SHA1 Overview

As I said lets have a look at inside the boxes, so that you will understand ,SHA 1 architecture will look like this, as I said we are going to have registers a,b,c,d,e . Imagine y0 is the first, if y0 is the first block ,this svq is going to be iv values given by the standards those numbers, you are going to enter those numbers, in the first round we are going to have 20 steps and I am going to have primitive functions, definitions are simple something like and or, and like a xor with b and xored with c and and with b kind of things ,it is given by the standards, if you are interested then the functions will be in this form, we are going to have some constants and we are going to call them k, and now we have extra words we have to create you have to do something right here called compression function and this time 2 is different and the words that you are using is different ,then its going to go to third round, fnction and words are different all have 20 steps and in 4th round again 20 steps and this iv value is going to be added to the output of 4th round to form new iv value for next round, and this addition right here is addition in mod 2 to power 32, modular bit addition, I am going to take this one and ia m going to take new iv,I m going to get the second block and I do the same thing and I am going to get the new iv and that's going to b third block its going to go like this until last block and if that is last block that iv is the fingerprint of the message or hash of the message or digest of the message ,as you see pretty complex, and you may get question like, what is inside this round as I said that s compression function and iam not going to go into details , we are going to see in SSL basically, when they calculate hash of the message they are going to calculate with md5 and SHA1 together .

MD5

They are using md5 for that compatibility some of the algorithm is still using md5, to make the whole protocol compatible, especially in SSL ,to see the secret they are using md5 and SHA1,if produces a 128 bit hash value which is very low now, until recently most widely used hash algorithm ,this is for SHA1 as SHA 1 is most widely used until recently used algorithm, they are going to change SHA1 to SHA512 in 2010, and in the cryptography class we are going to remove SHA1 and include SHA512 to modify the course basically, ,it is defined in RFC 1321, you can find all the algorithm, if you look at the md5 structure, md5 is also using 512 bit block sizes, instead of 5 registers it is going to use 4 registers ,and it is using 64 bit length value, it is using 4 rounds of operations ,instead of 20 steps md5 will have 16 steps and output is going to be added to the buffer and it will become input to the next abcd value ,so that the structure is same so most probably they are using these constants in other words compression part is different than SHA1,

RIPHEM 160

Other important part is called RIPHEM 160, its developed in Europe in 1996 and used 2 parallel lines of 5 rounds of 16 steps, so imagine md5 ,but imagine that you are running parallel to md5 and 5 rounds on SHA1 is 4 rounds 16 steps like md5 ,but SHA 1 has 20 steps, this guy looks more secure than MD 5 and SHA1, because it has 2 parallel lines as I said, if you have two parallel you are running 2 parallel algorithm and RIPHEM 160 could cause some performance problem, as I said it is slower more secure than SHA1, in some of the algorithm and if you look at the structure, block size is still 512 bits it has 5

registers abcde like SHA1 these are initial values for abcde and it is different than SHA1 value and produce 16 word 512 bit chunks 10 rounds, 5 in each side, they are going turn parallel and add output to the input to form new abcde value , if that is the last block then that will be the hash value ,this is the European version of hash ,if you compare this guy digest length is 128 in md5 and 160 in SHA1 and RIPHEM 160 ,block size is same 512 bits, number of steps it is 4 rounds of 16 in md5 and SHA1 4 rounds of 20 and RIPHEM 160 5 paired rounds of 16 , maximum message size is infinity but if you look at last block all of them are using 64 bit ,message is limited to 2 to power 64 that is huge, f functions is 4 in SHA1 and md5 and 5 In RIPHEM 160 additive constant is k values in 64 In md5 ,4 in SHA1 and 9 in RIPHEM ,but what is important for the security sake is SHA1 algorithm ,we are going to see SHA1 in major security algorithm and some of the major ones will see also md5 ,that is the comparison of hash function .

HMAC

This is going to be the standard that we are going to see in the security protocol and it is called HMAC, hash MAC, we are using hash algorithm with security, it is a internet standard RFC2104 ,and uses hash function on the message, I have a key this key will xoerd with a number that I cal it I pad and it is going to calculate the hash of the message and result is going to append to, my same key xored with another constant opad and all of them ,output of the hash will be input tot hash and output will be HMAC, padding elements are 5c and hexadecimal ,and 36 hexadecimal ,what I am saying is this I have a secret im xoring this key with ipad and iam creating a block and this block size may change depending on hash algorithm that you are using , HMAC doesn't specify which algorithm you are going to use, in other words, any hash algorithms is going to work ,so what is the block size of the hash algorithm that we saw in md5 and SHA512, when you xor it you have to create 12 bit block size ,lets say that you selected64 bit secret key and Ipad is 36 lets say and 36 is a 8 bit and 64 bit key and what are you going to do??,you are going to repeat and that pad is the repetition until you got 512 bit length , you are repeat the key and ipad and the result of ipad is 512 block and 1st block is the function of ipad ,remember the usage of the hash algorithm ,we put the s to the message and I m putting to the message and the first block is the function of the secret key that I am using for the whole calculation and the rest of the block is right here but when we go to SHA512 if im not mistaken the block size is 1024 bits, so then you are going to do same stuff and we have to repeat that key and same thing to get the that size of block , so now I have the whole message ,lets say that im using SHA1 what are the iv values whatever the value given by the standard and what is the size of iv value ,160 ,which means each buffer is 32 bits, so im going to calculate the hash of the message including the blocks and I m getting m bits of hash and the length of this block is 160 bit,160 is going to come here and im going to xor with 5 c and im going to create b bits ,this time block size defined by the output of the hash algorithm and what Is the block size over here is 160 so that s0 must be 160 ,I m having 160 bit over here ,no ..these two blocks will go hash algorithm if its SHA1 block size is 512 bits and I have 160 bit and with that I cannot create this 512 bit block size and I ma going to do the padding and this block size is 512 and this must be also 512 and im doing that 512 and I have now xor and padded out of hash and It is going to to go to another hash and this output right now from hash is going to be the HMAC output ,so this is the thing this is the standard we are going to use in HMAC, so HMAC is not a hash algorithm it is a kind of usable algorithm user mode of algorithm it

doesn't define which algorithm you are going to use, you may use SHA1,SHA512 ,md5 , some people asks this question is it possible to use md5 here or SHA1 here, I dint see indicating otherwise but at the same time I dint see the application that they are using the different hash function that's my personal belief it is upto the user whoever is the developing what type of protocol they may use 2 different algorithm or only one algorithm but most of the application are using only 1, so again in other words it is upto you , this is a secret key and it is not a session key and not a key for encryption and that keys are called HMAC keys or hash keys, lets say this is public ,why we are using this, the reason we are trying to provide authentication and with hash function we can provide authentication and we can Share something secret , if you make this keys public there will be no authentication and if you make public we are going to see the authentication and we are using it has secret MAC keys and we are going to generate those keys when we look at SSL , server right MAC keys and when we say right MAC key we are saying these key and we are not going to use MAC key for confidentiality ,for that we are going to use session keys , session keys, session keys are for encryption and MAC keys are for ,this made calculation there are two separate secret keys, there are lot of parameters to be distributed, the most important thing in symmetric key algorithm is the distribution of the secret keys along with tat if you are using CBC or CFB mode then will be secrecy in distributing , but if you are using HMAC right now another parameter , this one MAC key should be distributed ,session key iv and MAC keys and you may argue this way I may use this from point b to point a and from point a to point b I want use different key although b wants to use separate keys with authentication and integrity , he may say that I am going to use different keys and this way 3 more will be distributed and I may say client side keys, means anything can be coming form client to server and this iv will be used and this MAC key will be used and server to client they are going to use different MAC key , hash key and that's we are calling them client session key and server session key and client MAC key and server MAC key. That's why distribution is important.