Symmetric encryption schemes

2. 1 Symmetric Encryption Schemes:

With *symmetric-key encryption* , the encryption key can be calculated from the decryption key and vice versa. With most symmetric algorithms, the same key is used for both encryption and decryption, as shown in Figure 1. 1. Implementations of symmetric-key encryption can be highly efficient, so that users do not experience any significant time delay as a result of the encryption and decryption. Symmetric-key encryption also provides a degree of authentication, since information encrypted with one symmetric key cannot be decrypted with any other symmetric key. Thus, as long as the symmetric key is kept secret by the two parties using it to encrypt communications, each party can be sure that it is communicating with the other as long as the decrypted messages continue to make sense.

Encryption functions normally take a fixed-size input to a fixed-size output, so encryption of longer units of data must be done in one of two ways: either a block is encrypted at a time and the blocks are somehow joined together to make the cipher text, or a longer key is generated from a shorter one and XOR'd against the plaintext to make the cipher text. Schemes of the former type are called block ciphers, and schemes of the latter type are called stream ciphers.

2. 1. 1 Block ciphers

Block ciphers take as input the key and a block, often the same size as the key. Further, the first block is often augmented by a block called the initialization vector, which can add some randomness to the encryption.

2. 1. 1. 1 DES Algorithm:

The most widely used encryption scheme is based on Data Encryption Standard (DES). There are two inputs to the encryption function, the plain text to be encrypted and the key. The plain text must be 64 bits in length and key is of 56 bits. First, the 64 bits of plain text passes through an initial permutation that rearranges the bits. This is fallowed by 16 rounds of same function, which involves permutation & substitution functions. After 16 rounds of operation, the pre output is swapped at 32 bits position which is passed through final permutation to get 64 bit cipher text.

Initially the key is passed through a permutation function. Then for each of the 16 rounds, a sub key is generated by a combination of left circular shift and permutation.

At each round of operation, the plain text is divided to two 32 bit halves, and the fallowing operations are executed on 32 bit right halve of plain text. First it is expanded to 48 bits using a expansion table, then X-ORed with key, then processed in substitution tables to generate 32 bit output. This output is permuted using predefined table and XORed with left 32 bit plain text to form right 32 bit pre cipher text of first round. The right 32 bit plain text will form left 32 bit pre cipher text of first round.

Decryption uses the same algorithm as encryption, expect that the application of sub keys is reversed. A desirable property of any encryption algorithm is that a small change in either plain text or the key should produce a significant change in the cipher text. This effect is known as Avalanche effect which is very strong in DES algorithm. Since DES is a 56 bit

key encryption algorithm, if we proceed by brute force attack, the number of keys that are required to break the algorithm is 2^{56}. But by differential crypto analysis, it has been proved that the key can be broken in 2^{47} combinations of known plain texts. By linear crypto analysis it has been proved that, it could be broken by 2^{41} combinations of plain text.

The DES algorithm is a basic building block for providing data security. To apply DES in a variety of applications, four modes of operations have been defined. These four models are intended to cover all possible applications of encryption for which DES could be used. They involve using a initialization vector being used along with key to provided different cipher text blocks.

2. 1. 1. 1. 1 Electronic Code Book (ECB) mode: ECB mode divides the plaintext into blocks m1, m2, ..., mn, and computes the cipher text ci = Ei(mi). This mode is vulnerable to many attacks and is not recommended for use in any protocols. Chief among its defects is its vulnerability to splicing attacks, in which encrypted blocks from one message are replaced with encrypted blocks from another.

2. 1. 1. 1. 2Cipher Block Chaining (CBC)mode: CBC mode remedies some of the problems of ECB mode by using an initialization vector and chaining the input of one encryption into the next. CBC mode starts with an initialization vector iv and XORs a value with the plaintext that is the input to each encryption. So, c1 = Ek(iv XOR m1) and ci = Ek(ci-1 XOR mi). If a unique iv is used, then no splicing attacks can be performed, since each block depends on all previous blocks along with the initialization vector. The iv is a good

example of a nonce that needs to satisfy Uniqueness but not Unpredictability.

2. 1. 1. 1. 3 Cipher Feed-Back (CFB)mode: CFB mode moves the XOR of CBC mode to the output of the encryption. In other words, the cipher text c1 = p1 XOR S_j (E(IV)). This mode then suffers from failures of Non-Malleability, at least locally to every block, but changes to ciphertext do not propagate very far, since each block of ciphertext is used independently to XOR against a given block to get the plaintext.

These failures can be seen in the following example, in which a message m = m1 m2 ... mn is divided into n blocks, and encrypted with an iv under CFB mode to c1 c2 ... cn. Suppose an adversary substitutes c'2 for c2. Then, in decryption, m1 = Ek(iv) XOR c1, which is correct, but m'2 = Ek(c1) XOR c'2, which means that m'2 = m2 XOR c2 XOR c'2, since m2 = Ek(c1) XOR c2. Thus, in m2, the adversary can flip any bits of its choice. Then m'3 = Ek(c'2) XOR c3, which should lead to random looking message not under the adversary's control, since the encryption of c'2 should look random. But m4 = Ek(c3) XOR c4 and thereafter the decryption is correct.

2. 1. 1. 1. 4 Output Feed-Back (OFB) modeOFB mode modifies CFB mode to feed back the output of the encryption function to the encryption function without XOR-ing the cipher text.

2. 1. 1. 2 Triple DES:

Given the potential vulnerability of DES to brute force attack, a new mechanism is adopted which uses multiple encryptions with DES and

multiple keys. The simplest form of multiple encryptions has two encryption stages and two keys. The limitation with this mechanism is it is susceptible to meet in the middle attack. An obvious counter to meet in the middle attack and reducing the cost of increasing the key length, a triple encryption method is used, which considers only two keys with encryption with the first key, decryption with the second key and fallowed by encryption with the first key. Triple DES is a relatively popular alternative to DES and has been adopted for use in key management standards.

2. 1. 1. 3Homomorphic DES:

A variant of DES called a homophonic DES [7] is considered. The DES algorithm is strengthened by adding some random bits into the plaintext, which are placed in particular positions to maximize diffusion, and to resist differential attack. Differential attack makes use of the exclusive-or homophonic DES. In this new scheme, some random estimated bits are added to the plaintext. This increases the certain plaintext difference with respect to the cipher text.

A homophonic DES is a variant of DES that map search plaintext to one of many cipher texts (for a given key). In homophonic DES a desired difference pattern with the cipher text will be suggested with some key values including the correct one, oppositely wrong pairs of cipher text. For a difference pattern which 56-bit plaintext to a 64-bit cipher text using a 56-bit key. In this scheme, eight random bits are placed in specific positions of the 64-bit input data block to maximize diffusion.

For example, the random bits in HDESS are the bit- positions 25, 27, 29, 31, 57, 59, 61 and 63. In this algorithm, after the initial permutation and expansion permutation in the first round, these eight random bits will spread to bits 2, 6, 8, 12, 14, 18, 20, 24, 26, 30, 32, 36, 38, 42, 44, 48 of the 48-bit input block to the S-boxes and will affect the output of all the S-boxes. The 48 expanded bits must be exclusive-or'd with some key before proceeding to the S-boxes, thus two input bits into the S-boxes derived from the same random bit may have different values. This says that the random bits do not regularize the input to the S-boxes, that is, the property of confusion does not reduce while we try to maximize diffusion.

The decryption of the homophonic DES is similar to the decryption of DES. The only difference is that eight random bits must be removed to get the original plaintext (56 bits). A homophonic DES can easily be transformed into a triple-encryption version by concatenating a DES decryption and a DES encryption after the homophonic DES. Security analysis: Thus there is a probability of 1/256 between a pair of texts. The differential crypto analysis is also difficult on this mechanism. The diffusion of bits is also more in this mode. Thus this mechanism provides some probabilistic features to DES algorithm which makes it stronger from differential and linear crypto analysis.

2. 1. 1. 4 AES:

The Advanced Encryption Standard (AES) was chosen in 2001. AES is also an iterated block cipher, with 10, 12, or 14 rounds for key sizes 128, 192, and

256 bits, respectively. AES provides high performance symmetric key encryption and decryption.

2. 1. 1. 5 Dynamic substitution:

An apparently new cryptographic mechanism [34] which can be described as dynamic substitution is discussed in the fallowing topic. Although structurally similar to simple substitution, dynamic substitution has a second data input which acts to re-arrange the contents of the substitution table. The mechanism *combines* two data sources into a complex result; under appropriate conditions, a related inverse mechanism can then *extract* one of the data sources from the result. A dynamic substitution combiner can directly replace the exclusive-OR combiner used in Vernam stream ciphers. The various techniques used in Vernam ciphers can also be applied to dynamic substitution; any cryptographic advantage is thus due to the additional strength of the new combiner.

2. 1. 1. 5. 1 The Vernam Cipher: A Vernam cipher maps plaintext data with a pseudo-random sequence to generate cipher text. Since each ciphertext element from a Vernam combiner is the (mod 2) sum of two unknown values, the plaintext data is supposed to be safe. But this mode is susceptive to several cryptanalytic attacks, including known plain text and cipher text attacks. And if the confusion sequence can be penetrated and reproduced, the cipher is broken. Similarly, if the same confusion sequence is ever re-used, and the overlap identified, it becomes simple to break that section of the cipher.

2. 1. 1. 5. 2 Cryptographic Combiners: An alternate approach to the design of a secure stream cipher is to seek combining functions which can resist attack; such functions would act to hide the pseudo-random sequence from analysis.

The mechanism of this work is a new combining function which extends the weak classical concept of simple substitution into a stronger form suitable for computer cryptography.

2. 1. 1. 5. 3 Substitution Ciphers: In simple substitution ciphers each plain text character is replaced with fixed cipher text character. But this mechanism is weak from statistical analysis methods where by considering the rules of the language, the cipher can be broken. This work is concerned with the cryptographic strengthening of the fundamental substitution operation through *dynamic* changes to a substitution table. The substitution table can be represented as a function of not only input data but also a random sequence. This combination gives a cryptographic combining function; such a function may be used to combine plaintext data with a pseudo-random sequence to generate enciphered data.

2. 1. 1. 5. 4 Dynamic Substitution: A simple substitution table supported with combining function gives the idea of dynamic substitution. A substitution table is used to translate each data value into an enciphered value. But after each substitution, the table is re-ordered. At a minimum, it makes sense to exchange the just-used substitution value with some entry in the table selected at random. This generally changes the just-used substitution value to help prevent analysis, and yet retains the existence of an inverse, so that the cipher can be deciphered.

2. 1. 1. 5. 5 Black Box Analysis: Dynamic substitution may be considered to be a *black box* , with two input ports Data In and Random In, and one output port Combiner Out. In the simple version, each data path has similar width; evidently the mechanism inside the box in some way *combines* the two input streams to produce the output stream. It seems reasonable to analyze the output statistically, for various input streams.

2. 1. 1. 5. 6 Polyalphabetic Dynamic Substitution: A means to defend to known-plaintext and chosen-plaintext attacks would be to use multiple different dynamic substitution maps and to select between them using a hidden pseudo-random sequence. Thus the dynamic substitution if free from statistical attacks where each character of plain text is replaced with multiple characters of cipher text which makes the mechanism robust.

2. 1. 1. 5. 7 Internal State: Dynamic substitution contains internal data which after initialization is continuously re-ordered as a consequence of both incoming data streams; thus, the internal state is a function of initialization and all subsequent data and confusion values. The changing internal state of dynamic substitution provides necessary security to the data streams.

Thus dynamic substitution provides a probabilistic nature to the enciphering mechanism. The limitation with this scheme is, not only different dynamic substitution tables has to be maintained but also the pseudo random sequence which selects between these dynamic substitution tables has to be shared between sender and receiver.

2. 1. 1. 6 Nonces

A nonce [29] is a bit string that satisfies Uniqueness, which means that it has not occurred before in a given run of a protocol. Nonces might also satisfy Unpredictability, which effectively requires pseudo-randomness: no adversary can predict the next nonce that will be chosen by any principal. There are several common sources of nonces like counters, time slots and so on.

2. 1. 1. 6. 1 Nonce Based Encryption: In this work a different formalization for symmetric encryption is envisaged. The encryption algorithm is made to be a deterministic function, but it is supported with initialization vector (IV). Efficiency of the user is made success of this mode. The IV is a nonce like value, used at most once within a session. Since it is used at most once having any sort of crypto analysis is practically not possible which provides sufficient security.

2. 1. 1. 7 One-Time Pad Encryption

One more encryption mechanism for providing security to data is one time pad [13] encryption. The functions are computed as follows: A and B agree on a random number k that is as long as the message they later want to send.

Ek(x) = x XOR k

Dk(x) = x XOR k

Note that since k is chosen at random and not known to an adversary, the output of this scheme is indistinguishable to an adversary from a random number. But it suffers from several limitations. It is susceptible to chosen

plain text and chosen cipher text attacks. Again the limitation is here is sharing of one time keys by the participating parties of the encryption scheme. As a new key is always used for encryption, a continuous sharing of key mechanism has to be employed by the participating parties.

2. 1. 2 Stream ciphers

Unlike block ciphers, stream ciphers [14] (such as RC4) produce a pseudo-random sequence of bits that are then combined with the message to give an encryption. Since the combining operation is often XOR, naive implementations of these schemes can be vulnerable to the sort of bit-flipping attacks on Non-Malleability. Two types of stream ciphers exist: synchronous, in which state is kept by the encryption algorithm but is not correlated with the plaintext or cipher text, and self synchronizing, in which some information from the plaintext or cipher text is used to inform the operation of the cipher.

Ronald Rivest of RSA developed the RC4 algorithm, which is a shared key stream cipher algorithm requiring a secure exchange of a shared key. The algorithm is used identically for encryption and decryption as the data stream is simply XORed with the generated key sequence. The algorithm is serial as it requires successive exchanges of state entries based on the key sequence. Hence implementations can be very computationally intensive. In the algorithm the key stream is completely independent of the plaintext used. An 8 * 8 S-Box (S0 S255), where each of the entries is a permutation of the numbers 0 to 255, and the permutation is a function of the variable

length key. There are two counters i, and j, both initialized to 0 used in the algorithm.

2. 1. 2. 1. 1 Algorithm Features: 1. It uses a variable length key from 1 to 256 bytes to initialize a 256-byte state table. The state table is used for subsequent generation of pseudo-random bytes and then to generate a pseudo-random stream which is XORed with the plaintext to give the cipher text. Each element in the state table is swapped at least once.

2. The key is often limited to 40 bits, because of export restrictions but it is sometimes used as a 128 bit key. It has the capability of using keys between 1 and 2048 bits. RC4 is used in many commercial software packages such as Lotus Notes and Oracle Secure.

3. The algorithm works in two phases, key setup and ciphering. During a N-bit key setup (N being your key length), the encryption key is used to generate an encrypting variable using two arrays, state and key, and N-number of mixing operations. These mixing operations consist of swapping bytes, modulo operations, and other formulas.

2. 1. 2. 1. 2 Algorithm Strengths: The difficulty of knowing which location in the table is used to select each value in the sequence. A particular RC4 Algorithm key can be used only once and Encryption is about 10 times faster than DES. Algorithm Weakness: One in every 256 keys can be a weak key. These keys are identified by cryptanalysis that is able to find circumstances under which one of more generated bytes are strongly correlated with a few bytes of the key.

Thus some symmetric encryption algorithms have been discussed in this chapter. They varies from block ciphers like DES, Triple DES, Homomorphic DES to stream ciphers like RC4. To the symmetric encryption mechanisms concepts like application of Nounce and dynamic substitution are discussed which provides randomness to the encryption mechanism. This probabilistic nature to the encryption mechanism provides sufficient strength to the algorithms against Chosen Cipher text attacks(CCA). The security with all these mechanisms lies with proper sharing of keys among the different participating parties.

2. 1. 3 Adoptability of some mathematical functions in Cryptography:

Sign Function:[26, 27] This function when applied on when applied on a matrix of values, converts all the positive values to 1, negative values to -1 & zero with 0. The advantage of using this function in cryptography is it cannot be a reversible process ie we cannot get back to the original matrix by applying a reverse process.

Modular Arithmetic: One more function that is widely used in cryptography is modular arithmetic of a number with a base value. It will generate the remainder of a number with respect to the base value. This function is widely used in public key cryptography.

2. 2 Public-Key Encryption

The most commonly used implementations of public-key [13, 14] encryption are based on algorithms patented by RSA Data Security. Therefore, this section describes the RSA approach to public-key encryption.

Public-key encryption (also called *asymmetric encryption*) involves a pair of keys a *public key* and a *private key* , used for security & authentication of data. Each public key is published, and the corresponding private key is kept secret. Data encrypted with one key can be decrypted only with other key.

The scheme shown in Figure 1. 2 says public key is distributed and encryption being done using this key. In general, to send encrypted data, one encrypt’s the data with the receiver’s public key, and the person receiving the encrypted data decrypts it with his private key.

Compared with symmetric-key encryption, public-key encryption requires more computation and is therefore not always appropriate for large amounts of data. However, a combination of symmetric & Asymmetric schemes can be used in real time environment. This is the approach used by the SSL protocol.

As it happens, the reverse of the scheme shown in Figure 1. 2 also works: data encrypted with one’s private key can be decrypted only with his public key. This may not be an interesting way to encrypt important data, however, because it means that anyone with receiver’s public key, which is by definition published, could decipher the data. And also the important requirement with data transfer is authentication of data which is supported with Asymmetric encryption schemes, which is an important requirement for electronic commerce and other commercial applications of cryptography.

2. 2. 1 Key Length and Encryption Strength:

In general, the strength of encryption algorithm depends on difficulty in getting the key, which in turn depends on both the cipher used and the length of the key. For the RSA cipher, the strength depends on the difficulty

of factoring large numbers, which is a well-known mathematical problem. Encryption strength is often described in terms of the length of the keys used to perform the encryption, means the more the length of the key, the more the strength. Key length is measured in bits. For example, a RC4 symmetric-key cipher with key length of 128 bits supported by SSL provide significantly better cryptographic protection than 40-bit keys for use with the same cipher. It means 128-bit RC4 encryption is 3 x 10^{26} times stronger than 40-bit RC4 encryption. Different encryption algorithms require variable key lengths to achieve the same level of encryption strength.

Other ciphers, such as those used for symmetric key encryption, can use all possible values for a key of a given length, rather than a subset of those values. Thus a 128-bit key for use with a symmetric-key encryption cipher would provide stronger encryption than a 128-bit key for use with the RSA public-key encryption cipher.

This says that a symmetric encryption algorithm with a key length of 56 bits achieve a equal security to Asymmetric encryption algorithm with a key length of 512 bits,

2. 2. 2 RSA Key Generation Algorithm

1. Two large prime numbers are considered. Let them be p, q.
2. Calculate n = pq and (φ) phi = (p-1)(q-1).
3. Select e, such that 1 < e < phi and gcd(e, phi) = 1.
4. Calculate d, such that

   ed ≡ 1 (mod phi).
5. One key is (n, e) and the other key is (n, d). The values of p, q, and phi should also be kept secret.

- n is known as the modulus.
- e is known as the public key.
- d is known as the secret key.

Encryption

Sender A does the following:-

1. Get the recipient B's public key (n, e).
2. Identify the plaintext message as a positive integer m.
3. Calculate the ciphertext c = m ^e mod n.
4. Transmits the ciphertext c to receiver B.

Decryption

Recipient B does the following:-

1. Consider his own private key (n, d) to compute the plain text m = c ^d mod n.
2. Convert the integer to plain text form.

2. 2. 3 Digital signing

Sender A does the following:-

This concept can also be used in digital signing as well. The message to be transmitted is converted to some message digest form. This message digest is converted to encryption form using his private key. This encrypted message digest is transmitted to receiver.

Signature verification

Recipient B does the following:-

1. Using the sender's public key, the received message digest is decrypted. From the received message, the receiver independently computes the message digest of the information that has been signed.
2. If both message digests are identical, the signature is valid.

Compared with symmetric-key encryption, public-key encryption provides authentication & security to the data transmitted but requires more computation and is therefore not always appropriate for large amounts of data.

2. 3. Probabilistic encryption schemes

In public key encryption there is always a possibility of some information being leaked out. Because a crypto analyst can always encrypt random messages with a public key, he can get some information. Not a whole of information is to be gained here, but there are potential problems with allowing a crypto analyst to encrypt random messages with public key. Some information is leaked out every time to the crypto analyst, he encrypts a message.

With probabilistic encryption algorithms [6, 11], a crypto analyst can no longer encrypt random plain texts looking for correct cipher text. Since multiple cipher texts will be developed for one plain text, even if he decrypts the message to plain text, he does not know how far he had guessed the message correctly. To illustrate, assume a crypto analyst has a certain cipher text ci. Even if he guesses message correctly, when he encrypts message the result will be completely different cj. He cannot compare ci and cj and so cannot know that he has guessed the message correctly. Under this scheme, different cipher texts will be formed for one plain text. Also the

cipher text will always be larger than plain text. This develops the concept of multiple cipher texts for one plain text. This concept makes crypto analysis difficult to apply on plain text and cipher text pairs.

An encryption scheme consists of three algorithms: The encryption algorithm transforms plaintexts into cipher texts while the decryption algorithm converts cipher texts back into plaintexts. A third algorithm, called the key generator, creates pairs of keys: an encryption key, input to the encryption algorithm, and a related decryption key needed to decrypt. The encryption key relates encryptions to the decryption key. The key generator is considered to be a probabilistic algorithm, which prevents an adversary from simply running the key generator to get the decryption key for an intercepted message. The following concept is crucial to probabilistic cryptography:

2. 3. 1Definition [Probabilistic Algorithm]:

A probabilistic algorithm [11] is an algorithm with an additional command RANDOM that returns “ 0” or “ 1”, each with probability 1/2. In the literature, these random choices are often referred to as coin flips.

2. 3. 1. 1 Chosen Cipher Text Attack:

In the simplest attack model, known as Chosen Plaintext Attack (CPA) [5], the adversary has access to a machine that will perform arbitrary encryptions but will not reveal the shared key. This machine corresponds intuitively to being able to see many encryptions of many messages before trying to decrypt a new message. In this case, Semantic Security requires that it be computationally hard for any adversary to distinguish an encryption Ek(m) from Ek(m') for two arbitrarily chosen messages m and m'. Distinguishing these encryptions should be hard even if the adversary can request encryptions of arbitrary messages. Note that this property cannot be satisfied if the encryption function is deterministic! In this case, the

adversary can simply request an encryption of m and an encryption of m' and compare them. This is a point that one should all remember when implementing systems: encrypting under a deterministic function with no randomness in the input does not provide Semantic Security. One more crypto analytical model is Chosen Cipher text Attack (CCA) Model. Under the CCA model, an adversary has access to an encryption and a decryption machine and must perform the same task of distinguishing encryptions of two messages of its choice. First, the adversary is allowed to interact with the encryption and decryption services and choose the pair of messages. After it has chosen the messages, however, it only has access to an encryption machine. An advancement to CCA Model is Chosen Cipher text Attack 2 (CCA2). CCA2 security has the same model as CCA security, except that the adversary retains access to the decryption machine after choosing the two messages. To keep this property from being trivially violated, we require that the adversary not be able to decrypt the cipher text it is given to analyze.

To make these concepts of CCA & CCA2 adoptable in real time environment, recently Canetti, Krawczyk and Nielsen defined the notion of replayable adaptive chosen ciphertext attack [5] secure encryption. Essentially a cryptosystem that is RCCA secure has full CCA2 security except for the little detail that it may be possible to modify a ciphertext into another ciphertext containing the *s*