Mini-https

Confidential, Integral, Authentic

Public Parameters PP = (p, g).

p= 268435019; g=2;

(E, D), (Sign, Ver)

Hand shaking PuK_A →

AKAP ↔

G, $\tilde{G} = (r, s)$ ↔

$PrK_A = x =$ randi(p-1)

$PuK_A = a = g^x \bmod p$

k

$PuK_A = a$

k

$PrK_B = y =$ randi(p-1)

$PuK_B = b = g^y \bmod p$

$u \leftarrow$ randi $(p-1)$

>> u = int64(randi(p-1)

$k_A = g^u \bmod p$

>> kA = mod_exp(g, u, p)

Sign $(x, k_A) = \tilde{G}_A = (r_A, s_A)$

$i \leftarrow$ randi $(p-1)$

$r_A = g^i \bmod p$

$h_A =$ concat (k_A, r)

$s_A = (i + x \cdot h_A) = \tilde{G}_A = (r_A, s_A)$

$k_A, \tilde{G}_A = (r_A, s_A)$

PuK_A →

1. Verify if PuK_A is in Data Base.
2. Verify if $\tilde{G}_A$ on k_A is valid.

Ver $(PuK_A, \tilde{G}_A, k_A) =$ T

3. Generates $v \leftarrow$ randi $(p-1)$

computes $k_B = g^v \bmod p$.

Sign $(y, k_B) = \tilde{G}_B = (r_B, s_B)$

← $k_B, \tilde{G}_B$

PuK_B

$k_{AB} = (k_B)^u \bmod p =$

$= (g^v)^u \bmod p = g^{vu} \bmod p = k_{AB} = k = k_{BA}$

$k_{BA} = (k_A)^v \bmod p =$

$= (g^u)^v \bmod p = g^{uv} \bmod p$

A creates transaction T_x

$E(k, T_x) = C$

Encrypt & sign paradigm

Chosenc ciphertex security

CCS

$h_c = H(C)$

Sign $(Prk = x, h_c) = \tilde{G} = (r, s)$

By realizing Schnorr – sign

$i \leftarrow randi\ (p-1)$

$r = g^i \bmod p$

$h_c = H\ (C \| r)$

$s = (i + x \cdot h_c) \bmod p$

>> cc = concat(C,r)

>> hC = hd28(cc)

$C, \sigma = (r, s)$ →

1. $Ver(PuK_A = a, \sigma) = \{T, H\}$
2. $D(k, C) = Tx$
3. Performes money transf.

MINI-HTTPS

€5.00

After receiving ***Tx*** and **σ**, **Bob** according to (2.20) computes h_C

$$h_C = H(C \| r),$$

and verifies if

$$g^s \bmod p = r a^h \bmod p.$$

V1 V2

Symbolically this verification function we denote by

Ver(*a*,σ,*h*)=V∈{*True*, *False*}≡{1, 0}.

This function yields ***True*** if (2.22) is valid if:

PuK_A= *a* =*F*(PrK_A)= g^x *mod p*

1. Mentor sends you Public Parameters (p= 15728303; g= 5) of 24 bits length. Generate public and private keys PrK_A=x and PuK_A=a. Send public key [a] to the Mentor.

12677229

```
>> p=int64(15728303)
p = 15728303
>> g=5;
>> x=int64(randi(p-1))
x = 9712179
>> a=mod_exp(g,x,p)
a = 12677229
```

2. Compute random secret number u of 24 bit length and compute session public parameter K_A. Sign K_A with Schnorr signature scheme by computing two components of signature $Sig_A = (r, s)$. Be aware that parameter $h = hd24(concat(K_A, r))$. Send $[K_A, r, s]$ to the Mentor.

hd28

14438572,14538340,13707662

```
>> u=int64(randi(p-1))
u = 7085010
>> KA=mod_exp(g,u,p)
KA = 14438572
>> i=int64(randi(p-1))
i = 5104007
>> r=mod_exp(g,i,p)
r = 14538340
>> cc=concat(KA,r)
cc = 1443857214538340
```

hd28

```
>> h=int64(hd24(cc))
h = 12720923
>> hh=int64(hd24(concat(KA,r)))
ans = 12720923

>> xh=mod(x*h,p-1)
xh = 8603655
>> s=mod(i+xh,p-1)
s = 13707662
```

3. Mentor sends you (PuK_B=4670305, K_B =918922, R =8070925, S =6944326). Verify Mentor's signature $Sig_B = (R, S)$ on K_B. If signature is valid then compute secret session symmetric key $K_{AB} = K$. Transform K to the hexadecimal form K_h. Create the string of message variable m ='MMDD' consisting of the month and day of your birth. Encrypt message m using 1 round of AES128 cipher with key K_h by computing ciphertext
>> C=AES128(m,Kh,1,'e').
Send $[C]$ to the Mentor for decryption and for resending message m to your friend Bob2. C should be entered within ": ,C"

$\sigma_M = (R, S)$

```
>> PuKB=4670305;
>> KB=872215;
>> R=2705869;
>> S=4402137;
>> hd=int64(hd24(concat(KB,R)))
hd = 3684952
>> rez=sig_ver(p,g,R,S,PuKB,hd)
rez = TRUE: Signature correct
      Val=    11780819
```

```
>> K=mod_exp(KB,u,p)
K = 10027187
```

```
>> Kh=dec2hex(K,32)
key = 00000000000000000000000000009900B3
>> m='0501'
m = 0501
>> C=AES128(m,Kh,1,'e')
new= ........
C = 8c000e708c0069008cf20e00eb990eb3
```

'8c000e708c0069008cf20e00eb990eb3'

4. Ok, Bob2 informed me that all the sum of money he received from the Knowledge Bank he dedicates to buy the gift for your birthday. This sum I am sending you as a ciphertext (C_M ='8c000e3c8c0075008c8d0e008c990eb3'). Please decrypt and check it and then encrypt it again with added string 'ok' right after the sum by computing ciphertext C_1.
Send $[C_1]$ to the Mentor. C_1 should be entered within ": 'c1'

```
>> CM='8c000e3c8c0075008c8d0e008c990eb3'
CM = 8c000e3c8c0075008c8d0e008c990eb3
```

```
>> M=AES128(CM,Kh,1,'d')
Out = 00000000000000000000000000003437
M = 47
 >> Mok='47ok'
>> C1=AES128(Mok,key,1,'e')
new = .....
C1 = 8c000e028c00c5008c870e00f7990eb3
```

'8c000e028c00c5008c870e00f7990eb3' C_1

'8c000e708c0069008cf20e00eb990eb3' C

Realize 10 rounds of encryption for the same plaintexts.

Success! You have finished the task. Great job!

Get reward

Current Directory: C:\Octave\Octave 7.1.0\~Eli.m

C:/Octave/Octave 7.1.0/~Eli.m

Name

AES128.m

bin2hex.m

```
% AES128(in,kh32,NR,fun) Advanced Encryption Standard symmetric cipher with key length of 128 bits
%                 Encryption is performed for 1 block of length 128 bits or 16 ASCII symbols
%
% in - plaintext/ciphertext of string type: maximum 16 symbols or shorter
%
% kh32 - shared secret key in hexadecimal number of length=32 (128 bits)
% kh32 can be obtained when shared decimal key k is given using commands:
%     >> k=int64(randi(2^28))
%     k = 160966896
%     >> kh32=dec2hex(k,32)
%     kh32 = 00000000000000000000000099828F0
%
% NR - Number of Rounds (e.g. Nr = 10)
%    The smaller NR, the lower security of encryption but the speed of encryption is higher
%    The least number of NR is 1 and in this case security lack is evident
%
% fun - letter determining either encription: fun='e' or decryption: fun='d' functions
%
% Encryption example:
% >> in = 'Hello Bob';
```

```
% >> kh32 = '00000000000000000000000099828F0';
% >> NR = 10;
% >> Ch = AES128(in,kh32,NR,'e')
% ASCII_e =   ?1  ~mV                % ciphertext in ASCII format
% Ch = 0f9a2c08d191310fb27ed16d90f45686   % ciphertext in
hexadecimal format
%
% Decryption example:
% >> Dh = AES128(Ch,kh32,NR,'d')
% Dh = 00000000000048656c6c6f7720426f62   % decrypted message in
hex format
% D = Hello Bob                    % Decrypted message in ASCII format
%
```

Till this place

$$\begin{pmatrix}1 & 2\\ 3 & 4\end{pmatrix}^{x} \qquad \begin{pmatrix}a & b\\ c & d\end{pmatrix}^{w} \qquad \begin{pmatrix}5 & 6\\ 7 & 8\end{pmatrix}^{y}$$

$${}^{x}W = \begin{pmatrix}a^1 \cdot c^2 & b^1 \cdot d^2\\ a^3 \cdot c^4 & b^3 \cdot d^4\end{pmatrix}$$

$$\left({}^{x}W\right)^{y} = A$$

$$W^{y} = \begin{pmatrix}a & b\\ c & d\end{pmatrix}^{\begin{pmatrix}5 & 6\\ 7 & 8\end{pmatrix}} = \begin{pmatrix}a^5 b^7 & a^6 b^8\\ c^5 d^7 & c^6 d^8\end{pmatrix}$$