

PKI

Johannes Buchmann

April 10, 2005

Contents

1	Public-key cryptography	1
1.1	Applications	1
1.2	Public-key encryption	2
1.3	Digital signatures	4
1.4	Security of public-key cryptography	5
1.5	Key management in public-key infrastructures	6
1.6	Timestamping	6
1.7	Further applications	6
2	Network security	7
2.1	The OSI model	7
2.2	IP	8
2.3	IPSec	9
2.4	TCP	10
2.5	SSL	10
2.6	Application layer	11
	Bibliography	13
	Subject index	15

Chapter 1

Public-key cryptography

Public-key cryptography is used to make open computer networks such as the Internet more secure. Public-key cryptography helps to achieve the following security goals and provides the following security services.

Confidentiality The property that data are not made available or disclosed to unauthorized individuals, entities or processes.

Identification A process through which one ascertains the identity of another person or entity.

Integrity The property that data have not been altered or corrupted.

Authentication A process through which one ascertains the integrity and origin of data.

Non-repudiation A process that provides proof authenticity of data which can be verified by any third party.

Time stamping Proof that data existed at a given time.

The public-key techniques used for this purpose are public-key encryption and digital signatures. We first demonstrate the relevance of the security goals in a few application scenarios. Then we explain the core public-key techniques.

1.1 Applications

The world wide web (WWW) has become the most important information and communication medium in the world.

On web pages, people obtain relevant information. However, forging web pages is very simple. In 2003 a web page appeared on the Internet which looked exactly as a CNN web page and said “Microsoft chairman Bill gates murdered at Los Angeles charity event”. Of course, this page was a fake and not an original CNN page. However, the rumors caused the Korean stock market to drop by 1.5 % - a value loss of more than US \$ 3 billion - after a local TV was fooled into reporting that Bill Gates had been gunned down. This web page was not an *authentic* CNN page. But this was not easy to find out.

Bild einfügen: CNN Page.

Sometimes, web pages also need to be kept confidential. Pilots of airlines want to access their flight schedule on a web page from home. If this web page is not kept *confidential*, everybody can see the flight schedule. If a pilot lives by himself, has a long international

flight, and his flight schedule is public, then thieves can rob his apartment without being disturbed.

Modern operating systems such as Microsoft Windows XP support automatic updates. Automatic Update of Microsoft Windows XP checks the Windows Update Web site for critical updates and automates the process of downloading and installing the critical updates. If updates are not available, this feature is reset, and then it checks again in 24 hours for any new updates after the Internet connection is established. If a download operation is interrupted, this feature resumes at a later time when the Internet connection is re-established. It is crucial that the automatic Windows XP updates are *authentic*. If an attacker were able to insert faked updates into the automatic update process he could easily paralyze a serious part of the world's IT infrastructure. Also, the users may be interested in being able to prove that a certain update was in fact sent by the operating system manufacturer. So the updates should be *non-repudiable*.

Bild einfügen: Windows XP update

For email communication, *confidentiality*, *authenticity*, and *non-repudiation* is a big issue. For example, spam mail, that is, unsolicited “junk” e-mail sent to large numbers of people to promote products or services, has become a big problem for email users. Spam filters automatically identify spam mails and delete them or move them to junk folders. However, those spam filters are not error free. They sometimes eliminate important emails. A proof of authenticity could prevent this.

RNA analysis of thin sections of standard tumor biopsies are used to evaluate panels of genes that may predict breast cancer recurrence and response to chemotherapy. However, RNA data can be abused and must be kept confidential.

Further examples?

1.2 Public-key encryption

Traditionally, confidentiality of data was achieved by means of secret key cryptosystems. In such a cryptosystem, Alice and Bob exchange a secret key before they secretly communicate. For the key exchange, they use, for example, a secure channel or a courier. Alice uses that secret key to encrypt the data to be kept confidential (plaintext). The result of the encryption is the *ciphertext*. Bob uses the same key to decrypt the data. Without knowledge of the secret key, nobody can obtain information concerning the encrypted data. Such a cryptosystem is also called *symmetric*.

Symmetric cryptosystems are used in practice today and the time required to encrypt 1 GByte on an Athlon 1 GHz computer are shown in Table 1.1.

Table 1.1: Symmetric cryptosystems: Encryption of 1 GB on Athlon 1 GHz

DES-ede	AES	RC6	SERPENT	IDEA	MARS	TWOFISH
726 ms	173 ms	138 ms	200 ms	288 ms	394 ms	697 ms

Key exchange in an open computer network for symmetric cryptosystems is a very serious problem. If the network has n users and any two of them exchange a key, then $n(n-1)/2$ secret key exchanges are necessary and all those keys have to be stored securely. According to

[?] there were approximately $9 \cdot 10^8$ Internet users in 2005. If any two Internet users exchanged a secret key then $4 \cdot 10^{17}$ keys would be necessary. This would be impossible to organize.

Another possibility for organizing the key exchange for symmetric cryptosystems is to use a key center. Every user exchanges a secret key with this key center. If Alice wants to send a message to Bob, then she encrypts the message using her secret key and sends it to the key center. The center, knowing all secret keys, decrypts the message using Alice's key, encrypts it with Bob's key, and sends it to Bob. In this way, the number of key exchanges for n users is reduced to n . However, the key center gets to know all secret messages, and it must store all n keys securely.

Key management in public-key systems is much easier. In a *public-key cryptosystem*, the key e used for encryption is different from the key d used for decryption and the computation of d from e is infeasible. In such a system, the encryption key can be made public. If Alice wants to receive encrypted messages, she publishes an encryption key e (see Table 1.2) and keeps the corresponding decryption key d secret. That key is called the *private key*. Anybody can use the *public key* e to encrypt messages for Alice. But only Bob can decrypt the messages.

Public-key cryptosystems are also called *asymmetric cryptosystems*.

Table 1.2: A public key directory

Name	Public Key
Buchmann	13121311235912753192375134123
Maurer	84228349645098236102631135768
Alice	54628291982624638121025032510
Bob	27381253812351972497652990930
⋮	⋮

Unfortunately, the known public-key systems are not as efficient as many symmetric cryptosystems. The efficiency of two public-key cryptosystems is shown in Table 1.3. In practice,

Table 1.3: Public-key cryptosystems: Encryption of 1 GB on Athlon 1 GHZ

RSA	ElGamal
16 s	1826 s

hybrid cryptosystems, that is, combinations of public-key systems and symmetric systems are used. This works as follows. Alice wants to send a message m in encrypted form to Bob. She generates a *session key* for an efficient symmetric cryptosystem. Then she encrypts the message m using that session key and the symmetric system, obtaining the ciphertext c . This encryption is fast because an efficient symmetric cryptosystem has been used. Alice also encrypts the session key with Bob's public key, which she obtains from a public directory (see Table 1.2). Since the session key is small, this encryption is also fast, although the encryption function of the public-key system may not be very efficient. Then Alice sends the ciphertext c and the encrypted session key to Bob. Bob decrypts the session key using his private key. Then he decrypts the ciphertext c with the session key and obtains the original message m . Here, the public-key system is only used for the exchange of the session key. This combines

the elegant key management of the public-key system with the efficiency of the symmetric cryptosystem.

In order for a public-key cryptosystem to be secure it is not sufficient that computing the secret decryption key from the publicly known information such as the public encryption key is infeasible. A secure encryption scheme should provide *indistinguishability against chosen ciphertext attacks*: distinguishing the encryptions of two different plaintexts must be infeasible for anyone not knowing the secret decryption key even if he is able to decrypt other ciphertexts of his choice.

1.3 Digital signatures

While encryption is used to achieve confidentiality, the security services and goals integrity, authentication, non-repudiation and timestamping require digital signatures.

Digital signature schemes use *cryptographic hash functions* (see [2]). Such a hash function maps data (bit strings of arbitrary length) to short bit strings of fixed length. A *collision* of a hash function is a pair of documents with the same hash value. A cryptographic hash function is required to be *collision resistant*, that is, it must be infeasible to find a collision for this hash function. Today, the most popular hash function is SHA-1. Hash values of SHA-1 are 160-bit strings. However, there are doubts about the security of SHA-1. In [5] it is shown how to find collisions faster than with brute force search.

Cryptographic hash functions can be used to prove that data were not changed. The hash value of the data is computed and kept in a secure place. At any point in time, the hash value of the data can be recomputed and compared to the original hash value. Since the hash function is collision resistant, nobody can alter the data without changing the hash value. While hash functions can be used to prove the integrity of data, they cannot be used to identify the originator of the data.

The originator of data can be identified using a *message authentication code (MAC)*. In a MAC Alice and Bob share a secret key. Alice uses that key to compute a MAC of certain data. Like a hash value, a MAC is a short bit-string of fixed length. For example, MACs are 160-bit strings. While hash functions are public and hash values can be computed by anybody, computing a MAC without the knowledge of the key is infeasible. Upon receiving the data and the MAC, Bob uses the key to verify that the MAC is correct. MACs can guarantee integrity and authenticity. However, MACs cannot be used for non-repudiation, since verifying the MAC requires the knowledge of a secret key. That same key can also be used to compute a MAC. This is why the key has to be kept secret since otherwise anybody could produce valid MACs. But this means that verification is not possible for third parties who do not know the secret key. But to enable non-repudiation, verification must be possible for any third party.

Non-repudiation is provided by digital signature schemes. In a digital signature scheme, the signer has a secret *signing key* and publishes the corresponding public *verification key*. Also, the signer uses a publicly known cryptographic hash function. Using his signing key and the hash value of the data to be signed, the signer calculates the digital signature of the data. The verifier obtains the public verification key of the signer. He calculates the hash of the data. Using that hash and the verification key, the verifier verifies the correctness of the digital signature. Computing a valid signature without the knowledge of the secret signing key is infeasible. In fact, the security requirement for a digital signature scheme is even higher. Secure digital signature algorithms must be *existentially unforgeable under chosen message*

attacks: generating a new valid signature must be infeasible for anyone not knowing the secret signing key even if he can produce other valid signatures of his choice.

Digital signatures can be used for authentication and non-repudiation, since anybody can verify the digital signature for some data. The validity of that signature provides proof of the integrity and origin of data.

1.4 Security of public-key cryptography

The most popular public-key cryptosystem and digital signature scheme is RSA (see [2]). Its security relies on the integer factoring problem: given a large positive integer n which is known to be the product of two prime numbers p and q that are roughly of equal size. Find the prime factors p and q . It is believed today that the integer factoring problem is infeasible if n is at least a 1024-bit number. However, there is no proof that the integer factoring problem remains difficult. On the contrary. For many years the RSA Labs have published RSA challenge numbers. Those challenge numbers are the kind the people at RSA Labs believe to be the hardest to factor. These are the kind of numbers used in devising secure RSA cryptosystems. Table 1.4 shows the RSA challenge numbers factored so far. It demonstrates that there has been a lot of progress in factoring algorithms. It has also been shown by Shor [4] that a quantum computer can solve the integer factoring problem in polynomial time.

Table 1.4: RSA challenge numbers

number	dec. digits	factored
RSA-100	100	Apr. 1991
RSA-110	110	Apr. 1992
RSA-120	120	Jun. 1993
RSA-129	129	Apr. 1994
RSA-130	130	Apr. 1996
RSA-140	140	Feb. 1999
RSA-155	155	Aug. 1999
RSA-160	160	Apr. 2003
RSA-576	174	Dec. 2003

There are also Public-key cryptosystems and digital signature schemes whose security is based on the difficulty of computing discrete logarithms. The discrete logarithm problem DLP is the following (see also [2]). Given a group G and elements $g, a \in G$ such that a is a power of g . Find $x \in \mathbb{Z}$ such that $g^x = a$.

The group that is most relevant for practical applications is the group of points of an elliptic curve over a finite field. There are also challenges that demonstrate the difficulty of the elliptic curve DLP (see http://www.certicom.com/index.php?action=res,ecc_challenge). Also, it has been shown by Shor [4] that a quantum computer can solve this and all other cryptographically relevant discrete logarithm problems in polynomial time. Hence, if sufficiently large quantum computers can be built then most public key cryptosystems and digital signature schemes are insecure.

1.5 Key management in public-key infrastructures

In public-key systems, no key exchange between users is necessary. Encryption keys for public-key cryptosystems and verification keys for digital signature schemes are publically known. They can, for example, be listed in public directories. Although everybody may read those directories, they must be protected from unauthorized writing. If the attacker, Oscar, is able to replace Alice's public encryption key with his own, then he can decrypt the messages that are sent to Alice or sign data in the name of Alice.

1.6 Timestamping

Timestamping can be implemented using digital signature schemes. The details are described in [3] and [1]. The hash of the data to be timestamped are sent to a trusted timestamping service. That service digitally signs the hash together with the actual time. Since the timestamping service only sees the hash of the data, it learns nothing about the data. However, anyone who trusts the timestamping service can later verify that the data existed at the time given by the timestamp.

1.7 Further applications

There are further applications of public-key cryptography such as electronic voting schemes or electronic payment systems.

Chapter 2

Network security

In this chapter we show how PKI is used to achieve security in computer networks, in particular, the Internet.

2.1 The OSI model

The OSI, or Open System Interconnection, model defines a networking framework for implementing protocols in seven layers. Control is passed from one layer to the next, starting at the application layer in one station, proceeding to the bottom layer, over the channel to the next station and back up the hierarchy. In http://www.webopedia.com/quick_ref/OSI_Layers.asp the OSI model is explained as follows.

Layer 7: Application This layer supports application and end-user processes. Communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. Everything at this layer is application-specific. This layer provides application services for file transfers, e-mail, and other network software services. For example, Telnet and FTP are applications that exist entirely in the application level.

Layer 6: Presentation This layer provides independence from differences in data representation (e.g., encryption) by translating from application to network format, and vice versa. The presentation layer works to transform data into the form that the application layer can accept. This layer formats and encrypts data to be sent across a network, providing freedom from compatibility problems. It is sometimes called the syntax layer.

Layer 5: Session This layer establishes, manages and terminates connections between applications. The session layer sets up, coordinates, and terminates conversations, exchanges, and dialogues between the applications at each end. It deals with session and connection coordination.

Layer 4: Transport This layer provides transparent transfer of data between end systems, or hosts, and is responsible for end-to-end error recovery and flow control. It ensures complete data transfer.

Layer 3: Network This layer provides switching and routing technologies, creating logical paths, known as virtual circuits, for transmitting data from node to node. Routing and forwarding are functions of this layer, as well as addressing, internetworking, error handling, congestion control and packet sequencing.

Layer 2: Data Link At this layer, data packets are encoded and decoded into bits. It

furnishes transmission protocol knowledge and management and handles errors in the physical layer, flow control and frame synchronization. The data link layer is divided into two sublayers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. The MAC sublayer controls how a computer on the network gains access to the data and permission to transmit it. The LLC layer controls frame synchronization, flow control and error checking.

Layer 1: Physical This layer conveys the bit stream - electrical impulse, light or radio signal – through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier, including defining cables, cards and physical aspects. Fast Ethernet, RS232, and ATM are protocols with physical layer components.

On the Internet the top three layers of the OSI model are combined. So there are only five layers and the top layer is called *application layer*.

On each layer, attacks are possible.

On the physical layer, an attacker can intercept the bit stream. This is called *bit sniffing*. Bit sniffing is particularly easy in wireless networks. In principal, it is possible to prevent bit sniffing by using a stream cipher. However, in open networks, key management for such a solution is much too complex.

On the data link layer, attackers use network sniffing. Network sniffing is listening (with software) to the raw network device for frames or packets. When the sniffer sees a frame or a packet that fits certain criteria, it logs it to a file. Those data provide information about the communication. They may also be used to reconstruct the original communication. A network sniffer can be an invaluable tool for diagnosing network problems but is often employed by hackers to see what is going on behind the scenes, so to speak, during communication between two hosts. For example, the sniffer may collect all packets that contain words like "login" or "password" and may obtain secret login and password information.

2.2 IP

We discuss the network layer of the Internet in more detail.

On the network layer of the Internet, the Internet Protocol (IP) is used. (The following explanation is from http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci214031,00.html)

IP is the method or protocol by which data is sent from one computer to another on the Internet. Each computer (known as a host) on the Internet has at least one IP address that uniquely identifies it from all other computers on the Internet. When you send or receive data (for example, an e-mail note or a Web page), the message gets divided into little chunks called IP packets. Each of these packets contains both the sender's Internet address and the receiver's address in his IP header.

Bild: IP packet

Any packet is sent first to a gateway computer that understands a small part of the Internet. The gateway computer reads the destination address and forwards the packet to an adjacent gateway that in turn reads the destination address and so forth across the Internet until one gateway recognizes the packet as belonging to a computer within its immediate neighborhood or domain. That gateway then forwards the packet directly to the computer whose address is specified.

Because a message is divided into a number of packets, each packet can, if necessary, be sent by a different route across the Internet. Packets can arrive in a different order than the

order they were sent in. The Internet Protocol just delivers them. It's up to another protocol, the Transmission Control Protocol (TCP) to put them back in the right order.

IP is a connectionless protocol, which means that there is no continuing connection between the end points that are communicating. Each packet that travels through the Internet is treated as an independent unit of data without any relation to any other unit of data. (The reason the packets do get put in the right order is because of TCP, the connection-oriented protocol that keeps track of the packet sequence in a message.)

The most widely used version of IP today is Internet Protocol Version 4 (IPv4). However, IP Version 6 (IPv6) is also beginning to be supported. IPv6 provides for much longer addresses and therefore for the possibility of many more Internet users. IPv6 includes the capabilities of IPv4 and any server that can support IPv6 packets can also support IPv4 packets.

IP has no security features. Packets can be intercepted and replaced without the communication partners noticing. Common attacks are IP sniffing and IP spoofing. IP sniffing means listening to IP packets and combining them to obtain as much information as possible. IP spoofing is a technique whereby an intruder alters a packets IP address. For example, this is done to make the packet appear as though it has originated in a part of the network with higher access privileges.

2.3 IPsec

IPsec, a variant of IP, supports security. To prevent IP-packet sniffing, the ESP (Encapsulated Security Payload) modes of IPsec can be used. They are *ESP transport mode* and *ESP tunnel mode*

In the ESP transport mode, symmetric encryption is used to guarantee the confidentiality of the payload. However, the IP-header which contains information about the source and the destination is not kept confidential. This information may still be very interesting for an attacker.

Bild: ESP transport mode packet

In the ESP tunnel mode, symmetric encryption is used to guarantee the confidentiality of the whole IP packet including the IP-header. So the information in that header ist kept confidential.

To ensure authenticity, the AH (authentication header) modes of IPsec are used.

In the AH transport mode, symmetric cryptography is used to guarantee the authenticity of the payload but not of the IP-header.

Bild: AH transport mode packet

In the AH tunnel mode, symmetric cryptography is used to guarantee the authenticity of the whole IP packet including the IP-header.

Bild: AH tunnel mode packet

IPsec security is security of the IP-packets between the originating computer on which IPsec is implemented (this may be a gateway) and the destination computer on which IPsec runs (this may also be a gateway). IPsec security is not end-to-end security between users and IPsec cannot authenticate documents and cannot be used for non-repudiation of documents.

Later, we will describe IPsec in more detail.

In IPSec, public-key cryptography is used for the key agreement.

2.4 TCP

From http://searchnetworking.techtarget.com/sDefinition/0,290660,sid7_gci214172,00.html

TCP (Transmission Control Protocol) is a set of rules (protocol) used along with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. While IP takes care of handling the actual delivery of the data, TCP takes care of keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

For example, when an HTML file is sent to you from a Web server, the Transmission Control Protocol (TCP) program layer in that server divides the file into one or more packets, numbers the packets, and then forwards them individually to the IP program layer. Although each packet has the same destination IP address, it may get routed differently through the network. At the other end (the client program in your computer), TCP reassembles the individual packets and waits until they have arrived to forward them to you as a single file.

TCP is known as a connection-oriented protocol, which means that a connection is established and maintained until such time as the message or messages to be exchanged by the application programs at each end have been exchanged. TCP is responsible for ensuring that a message is divided into the packets that IP manages and for reassembling the packets back into the complete message at the other end. In the Open Systems Interconnection (OSI) communication model, TCP is in layer 4, the Transport Layer.

There is no security in TCP.

2.5 SSL

The Secure Socket Layer protocol (SSL) is commonly used to add security to TCP. SSL has recently been succeeded by Transport Layer Security (TLS), which is based on SSL. SSL uses a program layer located between the Internet's Hypertext Transfer Protocol (HTTP) and Transport Control Protocol (TCP) layers. SSL establishes a secure connection between sockets. SSL has various modes. In its basic mode, SSL ensures confidentiality of the data that are sent between the sockets. Other modes also guarantee the authenticity of the server or the client. Since SSL/TLS security is socket-to-socket connection security, SSL/TLS cannot be used for non-repudiation.

SSL is included as part of both the Microsoft and Netscape browsers and most Web server products. Developed by Netscape, SSL also gained the support of Microsoft and other Internet client/server developers as well and became the de facto standard until evolving into Transport Layer Security. The "sockets" part of the term refers to the sockets method of passing data back and forth between a client and a server program in a network or between program layers in the same computer.

SSL/TLS is composed of two layers: the SSL/TLS Record Protocol and the SSL/TLS Handshake Protocol. The TLS Record Protocol provides connection security (authenticity and/or confidentiality) and uses symmetric cryptography. The SSL/TLS Handshake Protocol allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before data is exchanged. That protocol uses public-key cryptography.

SSL and TLS are discussed in more detail later.

2.6 Application layer

There are many applications on the (combined) application layer of the Internet. We give a few important examples.

The *Hyper Text Transport Protocol (HTTP)* is the communication protocol used to connect to servers on the World Wide Web. The primary function of HTTP is to establish a connection between a web browser (client) and a Web server and to transmit HTML pages to browser.

The *File Transfer Protocol (FTP)* is standard method for sending files from one computer to another on TCP/IP networks such as the Internet.

The *Post Office Protocol (POP)* Short for Post Office Protocol is used to retrieve e-mail from a mail server. Many e-mail applications use POP, although some can use the newer *Internet Message Access Protocol (IMAP)*.

Let us discuss the example of Internet banking to explain possible attacks on the application layer. The user starts Internet banking by entering the web address of his bank in his browser. The browser sends a request to the Domain Name Server (DNS) server. The DNS server is a computer that determines Internet Protocol (IP) numeric addresses from domain names presented in a convenient, readable form. The DNS server returns the IP address of the bank. The browser of the user sends a http request to the IP address of the bank and establishes a http connection. The bank sends its Internet banking page. The user proceeds to do Internet banking using the http connection.

In this scenario, a possible attack is *DNS spoofing*. The DNS server is compromised and returns a wrong IP address.

DNS Server The DNS (Domain Name Server) server is a computer that determines Internet Protocol (IP) numeric addresses from domain names presented in a convenient, readable form.

Frame The basic unit of communication between two Ports. Frames are composed of a starting delimiter (SOF), a header, the payload, the Cyclic Redundancy Check (CRC), and an ending delimiter (EOF). The SOF and EOF contain the Special Character and are used to indicate where the frame begins and ends. The 24-byte header contains information about the frame, including the sender ID, destination ID, routing information, the type of data contained in the payload, and sequence/exchange management information. The payload contains the actual data to be transmitted, and may be 0-2112 bytes in length. The CRC is a 4-byte field used for detecting bit errors in the received frame.

Protocol A formal description of message formats and the rules two computers must follow to exchange those messages.

Bibliography

- [1] BAYER, D., HABER, S., AND STORNETTA, W. Improving the efficiency and reliability of digital timestamping. In *Proceedings Sequences II: Methods in Communication, Security, and Computer Science* (1993), Springer-Verlag, pp. 329–334.
- [2] BUCHMANN, J. *Introduction to Cryptography*, Second Edition ed. Springer-Verlag, New York, 2004.
- [3] HABER, S., AND STORNETTA, W. How to timestamp a digital document. *Journal of Cryptology* 2 (1991), 99–111.
- [4] SHOR, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing* 26 (1997), 1484–1509.
- [5] WANG, X., YIN, Y. L., AND YU, H. Collision search attacks on sha1. <http://theory.csail.mit.edu/~yiqun/shanote.pdf>, February 2005.

Index

asymmetric cryptosystem, 2
authtication, 1

chosen message attack, 4
ciphertext, 1
collision, 3
confidentiality, 1
cryptographic hash function, 3
cryptosystem
 asymmetric, 2
 secret key, 1
 symmetric, 1

hash function, 3
hybrid encryption, 2

identification, 1
integrity, 1

MAC, 3
message authentication code, 3

plaintext, 1
private key, 2
public key, 2

secret key cryptosystem, 1
session key, 2
signing key, 3
symmetric cryptosystem, 1

time stamping, 1

verification key, 3