**3A Identify requirements (e.g. confidentiality, integrity, non-repudiation).**

**Cryptography** is a detective control in the fact that it allows the detection of fraudulent insertion, deletion or modification. It also is a preventive control is the fact that it prevents disclosure, but it usually does not offers any means of detecting disclosure.

The cryptography domain addresses the principles, means, and methods of disguising information to ensure its integrity, confidentiality, and authenticity. Unlike the other domains, cryptography does not completely support the standard of availability.

Availability

Cryptography supports all three of the core principles of information security. Many access control systems use cryptography to limit access to systems through the use of passwords. Many token-based authentication systems use cryptographic-based hash algorithms to compute one-time passwords. Denying unauthorized access prevents an attacker from entering and damaging the system or network, thereby denying access to authorized users if they damage or currupt the data.

Confidentiality

Cryptography provides confidentiality through altering or hiding a message so that ideally it cannot be understood by anyone except the intended recipient.

Integrity

Cryptographic tools provide integrity checks that allow a recipient to verify that a message has not been altered. Cryptographic tools cannot prevent a message from being altered, but they are effective to detect either intentional or accidental modification of the message.

Additional Features of Cryptographic Systems In addition to the three core principles of information security listed above, cryptographic tools provide several more benefits.

Nonrepudiation

In a trusted environment, the authentication of the origin can be provided through the simple control of the keys. The receiver has a level of assurance that the message was encrypted by the sender, and the sender has trust that the message was not altered once it was received. However, in a more stringent, less trustworthy environment, it may be necessary to provide assurance via a third party of who sent a message and that the message was indeed delivered to the right recipient. This is accomplished through the use of digital signatures and public key encryption. The use of these tools provides a level of nonrepudiation of origin that can be verified by a third party.

Once a message has been received, what is to prevent the recipient from changing the message and contesting that the altered message was the one sent by the sender? The nonrepudiation of delivery prevents a recipient from changing the message and falsely claiming that the message is in its original state. This is also accomplished through the use of public key cryptography and digital signatures and is verifiable by a trusted third party.

Authentication

Authentication is the ability to determine if someone or something is what it declares to be. This is primarily done through the control of the keys, because only those with access to the key are able to encrypt a message. This is not as strong as the nonrepudiation of origin, which will be reviewed shortly Cryptographic functions use several methods to ensure that a message has not been changed or altered. These include hash functions, digital signatures, and message authentication codes (MACs). The main concept is that the recipient is able to detect any change that has been made to a message, whether accidentally or intentionally.

Access Control

Through the use of cryptographic tools, many forms of access control are supported—from log-ins via passwords and passphrases to the prevention of access to confidential files or messages. In all cases, access would only be possible for those individuals that had access to the correct cryptographic keys.

**NOTE FROM CLEMENT:**

As you have seen this question was very recently updated with the latest content of the Official ISC2 Guide (OIG) to the CISSP CBK, Version 3.

Myself, I agree with most of you that cryptography does not help on the availability side and it is even the contrary sometimes if you loose the key for example. In such case you would loose access to the data and negatively impact availability. But the ISC2 is not about what I think or what you think, they have their own view of the world where they claim and state clearly that cryptography does address availability even thou it does not fully address it.

They look at crypto as the ever emcompassing tool it has become today. Where it can be use for authentication purpose for example where it would help to avoid corruption of the data through illegal access by an unauthorized user.

The question is worded this way in purpose, it is VERY specific to the CISSP exam context where ISC2 preaches that cryptography address availability even thou they state it does not fully address it. This is something new in the last edition of their book and something you must be aware of.

**Strong encryption refers to an encryption process that uses at least a 128-bit key.**

Historically**, a code** refers to a cryptosystem that deals with linguistic units: words, phrases, sentences, and so forth. Codes are only useful for specialized circumstances where the message to transmit has an already defined equivalent ciphertext word.

The **European Union on Electronic Signatures** deal with **Non repudiation.**

**KEY LENGTH -** since a large keyspace allows for more possible keys it is the BEST option here.

The objective is to perform the maximum number of possible different keys generated, thus providing more security to the ciphering making harder for intruders to figure them out.

The One Time Pad, the Gilbert Cipher, The Vernam Cipher

One time pad is the best scenario in cryptography, it is known as the unbreakable cipher. Since a key of the same size as the message size will be used only once and then it is destroyed and never reused.

This is most likely a theorical scenario only, we should use the highiest amount of bits and larger keysize so the combination resulted from the algorith used will be higher. This would increase an attacker’s chances of figuring out the key value and deciphering the protected information.

**Hybrid Encryption Methods** are when Asymmetric and Symmetric Algorithms used together. In the hybrid approach, the two technologies are used in a complementary manner, with each performing a different function. A symmetric algorithm creates keys that are used for encrypting bulk data, and an asymmetric algorithm creates keys that are used for automated key distribution.

**SYMMETRIC VS ASYMMETRIC Performance**

Because of the amount of computation involved in public key cryptography, a DES hardware implementation of secret key cryptography is on the order of 1000 to 10000 times faster than RSA public key cryptography.

It is important to understand WHY it is faster and NOT only that it is faster. Symmetric uses binary addition of bits, substitution, permutation, shifing columns, shifing rows, which requires very little processing power. Asymmetric uses very complex mathematical problems such as the Discrete Logarithm problem in a finite field, factoring large numbers into the two prime numbers used to create the large numbers, which all requires a lot of processsing power.

So SPEED is definitively an advantage of Symmetric ciphers and this is WHY the bulk of the data is always encrypted using Symmetric ciphers versus Asymmetric ciphers.

Sometimes even within the same book there are contradictions between authors as far as the exact number of times it would be faster. Know WHY it is faster and where they would be used as this is the important thing to know for the purpose of the exam.

The confusion often times comes from the fact that books do not specify if it is hardware or software implementation. The RSA website does specify the following:

By comparison, DES is much faster than RSA. In software, DES is generally at least 100 times as fast as RSA. In hardware, DES is between 1,000 and 10,000 times as fast, depending on the implementation.

So if we do not know if software or hardware is being use, it is hard to make sense of the question. The answer really depend on the implementation and whether or not it is software or hardware.

**The keyspace size** uses a simple formula which is 2 the power of of the key size or in this case 2 to the power of 8.

**3B Determine usage (e.g. intransit, at rest)**

**3C Identify Cryptographic design considerations and constraints**

**SYMMETRIC**

When using symmetric cryptography, both parties will be using the same key for encryption and decryption. Symmetric cryptography is generally fast and can be hard to break, but it offers limited overall security in the fact that it can **only provide confidentiality.**

**DES - block**

**NSA** took the 128-bit algorithm **Lucifer that IBM developed**, reduced the key size to 64 bits and with that developed **DES.**

Data Encryption Standard (DES) is a symmetric key algorithm. Originally developed by IBM, under project name Lucifer, this 128-bit algorithm was accepted by the NIST in 1974, but the total key size was reduced to 64 bits, 56 of which make up the *effective* key, plus and extra 8 bits for parity. It somehow became a national cryptographic standard in 1977, and an American National Standard Institute (ANSI) standard in 1978.

The characters are put through **16 rounds of transposition** and substitution functions. **Triple DES uses 48 rounds.** Triple DES **encrypts a message three times.** This encryption can be accomplished in several ways. The most secure form of triple DES is when the three encryptions are performed with three different keys.

**Substitution** is not a mode of DES.

**There is no DES mode called DES-EEE1.** It does not exist.

The following are the correct modes for **triple-DES (3DES):**

**DES-EEE3** uses three keys for encryption and the data is encrypted, encrypted, encrypted;

**DES-EDE3** uses three keys and encrypts, decrypts and encrypts data. **THE MOST SECURE!!!** Triple DES with three distinct keys is the most secure form of triple-DES encryption. It can either be DES-EEE3 (encrypt-encrypt-encrypt) or DES-EDE3 (encrypt-decrypt-encrypt). DES-EDE1 is not defined and would mean using a single key to encrypt, decrypt and encrypt again, equivalent to single DES. DES-EEE4 is not defined and DES-EDE2 uses only 2 keys (encrypt with first key, decrypt with second key, encrypt with first key again).

**DES-EEE2** and DES-EDE2 are the same as the previous modes, but the first and third operations use the same key.

DES **(BLOCK CIPHER)** operates in four mode:

**a) Cipher Block Chaining (CBC)**

The previous DES output is used as input. This is a characteristic of Cipher Block Chaining. Cipher Block Chaining uses the output from the previous block to encrypt the next block.

CBC mode of operation was invented by IBM in 1976. In the cipher-block chaining (CBC) mode, each block of plaintext is XORed with the previous ciphertext block before being encrypted. This way, each ciphertext block is dependent on all plaintext blocks processed up to that point. Also, to make each message unique, an initialization vector must be used in the first block.

CBC is a block cipher system in which the first plain text data block is exclusive-ORed with a block of pseudo-random data prior to being processed through the DES. The resulting cipher text block is then exclusive-ORed with the next plain text data block to form the next input block to the DES, thus chaining together blocks of cipher text. The chaining of cipher text blocks provides an error extension characteristic which is valuable in protecting against fraudulent data alteration. A CBC authentication technique is described in Appendix F.

The CBC mode produces the same cipher text whenever the same plain text is encrypted using the same key and IV. Users who are concerned about this characteristic should incorporate a unique identifier (e.g., a one-up counter) at the beginning of each CBC message within a cryptographic period in order to insure unique cipher text. If the key and the IV are the same and no identifier precedes each message, messages that have the same beginning will have the same cipher text when encrypted in the CBC mode until the blocks that differ in the two messages are encrypted.

Since the CBC mode is a block method of encryption, it must operate on 64-bit data blocks. Partial data blocks (blocks of less than 64 bits) require special handling. One method of encrypting a final partial data block of a message is described below. Others may be defined for special applications.

**b) Electronic Code Book (ECB)**

A given block of plaintext and a given key will always produce the same ciphertext. BEST FOR DATABASES. Because ECB works with blocks of data independently, data within files does not have to be encrypted in a certain order. This is very helpful when using encryption in databases. A database has different pieces of data accessed in a random fashion. If it is encrypted with ECB mode, then any record or table can be added, encrypted, deleted, or decrypted independently of any other table or record.

It is important to note that ECB does not offer randomness and should NOT be used to encrypt large quantity of data.

The Electronic Codebook (ECB) mode is a basic, block, cryptographic method which transforms 64 bits of input to 64 bits of output u specified in FIPS PUB 46.

The analogy to a codebook arises because the same plain text block always produces the same cipher text block for a given cryptographic key. Thus a list (or codebook) of plain text blocks and corresponding cipher text blocks theoretically could be constructed for any given key. In electronic implementation the codebook entries are calculated each time for the plain text to be encrypted and, inversely, for the cipher text to be decrypted.

**Electronic Code Book** A given block of plaintext and a given key **will always produce the same ciphertext.**

**c) Cipher Feedback (CFB)**

Individual characters are encoded by combining output from earlier encryption routines with plaintext. This is a characteristic of Cipher Feedback. Cipher Feedback the ciphertext is run through a key-generating device to create the key for the next block of plaintext. The cipher feedback (CFB) mode, a close relative of CBC, makes a block cipher into a self-synchronizing stream cipher. Operation is very similar; in particular, CFB decryption is almost identical to CBC encryption performed in reverse.

The CFB mode is a stream method of encryption in which the DES is used to generate pseudorandom bits which are exclusive-ORed with binary plain text to form cipher text. The cipher text is fed back to form the next DES input block. Identical messages that are encrypted using the CFB mode and different IVs will have different cipher texts. IVs that are shorter than 64 bits should be put in the least significant bits of the first DES input block and the unused, most significant, bits initialized to "0's."

In the CFB mode, errors in any K-bit unit of cipher text will affect the decryption of the garbled cipher text and also the decryption of succeeding cipher text until the bits in error have been shifted out of the CFB input block. The first af fected K-bit unit of plain text will be garbled in exactly those places where the cipher text is in error. Succeeding decrypted plain text will have an average error rate of fifty percent until all errors have been shifted out of the DES input block. Assuming no additional errors are encountered during this time, the correct plain text will then be obtained.

**d) Output Feedback (OFB)**

The output feedback (OFB) mode makes a block cipher into a synchronous stream cipher. It generates keystream blocks, which are then XORed with the plaintext blocks to get the ciphertext. Just as with other stream ciphers, flipping a bit in the ciphertext produces a flipped bit in the plaintext at the same location. This property allows many error correcting codes to function normally even when applied before encryption.

The Output Feedback (OFB) mode is an additive stream cipher in which errors in the cipher text are not extended to cause additional errors in the decrypted plain text. One bit in error in the cipher text causes only one bit to be in error in the decrypted plain text. Therefore, this mode cannot be used for data authentication but is useful in applications where a few errors in the decrypted plain text are acceptable.

In the OFB mode, the same K bits of the DES output block that are used to encrypt a K-bit unit of plain text are fed back for the next input block. This feedback is completely independent of all plain text and all cipher text. As a result, there is no error extension in OFB mode.

If cryptographic synchronization is lost in the OFB mode, then cryptographic initialization must be performed. The OFB mode is not a self-synchronizing cryptographic mode.

**AES – block**

The Rijndael algorithm, chosen as the Advanced Encryption Standard (AES) to replace DES, can be categorized as an iterated block cipher with a variable block length and key length that can be independently chosen as **128, 160, 192, 224, and 256.**

Characteristic:

* **It employs a round transformation that is comprised of three layers of distinct and invertible transformations.**
* **It is suited for high speed chips with no area restrictions.**
* **It could be used on a smart card.**
* **Rijndael does not support multiples of 64 bits but multiples of 32 bits in the range of 128 bits to 256 bits. Key length could be 128, 160, 192, 224, and 256.**
* **The key sizes may be any multiple of 32 bits**
* **Maximum block size is 256 bits**
* **The key size does not have to match the block size**

The Rijndael algorithm was chosen by NIST as a replacement standard for DES.

It is a block cipher with a variable block length and key length.

It employs a round transformation that is comprised of three layers of distinct and invertible transformations:

**The non-linear layer,**

**the linear mixing layer, and**

**the key addition layer.**

The Rijndael algorithm is a new generation symmetric block cipher that supports key sizes of 128, 192 and 256 bits, with data handled in 128-bit blocks - however, in excess of AES design criteria, the block sizes can mirror those of the keys. Rijndael uses a variable number of rounds, depending on key/block sizes, as follows:

**10 rounds if the key/block size is 128 bits**

**12 rounds if the key/block size is 192 bits**

**14 rounds if the key/block size is 256 bits**

**The Rijndael Cipher**

Rijndael is a block cipher, designed by Joan Daemen and Vincent Rijmen as a candidate algorithm for the Advanced Encryption Standard (AES) in the United States of America. The cipher has a variable block length and key length.

Rijndael can be implemented very efficiently on a wide range of processors and in hardware.

The design of Rijndael was strongly influenced by the design of the block cipher Square.

**The Advanced Encryption Standard (AES)**

The Advanced Encryption Standard (AES) keys are defined to be either 128, 192, or 256 bits in accordance with the requirements of the AES.

The number of rounds, or iterations of the main algorithm, can vary from 10 to 14 within the Advanced Encryption Standard (AES) and is dependent on the block size and key length. 128 bits keys uses 10 rounds or encryptions, 192 bits keys uses 12 rounds of encryption, and 256 bits keys uses 14 rounds of encryption.

The low number of rounds has been one of the main criticisms of Rijndael, but if this ever becomes a problem the number of rounds can easily be increased at little extra cost performance wise by increasing the block size and key length.

Range of key and block lengths in Rijndael and AES

Rijndael and AES differ only in the range of supported values for the block length and cipher key length.

For Rijndael, the block length and the key length can be independently specified to any multiple of 32 bits, with a minimum of 128 bits, and a maximum of 256 bits. The support for block and key lengths 160 and 224 bits was introduced in Joan Daemen and Vincent Rijmen, AES submission document on Rijndael, Version 2, September 1999 available at http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf

AES fixes the block length to 128 bits, and supports key lengths of 128, 192 or 256 bits only.

**IDEA - block**

is a block cipher and operates on **64 bit blocks** of data with a **128-bit key.** The data blocks are divided into 16 smaller blocks and each has **8 rounds of mathematical functions performed on it**. It is used in the PGP encryption software. **It is a SYMMETRIC algorithm.**

**RC2 - block -** works with **64-bit blocks** and variable key lengths. RC2 with a **40-bit key size** was treated favourably under US export regulations for cryptography.

**RC5 - block** is a symmetric encryption algorithm. It is a block cipher of variable block length, encrypts through integer addition, the application of a bitwise Exclusive OR (XOR), and variable rotations.

RC5 is a fast block cipher created by Ron Rivest and analyzed by RSA Data Security, Inc.

It is a parameterized algorithm with a variable block size, a variable key size, and a variable number of rounds.

Allowable choices for the block size are 32 bits (for experimentation and evaluation purposes only), 64 bits (for use a drop-in replacement for DES), and 128 bits.

The number of rounds can range from **0 to 255,** while the key can range from 0 bits to **2040 bits in size.**

Please note that some sources such as the latest Shon Harris book mentions that RC5 maximum key size is of 2048, not 2040 bits. I would definitively use RSA as the authoritative source which specifies a key of 2040 bits. It is an error in Shon's book.

**RC4 - stream**

RC4 ' as it is an algorithm used for encryption and does not provide hashing functions , it is also commonly implemented ' Stream Ciphers '. RC4 was initially a trade secret, but in September 1994 a description of it was anonymously posted to the Cypherpunks mailing list. It was soon posted on the sci.crypt newsgroup, and from there to many sites on the Internet. The leaked code was confirmed to be genuine as its output was found to match that of proprietary software using licensed RC4. Because the algorithm is known, it is no longer a trade secret. The name RC4 is trademarked, so RC4 is often referred to as ARCFOUR or ARC4 (meaning alleged RC4) to avoid trademark problems. RSA Security has never officially released the algorithm; Rivest has, however, linked to the English Wikipedia article on RC4 in his own course notes. RC4 has become part of some commonly used encryption protocols and standards, including WEP and WPA for wireless cards and TLS.

**RC6 – block -** was designed to fix a flaw in RC5. RC6 proper has a block size of **128 bits and supports key sizes of 128, 192 and 256 bits,** but, like RC5, it can be parameterised to support a wide variety of word-lengths, key sizes and number of rounds.

**Skipjack – block -** Skipjack uses an 80-bit key to encrypt or decrypt 64-bit data blocks. It is an unbalanced Feistel network with 32 rounds.[3] It was designed to be used in secured phones.

**Blowfish - BLOCK -** is a keyed, symmetric block cipher, designed in 1993 by Bruce Schneier and included in a large number of cipher suites and encryption products. Blowfish provides a good encryption rate in software and no effective cryptanalysis of it has been found to date. However, the Advanced Encryption Standard now receives more attention.

Schneier designed Blowfish as a general-purpose algorithm, intended as an alternative to the aging DES and free of the problems and constraints associated with other algorithms. At the time Blowfish was released, many other designs were proprietary, encumbered by patents or were commercial/government secrets. Schneier has stated that, "Blowfish is unpatented, and will remain so in all countries. The algorithm is hereby placed in the public domain, and can be freely used by anyone."

Blowfish has a **64-bit block size** and a variable key length from **32 bits up to 448 bits.**

**STREAM CIPHER**

A stream cipher is a type of symmetric encryption algorithm that operates on continuous streams of plain text and is appropriate for hardware-based encryption due to their higher processing power requirement. Stream ciphers can be designed to be exceptionally fast.

A stream cipher encrypts individual bits, whereas a block cipher encrypts blocks of bits. Block ciphers are commonly implemented at the software level because they require less processing power. Stream ciphers, on the other hand,

require more randomness and processing power, making them more suitable for hardware-level encryption.

A strong stream cipher is characterized by the following:

■ Long portions of bit patterns without repeating patterns within keystream values. The keystream must generate random bits.

■ A keystream independent of the key. An attacker should not be able to determine the key value based on the keystream.

■ An unpredictable keystream. The keystream must generate statistically unpredictable bits.

■ An unbiased keystream. There should be as many 0s as there are 1s in the keystream. Neither should dominate.

A stream cipher is a type of symmetric encryption algorithm that operates on continuous streams of plain text and is appropriate for hardware-based encryption.

Stream ciphers can be designed to be exceptionally fast, much faster than any block cipher. A stream cipher generates what is called a keystream (a sequence of bits used as a key).

Stream ciphers can be viewed as approximating the action of a proven unbreakable cipher, the one-time pad (OTP), sometimes known as the Vernam cipher. A one-time pad uses a keystream of completely random digits. The keystream is combined with the plaintext digits one at a time to form the ciphertext. This system was proved to be secure by Claude Shannon in 1949. However, the keystream must be (at least) the same length as the plaintext, and generated completely at random. This makes the system very cumbersome to implement in practice, and as a result the one-time pad has not been widely used, except for the most critical applications.

A stream cipher makes use of a much smaller and more convenient key — 128 bits, for example. Based on this key, it generates a pseudorandom keystream which can be combined with the plaintext digits in a similar fashion to the one-time pad. However, this comes at a cost: because the keystream is now pseudorandom, and not truly random, the proof of security associated with the one-time pad no longer holds: it is quite possible for a stream cipher to be completely insecure if it is not implemented properly as we have seen with the Wired Equivalent Privacy (WEP) protocol.

Encryption is accomplished by combining the keystream with the plaintext, usually with the bitwise XOR operation.

**Synchronous stream ciphers**

In a synchronous stream cipher a stream of pseudo-random digits is generated independently of the plaintext and ciphertext messages, and then combined with the plaintext (to encrypt) or the ciphertext (to decrypt). In the most common form, binary digits are used (bits), and the keystream is combined with the plaintext using the exclusive or operation (XOR). This is termed a binary additive stream cipher.

In a synchronous stream cipher, the sender and receiver must be exactly in step for decryption to be successful. If digits are added or removed from the message during transmission, synchronisation is lost. To restore synchronisation, various offsets can be tried systematically to obtain the correct decryption. Another approach is to tag the ciphertext with markers at regular points in the output.

If, however, a digit is corrupted in transmission, rather than added or lost, only a single digit in the plaintext is affected and the error does not propagate to other parts of the message. This property is useful when the transmission error rate is high; however, it makes it less likely the error would be detected without further mechanisms. Moreover, because of this property, synchronous stream ciphers are very susceptible to active attacks: if an attacker can change a digit in the ciphertext, he might be able to make predictable changes to the corresponding plaintext bit; for example, flipping a bit in the ciphertext causes the same bit to be flipped in the plaintext.

**Self-synchronizing stream ciphers**

Another approach uses several of the previous N ciphertext digits to compute the keystream. Such schemes are known as self-synchronizing stream ciphers, asynchronous stream ciphers or ciphertext autokey (CTAK). The idea of self-synchronization was patented in 1946, and has the advantage that the receiver will automatically synchronise with the keystream generator after receiving N ciphertext digits, making it easier to recover if digits are dropped or added to the message stream. Single-digit errors are limited in their effect, affecting only up to N plaintext digits.

An example of a self-synchronising stream cipher is a block cipher in cipher feedback (CFB) mode.

**BLOCK CIPHER**

In cryptography, a block cipher is a deterministic algorithm operating on fixed-length groups of bits, called blocks, with an unvarying transformation that is specified by a symmetric key. Block ciphers are important elementary components in the design of many cryptographic protocols, and are widely used to implement encryption of bulk data.

Even a secure block cipher is suitable only for the encryption of a single block under a fixed key. A multitude of modes of operation have been designed to allow their repeated use in a secure way, commonly to achieve the security goals of confidentiality and authenticity. However, block ciphers may also be used as building blocks in other cryptographic protocols, such as universal hash functions and pseudo-random number generators.

**The Cesar cipher** is a simple substitution cipher that involves shifting the alphabet three positions to the right.

**ROT13** is a substitution cipher that shifts the alphabet by 13 places.

**Polyalphabetic cipher** refers to using multiple alphabets at a time.

**Transposition cipher** - In cryptography, a transposition cipher is a method of encryption by which the positions held by units of plaintext (which are commonly characters or groups of characters) are shifted according to a regular system, so that the ciphertext constitutes a permutation of the plaintext. That is, the order of the units is changed. Mathematically a bijective function is used on the characters' positions to encrypt and an inverse function to decrypt:

* **Rail Fence cipher**
* **Route cipher**
* **Columnar transposition**
* **Double transposition**
* **Myszkowski transposition**
* **Disrupted transposition**

**ASYMMETRIC**

**RSA**

RSA can be used for encryption, key exchange, and digital signatures.

The correct answer is ' RSA ' named after its inventors Ron Rivest , Adi Shamir and Leonard Adleman is based on the difficulty of factoring large prime numbers.

Factoring a number means representing it as the product of prime numbers. Prime numbers, such as 2, 3, 5, 7, 11, and 13, are those numbers that are not evenly divisible by any smaller number, except 1. A non-prime, or composite number, can be written as the product of smaller primes, known as its prime factors. 665, for example is the product of the primes 5, 7, and 19. A number is said to be factored when all of its prime factors are identified. As the size of the number increases, the difficulty of factoring increases rapidly.

**PKCS #1: RSA Cryptography Standard**

This document provides recommendations for the implementation of public-key cryptography based on the RSA algorithm, covering the following aspects: cryptographic primitives; encryption schemes; signature schemes with appendix; ASN.1 syntax for representing keys and for identifying the schemes.

The computations involved in selecting keys and in enciphering data are complex, and are not practical for manual use. However, using mathematical properties of modular arithmetic and a method known as computing in **Galois fields**, RSA is quite feasible for computer use.

**ECC – Elliptic curve cryptosystem**

It is believed to require shorter keys for equivalent security. Some experts believe that ECC with key length 160 bits is equivalent to RSA with key length 1024 bits. It computes discrete logarithms of elliptic curves. Elliptic curves are rich mathematical structures that have shown usefulness in many different types of applications. An elliptic curve cryptosystem (ECC) provides much of the same functionality that RSA provides: digital signatures, secure key distribution,and encryption. **One differing factor is ECC’s efficiency. ECC is more efficient that RSA and any other asymmetric algorithm.**

**Elliptic Curve Cryptosystems (ECCs) computes discrete logarithms of elliptic curves.**

**El Gamal**

is based on the discrete logarithms in a finite field. El Gamal is a public key algorithm that can be used for digital signatures, encryption,and key exchange. It is based not on the difficulty of factoring large numbers but on calculating discrete logarithms in a finite field. El Gamal is actually an extension of the Diffie-Hellman algorithm. Although El Gamal provides the same type of functionality as some of the other asymmetric algorithms, its main drawback is **performance.** When compared to other algorithms, this algorithm is usually the slowest.

**El Gamal is based on the discrete logarithms in a finite field.**

**HASHING**

**'A Message Digest '** as when a hash algorithm is applied on a message , it produces a message digest.

The other answers are incorrect because :

**A digital signature is a hash value** that has been encrypted with a sender's private key.

A ciphertext is a message that appears to be unreadable.

A plaintext is a readable data.

**SHA-1**

The Secure Hash Algorithm (SHA-1) computes a fixed length message digest from a variable length input message.

SHA-1 produces a 160 bit message digest or hash value. From the nist.gov document referenced above:

This standard specifies four secure hash algorithms, SHA-1, SHA-256, SHA-384, and SHA- 512. All four of the algorithms are iterative, one-way hash functions that can process a messageto produce a condensed representation called a message digest. These algorithms enable the determination of a message’s integrity: any change to the message will, with a very high probability, result in a different message digest. This property is useful in the generation and verification of digital signatures and message authentication codes, and in the generation of random numbers (bits).

Each algorithm can be described in two stages: preprocessing and hash computation. Preprocessing involves padding a message, parsing the padded message into m-bit blocks, and setting initialization values to be used in the hash computation. The hash computation generates a message schedule from the padded message and uses that schedule, along with functions, constants, and word operations to iteratively generate a series of hash values. The final hash vlue generated by the hash computation is used to determine the message digest.

The four algorithms differ most significantly in the number of bits of security that are provided or the data being hashed – this is directly related to the message digest length. When a secure hash algorithm is used in conjunction with another algorithm, there may be requirements specified elsewhere that require the use of a secure hash algorithm with a certain number of bits of security. For example, if a message is being signed with a digital signature algorithm that provides 128 bits of security, then that signature algorithm may require the use of a secure hash algorithm that also provides 128 bits of security (e.g., SHA-256).

Additionally, the four algorithms differ in terms of the size of the blocks and words of data that are used during hashing.

SHA-1 is a one-way hashing algorithms. SHA-1 is a cryptographic hash function designed by the United States National Security Agency and published by the United States NIST as a U.S. Federal Information Processing Standard. SHA stands for "secure hash algorithm".

The three SHA algorithms are structured differently and are distinguished as SHA-0, SHA-1, and SHA-2. SHA-1 is very similar to SHA-0, but corrects an error in the original SHA hash specification that led to significant weaknesses. The SHA-0 algorithm was not adopted by many applications. SHA-2 on the other hand significantly differs from the SHA-1 hash function.

SHA-1 is the most widely used of the existing SHA hash functions, and is employed in several widely used security applications and protocols. In 2005, security flaws were identified in SHA-1, namely that a mathematical weakness might exist, indicating that a stronger hash function would be desirable. Although no successful attacks have yet been reported on the SHA-2 variants, they are algorithmically similar to SHA-1 and so efforts are underway to develop improved alternatives. A new hash standard, SHA-3, is currently under development — an ongoing NIST hash function competition is scheduled to end with the selection of a winning function in 2012.

SHA-1 produces a 160-bit message digest based on principles similar to those used by Ronald L. Rivest of MIT in the design of the MD4 and MD5 message digest algorithms, but has a more conservative design.

**MD5 -** was also created by Ron Rivest and is the newer version of MD4. It still produces a 128-bit hash, but the algorithm is more complex, which makes it harder to break.MD5 was also created by Ron Rivest and is the newer version of MD4. It still produces a 128-bit hash, but the algorithm is more complex, which makes it harder to break.MD5 added a fourth round of operations to be performed during the hashing functions and makes several of its mathematical operations carry out more steps or more complexity to provide a higher level of security.

A hash algorithm (alternatively, hash "function") takes binary data, called the message, and produces a condensed representation, called the message digest. A cryptographic hash algorithm is a hash algorithm that is designed to achieve certain security properties. The Federal Information Processing Standard 180-3, Secure Hash Standard, specifies five cryptographic hash algorithms - SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512 for federal use in the US; the standard was also widely adopted by the information technology industry and commercial companies.

The MD5 Message-Digest Algorithm is a widely used cryptographic hash function that produces a **128-bit (16-byte) hash value.** Specified in RFC 1321, MD5 has been employed in a wide variety of security applications, and is also commonly used to check data integrity. MD5 was designed by Ron Rivest in 1991 to replace an earlier hash function, MD4. An MD5 hash is typically expressed as a 32-digit hexadecimal number.

However, it has since been shown that MD5 is not collision resistant; as such, MD5 is not suitable for applications like SSL certificates or digital signatures that rely on this property. In 1996, a flaw was found with the design of MD5, and while it was not a clearly fatal weakness, cryptographers began recommending the use of other algorithms, such as SHA-1 - which has since been found also to be vulnerable. In 2004, more serious flaws were discovered in MD5, making further use of the algorithm for security purposes questionable - specifically, a group of researchers described how to create a pair of files that share the same MD5 checksum. Further advances were made in breaking MD5 in 2005, 2006, and 2007. In December 2008, a group of researchers used this technique to fake SSL certificate validity, and US-CERT now says that MD5 "should be considered cryptographically broken and unsuitable for further use." and most U.S. government applications now require the SHA-2 family of hash functions.

**MD2 -** is a one-way hash function designed by Ron Rivest that creates a 128-bit message digest value. It is not necessarily any weaker than the other algorithms in the "MD" family, but it is much slower.

**HAVAL -** is a one-way hashing algorithms. HAVAL is a cryptographic hash function. Unlike MD5, but like most modern cryptographic hash functions, HAVAL can produce hashes of different lengths. HAVAL can produce hashes in lengths of 128 bits, 160 bits, 192 bits, 224 bits, and 256 bits. HAVAL also allows users to specify the number of rounds (3, 4, or 5) to be used to generate the hash.

**One-Time-PAD**

A one-time pad is an encryption scheme using a random key of **the same size as the message** and is **used only once**. It is said to be unbreakable, even with infinite resources. A running key cipher uses articles in the physical world rather than an electronic algorithm. Steganography is a method where the very existence of the message is concealed. Cipher block chaining is a DES operating mode.

**it is easy (but not necessarily quick) to compute the hash value for any given message**

**it is infeasible to generate a message that has a given hash**

**it is infeasible to modify a message without changing the hash**

**it is infeasible to find two different messages with the same hash**

**DoD Model layers:**

The Application Layer determines the identity of the communication partners and this is where Non-Repudiation service would be provided as well.

network layer. Is incorrect because the Network Layer mostly has routing protocols, ICMP, IP, and IPSEC. It it not a layer in the DoD Model. It is called the Internet Layer within the DoD model.

transport layer. Is incorrect because the Transport layer provides transparent transfer of data between end users. This is called Host-to-Host on the DoD model but sometimes some books will call it Transport as well on the DoD model.

data link layer. Is incorrect because the Data Link Layer defines the protocols that computers must follow to access the network for transmitting and receiving messages. It is part of the OSI Model. This does not exist on the DoD model, it is called the Link Layer on the DoD model.

**Digital Signature**

A digital signature directly addresses both confidentiality and integrity of the CIA triad. It does not directly address availability, which is what denial-of-service attacks.

Digital Signature Standard (DSS) specifies a Digital Signature Algorithm (DSA) appropriate for applications requiring a digital signature, providing the capability to generate signatures (with the use of a private key) and verify them (with the use of the corresponding public key).

DSS provides Integrity, digital signature and **Authentication, but does not provide Encryption.**

**The steps to create a Digital Signature are very simple:**

1. You create a Message Digest of the message you wish to send

2. You encrypt the message digest using your Private Key which is the action of Signing

3. You send the Message along with the Digital Signature to the recipient

**To validate the Digital Signature the recipient will make use of the sender Public Key. Here are the steps:**

1. The receiver will decrypt the Digital Signature using the sender Publick Key producing a clear text message digest.

2. The receiver will produce his own message digest of the message received.

3. At this point the receiver will compare the two message digest (the one sent and the one produce by the receiver), if the two matches, it proves the authenticity of the message and it confirms that the message was not modified in transit validating the integrity as well. Digital Signatures provides for Authenticity and Integrity only. There is no confidentiality in place, if you wish to get confidentiality it would be needed for the sender to encrypt everything with the receiver public key as a last step before sending the message.

**A digital envelope** for a recipient is a combination of encrypted data and its encryption key in an encrypted form that has been prepared for use of the recipient.

It consists of a hybrid encryption scheme in sealing a message, by encrypting the data and sending both it and a protected form of the key to the intended recipient, so that one else can open the message.

In PKCS #7, it means first encrypting the data using a symmetric encryption algorithm and a secret key, and then encrypting the secret key using an asymmetric encryption algorithm and the public key of the intended recipient.

RFC 2828 (Internet Security Glossary) defines **digital watermarking** as computing techniques for inseparably embedding unobtrusive marks or labels as bits in digital data-text, graphics, images, video, or audio#and for detecting or extracting the marks later. The set of embedded bits (the digital watermark) is sometimes hidden, usually imperceptible, and always intended to be unobtrusive. It is used as a measure to protect intellectual property rights. Steganography involves hiding the very existence of a message. A digital signature is a value computed with a cryptographic algorithm and appended to a data object in such a way that any recipient of the data can use the signature to verify the data's origin and integrity. A digital envelope is a combination of encrypted data and its encryption key in an encrypted form that has been prepared for use of the recipient.

**Steganography** is a secret communication where the very existence of the message is hidden. For example, in a digital image, the least significant bit of each word can be used to comprise a message without causing any significant change in the image. Key clustering is a situation in which a plaintext message generates identical ciphertext messages using the same transformation algorithm but with different keys. Cryptology encompasses cryptography and cryptanalysis. The Vernam Cipher, also called a one-time pad, is an encryption scheme using a random key of the same size as the message and is used only once. It is said to be unbreakable, even with infinite resources.

**Known plaintext attack,** the attacker has the plaintext and ciphertext of one or more messages. The goal is to discover the key used to encrypt the messages so that other messages can be deciphered and read. The known-plaintext attack (KPA) or crib is an attack model for cryptanalysis where the attacker has samples of both the plaintext and its encrypted version (ciphertext), and is at liberty to make use of them to reveal further secret information such as secret keys and code books. The term "crib" originated at Bletchley Park, the British World War II decryption operation.

**An analytic attack** refers to using algorithm and algebraic manipulation weakness to reduce complexity. A statistical attack uses a statistical weakness in the design. A brute-force attack is a type of attack under which every possible combination of keys and passwords is tried. In a codebook attack, an attacker attempts to create a codebook of all possible transformations between plaintext and ciphertext under a single key.

**A Birthday attack** is usually applied to the probability of two different messages using the same hash function producing a common message digest.

The term "birthday" comes from the fact that in a room with 23 people, the probability of two of more people having the same birthday is greater than 50%.

Linear cryptanalysis is a general form of cryptanalysis based on finding affine approximations to the action of a cipher. Attacks have been developed for block ciphers and stream ciphers. Linear cryptanalysis is one of the two most widely used attacks on block ciphers; the other being differential cryptanalysis.

**A Brute force attack** or exhaustive key search is a strategy that can in theory be used against any encrypted data by an attacker who is unable to take advantage of any weakness in an encryption system that would otherwise make his task easier. It involves systematically checking all possible keys until the correct key is found. In the worst case, this would involve traversing the entire key space, also **called search space.**

**Countermeasure = Session keys -** If we assume a crytpo-system with a large key (and therefore a large key space) a brute force attack will likely take a good deal of time - anywhere from several hours to several years depending on a number of variables. If you use a session key for each message you encrypt, then the brute force attack provides the attacker with only the key for that one message. So, if you are encrypting 10 messages a day, each with a different session key, but it takes me a month to break each session key then I am fighting a loosing battle.

**Differential Cryptanalysis** is a potent cryptanalytic technique introduced by Biham and Shamir. Differential cryptanalysis is designed for the study and attack of DES-like cryptosystems. A DES-like cryptosystem is an iterated cryptosystem which relies on conventional cryptographic techniques such as substitution and diffusion.

Differential cryptanalysis is a general form of cryptanalysis applicable primarily to block ciphers, but also to stream ciphers and cryptographic hash functions. In the broadest sense, it is the study of how differences in an input can affect the resultant difference at the output. In the case of a block cipher, it refers to a set of techniques for tracing differences through the network of transformations, discovering where the cipher exhibits non-random behaviour, and exploiting such properties to recover the secret key.

**A chosen-ciphertext attack** is one in which cryptanalyst may choose a piece of ciphertext and attempt to obtain the corresponding decrypted plaintext. This type of attack is generally most applicable to public-key cryptosystems.

A chosen-ciphertext attack (CCA) is an attack model for cryptanalysis in which the cryptanalyst gathers information, at least in part, by choosing a ciphertext and obtaining its decryption under an unknown key. In the attack, an adversary has a chance to enter one or more known ciphertexts into the system and obtain the resulting plaintexts. From these pieces of information the adversary can attempt to recover the hidden secret key used for decryption.

A number of otherwise secure schemes can be defeated under chosen-ciphertext attack. For example, the El Gamal cryptosystem is semantically secure under chosen-plaintext attack, but this semantic security can be trivially defeated under a chosen-ciphertext attack. Early versions of RSA padding used in the SSL protocol were vulnerable to a sophisticated adaptive chosen-ciphertext attack which revealed SSL session keys. Chosen-ciphertext attacks have implications for some self-synchronizing stream ciphers as well. Designers of tamper-resistant cryptographic smart cards must be particularly cognizant of these attacks, as these devices may be completely under the control of an adversary, who can issue a large number of chosen-ciphertexts in an attempt to recover the hidden secret key.

Cryptanalytic attacks are generally classified into six categories that distinguish the kind of information the cryptanalyst has available to mount an attack. The categories of attack are listed here roughly in increasing order of the quality of information available to the cryptanalyst, or, equivalently, in decreasing order of the level of difficulty to the cryptanalyst. The objective of the cryptanalyst in all cases is to be able to decrypt new pieces of ciphertext without additional information. The ideal for a cryptanalyst is to extract the secret key.

**A ciphertext-only attack** is one in which the cryptanalyst obtains a sample of ciphertext, without the plaintext associated with it. This data is relatively easy to obtain in many scenarios, but a successful ciphertext-only attack is generally difficult, and requires a very large ciphertext sample. Such attack was possible on cipher using Code Book Mode where frequency analysis was being used and even thou only the ciphertext was available, it was still possible to eventually collect enough data and decipher it without having the key.

**A known-plaintext attack** is one in which the cryptanalyst obtains a sample of ciphertext and the corresponding plaintext as well. The known-plaintext attack (KPA) or crib is an attack model for cryptanalysis where the attacker has samples of both the plaintext and its encrypted version (ciphertext), and is at liberty to make use of them to reveal further secret information such as secret keys and code books.

**A chosen-plaintext attack** is one in which the cryptanalyst is able to choose a quantity of plaintext and then obtain the corresponding encrypted ciphertext. A chosen-plaintext attack (CPA) is an attack model for cryptanalysis which presumes that the attacker has the capability to choose arbitrary plaintexts to be encrypted and obtain the corresponding ciphertexts. The goal of the attack is to gain some further information which reduces the security of the encryption scheme. In the worst case, a chosen-plaintext attack could reveal the scheme's secret key.

This appears, at first glance, to be an unrealistic model; it would certainly be unlikely that an attacker could persuade a human cryptographer to encrypt large amounts of plaintexts of the attacker's choosing. Modern cryptography, on the other hand, is implemented in software or hardware and is used for a diverse range of applications; for many cases, a chosen-plaintext attack is often very feasible. Chosen-plaintext attacks become extremely important in the context of public key cryptography, where the encryption key is public and attackers can encrypt any plaintext they choose.

Any cipher that can prevent chosen-plaintext attacks is then also guaranteed to be secure against known-plaintext and ciphertext-only attacks; this is a conservative approach to security.

Two forms of chosen-plaintext attack can be distinguished:

**\* Batch chosen-plaintext attack,** where the cryptanalyst chooses all plaintexts before any of them are encrypted. This is often the meaning of an unqualified use of "chosen-plaintext attack".

**\* Adaptive chosen-plaintext attack,** is a special case of chosen-plaintext attack in which the cryptanalyst is able to choose plaintext samples dynamically, and alter his or her choices based on the results of previous encryptions. The cryptanalyst makes a series of interactive queries, choosing subsequent plaintexts based on the information from the previous encryptions.Non-randomized (deterministic) public key encryption algorithms are vulnerable to simple "dictionary"-type attacks, where the attacker builds a table of likely messages and their corresponding ciphertexts. To find the decryption of some observed ciphertext, the attacker simply looks the ciphertext up in the table. As a result, public-key definitions of security under chosen-plaintext attack require probabilistic encryption (i.e., randomized encryption). Conventional symmetric ciphers, in which the same key is used to encrypt and decrypt a text, may also be vulnerable to other forms of chosen-plaintext attack, for example, differential cryptanalysis of block ciphers.

An adaptive-chosen-ciphertext is the adaptive version of the above attack. A cryptanalyst can mount an attack of this type in a scenario in which he has free use of a piece of decryption hardware, but is unable to extract the decryption key from it.

An adaptive chosen-ciphertext attack (abbreviated as CCA2) is an interactive form of chosen-ciphertext attack in which an attacker sends a number of ciphertexts to be decrypted, then uses the results of these decryptions to select subsequent ciphertexts. It is to be distinguished from an indifferent chosen-ciphertext attack (CCA1).

The goal of this attack is to gradually reveal information about an encrypted message, or about the decryption key itself. For public-key systems, adaptive-chosen-ciphertexts are generally applicable only when they have the property of ciphertext malleability — that is, a ciphertext can be modified in specific ways that will have a predictable effect on the decryption of that message.

**Frequency Analysis**

Simple substitution and transposition ciphers are vulnerable to attacks that perform frequency analysis.

In every language, there are words and patterns that are used more than others.

Some patterns common to a language can actually help attackers figure out the transformation between plaintext and ciphertext, which enables them to figure out the key that was used to perform the transformation. **Polyalphabetic ciphers use** different alphabets to defeat frequency analysis.

The ceasar cipher is a very simple substitution cipher that can be easily defeated and it does show repeating letters.

Out of list presented, it is the Polyalphabetic cipher that would provide the best protection against simple frequency analysis attacks.

**Cross Certification**

The correct answer is: Creating trust between different PKIs More and more organizations are setting up their own internal PKIs. When these independent PKIs need to interconnect to allow for secure communication to take place (either between departments or different companies), there must be a way for the two root CAs to trust each other. These two CAs do not have a CA above them they can both trust, so they must carry out cross certification. A cross certification is the process undertaken by CAs to establish a trust relationship in which they rely upon each other's digital certificates and public keys as if they had issued them themselves. When this is set up, a CA for one company can validate digital certificates from the other company and vice versa.

Cross-certification is the act or process by which two CAs each certifiy a public key of the other, issuing a public-key certificate to that other CA, enabling users that are certified under different certification hierarchies to validate each other's certificate.

In order to protect against fraud in electronic fund transfers (EFT), the **Message Authentication Code (MAC),** ANSI X9.9, was developed. The MAC is a check value, which is derived from the contents of the message itself, that is sensitive to the bit changes in a message. It is similar to a Cyclic Redundancy Check (CRC).

The aim of message authentication in computer and communication systems is to verify that he message comes from its claimed originator and that it has not been altered in transmission. It is particularly needed for EFT Electronic Funds Transfer). The protection mechanism is generation of a Message Authentication Code (MAC), attached to the message, which can be recalculated by the receiver and will reveal any alteration in transit. One standard method is described in (ANSI, X9.9). Message authentication mechanisms an also be used to achieve non-repudiation of messages.

A Message Authentication Code (MAC) is an authentication checksum derived by applying an authentication scheme, together with a secret key, to a message.

Unlike digital signatures, MACs are computed and verified with the same key, so that they can only be verified by the intended recipient.

There are four types of MACs:

**(1) unconditionally secure,**

**(2) hash function based,**

**(3) stream cipher-based and**

**4) block cipher-based.**

The correct answer is ' To detect any alteration of the message ' as the message digest is calculated and included in a digital signature to prove that the message has not been altered since the time it was created by the sender.

A keyed hash also called a MAC (message authentication code) is used for integrity protection and authenticity.

In cryptography, a message authentication code (MAC) is a generated value used to authenticate a message. A MAC can be generated by HMAC or CBC-MAC methods. The MAC protects both a message’s integrity (by ensuring that a different MAC will be produced if the message has changed) as well as its authenticity, because only someone who knows the secret key could have modified the message.

MACs differ from digital signatures as MAC values are both generated and verified using the same secret key. This implies that the sender and receiver of a message must agree on the same key before initiating communications, as is the case with symmetric encryption. For the same reason, MACs do not provide the property of non-repudiation offered by signatures specifically in the case of a network-wide shared secret key: any user who can verify a MAC is also capable of generating MACs for other messages.

**HMAC**

When using HMAC the symmetric key of the sender would be concatenated (added at the end) with the message. The result of this process (message + secret key) would be put through a hashing algorithm, and the result would be a MAC value. This MAC value is then appended to the message being sent. If an enemy were to intercept this message and modify it, he would not have the necessary symmetric key to create a valid MAC value. The receiver would detect the tampering because the MAC value would not be valid on the receiving side.

**CBC-MAC**

If a CBC-MAC is being used, the message is encrypted with a symmetric block cipher in CBC mode, and the output of the final block of ciphertext is used as the MAC. The sender does not send the encrypted version of the message, but instead sends the plaintext version and the MAC attached to the message. The receiver receives the plaintext message and encrypts it with the same symmetric block cipher in CBC mode and calculates an independent MAC value. The receiver compares the new MAC value with the MAC value sent with the message. This method does not use a hashing algorithm as does HMAC.

**Cipher-Based Message Authentication Code (CMAC)**

Some security issues with CBC-MAC were found and they created Cipher-Based Message Authentication Code (CMAC) as a replacement. CMAC provides the same type of data origin authentication and integrity as CBC-MAC, but is more secure mathematically. CMAC is a variation of CBC-MAC. It is approved to work with AES and Triple DES. HMAC, CBC-MAC, and CMAC work higher in the network stack and can identify not only transmission errors (accidental), but also more nefarious modifications, as in an attacker messing with a message for her own benefit. This means all of these technologies can identify intentional, unauthorized modifications and accidental changes— three in one.

**3C1 Vetting of proprietary cryptography**

**3C2 Computational overhead**

**3C3 Useful life**

**LIFESPAN -** The correct answer to this question is "Session Key" (**SHORTEST**) as session key is a symmetric key that is used to encrypt messages between two users. A session key is only good for one communication session between users.

For example , If Tanya has a symmetric key that she uses to encrypt messages between Lance and herself all the time , then this symmetric key would not be regenerated or changed. They would use the same key every time they communicated using encryption. However , using the same key repeatedly increases the chances of the key being captured and the secure communication being compromised. If , on the other hand , a new symmetric key were generated each time Lance and Tanya wanted to communicate , it would be used only during their dialog and then destroyed. if they wanted to communicate and hour later , a new session key would be created and shared.

The other answers are not correct because :

Public Key can be known to anyone.

Private Key must be known and used only by the owner.

Secret Keys are also called as Symmetric Keys, because this type of encryption relies on each user to keep the key a secret and properly protected.

**3C4 Design testable cryptographic system**

**3D Define key management lifecycle (e.g. creation, distribution, ecrow, recovery)**

**Key clustering** happens when a plaintext message generates identical ciphertext messages using the same transformation algorithm, but with different keys.

**Internet Security Association Key Management Protocol (ISAKMP**) is a key management protocol used by IPSec. ISAKMP (Internet Security Association and Key Management Protocol) is a protocol defined by RFC 2408 for establishing Security Associations (SA) and cryptographic keys in an Internet environment. ISAKMP only provides a framework for authentication and key exchange. The actual key exchange is done by the Oakley Key Determination Protocol which is a key-agreement protocol that allows authenticated parties to exchange keying material across an insecure connection using the Diffie-Hellman key exchange algorithm.

The Internet Security Association and Key Management Protocol (ISAKMP) is a framework that defines the phases for establishing a secure relationship and support for negotiation of security attributes, it does not establish sessions keys by itself, it is used along with the Oakley session key establishment protocol. The Secure Key Exchange Mechanism (SKEME) describes a secure exchange mechanism and Oakley defines the modes of operation needed to establish a secure connection.

ISAKMP provides a framework for Internet key management and provides the specific protocol support for negotiation of security attributes. Alone, it does not establish session keys. However it can be used with various session key establishment protocols, such as Oakley, to provide a complete solution to Internet key management.

**Diffie-Hellman** and one variation of the Diffie-Hellman algorithm called the Key Exchange Algorithm (KEA) are also key exchange protocols. Key exchange (also known as "key establishment") is any method in cryptography by which cryptographic keys are exchanged between users, allowing use of a cryptographic algorithm. Diffie–Hellman key exchange (D–H) is a specific method of exchanging keys. It is one of the earliest practical examples of key exchange implemented within the field of cryptography. The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher. **It deals with discrete logarithms**

**MQV (Menezes–Qu–Vanstone)** is an authenticated protocol for key agreement based on the Diffie–Hellman scheme. Like other authenticated Diffie-Hellman schemes, MQV provides protection against an active attacker. The protocol can be modified to work in an arbitrary finite group, and, in particular, elliptic curve groups, where it is known as elliptic curve MQV (ECMQV).

Both parties in the exchange calculate an implicit signature using its own private key and the other's public key.

It uses **implicit signatures.**

**Clipper Chip,** except that keys are not escrowed by law enforcement agencies, but by specialized escrow agencies.

There is nonetheless a concern with these agencies' ability to protect escrowed keys, and whether they may divulge them in unauthorized ways.

The fact that the Skipjack algorithm (used in the Clipper Chip) was never opened for public review is a concern in the fact that it has never been publicly tested to ensure that the developers did not miss any important steps in building a complex and secure mechanism.

Because the algorithm was never released for public review, many people in the public did not initially trust its effectiveness. It was declassified on 24 June 1998 and became available for public scrutiny at that point.

The Clipper Chip is a NSA designed tamperproof chip for encrypting data and it uses the **SkipJack algorithm.** Each Clipper Chip has a unique serial number and a copy of the unit key is stored in the database under this serial number. The sending Clipper Chip generates and sends a Law Enforcement Access Field (LEAF) value included in the transmitted message. It is based on **a 80-bit key and a 16-bit checksum.**

**3E Design integrated cryptographic solutions (e.g. public key infrastructure (PKI), API selection, identity system intergration)**

**PKI stands for Public Key Infrastructure.**

It supports public key exchange and it is responsible for **issuing, locating, trusting, renewing, and revoking certificates.**

**A Public Key Infrastructure (PKI) provides confidentiality, access control, integrity, authentication and non-repudiation.**

**It does not provide reliability services.**

**Public key certificate (or identity certificate)** is an electronic document which incorporates a digital signature to bind together a public key with an identity — information such as the name of a person or an organization, their address, and so forth. The certificate can be used to verify that a public key belongs to an individual.

In a typical public key infrastructure (PKI) scheme, the signature will be of a certificate authority (CA). In a web of trust scheme, the signature is of either the user (a self-signed certificate) or other users ("endorsements"). In either case, the signatures on a certificate are attestations by the certificate signer that the identity information and the public key belong together.

In computer security, an authorization certificate (also known as an attribute certificate) is a digital document that describes a written permission from the issuer to use a service or a resource that the issuer controls or has access to use. The permission can be delegated.

Some people constantly confuse PKCs and ACs. An analogy may make the distinction clear. A PKC can be considered to be like a passport: it identifies the holder, tends to last for a long time, and should not be trivial to obtain. An AC is more like an entry visa: it is typically issued by a different authority and does not last for as long a time. As acquiring an entry visa typically requires presenting a passport, getting a visa can be a simpler process.

A real life example of this can be found in the mobile software deployments by large service providers and are typically applied to platforms such as Microsoft Smartphone (and related), Symbian OS, J2ME, and others.

In each of these systems a mobile communications service provider may customize the mobile terminal client distribution (ie. the mobile phone operating system or application environment) to include one or more root certificates each associated with a set of capabilities or permissions such as "update firmware", "access address book", "use radio interface", and the most basic one, "install and execute". When a developer wishes to enable distribution and execution in one of these controlled environments they must acquire a certificate from an appropriate CA, typically a large commercial CA, and in the process they usually have their identity verified using out-of-band mechanisms such as a combination of phone call, validation of their legal entity through government and commercial databases, etc., similar to the high assurance SSL certificate vetting process, though often there are additional specific requirements imposed on would-be developers/publishers.

Once the identity has been validated they are issued an identity certificate they can use to sign their software; generally the software signed by the developer or publisher's identity certificate is not distributed but rather it is submitted to processor to possibly test or profile the content before generating an authorization certificate which is unique to the particular software release. That certificate is then used with an ephemeral asymmetric key-pair to sign the software as the last step of preparation for distribution. There are many advantages to separating the identity and authorization certificates especially relating to risk mitigation of new content being accepted into the system and key management as well as recovery from errant software which can be used as attack vectors.

**A Certificate authority** is incorrect as it is a part of PKI in which the certificate is created and signed by a trusted 3rd party.

**A Registration authority** is incorrect as it performs the certification registration duties in PKI.

A X.509 certificate is incorrect as a certificate is the mechanism used to associate a public key with a collection of components in a manner that is sufficient to uniquely identify the claimed owner.

Public key infrastructure (PKI) consists of programs, data formats, procedures, communication protocols, security policies, and public key cryptographic mechanisms working in a comprehensive manner to enable a wide range of dispersed people to communicate in a secure and predictable fashion. In other words, a PKI establishes a level of trust within an environment. PKI is an ISO authentication framework that uses public key cryptography and the X.509 standard. The framework was set up to enable authentication to happen across different networks and the Internet. Particular protocols and algorithms are not specified, which is why PKI is called a framework and not a specific technology.

PKI provides authentication, confidentiality, nonrepudiation, and integrity of the messages exchanged. PKI is a hybrid system of symmetric and asymmetric key algorithms and methods.

PKI is made up of many different parts: certificate authorities, registration authorities, certificates, keys, and users.

Each person who wants to participate in a PKI requires a digital certificate, which is a credential that contains the public key for that individual along with other identifying information. The certificate is created and signed (digital signature) by a trusted third party, which is a certificate authority (CA). When the CA signs the certificate, it binds the individual’s identity to the public key, and the CA takes liability for the authenticity of that individual. It is this trusted third party (the CA) that allows people who have never met to authenticate to each other and communicate in a secure method. If Kevin has never met David, but would like to communicate securely with him, and they both trust the same CA, then Kevin could retrieve David’s digital certificate and start the process.

Public keys are published through digital certificates, signed by certification authority (CA), binding the certificate to the identity of its bearer.

**Revocation Request Grace Period -** The length of time between the Issuer’s receipt of a revocation request and the time the Issuer is required to revoke the certificate should bear a reasonable relationship to the amount of risk the participants are willing to assume that someone may rely on a certificate for which a proper evocation request has been given but has not yet been acted upon.

How quickly revocation requests need to be processed (and CRLs or certificate status databases need to be updated) depends upon the specific application for which the Policy Authority is rafting the Certificate Policy.

A Policy Authority should recognise that there may be risk and lost tradeoffs with respect to grace periods for revocation notices.

If the Policy Authority determines that its PKI participants are willing to accept a grace period of a few hours in exchange for a lower implementation cost, the Certificate Policy may reflect that decision.

Thanks to Thomas Fung for finding a mistake in this question and providing a second reference on the subject.

Thanks to Vince Martinez for reporting issues with words that were truncated.

**Digital certificate** helps others verify that the public keys presented by users are genuine and valid.

A digital certificate is an electronic "credit card" that establishes your credentials when doing business or other transactions on the Web.

It is issued by a certification authority (CA). It contains your name, a serial number, expiration dates, a copy of the certificate holder's public key (used for encrypting messages), and the digital signature of the certificate-issuing authority so that a recipient can verify that the certificate is real. Some digital certificates conform to a standard, X.509. Digital certificates can be kept in registries so that authenticating users can look up other users' public keys.

A Digital Certificate is not like same as a digital signature, they are two different things, a digital Signature is created by using your Private key to encrypt a message digest and a Digital Certificate is issued by a trusted third party who vouch for your identity.

There are many other third parties which are providing Digital Certifictes and not just Verisign, RSA.

In cryptography, a public key certificate (also known as a digital certificate or identity certificate) is an electronic document which uses a digital signature to bind together a public key with an identity — information such as the name of a person or an organization, their address, and so forth. The certificate can be used to verify that a public key belongs to an individual.

In a typical public key infrastructure (PKI) scheme, the signature will be of a certificate authority (CA). In a web of trust scheme such as PGP or GPG, the signature is of either the user (a self-signed certificate) or other users ("endorsements") by getting people to sign each other keys. In either case, the signatures on a certificate are attestations by the certificate signer that the identity information and the public key belong together.

An entity that issues digital certificates (especially X.509 certificates) and vouches for the binding between the data items in a certificate.

An authority trusted by one or more users to create and assign certificates. Optionally, the certification authority may create the user's keys.

X509 Certificate users depend on the validity of information provided by a certificate. Thus, a CA should be someone that certificate users trust, and usually holds an official position created and granted power by a government, a corporation, or some other organization. A CA is responsible for managing the life cycle of certificates and, depending on the type of certificate and the CPS that applies, may be responsible for the life cycle of key pairs associated with the certificates

Users can obtain certificates with various levels of assurance.

**Class 1/Level 1** for individuals, intended for email, no proof of identityFor example, level 1 certificates verify electronic mail addresses. This is done through the use of a personal information number that a user would supply when asked to register. This level of certificate may also provide a name as well as an electronic mail address; however, it may or may not be a genuine name (i.e., it could be an alias). This proves that a human being will reply back if you send an email to that name or email address.

**Class 2/Level 2** is for organizations and companies for which proof of identity is requiredLevel 2 certificates verify a user's name, address, social security number, and other information against a credit bureau database.

**Class 3/Level 3** is for servers and software signing, for which independent verification and checking of identity and authority is done by the issuing certificate authorityLevel 3 certificates are available to companies. This level of certificate provides photo identification to accompany the other items of information provided by a level 2 certificate.

**Class 4** for online business transactions between companies

**Class 5** for private organizations or governmental security

The Internet Security Glossary (RFC2828) defines **an attribute certificate** as a digital certificate that binds a set of descriptive data items, other than a public key, either directly to a subject name or to the identifier of another certificate that is a public-key certificate. A public-key certificate binds a subject name to a public key value, along with information needed to perform certain cryptographic functions. Other attributes of a subject, such as a security clearance, may be certified in a separate kind of digital certificate, called an attribute certificate. A subject may have multiple attribute certificates associated with its name or with each of its public-key certificates.

**CRL**

Certificate revocation is the process of revoking a certificate before it expires.

A certificate may need to be revoked because it was stolen, an employee moved to a new company, or someone has had their access revoked. A certificate revocation is handled either through a Certificate Revocation List (CRL) or by using the Online Certificate Status Protocol (OCSP).

A repository is simply a database or database server where the certificates are stored. The process of revoking a certificate begins when the CA is notified that a particular certificate needs to be revoked. This must be done whenever the private key becomes known/compromised.

The owner of a certificate can request it be revoked at any time, or the request can be made by the administrator. The CA marks the certificate as revoked. This information is published in the CRL. The revocation process is usually very quick; time is based on the publication interval for the CRL.

Disseminating the revocation information to users may take longer. Once the certificate has been revoked, it can never be used—or trusted—again. The CA publishes the CRL on a regular basis, usually either hourly or daily. The CA sends or publishes this list to organizations that have chosen to receive it; the publishing process occurs automatically in the case of PKI. The time between when the CRL is issued and when it reaches users may be too long for some applications. This time gap is referred to as latency.

OCSP solves the latency problem: If the recipient or relaying party uses OCSP for verification, the answer is available immediately.

**ARL -** The Internet Security Glossary (RFC2828) defines the **Authority Revocation List (ARL)** as a data structure that enumerates digital certificates that were issued to CAs but have been invalidated by their issuer prior to when they were scheduled to expire.

Do not to confuse with an ARL with a Certificate Revocation List (CRL). A certificate revocation list is a mechanism for distributing notices of certificate revocations.

**Registration Authority (RA)** A registration authority (RA) is an authority in a network that verifies user requests for a digital certificate and tells the certificate authority (CA) to issue it. RAs are part of a public key infrastructure (PKI), a networked system that enables companies and users to exchange information and money safely and securely. The digital certificate contains a public key that is used to encrypt and decrypt messages and digital signatures.

**Recovery agent** Sometimes it is necessary to recover a lost key. One of the problems that often arises regarding PKI is the fear that documents will become lost forever—irrecoverable because someone loses or forgets his private key. Let’s say that employees use Smart Cards to hold their private keys. If a user was to leave his Smart Card in his or her wallet that was left in the pants that he or she accidentally threw into the washing machine, then that user might be without his private key and therefore incapable of accessing any documents or e-mails that used his existing private key.

Many corporate environments implement a key recovery server solely for the purpose of backing up and recovering keys. Within an organization, there typically is at least one key recovery agent. A key recovery agent has the authority and capability to restore a user’s lost private key. Some key recovery servers require that two key recovery agents retrieve private user keys together for added security. This is similar to certain bank accounts, which require two signatures on a check for added security. Some key recovery servers also have the ability to function as a key escrow server, thereby adding the ability to split the keys onto two separate recovery servers, further increasing security.

**Key escrow** (also known as a “fair” cryptosystem) is an arrangement in which the keys needed to decrypt encrypted data are held in escrow so that, under certain circumstances, an authorized third party may gain access to those keys. These third parties may include businesses, who may want access to employees' private communications, or governments, who may wish to be able to view the contents of encrypted communications.

**----------------------------------**

**OTHER**

The **Internet Key Exchange (IKE) protocol** is a key management protocol standard that is used in conjunction with the IPSec standard. IKE enhances IPSec by providing additional features, flexibility, and ease of configuration for the IPSec standard. IPSec can however, be configured without IKE by manually configuring the gateways communicating with each other for example.

A security association (SA) is a relationship between two or more entities that describes how the entities will use security services to communicate securely.

In phase 1 of this process, IKE creates an authenticated, secure channel between the two IKE peers, called the IKE security association. The Diffie-Hellman key agreement is always performed in this phase.

In phase 2 IKE negotiates the IPSec security associations and generates the required key material for IPSec. The sender offers one or more transform sets that are used to specify an allowed combination of transforms with their respective settings.

**Benefits provided by IKE include:**

* **Eliminates the need to manually specify all the IPSec security parameters in the crypto maps at both peers.**
* **Allows you to specify a lifetime for the IPSec security association.**
* **Allows encryption keys to change during IPSec sessions.**
* **Allows IPSec to provide anti-replay services.**
* **Permits Certification Authority (CA) support for a manageable, scalable IPSec implementation.**
* **Allows dynamic authentication of peers.**

RFC 2828 (Internet Security Glossary) defines IKE as an Internet, IPsec, key-establishment protocol (partly based on OAKLEY) that is intended for putting in place authenticated keying material for use with ISAKMP and for other security associations, such as in AH and ESP.

**IKE does not need a Public Key Infrastructure (PKI) to work.** Internet Key Exchange (IKE or IKEv2) is the protocol used to set up a security association (SA) in the IPsec protocol suite. IKE builds upon the Oakley protocol and ISAKMP. IKE uses X.509 certificates for authentication which are either pre-shared or distributed using DNS (preferably with DNSSEC) and a Diffie–Hellman key exchange to set up a shared session secret from which cryptographic keys are derived.

**OAKLEY -** RFC 2828 (Internet Security Glossary) defines OAKLEY as a key establishment protocol (proposed for IPsec but superseded by IKE) based on the Diffie-Hellman algorithm and designed to be a compatible component of ISAKMP.

ISAKMP is an Internet IPsec protocol to negotiate, establish, modify, and delete security associations, and to exchange key generation and authentication data, independent of the details of any specific key generation technique, key establishment protocol, encryption algorithm, or authentication mechanism.

The Oakley protocol uses a hybrid Diffie-Hellman technique to establish session keys on Internet hosts and routers. Oakley provides the important security property of Perfect Forward Secrecy (PFS) and is based on cryptographic techniques that have survived substantial public scrutiny. Oakley can be used by itself, if no attribute negotiation is needed, or Oakley can be used in conjunction with ISAKMP. When ISAKMP is used with Oakley, key escrow is not feasible.

The ISAKMP and Oakley protocols have been combined into a hybrid protocol. The resolution of ISAKMP with Oakley uses the framework of ISAKMP to support a subset of Oakley key exchange modes. This new key exchange protocol provides optional PFS, full security association attribute negotiation, and authentication methods that provide both repudiation and non-repudiation. Implementations of this protocol can be used to establish VPNs and also allow for users from remote sites (who may have a dynamically allocated IP address) access to a secure network.

**SKEME** describes a versatile key exchange technique which provides anonymity, repudiability, and quick key refreshment. SKEME constitutes a compact protocol that supports a variety of realistic scenarios and security models over Internet. It provides clear tradeoffs between security and performance as required by the different scenarios without incurring in unnecessary system complexity. The protocol supports key exchange based on public key, key distribution centers, or manual installation, and provides for fast and secure key refreshment. In addition, SKEME selectively provides perfect forward secrecy, allows for replaceability and negotiation of the underlying cryptographic primitives, and addresses privacy issues as anonymity and repudiatability

SKEME's basic mode is based on the use of public keys and a Diffie-Hellman shared secret generation.

However, SKEME is not restricted to the use of public keys, but also allows the use of a pre-shared key. This key can be obtained by manual distribution or by the intermediary of a key distribution center (KDC) such as Kerberos.

In short, SKEME contains four distinct modes:

Basic mode, which provides a key exchange based on public keys and ensures PFS thanks to Diffie-Hellman. A key exchange based on the use of public keys, but without Diffie-Hellman. A key exchange based on the use of a pre-shared key and on Diffie-Hellman. A mechanism of fast rekeying based only on symmetrical algorithms. In addition, SKEME is composed of three phases: SHARE, EXCH and AUTH.

During the SHARE phase, the peers exchange half-keys, encrypted with their respective public keys. These two half-keys are used to compute a secret key K. If anonymity is wanted, the identities of the two peers are also encrypted. If a shared secret already exists, this phase is skipped.

The exchange phase (EXCH) is used, depending on the selected mode, to exchange either Diffie-Hellman public values or nonces. The Diffie-Hellman shared secret will only be computed after the end of the exchanges.

The public values or nonces are authenticated during the authentication phase (AUTH), using the secret key established during the SHARE phase.

**SKIP** is a key distribution protocol that uses hybrid encryption to convey session keys that are used to encrypt data in IP packets. RFC 2828 (Internet Security Glossary) defines Simple Key Management for Internet Protocols (SKIP) as:

A key distribution protocol that uses hybrid encryption to convey session keys that are used to encrypt data in IP packets.

SKIP is an hybrid Key distribution protocol similar to SSL, except that it establishes a long-term key once, and then requires no prior communication in order to establish or exchange keys on a session-by-session basis. Therefore, no connection setup overhead exists and new keys values are not continually generated. SKIP uses the knowledge of its own secret key or private component and the destination's public component to calculate a unique key that can only be used between them.

**The Key Exchange Algorithm (KEA)** is defined as a key agreement algorithm that is similar to the Diffie-Hellman algorithm, uses 1024-bit asymmetric keys, and was developed and formerly classified at the secret level by the NSA.

**FIPS-140 -** Hardware and software cryptographic modules

**SSL - The Secure Socket Layer (SSL)** and also the Transport Layer Security (TLS) protocols are used for the encryption of Hypertext Transport Protocol (HTTP) data between a Web Browser and a Web Server.

SSL/TLS and The Internet Protocol Security (IPSec) protocol suite both provides a method of setting up a secure channel for protecting data exchange between two entities wishing to communicate securely with each other.

The biggest difference between IPSEC and SSL is:

Using IPSEC the encryption is done at the Network Layer of the OSI model. The IPSEC devices that share this secure channel can be two servers, two routers, a workstation and a server, or two gateways between different networks. It is always from a HOST to another HOST.

SSL/TLS is used for APPLICATION to APPLICATION secure channels. The question was making reference specifically to a Web Browser, being an Application this ruled out IPSEC as a valid choice.

For the purpose of the exam you must understand these differences.

SSL provides security services at the Transport Layer of the OSI model.

**SSL 3.0 (Secure Socket Layer) and TLS 1.1 (Transport Layer Security)** are essentially fully compatible, with SSL being a session encryption tool originally developed by Netscape and TLS 1.1 being the open standard IETF version of SSL 3.0.

SSL is one of the most common protocols used to protect Internet traffic. It encrypts the messages using symmetric algorithms, such as IDEA, DES, 3DES, and Fortezza, and also calculates the MAC (Message Authentication Code) for the message using MD5 or SHA-1.

The MAC is appended to the message and encrypted along with the message data. The exchange of the symmetric keys is accomplished through various versions of Diffie–Hellmann or RSA. TLS is the Internet standard based on SSLv3. TLSv1 is backward compatible with SSLv3. It uses the same algorithms as SSLv3; however, it computes an HMAC instead of a MAC along with other enhancements to improve security.

Transport The protocols at the transport layer handle end-to-end transmission and segmentation of a data stream. The following protocols work at this layer: • Transmission Control Protocol (TCP) • User Datagram Protocol (UDP) • Secure Sockets Layer (SSL)/ Transport Layer Security (TLS) • Sequenced Packet Exchange (SPX).

Once the merchant server has been authenticated by the browser client, the **browser generates a master secret that** is to be shared only between the server and client. This secret serves as a seed to generate the session (private) keys. The master secret is then encrypted with the merchant's public key and sent to the server. The fact that the master secret is generated by the client's browser provides the client assurance that the server is not reusing keys that would have been used in a previous session with another client.

**Cryptology** is the science that includes both cryptography and cryptanalysis and is not directly concerned with key management. Cryptology is the mathematics, such as number theory, and the application of formulas and algorithms, that underpin cryptography and cryptanalysis.

**The Secure Electronic Transaction (SET)** was developed by a consortium including MasterCard and VISA as a means of preventing fraud from occurring during electronic payment.

**Kerberos** depends on Secret Keys or Symmetric Key cryptography.

Kerberos a third party authentication protocol. It was designed and developed in the mid 1980's by MIT. It is considered open source but is copyrighted and owned by MIT. It relies on the user's secret keys. The password is used to encrypt and decrypt the keys.

This question asked specifically about encryption methods. Encryption methods can be SYMMETRIC (or secret key) in which encryption and decryption keys are the same, or ASYMMETRIC (aka 'Public Key') in which encryption and decryption keys differ.

'Public Key' methods must be asymmetric, to the extent that the decryption key CANNOT be easily derived from the encryption key. Symmetric keys, however, usually encrypt more efficiently, so they lend themselves to encrypting large amounts of data. Asymmetric encryption is often limited to ONLY encrypting a symmetric key and other information that is needed in order to decrypt a data stream, and the remainder of the encrypted data uses the symmetric key method for performance reasons. This does not in any way diminish the security nor the ability to use a public key to encrypt the data, since the symmetric key method is likely to be even MORE secure than the asymmetric method.

For symmetric key ciphers, there are basically two types: BLOCK CIPHERS, in which a fixed length block is encrypted, and STREAM CIPHERS, in which the data is encrypted one 'data unit' (typically 1 byte) at a time, in the same order it was received in.

**IPSEC**

The IETF's IPSec Working Group develops standards for IP-layer security mechanisms for both IPv4 and IPv6. The group also is developing generic key management protocols for use on the Internet. For more information, refer to the IP Security and Encryption Overview.

IPSec is a framework of open standards developed by the Internet Engineering Task Force (IETF) that provides security for transmission of sensitive information over unprotected networks such as the Internet. It acts at the network level and implements the following standards:

•**IPSec**

**•Internet Key Exchange (IKE)**

**•Data Encryption Standard (DES)**

**•MD5 (HMAC variant)**

**•SHA (HMAC variant)**

**•Authentication Header (AH)**

**•Encapsulating Security Payload (ESP)**

IPSec services provide a robust security solution that is standards-based. IPSec also provides data authentication and anti-replay services in addition to data confidentiality services.

**Why even bother with AH then?**

In most cases, the reason has to do with whether the environment is using network address translation (NAT). IPSec will generate an integrity check value (ICV), which is really the same thing as a MAC value, over a portion of the packet. Remember that the sender and receiver generate their own values. In IPSec, it is called an ICV value. The receiver compares her ICV value with the one sent by the sender. If the values match, the receiver can be assured the packet has not been modified during transmission. If the values are different, the packet has been altered and the receiver discards the packet.

The AH protocol calculates this ICV over the data payload, transport, and network headers. If the packet then goes through a NAT device, the NAT device changes the IP address of the packet. That is its job. This means a portion of the data (network header) that was included to calculate the ICV value has now changed, and the receiver will generate an ICV value that is different from the one sent with the packet, which means the packet will be discarded automatically.

The ESP protocol follows similar steps, except it does not include the network header portion when calculating its ICV value. When the NAT device changes the IP address, it will not affect the receiver’s ICV value because it does not include the network header when calculating the ICV.

**Here is a tutorial on IPSEC from the Shon Harris Blog:**

The Internet Protocol Security (IPSec) protocol suite provides a method of setting up a secure channel for protected data exchange between two devices. The devices that share this secure channel can be two servers, two routers, a workstation and a server, or two gateways between different networks. IPSec is a widely accepted standard for providing network layer protection. It can be more flexible and less expensive than end-to end and link encryption methods.

IPSec has strong encryption and authentication methods, and although it can be used to enable tunneled communication between two computers, it is usually employed to establish virtual private networks (VPNs) among networks across the Internet.

IPSec is not a strict protocol that dictates the type of algorithm, keys, and authentication method to use. Rather, it is an open, modular framework that provides a lot of flexibility for companies when they choose to use this type of technology. IPSec uses two basic security protocols: Authentication Header (AH) and Encapsulating Security Payload (ESP). AH is the authenticating protocol, and ESP is an authenticating and encrypting protocol that uses cryptographic mechanisms to provide source authentication, confidentiality, and message integrity.

IPSec can work in one of two modes: transport mode, in which the payload of the message is protected, and tunnel mode, in which the payload and the routing and header information are protected. ESP in transport mode encrypts the actual message information so it cannot be sniffed and uncovered by an unauthorized entity. Tunnel mode provides a higher level of protection by also protecting the header and trailer data an attacker may find useful. Figure 8-26 shows the high-level view of the steps of setting up an IPSec connection.

Each device will have at least one security association (SA) for each VPN it uses. The SA, which is critical to the IPSec architecture, is a record of the configurations the device needs to support an IPSec connection. When two devices complete their handshaking process, which means they have agreed upon a long list of parameters they will use to communicate, these data must be recorded and stored somewhere, which is in the SA.

The SA can contain the authentication and encryption keys, the agreed-upon algorithms, the key lifetime, and the source IP address. When a device receives a packet via the IPSec protocol, it is the SA that tells the device what to do with the packet. So if device B receives a packet from device C via IPSec, device B will look to the corresponding SA to tell it how to decrypt the packet, how to properly authenticate the source of the packet, which key to use, and how to reply to the message if necessary.

SAs are directional, so a device will have one SA for outbound traffic and a different SA for inbound traffic for each individual communication channel. If a device is connecting to three devices, it will have at least six SAs, one for each inbound and outbound connection per remote device. So how can a device keep all of these SAs organized and ensure that the right SA is invoked for the right connection? With the mighty secu rity parameter index (SPI), that’s how. Each device has an SPI that keeps track of the different SAs and tells the device which one is appropriate to invoke for the different packets it receives. The SPI value is in the header of an IPSec packet, and the device reads this value to tell it which SA to consult.

IPSec can authenticate the sending devices of the packet by using MAC (covered in the earlier section, “The One-Way Hash”). The ESP protocol can provide authentication, integrity, and confidentiality if the devices are configured for this type of functionality.

So if a company just needs to make sure it knows the source of the sender and must be assured of the integrity of the packets, it would choose to use AH. If the company would like to use these services and also have confidentiality, it would use the ESP protocol because it provides encryption functionality. In most cases, the reason ESP is employed is because the company must set up a secure VPN connection.

It may seem odd to have two different protocols that provide overlapping functionality. AH provides authentication and integrity, and ESP can provide those two functions and confidentiality. Why even bother with AH then? In most cases, the reason has to do with whether the environment is using network address translation (NAT). IPSec will generate an integrity check value (ICV), which is really the same thing as a MAC value, over a portion of the packet. Remember that the sender and receiver generate their own values. In IPSec, it is called an ICV value. The receiver compares her ICV value with the one sent by the sender. If the values match, the receiver can be assured the packet has not been modified during transmission. If the values are different, the packet has been altered and the receiver discards the packet.

The AH protocol calculates this ICV over the data payload, transport, and network headers. If the packet then goes through a NAT device, the NAT device changes the IP address of the packet. That is its job. This means a portion of the data (network header) that was included to calculate the ICV value has now changed, and the receiver will generate an ICV value that is different from the one sent with the packet, which means the packet will be discarded automatically.

The ESP protocol follows similar steps, except it does not include the network header portion when calculating its ICV value. When the NAT device changes the IP address, it will not affect the receiver’s ICV value because it does not include the network header when calculating the ICV.

Because IPSec is a framework, it does not dictate which hashing and encryption algorithms are to be used or how keys are to be exchanged between devices. Key management can be handled manually or automated by a key management protocol. The de facto standard for IPSec is to use Internet Key Exchange (IKE), which is a combination of the ISAKMP and OAKLEY protocols. The Internet Security Association and Key Management Protocol (ISAKMP) is a key exchange architecture that is independent of the type of keying mechanisms used. Basically, ISAKMP provides the framework of what can be negotiated to set up an IPSec connection (algorithms, protocols, modes, keys). The OAKLEY protocol is the one that carries out the negotiation process. You can think of ISAKMP as providing the playing field (the infrastructure) and OAKLEY as the guy running up and down the playing field (carrying out the steps of the negotiation).

IPSec is very complex with all of its components and possible configurations. This complexity is what provides for a great degree of flexibility, because a company has many different configuration choices to achieve just the right level of protection. If this is all new to you and still confusing, please review one or more of the following references to help fill in the gray areas.

**Pre Shared Keys**

In cryptography, a pre-shared key or PSK is a shared secret which was previously shared between the two parties using some secure channel before it needs to be used. To build a key from shared secret, the key derivation function should be used. Such systems almost always use symmetric key cryptographic algorithms. The term PSK is used in WiFi encryption such as WEP or WPA, where both the wireless access points (AP) and all clients share the same key.

The characteristics of this secret or key are determined by the system which uses it; some system designs require that such keys be in a particular format. It can be a password like 'bret13i', a passphrase like 'Idaho hung gear id gene', or a hexadecimal string like '65E4 E556 8622 EEE1'. The secret is used by all systems involved in the cryptographic processes used to secure the traffic between the systems.

**Certificate Based Authentication**

The most common form of trusted authentication between parties in the wide world of Web commerce is the exchange of certificates. A certificate is a digital document that at a minimum includes a Distinguished Name (DN) and an associated public key.

The certificate is digitally signed by a trusted third party known as the Certificate Authority (CA). The CA vouches for the authenticity of the certificate holder. Each principal in the transaction presents certificate as its credentials. The recipient then validates the certificate’s signature against its cache of known and trusted CA certificates. A “personal

certificate” identifies an end user in a transaction; a “server certificate” identifies the service provider.

Generally, certificate formats follow the X.509 Version 3 standard. X.509 is part of the Open Systems Interconnect

X.509 is used in digital certificates. X.400 is used in e-mail as a message handling protocol. X.25 is a standard for the network and data link levels of a communication network and X.75 is a standard defining ways of connecting two X.25 networks.

**Public Key Authentication**

Public key authentication is an alternative means of identifying yourself to a login server, instead of typing a password. It is more secure and more flexible, but more difficult to set up.

In conventional password authentication, you prove you are who you claim to be by proving that you know the correct password. The only way to prove you know the password is to tell the server what you think the password is. This means that if the server has been hacked, or spoofed an attacker can learn your password.

Public key authentication solves this problem. You generate a key pair, consisting of a public key (which everybody is allowed to know) and a private key (which you keep secret and do not give to anybody). The private key is able to generate signatures. A signature created using your private key cannot be forged by anybody who does not have a copy of that private key; but anybody who has your public key can verify that a particular signature is genuine.

So you generate a key pair on your own computer, and you copy the public key to the server. Then, when the server asks you to prove who you are, you can generate a signature using your private key. The server can verify that signature (since it has your public key) and allow you to log in. Now if the server is hacked or spoofed, the attacker does not gain your private key or password; they only gain one signature. And signatures cannot be re-used, so they have gained nothing.

There is a problem with this: if your private key is stored unprotected on your own computer, then anybody who gains access to your computer will be able to generate signatures as if they were you. So they will be able to log in to your server under your account. For this reason, your private key is usually encrypted when it is stored on your local machine, using a passphrase of your choice. In order to generate a signature, you must decrypt the key, so you have to type your passphrase.

**SPLIT KNOWLEDGE**

Split knowledge involves encryption keys being separated into two components, each of which does not reveal the other. Split knowledge is the other complementary access control principle to dual control.

In cryptographic terms, one could say dual control and split knowledge are properly implemented if no one person has access to or knowledge of the content of the complete cryptographic key being protected by the two rocesses.

The sound implementation of dual control and split knowledge in a cryptographic environment necessarily means that the quickest way to break the key would be through the best attack known for the algorithm of that key. The principles of dual control and split knowledge primarily apply to access to plaintext keys.

Access to cryptographic keys used for encrypting and decrypting data or access to keys that are encrypted under a master key (which may or may not be maintained under dual control and split knowledge) do not require dual control and split knowledge. Dual control and split knowledge can be summed up as the determination of any part of a key being protected must require the collusion between two or more persons with each supplying unique cryptographic materials that must be joined together to access the protected key.

Any feasible method to violate the axiom means that the principles of dual control and split knowledge are not being upheld.

Split knowledge is the unique “what each must bring” and joined together when implementing dual control. To illustrate, a box containing petty cash is secured by one combination lock and one keyed lock. One employee is given the combination to the combo lock and another employee has possession of the correct key to the keyed lock.

In order to get the cash out of the box both employees must be present at the cash box at the same time. One cannot open the box without the other. This is the aspect of dual control.

On the other hand, split knowledge is exemplified here by the different objects (the combination to the combo lock and the correct physical key), both of which are unique and necessary, that each brings to the meeting. Split knowledge focuses on the uniqueness of separate objects that must be joined together.

Dual control has to do with forcing the collusion of at least two or more persons to combine their split knowledge to gain access to an asset. Both split knowledge and dual control complement each other and are necessary functions that implement the segregation of duties in high integrity cryptographic environments.

**Dual control is** a procedure that uses two or more entities (usually persons) operating in concert to protect a system resource, such that no single entity acting alone can access that resource. Dual control is implemented as a security procedure that requires two or more persons to come together and collude to complete a process. In a cryptographic system the two (or more) persons would each supply a unique key, that when taken together, performs a cryptographic process. Split knowledge is the other complementary access control principle to dual control.

**PPTP -** PPTP is an encapsulation protocol based on PPP that works at OSI layer 2 (Data Link) and that enables a single point-to-point connection, usually between a client and a server. While PPTP depends on IP to establish its connection.

As currently implemented, PPTP encapsulates PPP packets using a modified version of the generic routing encapsulation (GRE) protocol, which gives PPTP to the flexibility of handling protocols other than IP, such as IPX and NETBEUI over IP networks.

PPTP does have some limitations:

It does not provide strong encryption for protecting data, nor does it support any token-based methods for authenticating users.

**L2TP is derived from L2F and PPTP, not the opposite.**

Thanks to Remigio Armano for providing feedback to improve the quality of this question.

Thanks to John Baker for finding two good answer to this question.

**L2F (Layer 2 Forwarding)** provides no authentication or encryption. It is a Protocol that supports the creation of secure virtual private dial-up networks over the Internet.

At one point L2F was merged with PPTP to produce L2TP to be used on networks and not only on dial up links.

IPSec is now considered the best VPN solution for IP environments.

**LDAP vs CA Based PKI**

The primary security concerns relative to LDAP servers are availability and integrity.

For example, denial of service attacks on an LDAP server could prevent access to the Certificate Revocation List and, thus, permit the use of a revoked certificate.

Below you have a small extract comparing LDAP Based PKI versus CA Based PKI comparing benefits and disadvantages.

When you compare / contrast the use of LDAP vs CA based PKI, you will find that:

LDAP is used as a general directory, retaining all kinds of information. For example, a large company may use LDAP to store:

- First, middle, and last name

- Office location including address and phone number

- Electronic addresses including email, web pages and so on

- Authentication information such as user name, password, key informationand so on.

CA based PKI is primarily to establish a trust relationship between two people, two companies, or an individual with a company. For xample, when I go to a secure web site such as https://answers.google.com/answers/myquestions

to check for questions I have asked, I get an assurance from my browser that the link established between my computer and Google's is secure and that Google is indeed the company I have contacted.

With regard to issuing, storing, sending, retrieving certificates, let's start with the fact that an LDAP server is set up by the company or organization that is responsible for it. There is no specific requirement for an external agency (or

company) such as a Certificate Authority be involved in the set up of an LDAP server. As a result of this, the following will summarize and contrast for each item:

To issue a certificate in LDAP anyone with authorization to update that part of the server at the issuing company can generate it. For a CA-PKI, there are recognized companies (the Certification Authority) that will issue the certificate for the company to use. In the latter case, the CA will also make available information to clients to verify that the certificate they receive is valid.

At this point with LDAP, the certificate information is stored on the LDAP server (along with the other information stored). It may also be delivered to other systems requiring that information. In CA-PKI, you have two parts:

- the first part goes to the company / person registered with the CA

- the second part goes to the client (or perhaps better, the client application)

so the two people / companies can establish the trust relationship.

With LDAP, there is a defined method to request the information. As an example, let's use an LDAP server to store user names / passwords for a group of workstation. The workstation will request the username and password, encode the password, and compare it to the value fetched from the LDAP server. If they match, the user is authenticated.

With CA-PKI, and you go to a secure web site (as an example), part of the setup of the connection will be an exchange of trust information. The two parts of the information distributed / stored above are used to help establish a secure connection and provide information that the user can use to verify everything is OK. For example, when I connect to my bank using a secure web browser, I can pull up a description of the certificate indicating:

- the name of the company I am dealing with

- the name of the Certification Authority

- the range of dates the certificate is valid

as well as a variety of other information that experts could use to verify the certificate (the application already said its OK).

What are the advantages / disadvantages?

This was answered in part above based on the different types of application domains. In terms of security, LDAP is only as secure as the server on which the information is stored. For many purposes, the LDAP server is the authoritative basis of the information (as above - the user authentication). This also works best when the access to the network is controlled so you can prevent "man in the middle" attacks and similar problems.

In CA-PKI systems, the security comes from

- an independent company / organization that identifies the company / person that will receive the certificate

- independent distribution of public key information (the private part at the company, the public part in each client application)

- strong encryption methods used to validate the identify of both systems this works on both secure networks as insecure ones (such as the Internet).

**Pretty Good Privacy (PGP)** was designed by Phil Zimmerman as a freeware e-mail security program and was released in 1991. It was the first widespread public key encryption program. PGP is a complete cryptosystem that uses cryptographic protection to protect e-mail and files. It can use RSA public key encryption for key management and **use IDEA symmetric cipher** for bulk encryption of data, although the user has the option of picking different types of algorithms for these functions. PGP can provide confidentiality by using the IDEA encryption algorithm, integrity by using the MD5 hashing algorithm, authentication by using the public key certificates, and nonrepudiation by using cryptographically signed messages. PGP initially used its own type of digital certificates rather than what is used in PKI, but they both have similar purposes. **Today PGP support X.509 V3 digital certificates.**

Notice that the question specifically asks what PGP uses to encrypt For this, PGP uses an symmetric key algorithm. PGP then uses an asymmetric key algorithm to encrypt the session key and then send it securely to the receiver. It is an hybrid system where both types of ciphers are being used for different purposes.

Whenever a question talks about the bulk of the data to be sent, Symmetric is always best to choice to use because of the inherent speed within Symmetric Ciphers. Asymmetric ciphers **are 100 to 1000 times slower than Symmetric Ciphers.**

**The OCSP** - (Online Certificate Status Protocol) provides real-time certificate checks and a Certificate Revocation List (CRL) has a delay in the updates. In lieu of or as a supplement to checking against a periodic CRL, it may be necessary to obtain timely information regarding the revocation status of a certificate (cf. [RFC2459], Section 3.3). Examples include high-value funds transfer or large stock trades.

The Online Certificate Status Protocol (OCSP) enables applications to determine the (revocation) state of an identified certificate. OCSP may be used to satisfy some of the operational requirements of providing more timely revocation information than is possible with CRLs and may also be used to obtain additional status information. An OCSP client issues a status request to an OCSP responder and suspends acceptance of the certificate in question until the responder provides a response.

This protocol specifies the data that needs to be exchanged between an application checking the status of a certificate and the server providing that status.

**SMIME -** Content security measures presumes that the content is available in cleartext on the central mail server.

Encrypted emails have to be decrypted before it can be filtered (e.g. to detect viruses), so you need the decryption key on the central "crypto mail server".

There are several ways for such key management, e.g. by message or key recovery methods. However, that would certainly require further processing in order to achieve such goal. It is **Public key based, hybrid encryption scheme.**

**LINK Encryption and END-TO-END ENCRYPTION**

In link encryption, each entity has keys in common with its two neighboring nodes in the transmission chain.

Thus, a node receives the encrypted message from its predecessor, decrypts it, and then re-encrypts it with a new key, common to the successor node. Obviously, this mode does not provide protection if anyone of the nodes along the transmission path is compromised.

Encryption can be performed at different communication levels, each with different types of protection and implications. Two general modes of encryption implementation are link encryption and end-to-end encryption.

Link encryption encrypts all the data along a specific communication path, as in a satellite link, T3 line, or telephone circuit. Not only is the user information encrypted, but the header, trailers, addresses, and routing data that are part of the packets are also encrypted. The only traffic not encrypted in this technology is the data link control messaging information, which includes instructions and parameters that the different link devices use to synchronize communication methods. Link encryption provides protection against packet sniffers and eavesdroppers.

In end-to-end encryption, the headers, addresses, routing, and trailer information are not encrypted, enabling attackers to learn more about a captured packet and where it is headed.

When using **link encryption**, packets **have to be decrypted at each hop and encrypted again.** Information staying encrypted from one end of its journey to the other is a characteristic of end-to-end encryption, not link encryption.

Link Encryption vs. End-to-End EncryptionLink encryption encrypts the entire packet, including headers and trailers, and has to be decrypted at each hop. End-to-end encryption does not encrypt the headers and trailers, and therefore does not need to be decrypted at each hop.Reference: All in one 4th Edition, Page 735 & Glossary

**WTLS**

Wireless Transport Layer Security (WTLS) is a communication protocol that allows wireless devices to send and receive encrypted information over the Internet. S-WAP is not defined. WSP (Wireless Session Protocol) and WDP (Wireless Datagram Protocol) are part of Wireless Access Protocol (WAP).

**Key recovery** as a process for learning the value of a cryptographic key that was previously used to perform some cryptographic operation.

Key encapsulation is one class of key recovery techniques and is defined as a key recovery technique for storing knowledge of a cryptographic key by encrypting it with another key and ensuring that that only certain third parties called "recovery agents" can perform the decryption operation to retrieve the stored key. Key encapsulation typically allows direct retrieval of the secret key used to provide data confidentiality.

The other class of key recovery technique is Key escrow, defined as a technique for storing knowledge of a cryptographic key or parts thereof in the custody of one or more third parties called "escrow agents", so that the key can be recovered and used in specified circumstances.