Chapter 4

4.1 Key Exchange protocols

In TLS data is encypted with symmetric block cipher AES

Before encryption with AES starts, the symmetric key must be agreed between the communicating parties. For this there exists three commonly used algorithms, which are called key exchange protocols

RSA Exchange is generally used in TLS connections

ECDHE ***(Elliptic Curve Diffie Hellman Exchange)*** is newer and has partly replaced RSA Exchange

DH Exchange ***(Diffie Hellman Exchange)*** is also widely used option

4.1.1 RSA Exchange

One of the communicating parties creates a random symmetric key and sends a copy to the other party encrypted with RSA

Possible security problems of RSA exchange

1. If the hacker manages to **factor recipients public RSA key,** which is permanent he may break the key excange messages containing AES also in future connections.

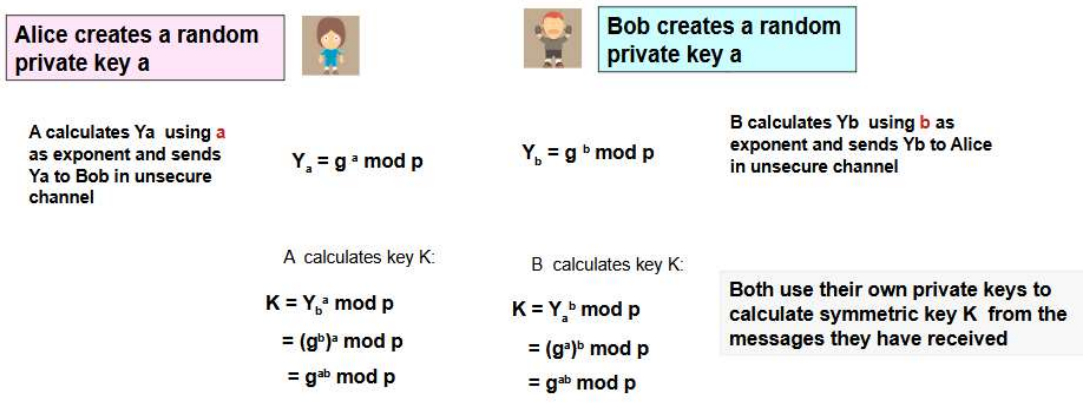
2. **Authorities** could require service providers to give private RSA keys to officials.

4.1.2 Diffie Hellman Exchange (DHE)

System parameters: Large (2048 bit) prime **p** and a generator **g** *)

) Number set Z_p with multiplication mod p forms a **cyclic group**. This means that there are elements g called "**group generators**", whose powers cover the entire set: $\{g,g^2,\ldots,g^{p-1}\} = \{1,2,\ldots, p-1\}$.

Users Alice and Bob agree on symmetric key in the way shown in the diagram.



To avoid Man in The Middle attack (A third party pretends Alice to be Bob), all user can have own system parameters. The public key of Bob is the triple (g_B,p_B, Y_B), and Bob's private key B. The public key (g_B,p_B, Y_B) is found in Bob's certificate.

The basis of security of DHE is called Discrete Logarithm Problem (DLP)

A hacker can get the public messages Ya = g^a mod p ja Yb = g^b mod p by listening the unsecure channel, but he cannot solve private keys **a** and **b** from these number.

Solving exponent x from y = g^x mod p is as hard as factoring large integers in RSA

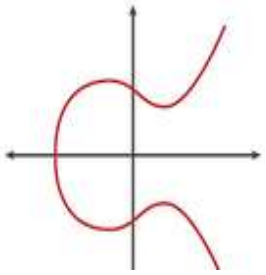
DLP = " Discrete Logarithm Problem" is the "hard problem", which is behind the security of DHE.

DLP: Solve integer x from $y = g^x$ mod p, where integers y, g and prime p are given

Usual recommendation for secure size of p is 2048 bits

4.1.3 ECDHE Elliptic Curve Diffie Hellman Exchange

A more recent key exhange is ECDHE , which has replaced RSA exchange in many web services (for example in some Finnish banks)



Elliptic curves used in ECC are curves

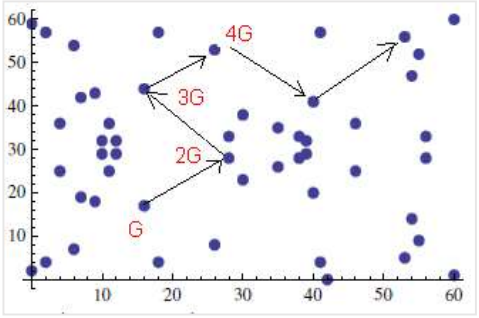
$y^2 = x^3 + a\,x + b$

If P and Q are points of the curve, it is possible to define an addition operation which gives a third point of the curve called the sum P + Q

Elliptic curve cryptography ECC uses discrete elliptic curves. Discrete curve consists of integer pairs (x,y) , which satisfy equation $y^2 = x^3 + a\,x + b \pmod q$, where a and b are integers and modulus q is prime.

Example of a discrete elliptic curve

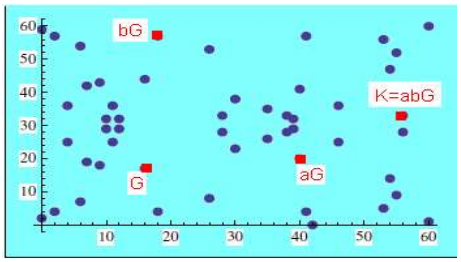
$y^2 = x^3 + 2x + 4 \pmod{61}$, where $0 < x,y < 60$



There are 61 integer pairs (x,y) satisfying the equation. It is possible to define addition of two point P + Q in a way that the curve forms a cyclic group. A generator of this group is point G = (2,4). All points are form k*G, where k is some integer between 1...61.

How ECDHE works ?

The curve $y^2=x^3 + ax+ b \pmod q$ and generator point G are given.



1. Alice and Bob choose their random private keys integers a and b
2. Both send each other as their public keys points aG and bG
3. Both calculate the shared symmetric key K, which is also a point of the curve:
Alice calculates: K = a(bG) and Bob calculates: K = b(aG)

In real ECDHE used in TLS, modulus q, and the private and public keys are about 250 bit integers.
The curve has also about 250 points.
The symmetric AES - key can be calculated from point K for example with hash function.

Security of ECDHE is based on a hard problem called ECDLP.
A hacker can listen unsecure channel and read the public keys bG and aG, but he is not able to calculate private keys a or b in a reasonable time.

ECDLP = " Elliptic Curve Discrete Logarithm Problem"

ECDLP: Solve integer n from P = n G, where G and P are points of an elliptic curve

Secure curve has about 250 points.

4.2 Cryptographic Hash functions

Definition: Hash functions are one-way functions, which produce a fixed length digest from the message

Uses of hash functions:

1.Hash value **ensures the integrity of data transmission** (that data is not changed during transmission).

2. **Password files** of servers do not store passwords, but their **hash values**

3. **Digital signature** is calculate using hash value of message as input .

4. **Checksums** are shorter hashes, which f.e antivirus software calculates for folders and files.

Hash values are also called "message digests"

Requirements for a secure hash function

Notation: Hash function is h(m), where argument m is message

h(m) is a **one-way function**. Is must not be possible to calculate the message m from its hash value h(m).

2. **Collision resistance 1**. If the hash h(m1) for one message m1 is known, it should be impossible to generate another message m2 with same hash value.
(Other messages surely exists, but there should be no method for finding them)

3. **Collision resistance 2**. It should be impossible to generate two message m1 and m2 with same hash value.

Some known hash functions

Chinese scientist in 2005 showed the vulnerability of many common hash functions by showing how to generate another message with a given hash value.

Older hash functions, which are compromized	
MD5	**- broken**
SHA-1 (160 bit)	- **not recommended**
Following hash functions are regarded safe	
SHA256	
SHA384	
SHA512	

Wolfram Alpha command Hash["Tämä on koeviesti, josta lasketaan tiiviste","SHA256"] gives

32134986657396586487192643852384396035054468584117070131813446773863278924851

Minimum length of secure hash. Birthday paradox

A secure hash should be twice as long as the key of block cipher

Thus the security of SHA258 hash is equivalent to security of AES128.

The reason is related to "**birthday paradox"** , which is formulated as a following question:
"What is the mimimum size of a school class, so that the probability that at least two have the same birthday would be more than 50 percent?". Answer is 23. Calculation below proves the result.

$$P = 1 - \prod_{k=1}^{23} \frac{365 - k + 1}{365} = 0.51$$

Similar calculation can be done for probabilty of collisions of hash values:

If the size of "hash space" is n, there will be collisions at probability of 50% among approximately $\sqrt{\mathbf{n}}$ hash values => The number of 256 bit hashes is 2^{256} . The square root is 2^{128} which is the number of hashes where appears collision at >50% probability.

4.3 MAC functions

MAC = message authentication code

MAC means hash function with symmetric key K as extra input in addition to message m

MAC ensures not only the integrity of message, but also authenticates the sender of message

MAC is used in the same way as digital signatures at the end of transmitted data packets to quarantee integrity and sender's authenticity

sha-HMAC

= most common MAC which use RSA to calculate value from message m ja and symmetric key K

$$HMAC(K, m) = sha(K \oplus opad || sha(K \oplus ipad || m)$$

⊕ = XOR sum

K = symmetric key

Ipad and Opad are 64 bit constants, in hexadecimal form

Ipad = 5c5c5c5c , Opad = 36363636

|| means concatenation of strings ("auto"||"mat" = "automat")

sha-HMAC – is used as pseudorandom number generator which produce one-time passwords. (More about this in the last chapter)

4.4 Digital Signature

A digital signature enables secure online transactions.
Digitally signed document is juridically equal to documents which are manually signed at presence of wittnesses

The goal of digital signature is to ensure that

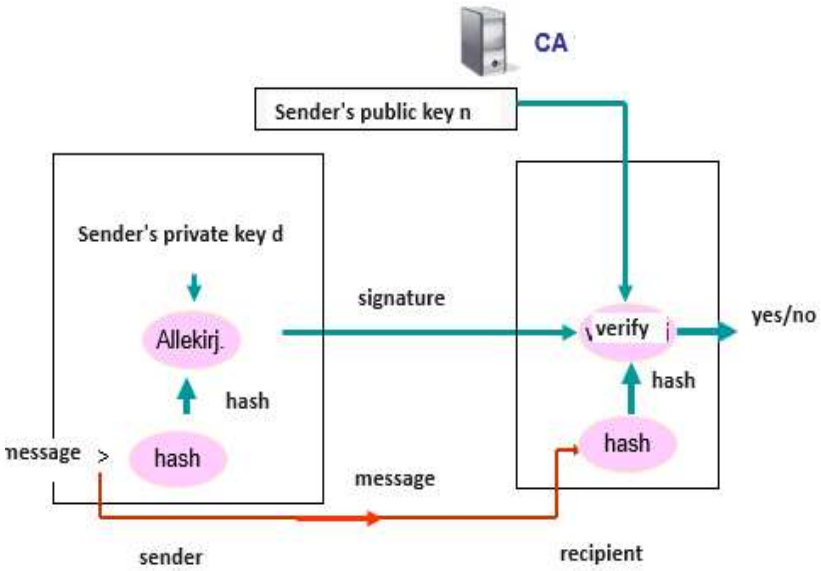
1) document/message is not changed

2) sender/author is authenticated

Digital signature algorithm usually uses hash function and public key encryption

Typical signature in TLS is **sha256RSA.** Another signature is ECDSA

Diagram of Digital Signature



The recipient receives the pair (m,s), where m = message and s = digital signature. This pair is mostly encrypted with recipients public key. The recipient needs in that case decrypt the pair before verification of signature.

The recipient verifies the signature in the following way

Decrypts the signature s with sender's public key. Result is the hash value of message.

He calculates another hash from the message itself.

If both hash values match, message is unchanged and sender authenticated

Digital signature sha256RSA

= the messages sha hash value encrypted with senders private key :

$S = sha256(m)^d \bmod n$

S = signature
m = message
sha256(m) = hash of the message
d = senders private key
n = senders public key

Calculated example of sha256RSA

In the example RSA - modulus n is short. In real situation n should be larger than sha – hash. That is why in this example we use shorter hash which is 4 digits from the beginning of real SHA. (WolframAlpha's Hash function is used to calculate SHA)

a) Alice sends Bob a message ”Tämä on koeviesti” with shaRSA digital signature.
Her RSA public key is n = 308911 and private key d = 133073.
Calculate the message-signature pair she sends.

At first we calculate SHA-value with W.A command: SHA ”Tämä on koeviesti” which returns 91264211644843524613812498926339549197338756350 5. We use only 4 digits hash : **9126**
Then we calculate signature $S = sha(m)^d \bmod n = 9126^{\wedge}133073 \bmod 308911 = 298954$

Alice sends pair (m,S) = (”Tämä on koeviesti”, 298954)

b) Bob receives (”Tämä on koeviesti”, 298954) from Alice. Show how he verifies the signature.

Firstly Bob decrypts signature using Alices public key: $298954^{65537} \bmod 308911 = 9126$
Then Bob calculates the signature from the message: SHA ”Tämä on koeviesti” , which returns 9126…..
Comparison shows that both signatures match. Signature is verified (integrity and authenticity proved)