AES Origins

As I said clear replacement of DES was needed and can also use triple DES. I will explain triple DES later on. With triple DES you are going to do triple encryption for one plaintext block. Instead you are increasing encryption time 3 times. But it is slow, but if you are using small blocks it will be ok, if not it will be really slow. Anyways that is a standard and we are going to see that standard later on, US NIST issued call for ciphers in 1997, 15 candidates accepted in 1998 and 5 were shortlisted in aug 1999,that means this is nearly jus 10 years old. Rijndael was selected in AES in oct 2000, jus a few years ago. Issued as FIPS PUB197 standard in Nov 2001, so that is the standard.

AES requirements

What were the requirements of AES, the requirement of new algorithm was this, it should be private key symmetric block cipher and data size or data block, that is 128 bit data, 128/192/256 bit key size, block size is fixed but the key size may come with three options ,it should be stronger and faster than triple DES, active life of 20 – 30 years, plus archival use. it should provide full specification details, nothing should be secret , it should implement both c and java implementations, NIST have released all submission and unclassified analysis, in other words everything is known.

AES evaluation criteria

AES evaluation criteria was this, initial criteria, security, cost and implementation and algorithm characterisitics. In final criteria, general security , software and hardware implementation ease, implementation is very important for both initial and final criteria, using the same hardware and software implementations ,that's very important and you will see in AES that it Is very important is it is very easy and fast. You need only 4 tables to implement the algorithm ,we are going to see that. Implementation attacks you have to look at it, if you have 4 look up tables then there will be no implementation attacks. And in flexibility, encryption, decryption keying and other factors must be easy.

AES shortlist

It was by MARS by IBM, RC6 by USA, RIJNDAEL from Belgium, Serpent from Europe, Twofish from USA. So these where the 5 algorithms that is needed for AES algorithm. If you look at Rijndael then you can see what he does, clean, fast and good security margin. If you look at MARS it says says high security margin, so probably data will choose mars, so if you are in some committee or something which algorithm do you choose, most probably you should know which algorithm you have to choose, most probably you will know. But actually you will not see high security margin and all ,as it is commercial algorithm. Good margin ,good implementation and good security is the best all the time. If you increase the security margin then you will increase the complexity of the algorithm, if the algorithm is too complex, that means during implementation you are going to have problems for sure. That's why these guys selected rijndael algorithm, even they could have gone with two fish which is also good, but It has high security, and complexity will be the problem, actually these blowfish guys selected two fish, we are going to see blowfish ,but we are not going to see twofish. Serpent from Europe were slow ,so forget it, then it came RC6 , RC series we are going to see, RC is the series of, RC2,RC3,RC4,RC5,RC6, are all series

of algorithm ,when you see these RC series you are also going to see that these algorithms are so fast, because these are developed basically for microprocessor implementation . it is only shifts an exclusive OR's and we are going to see RC4 and RC5 anyways. This rijn dael got selected for good security margin, clean and fast. They selected from lot of algorithms and rijndael became the best. And they selected it.

AES cipher – Rijndael

It was designed by these guys Rindael- Daemen in Belgium, so its European algorithm it has 128/192/256 bit keys with 128 bit data.it is iterative rather than a feistel cipher, in other words it is not a feistel cipher. DES was a feistel cipher but these was not feistel cipher this algorithm treats 4 groups of 4 bytes, that means if I have 32 bit, then I am going to have 32 words. it operated in entire block in every round, in feistel we were operating, on half of the block . we didn't process the whole block in each round ,we processed either right or left block. So it is iterative rather than feistel, so we are going to operate on entire block of data. It is designed to resistant against known attacks ,speed and compactness on many CPUs that is very important ,we are going to see that, design simplicity. Key size is 128 bits, and 16 bytes and if the word size is 32 bits then you have 4 words. key size is 128 bit for all of them ,but look at the, number of round it It is 10 for 64 bits 12 for 128 bits and 14 for 256 bits. So numberof rounds are going to increase if you increase the key size we are going to see this option, 128 bit key, 128 bit ,plaintext block and 10 rounds, round key size are 128 bit for everything, and expanded key size are 176 bits expanded key size is this look at the words, 44 words. I have 10 rounds and in each round I am going to use 4 words subkey. Means 128 bit keys, so how many keys I need. You need 40 words, that is the key I am going to use and iam going touse the first, 4 words before the first round so I need 44 rounds.so in round Iam going to use 4 rounds as subkey. That is the parameter that we are going touse in AES algorithm, so rijndael

RIJNDAEL

It is going to process data ,in 4 groups of 4 bytes. And I am going to call that states. Each bit 32 bits. We are going to use 2 dimensional data. 10/12/14 bytes it is going to this side,byte substitution. We will have I big substitution cipher and in AES we will know how to construct that substitution table.it is not like DES, it is like this so don't use this table. But over here you know why the number in that s box is that number. Because you know how to construct s box , so first of all we know it is substitution and now we are going to go to shift rows. Shift rows are going to change the rows, imagine we are going to have data in matrix form. That is shift row and then we are going to mix columns that is most important par t in AES algorithm, that is multiplication part and this multiplication is GF^n multiplication. So it is totally different stuff, so when I come there I am going to explain how to do those calculations. And then comes the add round keys, XOR the subkeys to the result so if you look at these. If you look at the 10 rounds only the last round will not have the initial XOR will have incomplete last round. In the 9 rounds I am going to do these guys, mix columns, byte substitution and shift rows. So you guys tell me what was the rule for modern ciphers. Exactly we are going to use substitution and transposition. What is the multiplication part ., it is shift rows, and mix column is the mixture of both multiplication and substitution.when you multiply something then the result is the substitute for the data basically . add round key is also substitution. So modern cipher consists of both substitution and permutation.

Rijndeal

So lets have a look at the algorithm, plaintext 128 bit will come to add round key ,so no where else it will come to add round key and then the first key whose size is 128 bit, the compete key will be XORed with the plaintext. Word size is 32 bits and how many words I am using, 0 to 3 total 4 words are going to be added to the plaintext and this is you actual key ,then I am bringing to substitution bytes then mixed rows and then shift rows and then to add round key. I used the key already , so what I am goingto do is by using my , original key I am going to have my own algorithm and it is called expand key. I will explain to you. By using that algorithm you will be able to generate rest of the required keys. By word by word. How many 32 bit I need 32, because in each I am going to use 4 words. but the subkey size is fixed ,128 bits. If you look at the last round you don't see mixed columns, so I call it incomplete keys. So this is the encryption part, what about the decryption part??..this is not feistel cipher so our rules for decryption may not be the same. So plaintext is going to add round key and in the reverse order and instead of shift rows It is reverse shift rows. We have inverse substitute bytes and it will go to add round key and for mix columns we have inverse mix columns. so everything that we did in encryption we are going to do in the decryption. It is the inverse shift rows here. If the shift rows was one byte shift in the encryption, 1 bit right shift in decryption and s box that we constructed in encryption, will be different s box in decryption and it will be inverse s box. So it will be inverse s box. Mix columns if you multiplying your data with some matrix ,then you are going to multiply with, a inverse matrix that you are going to use here. If you compare this with , here then decryption side of the algorithm is different than the encryption side. In DES it was same , only thing was it was reverse order of the subkeys. So here every element within the round will be different. It requires different implementation for decipherment.