISO DIS 8730. *Banking requirements for message authentication (wholesale).*
122. [ISO91] ISO/IEC 9979. *Data Cryptographic Techniques - Procedures for the Registration of Cryptographic Algorithms*. 1991.
123.[ISO92a] ISO/IEC 9798. *Entity authentication mechanisms using symmetric techniques*.
124.[ISO92b] ISO/IEC 10116. *Modes of operation for an n-bit block cipher algorithm.* 1992.
125.[ISO92c] ISO/IEC 10118. *Information technology - Security techniques - Hash functions*.
126. [JML93] D.B. Johnson, S.M. Matyas, A.V. Le, and J.D. Wilkins. Design of the commercial data masking facility data privacy algorithm. In *Proceedings of the 1st ACM Conference on Communications and Computer Security*, ACM Press, VA, 1993.
127.[Jue83] R.R. Jueneman. Analysis of certain aspects of output feedback mode. In *Advances in Cryptology: Proceedings of Crypto '82*, pages 99–127, Plenum Press, 1983.
128.[Kah67] D. Kahn. *The Codebreakers*. Macmillan Co., New York, 1967.
129.[Kal92] B.S. Kaliski Jr. *RFC 1319: The MD2 Message-Digest Algorithm*. RSA Laboratories, April 1992.
130.[Kal93a] B.S. Kaliski Jr. *RFC 1424: Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services*. RSA Laboratories, February 1993.

131.[Kal93b] B.S. Kaliski Jr. A survey of encryption standards. *IEEE Micro*, 13(6): 74–81, December 1993.
132.[Kal95] B.S. Kaliski Jr. A chosen message attack on Demytko's cryptosystem. Journal of Cryptology. To appear.
133.[Ken93] S. Kent. *RFC 1422: Privacy Enhancement for Internet Electronic Mail, Part II: Certificate-Based Key Management*. Internet Activities Board, February 1993.
134.[KMS95] P. Karn, P. Metzger, and W. Simpson. *RFC 1829: The ESP DES-CBC Transform*. Qualcomm, Piermont, and Daydreamer, August 1995.
135.[KN93] J. Kohl and B. Neuman. *The Kerberos Network Authentication Service*. Network Working Group RFC 1510, 1993.
136.[KNT94] J. Kohl, B. Neuman, and T. Tso. The evolution of the Kerberos authentication service. *Distributed Open Systems*, IEEE Press, 1994.
137.[Knu81] D.E. Knuth. *The Art of Computer Programming*, volume 2, *Seminumerical Algorithms*. Addison-Wesley, 2nd edition, 1981.
138.[Knu93] L.R. Knudsen. Practically secure Feistel ciphers. In *Proceedings of 1st Workshop* on *Fast Software Encryption*, pages 211–221, Springer-Verlag, 1993.
139.[Knu95] L.R. Knudsen. A key-schedule weakness in SAFER K-64. In *Advances in Cryptology — Crypto '95*, pages 274–286, Springer-Verlag, 1995.
140.[KO95] K. Kurosawa and K. Okada. Low exponent attack against elliptic curve RSA. In *Advances in Cryptology — Asiacrypt '94*, pages 376–383, Springer-Verlag,
141.[Kob87] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48: 1987.
142.[Kob94] N. Koblitz. *A Course in Number Theory and Cryptography*. Springer-Verlag,
143.[Koc94] H.K. Κοη. *High-Speed RSA Implementation*. Technical Report TR-201, version RSA Laboratories, November 1994.
144.[KR94] B.S. Kaliski Jr. and M.J.B. Robshaw. Linear cryptanalysis using multiple approximations. In *Advances in Cryptology — Crypto '94*, pages 26–39, Springer-Verlag, 1994.
145.[KR95a] B.S. Kaliski Jr. and M.J.B. Robshaw. Linear cryptanalysis using multiple approximations and FEAL. In *Proceedings of 2nd Workshop on Fast Software Encryption*, pages 249–264, Springer-Verlag, 1995.
146.[KR95b] B.S. Kaliski Jr. and M.J.B. Robshaw. Message authentication with MD5. *CryptoBytes,* 1(1): 5–8, 1995.
147.[KR95c] B.S. Kaliski Jr. and M.J.B. Robshaw. The secure use of RSA. *CryptoBytes,* : 7–13, 1995.
148.[KR96] B.S. Kaliski Jr. and M.J.B. Robshaw. *Multiple encryption: weighing up security and performance*. *Dr. Dobb's Journal*, #243, pages 123–127, January 1996.
149.[Kra93] D. Kravitz. Digital signature algorithm. U.S. Patent #5,231,668, July 27,
150.[KRS88] B.S. Kaliski Jr., R.L. Rivest, and A.T. Sherman. Is the data encryption standard a group? *Journal of Cryptology*, 1: 3–36, 1988.
151.[KT91] V.I. Korzhik and A.I. Turkin. Cryptanalysis of McEliece's public-key cryptosystem. In *Advances in Cryptology — Eurocrypt '91*, pages 68–70, Springer-Verlag, 1991.
152.[KY95] B.S. Kaliski Jr. and Y.L. Yin. On differential and linear cryptanalysis of the RC5 encryption algorithm. In *Advances in Cryptology — Crypto '95*, pages Springer-Verlag, 1995.
153.[Lan88] S. Landau. Zero knowledge and the Department of Defense. *Notices of the American Mathematical Society*, 35: 5–12, 1988.
154.[Len87] H.W. Lenstra Jr. Factoring integers with elliptic curves. *Annuals of Mathematics.*, 126: 649–673, 1987.

155.[LH94] S.K. Langford and M.E. Hellman. Differential-linear cryptanalysis. In *Advances in Cryptology — Crypto '94*, pages 17–25, Springer-Verlag, 1994.
156.[Lin93] J. Linn. *RFC 1508: Generic Security Services Application Programming Interface*. Geer Zolot Associates, September 1993.
157.[Lip94] R.J. Lipton. Speeding up computations via molecular biology. Princeton University, draft, December 1994.
158.[LL90] A.K. Lenstra and H.W. Lenstra Jr. Algorithms in number theory. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, pages MIT Press/Elsevier, Amsterdam, 1990.
159.[LLM93] A.K. Lenstra, H.W. Lenstra Jr., M.S. Manasse, and J.M. Pollard. The factorization of the ninth Fermat number. *Mathematics of Computation*, : 319–349, 1993.
160.[LM91a] X. Lai and J.L. Massey. A proposal for a new block encryption standard. In *Advances in Cryptology — Eurocrypt '90*, pages 389–404, Springer-Verlag,
161.[LM91b] A.K. Lenstra and M.S. Manasse. Factoring with two large primes. In *Advances in Cryptology — Eurocrypt '90*, pages 72–82, Springer-Verlag, 1991.
162.[LMM92] X. Lai, J.L. Massey and S. Murphy. Markov ciphers and differential cryptanalysis. In *Advances in Cryptology — Eurocrypt '91*, pages 17–38, Springer-Verlag, 1992.
163.[LO91] B.A. LaMacchia and A.M. Odlyzko. Computation of discrete logarithms in prime fields. *Designs, Codes and Cryptography*, 1: 47–62, 1991.
164.[LRW92] X. Lai, R.A. Rueppel, and J. Woollven. A fast cryptographic checksum algorithm based on stream ciphers. In *Advances in Cryptology — Auscrypt '92*, Springer-Verlag, 1992.
165.[Mas93] J.L. Massey. SAFER K-64: A byte-oriented block ciphering algorithm. In *Proceedings of 1st Workshop on Fast Software Encryption*, pages 1–17, Springer-Verlag, 1993.
166.[Mas95] J.L. Massey. SAFER K-64: One year later. In *Proceedings of 2nd Workshop on Fast Software Encryption*, pages 212–241, Springer-Verlag, 1995.
167.[Mat93] M. Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology — Eurocrypt '93*, pages 386–397, Springer-Verlag, 1993.
168.[Mat94] M. Matsui. The first experimental cryptanalysis of the data encryption standard. In *Advances in Cryptology — Crypto '94*, pages 1–11, Springer-Verlag,
169.[Mat96] T. Matthews. Suggestions for random number generation in software. Bulletin No. 1, RSA Laboratories, January 1996.
170.[Mau94] U. Maurer. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms. In *Advances in Cryptology — Crypto '94*, pages 271–281, Springer-Verlag, 1994.
171.[Mce78] R.J. McEliece. A public-key cryptosystem based on algebraic coding theory. *JPL DSN Progress Report 42–44*, pages 114–116, 1978.
172.[Mcn95] F.L. McNulty. Clipper – Alive and well as a voluntary government standard for telecommunications. *The 1995 RSA Data Security Conference*, January 1995.
173.[Men93] A. Menezes. *Elliptic Curve Public Key Cryptosystems*. Kluwer Academic Publishers, 1993.
174.[Mer79] R.C. Merkle. Secrecy, authentication and public-key systems. Ph. D. Thesis, Stanford University, 1979.
175.[Mer90a] R.C. Merkle. One way hash functions and DES. In *Advances in Cryptology — Crypto '89*, pages 428–446, Springer-Verlag, 1990.
176.[Mer90b] R.C. Merkle. A digital signature based on a conventional encryption function. In *Advances in Cryptology — Crypto '89*, pages 428–446, Springer-Verlag,

177.[Mer91] R.C. Merkle. Fast software encryption functions. In *Advances in Cryptology — Crypto '90*, pages 627–638, Springer-Verlag, 1991.
178.[MH78] R.C. Merkle and M.E. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, IT-24: 525–530,
179.[MH81] R.C. Merkle and M.E. Hellman. On the security of multiple encryption. *Communications of the ACM*, 24: 465–467, July 1981.
180.[Mic93] S. Micali. Fair public-key cryptosystems. In *Advances in Cryptology — Crypto '92*, pages 113–138, Springer-Verlag, 1993.
181.[Mic95] Microsoft Corporation. *STT Wire Formats and Protocols*. Version 0.902, Redmond, WA, October 5, 1995. <http://www.microsoft.com/windows/ ie/STT.htm>
182.[Mil86] V.S. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology — Crypto '85*, pages 417–426, Springer-Verlag, 1986.
183.[MOV90] A. Menezes, T. Okamoto, and S. Vanstone. *Reducing elliptic curve logarithms to logarithms in a finite field.* Unpublished manuscript, September 1990.
184.[MQV95] A. Menezes, M. Qu, and S. Vanstone. Some new key agreement protocols providing implicit authentication. In *Preproceedings of Workshops on Selected Areas in Cryptography*, 1995.
185.[MS95] P. Metzger and W. Simpson. *RFC 1828: IP Authentication using Keyed MD5*. Piermont and Daydreamer, August 1995.
186.[MS95] W. Meier and O. Staffelbach. The self-shrinking generator. In *Advances in Cryptology — Eurocrypt '94*, pages 205–214, Springer-Verlag, 1995.
187.[Mur90] S. Murphy. The cryptanalysis of FEAL-4 with 20 chosen plaintexts. *Journal of Cryptology*, 2(3): 145–154, 1990.
188.[MY92] M. Matsui and A. Yamagishi. A new method for known plaintext attack of FEAL cipher. In *Advances in Cryptology — Eurocrypt '92*, pages 81–91, Springer-Verlag, 1992.
189.[NIS80] National Institute of Standards and Technology (NIST). *FIPS Publication 81: DES Modes of Operation*. December 2, 1980. Originally issued by National Bureau of Standards.
190.[NIS85] National Institute of Standards and Technology (NIST). *FIPS Publication Computer Data Authentication*. 1985.
191.[NIS92] National Institute of Standards and Technology (NIST). The Digital Signature Standard, proposal and discussion. *Communications of the ACM*, : 36–54, July 1992.
192.[NIS93a] National Institute of Standards and Technology (NIST). *FIPS Publication Secure Hash Standard (SHS)*. May 1993.
193.[NIS93b] National Institute of Standards and Technology (NIST). *FIPS Publication 46- Data Encryption Standard*. December 1993.
194.[NIS94a] National Institute of Standards and Technology (NIST). *FIPS Publication Escrowed Encryption Standard*. February 1994.
195.[NIS94b] National Institute of Standards and Technology (NIST). *FIPS Publication Digital Signature Standard (DSS)*. May 1994.
196.[NIS94c] National Institute of Standards and Technology (NIST). *Announcement of Weakness in the Secure Hash Standard*. May 1994.
197.[NK95] K. Nyberg and L.R. Knudsen. Provable security against a differential attack. *Journal of Cryptology*, 8(1): 27–37, 1995.
198.[NMR94] D. Naccache, D. M'raohi, D. Raphaeli, and S. Vaudenay. Can D.S.A. be improved? Complexity trade-offs with the Digital Signature Standard. In *Advances in Cryptology — Eurocrypt '94*, pages 77–85, Springer-Verlag, 1994.
199.[NS78] R.M. Needham and M.D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21: 993–999,

200.[NS94] M. Naor and A. Shamir. Visual cryptography. In *Advances in Cryptology — Eurocrypt* ’94, pages 1–12, Springer-Verlag, 1994.
201.[NSA95] NSA Cross Organization CAPI Team. *Security Service API: Cryptographic API Recommendation*, 1995.
202.[Nyb95] K. Nyberg. Linear approximation of block ciphers. In *Advances in Cryptology — Eurocrypt ’94* (rump session), pages 439–44, Springer-Verlag,
203.[OA94] K. Ohta and K. Aoki. Linear cryptanalysis of the fast data encipherment algorithm. *In Advances in Cryptology — Crypto ’94*, pages 12–16, Springer-Verlag,
204.[Oco95] L. O’Connor. A unified markov approach to differential and linear cryptanalysis. In *Advances in Cryptology — Asiacrypt ’94*, pages 387–397, Springer-Verlag, 1995.
205.[Odl84] A.M. Odlyzko. Discrete logarithms in finite fields and their cryptographic significance. In *Advances in Cryptology — Eurocrypt ’84*, pages 224–314, Springer-Verlag, 1984.
206.[Odl95] A.M. Odlyzko. The future of integer factorization. *CryptoBytes,* 1(2): 5–12,
207.[Oka93] T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *Advances in Cryptology — Crypto ’92*, pages 31–53, Springer-Verlag, 1993.
208.[OPS93] Office of the Press Secretary. *Statement*. The White House, April 16, 1993.
209.[Pol74] J. Pollard. Theorems of factorization and primality testing. *Proceedings of Cambridge Philosophical Society*, 76: 521–528, 1974.
210.[Pol75] J. Pollard. Monte Carlo method for factorization. *BIT*, 15: 331–334, 1975.
211.[Pre93] B. Preneel. *Analysis and Design of Cryptographic Hash Functions*. Ph.D. Thesis, Katholieke University Leuven, 1993.
212.[Pre94] B. Preneel. The State of DES. *1994 RSA Laboratories Seminar Series*, August
213.[QG90] J.J. Quisquater and L. Guillou. How to explain zero-knowledge protocols to your children. In *Advances in Cryptology — Crypto ’89*, pages 628–631, Springer-Verlag, 1990.
214.[Rab79] M.O. Rabin. *Digitalized signatures and public-key functions as intractable as factorization*. Technical Report MIT/LCS/TR-212, MIT, 1979.
215.[RC93] P. Rogaway and D. Coppersmith. A software-optimized encryption algorithm. In *Proceedings of 1st Workshop on Fast Software Encryption*, pages Springer-Verlag, 1993.
216.[RC95] N. Rogier and P. Chauvaud. The compression function of MD2 is not collision free. Presented at *Selected Areas in Cryptography ’95*, Ottawa, Canada, May 18–19, 1995.
217.[RG91] D. Russell and G.T. Gangemi Sr. *Computer Security Basics*. O’Reilly & Associates, Inc., 1991.
218.[Riv90] R.L. Rivest. Cryptography. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, pages 719–755, MIT Press/Elsevier, Amsterdam, 1990.
219.[Riv91a] R.L. Rivest. Finding four million random primes. In *Advances in Cryptology — Crypto ’90*, pages 625–626, Springer-Verlag, 1991.
220.[Riv91b] R.L. Rivest. The MD4 message digest algorithm. In *Advances in Cryptology — Crypto ’90*, pages 303–311, Springer-Verlag, 1991.
221.[Riv92a] R.L. Rivest. Response to NIST’s proposal. *Communications of the ACM*, 35: July 1992.
222.[Riv92b] R.L. Rivest. *RFC 1320: The MD4 Message-Digest Algorithm*. Network Working Group, April 1992.
223.[Riv92c] R.L. Rivest. *RFC 1321: The MD5 Message-Digest Algorithm*. Internet Activities Board, April 1992.

224.[Riv95] R.L. Rivest. The RC5 encryption algorithm. *CryptoBytes,* 1(1): 9–11, 1995.
225.[Rob95a] M.J.B. Robshaw. *Block Ciphers*. Technical Report TR-601, version 2.0, RSA Laboratories, August 1995.
226.[Rob95b] M.J.B. Robshaw. *Stream Ciphers*. Technical Report TR-701, version 2.0, RSA Laboratories, July 1995.
227.[Rob95c] M.J.B. Robshaw. *MD2, MD4, MD5, SHA and Other Hash Functions.*Technical Report TR-101, version 4.0, RSA Laboratories, July 1995.
228.[Rob95d] M.J.B. Robshaw. *Security estimates for 512-bit RSA*. Technical Note, RSA Laboratories, June 1995.
229.[RS95] E. Rescorla and A. Schiffman. *The Secure HyperText Transfer Protocol*. Internet-Draft, EIT, July 1995.
230.[RSA78] R.L. Rivest, A. Shamir, and L.M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, : 120–126, February 1978.
231.[RSA95] RSA Laboratories. *PKCS #11: Cryptographic Token Interface Standard*. Version April 1995.
232.[Rue92] R.A. Rueppel. Stream ciphers. In *Contemporary Cryptology — The Science of Information Integrity*. IEEE Press, 1992.
233.[SB93] M.E. Smid and D.K. Branstad. Response to comments on the NIST proposed Digital Signature Standard. In *Advances in Cryptology — Crypto '92*, pages 76–87, Springer-Verlag, 1993.
234.[Sch83] I. Schaumuller-Bichl. Cryptanalysis of the Data Encryption Standard by a method of formal coding. *Cryptography, Proc. Burg Feuerstein 1982*, 149: 235–Berlin,1983.
235.[Sch90] C.P. Schnorr. Efficient identification and signatures for smart cards. In *Advances in Cryptology — Crypto '89*, pages 239–251, Springer-Verlag, 1990.
236.[Sch91] C.P. Schnorr. Method for identifying subscribers and for generating and verifying electronic signatures in a data exchange system. U.S. Patent February 19, 1991.
237.[Sch93] B. Schneier. Description of a new variable-length key, 64-bit block cipher (Blowfish). In *Proceedings of 1st Workshop on Fast Software Encryption*, pages Springer-Verlag, 1993.
238.[Sch95a] B. Schneier. The Blowfish encryption algorithm: one year later. *Dr. Dobb's Journal*, No. 234, pages 137–138, September 1995.
239.[Sch95b] B. Schneier. *Applied Cryptography*: *Protocols, Algorithms, and Source Code in C.* Wiley, 2nd Edition, 1995.
240.[SH95] C.P. Schnorr and H.H. Hφrner. Attacking the Chor-Rivest cryptosystem by improved lattice reduction. In *Advances in Cryptology — Eurocrypt '95*, pages 1–12, Springer-Verlag, 1995.
241.[Sha49] C.E. Shannon. Communication Theory of Secrecy Systems. *Bell Systems Technical Journal*, 28: 656–715, October 1949.
242.[Sha79] A. Shamir. How to share a secret. *Communications of the ACM*, 22: 612–613,
243.[Sha84] A. Shamir. A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem. *IEEE Transactions on Information Theory*, IT-30(5):September 1984.
244.[Sha95] M. Shand. Personal communication. 1995.
245.[Sho94] P.W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Annual IEEE Symposium on the Foundations of Computer Science*, pages 124–134, 1994.
246.[Sil87] R.D. Silverman. The multiple polynomial quadratic sieve. *Mathematics of Computation*, 48: 329–339, 1987.

247.[Sim92] G.J. Simmons, editor. *Contemporary Cryptology — The Science of Information Integrity*. IEEE Press, 1992.
248.[SM88] A. Shimizu and S. Miyaguchi. Fast data encipherment algorithm FEAL. In *Advances in Cryptology — Eurocrypt '87*, pages 267–280, Springer-Verlag,
249.[SPC95] M. Stadler, J.M. Piveteau, and J. Carmenisch. Fair blind signatures. In *Advances in Cryptology — Eurocrypt '95*, pages 209–219, Springer-Verlag,
250.[SS95] P. Smith and C. Skinner. A public-key cryptosystem and a digital signature system based on the Lucas function analogue to discrete logarithms. In *Advances in Cryptology — Asiacrypt '94*, pages 357–364, Springer-Verlag,
251.[Sta95] W. Stallings. *Network and Internetwork Security – Principles and Practice*. Prentice-Hall, New Jersey, 1995.
252.[Sti95] D.R. Stinson. *Cryptography — Theory and Practice*. CRC Press, Boca Raton,
253.[SV93] M. Shand and J. Vuillemin. Fast implementations of RSA cryptography. In *Proceedings of the 11th IEEE Symposium on Computer Arithmetic*, pages 252– IEEE Computer Society Press, 1993.
254.[Ver26] G.S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *J. Amer. Inst. Elec. Eng.*, vol. 45, pages 109–115,
255.[Vis95] Visa International. *Secure Transaction Technology Specifications*, Version 1.0, September 26, 1995. (http://www.visa.com/visa-stt/)
256.[VP92] E. van Heyst and T.P. Pederson. How to make efficient fail-stop signatures. In *Advances in Cryptology — Eurocrypt '92*, pages 366–377, Springer-Verlag,
257.[VW91] P. van Oorschot and M. Wiener. A known plaintext attack on two-key triple encryption. In *Advances in Cryptology — Eurocrypt '90*, pages 318–325, Springer-Verlag, 1991.
258.[VW94] P. van Oorschot and M. Wiener. Parallel collision search with application to hash functions and discrete logarithms. In *Proceedings of 2nd ACM Conference on Computer and Communication Security*, 1994.
259.[Wie94] M.J. Wiener. Efficient DES key search. Technical Report TR–244, School of Computer Science, Carleton University, Ottawa, Canada, May 1994.
260.[Xop95] X/Open Company Ltd. *Generic Cryptographic Service API (GCS-API).* Base – Draft 3, April 1995.
261.[Yuv79] G. Yuval. How to swindle Rabin. *Cryptologia*, July 1979.
262.[ZPS93] Y. Zheng, J. Pieprzyk and J. Seberry. HAVAL - a one-way hashing algorithm with variable length output. In *Advances in Cryptology — Auscrypt '92*, pages 83–104, Springer-Verlag, 1993.