Joseph Eckl

CIS 483

Dr. Im

4/18/19

Encryption Algorithms in SQL Server

This paper will provide an overview of the encryption algorithms that are provided by SQL server; this includes their technical operation, security considerations, and other pertinent information regarding their use in SQL server applications. There will not be any discussion of how to use these algorithms in practice for database purposes, only recommendations of which algorithms should be used based on their encryption strength.

Microsoft provides several encryption algorithms to choose from when using encryption on a SQL server database. There are four main algorithms and several variations. The four main encryption methods offered by SQL server are: DES, RC2, RC4, and AES. In the forthcoming sections of this paper, these four algorithms will be focused on.

The point of an effective encryption algorithm is to be strong. This is accomplished by making any attempt to work backwards through the algorithm by guessing every possible key as long as possible. The primary factor that makes an encryption algorithm strong is its key size. A longer key size means more possible keys. The processing methods in the algorithm also contribute to increasing the time of a possible attack. Using different techniques to mix up the processed bytes is more secure than repeating a similar technique throughout the algorithm.

Important Encryption Terms

There are several terms that must be discussed in preface of the algorithms themselves. At the most basic level, there are three simple components that make up the algorithm. The first is the input value. This is the plaintext message that is to be encrypted, along with a string of a certain length that will be used as the encryption key. The last is the resulting ciphertext. This is the message after undergoing the encryption process. The middle portion is the most important; this is where the data is transformed. In this middle section of the process, there are many types of operations that can be performed on the plaintext value using the provided key.

The first step of the transformation is process is key expansion. The initial key must be expanded for the sake of security. Using only the initial key throughout the process would be extremely insecure and would render the algorithm pointless. Therefore, various subkeys are created. Furthermore, these algorithms are symmetrical. The same key can be used to perform encryption and decryption, which is most feasible for SQL server database use cases. In a typical use case, one user will modify the data, encrypt it, then save it. The same user will likely come back to modify the data, making a symmetric process more efficient.

Using the subkeys, the data can now be processed. In this stage, a loop will be used to effectively transform the data. Several iterations of a combination of the following elements will be performed and loops are the only way to achieve this task. Loops are invaluable to encryption, and without them, encryption would be practically unfeasible. Alongside and within the loop are several common elements that perform the transformation. Permutation patterns and substitution tables (S-boxes) are used to generate new variations of the data. See the appendix for any tables mentioned in the processes. These are uniquely defined in each algorithm. Shifting (at the bit or byte level) is a simpler operation of moving a piece of data over that achieves a similar effect. There are mathematical operations, namely byte addition and modulo division (the remainder of a traditional division operation), that are commonly used. An encryption algorithm may also employ the use of calculated results of predefined equations. Additionally, some logical functions are used. The AND function and XOR (exclusive OR) function are used; in particular, the XOR function is used often.

DES

Data Encryption Standard is the first algorithm offered by SQL server. It is the oldest of the main four methods and was in widespread use as a standard for the longest. It was developed in 1975 as the first standard for encryption and was used officially until 2002. DES was replaced as an encryption standard because it was deemed insecure for practical use.

DES is a 64-bit block cipher. It uses a key of 64-bit size but is effectively limited to 56 bits due to hardware limitations at the time it was created. Since its creation, it has been broken in practice (in 1999), mainly due to the small key size.

DES accepts a plaintext message and a 64-bit key as input. The key combination step will occur in 16 groups of 4 bits totaling 64 bits, so 16 subkeys will be necessary. The key K will first be permutated by table PC-1. This table only has 56 places, thus limiting the key size. The resulting K+ is split in halves as C0 and D0. Each half will be left shifted according to a table, and each iteration is stored after concatenation into C0D0. The concatenated pair is then permutated with table PC-2 and stored as subkey K1-16.

With the subkeys now generated, the message can begin to be encrypted. The 64-bit message block M is permutated according to table IP. IP is then split in halves L0 andR0. The looping section of the algorithm will now operate on L, R and K. For iteration number N 1-16 the values are defined as follows: LN = RN-1; RN = LN-1 + func(RN-1,KN) with + denoting bit addition. The function must be further defined into steps. RN-1 is expanded to 48 bits using table E. The resulting E(RN-1) is combined with KN by XOR. KN XOR E(RN-1) is split into groups of 6 bits and run through an S-Box. Each of the bytes has a separate S-Box and uses the outer two bits to determine the S-Box address. The result (now 32 bits) is permutated with table P and returned to the loop.

Each new combination of LNRN is swapped to RNLN. This combination is permutated a final time with table P, written in hexadecimal, and returned as ciphertext. The encryption process has been completed. Decryption is performed by reversing the process.

As a note, there are other variants of DES present in SQL server. 3DES is more secure and has been approved for use through 2030. 3DES is done by completing the DES process three times. 3DES-3KEY is 3DES done using three unique keys.

RC2

Rivest Cipher 2 was designed in 1989 by Ron Rivest. It is a 64-bit block cipher and uses a default key size of 64 bits. As DES was still in use, RC2 and RC4 were not used as standards.

Key expansion in RC2 is done through use of a table based on pi (PITABLE), the length in bytes T8 (8 default), length in bits T1 (64 default), and a mask TM declared as: 255 MOD 2^(8 + T1 - 8\*T8). These values held in a buffer L after being run through the PITABLE according to the following loop:

for i = T, T+1, ..., 127: L[i] = PITABLE[L[i-1] + L[i-T]],

an intermediate step:

L[128-T8] = PITABLE[L[128-T8] & TM],

and a final loop:

L[i] = PITABLE[L[i+1] XOR L[i+T8]].

L is combined into a series of 16 bit words K and the message block is split into 4 16 bit words R[0-3]. The encryption process begins.

The first process is called Mix. It is defined by this loop:

R[i] = R[i] + K[j] + (R[i-1] & R[i-2]) + ((~R[i-1]) & R[i-3]);   
j = j + 1;   
R[i] = R[i] rol s[i];

This loop takes a word and adds the next word in K, the AND of the previous two words in R, the AND of the third previous word and the inverse of the previous word. J is a global variable that keeps track of the current word in K. R[i] is then rotated left according to this series:

s[0] = 1, s[1] = 2, s[2] = 3, and s[3] = 5

The other process is Mash, defined by this statement:

R[i] = R[i] + K[R[i-1] & 63]

This statement is added to a word in the expanded key. The encryption then loops through a pattern. Rounds of 5 mix, 1 mash, 6 mix, 1 mash, 5 mix result in the final cipher text.

RC4

RC4 is another algorithm made by Ron Rivest. This is stream cipher operating byte by byte. It is comprised of only two loops, but uses up to a 256 bit key.

This loop creates a permutation of the values 0-255.

For I from 0 to 255: S[i] := i

j = 0

for I from 0 to 255 j := (j + S[i] + key[i mod 256]) mod 256

swap(S[i],S[j])

The second loop is used to create the output.

i := 0, j := 0

while GeneratingOutput:

i := (i + 1) mod 256, j := (j + S[i]) mod 256

swap(S[i],S[j])

output S[(S[i] + S[j]) mod 256]

The result of this loop is XORed with a byte of the plaintext to produce the ciphertext.

AES

Advanced Encryption Standard was created in 2001 to replace DES the following year. It is a 128-bit block cipher using a 128, 192, or 256-bit key. It is currently thought to be unbreakable for the next 50 years at least. AES has been adopted as the new standard for encryption and is the only one of these algorithms Microsoft recommends using.

First, the key must be expanded into 11 round keys. AES uses rounds of encryption with separate keys plus one more key. For key expansion, several values must be declared. N = number of 32 bit/4-byte words in the key = 4. K (0-3) = the 4 words of the provided key. R = number of keys needed = 11. W(#) = an individual word, # is values from 0 to 4\*R-1, 44 words. RotWord – a word shifted left one byte. SubWord – a word with each byte having been run through an S-box. Rcon = a polynomial function given by a finite field equation.

Now a loop will create each word for the 11 keys. 3 logic statements are used.

If i < N (4): WN = KN. The first four words are the four words of the key.

If I >= 4 AND i MOD N = 0:

W(N) = Wi-N XOR SUBWORD(ROTWORD(Wi-1)) XOR rcon1/N

Every fourth word is a combination of the 4th previous word, the substituted rotated previous word, and the constant that changes every 4 words.

Else WN = Wi-N XOR Wi-1

Otherwise, the word is a combination of the 4th previous word and the previous word.

The words are then combined in groups of 4 to create the 11 keys.

The encryption process consists of 10 rounds of four steps. The data block is broken into a 2D array of 16 bytes. The rounds are performed on this array.

SubBytes: Bytes are substituted with an S-Box. This is the same S-Box used for key expansion.

ShiftRows: Each byte in rows 1,2,3 are shifted left by 1,2,3 spaces respectively

MixColumns: Each column is multiplied with the fixed polynomial from the key expansion.

AddRoundKey: one of the round keys is added.

In the last round, the MixColumns step is omitted. This results in the final ciphertext.

These are the encryption algorithms offered in SQL server. Then can be applied when choosing an encryption method for a database object.

Sources

J. Orlin Grabbe. *The DES Algorithm Illustrated*. Technische Universität Berlin. 1998.

Ron Rivest. *A Description of the RC2(r) Encryption Algorithm.* IPA Japan. 1998.

Ralph Rise, Suk-Hyun Cho, Devin Kaylor. *RC4 Encryption.* Washington University, 2008.

NIST. *Federal Information Processing Standards Publication 197: Announcing the ADVANCED ENCRYPTION STANDARD (AES)*. NIST. 2001.

Appendix









