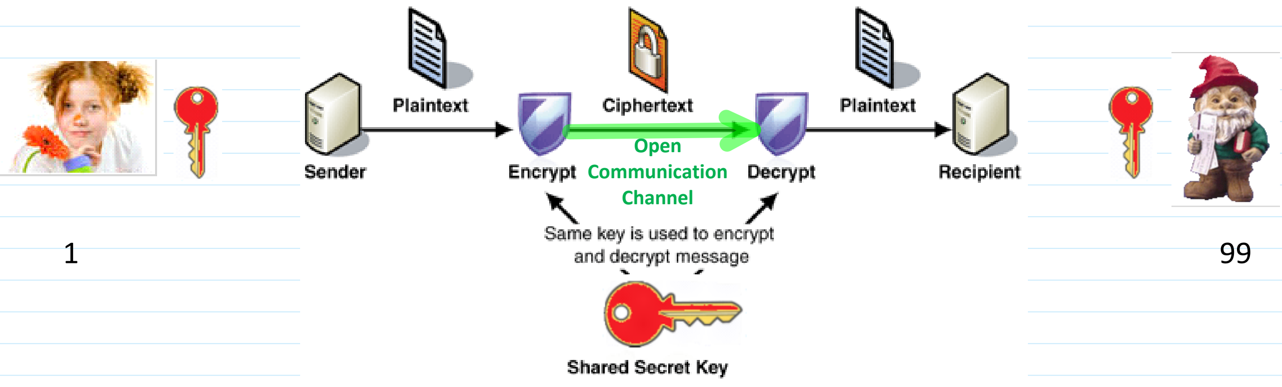
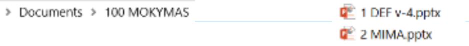


δ

$$C^{2}_{100} = \frac{100*99}{2} = 4950$$

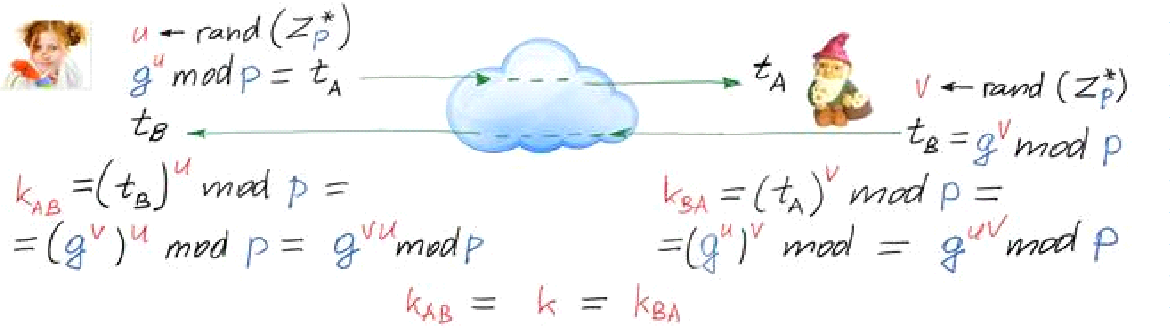


› Documents › 100 MOKYMAS

1 DEF v-4.pptx
2 MIMA.pptx

Diffie-Hellman Key Agreement Protocol (DH KAP)

Public Parameters PP=(*p*,*g*)



$u \leftarrow rand(Z_p^*)$

$g^u \bmod p = t_A$ → t_A

$v \leftarrow rand(Z_p^*)$

t_B ← $t_B = g^v \bmod p$

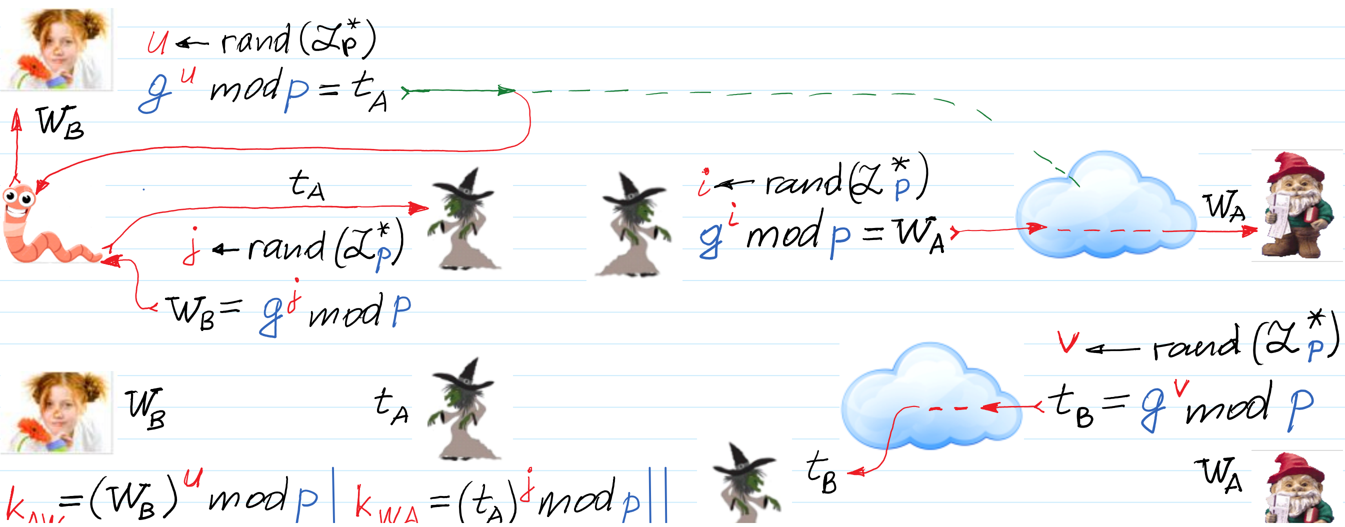
$k_{AB} = (t_B)^u \bmod p =$
$= (g^v)^u \bmod p = g^{vu} \bmod p$

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

$k_{AB} = k = k_{BA}$

Man in the Middle Attack - MiMA - Impersonation Attack

C:\Users\Eligijus\Documents\100 MOKYMAS



$u \leftarrow rand(\mathcal{Z}_p^*)$

$g^u \bmod p = t_A$

W_B

t_A

$j \leftarrow rand(\mathcal{Z}_p^*)$

$W_B = g^j \bmod p$

$i \leftarrow rand(\mathcal{Z}_p^*)$

$g^i \bmod p = W_A$

W_A

$v \leftarrow rand(\mathcal{Z}_p^*)$

$t_B = g^v \bmod p$

W_B t_A

t_B

W_A

$k_{AW} = (W_B)^u \bmod p \mid k_{WA} = (t_A)^j \bmod p \|$

$$k_{AW} = (W_B)^u \bmod p = (g^j)^u \bmod p = g^{ju} \bmod p \quad \Big| \quad k_{WA} = (t_A)^j \bmod p = (g^u)^j \bmod p = g^{uj} \bmod p$$

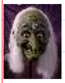
$$k_{AW} = k_1 = k_{WA}$$

t_B

W_A

$$k_{WB} = (t_B)^i \bmod p = (g^v)^i \bmod p = g^{vi} \bmod p \quad \Big| \quad k_{BW} = (W_A)^v \bmod p = (g^i)^v \bmod p = g^{iv} \bmod p$$

$$k_{WB} = k_2 = k_{BW}$$



It is an example of very actual so far kind of active attack directed to KAP. The actuality of this attack remains high due to the lack of identification from the ordinary customer side. According to this scenario the protocol is executed in the following way.

Alice chooses at random $\boldsymbol{u} \leftarrow \text{rand}(\boldsymbol{Z_p}^*)$, computes

$t_A = g^u \bmod p$,

and sends t_A thinking that it is sent to Bob but actually it is sent to Zoe.

Zoe after receiving t_A from Alice chooses at random $\boldsymbol{j} \leftarrow \text{rand}(\boldsymbol{Z_p}^*)$, computes

$W_B = g^j \bmod p$,

and sends W_B to Alice thus impersonating Bob.

Alice and Zoe after receiving t_A and W_B computes their secret keys k_{AW} and k_{WA} respectively.

$k_{AW} = (W_B)^u \bmod p = (g^j)^u \bmod p = g^{ju} \bmod p$.

$k_{WA} = (t_A)^j \bmod p = (g^u)^j \bmod p = g^{uj} \bmod p$.

Analogously to and Alice and Zoe agreed on the same secret key

$k_{AW} = k_1 = k_{WA}$.

Zoe continues computations with Bob in the similar way. Zoe chooses at random $\boldsymbol{i} \leftarrow \text{rand}(\boldsymbol{Z_p}^*)$, computes

$W_A = g^i \bmod p$,

and sends W_A to Bob thus impersonating Alice.

Bob does not suspecting any badness, as usual, chooses at random $\boldsymbol{v} \leftarrow \text{rand}(\boldsymbol{Z_p}^*)$, computes

$t_B = g^v \bmod p$,

and sends t_B to Zoe thinking that he have sent it to Alice.

Zoe and Bob after receiving t_B and W_B computes their secret keys k_{WB} and k_{BW} respetively

$k_{WB} = (t_B)^i \bmod p = (g^v)^i \bmod p = g^{vi} \bmod p$.

$k_{BW} = (W_B)^v \bmod p = (g^i)^v \bmod p = g^{iv} \bmod p$.

And again, analogously to and Zoe and Bob agreed on the same secret key.

$k_{BW} = k_2 = k_{WB}$.

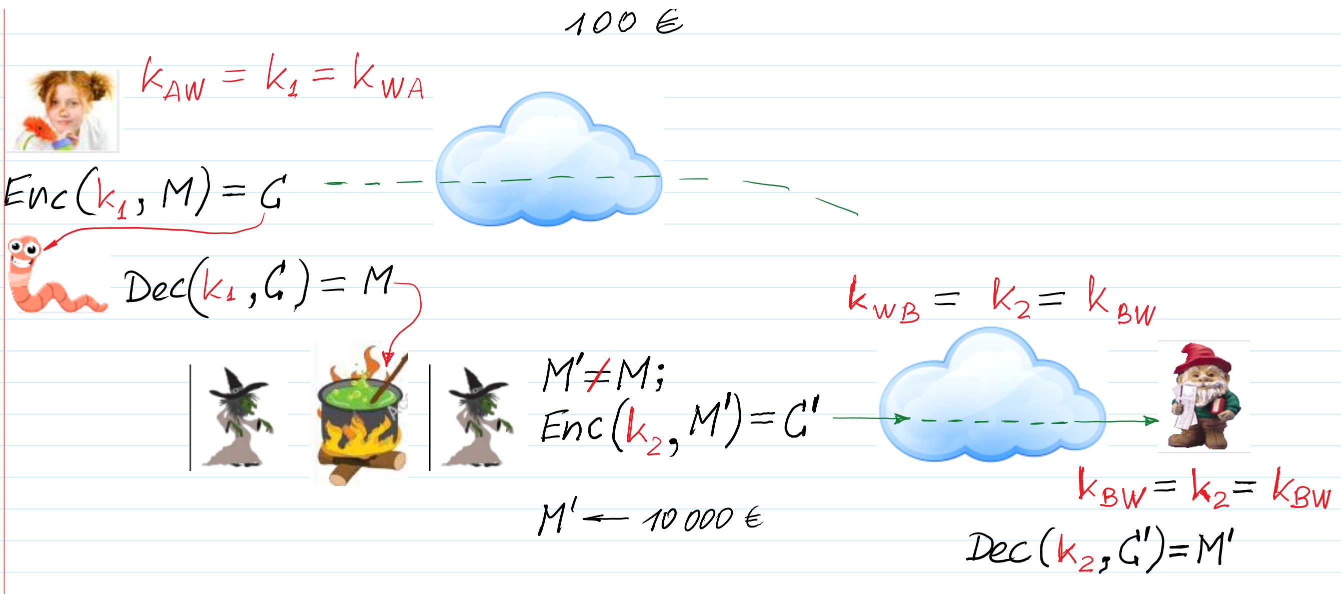
As an outcome of MiM Attack parties have agreeded two secret keys: key k_1 between Alice and Zoe and k_2 between Zoe and Bob.

M - message to be encrypted.

M = Account No From || Money amount || Account No To

100 €

$k_{AW} = k_1 = k_{WA}$



100 €

$k_{AW} = k_1 = k_{WA}$

$Enc(k_1, M) = C$

$Dec(k_1, C) = M$

$M' \neq M;$

$Enc(k_2, M') = C'$

$M' \leftarrow 10\,000$ €

$k_{WB} = k_2 = k_{BW}$

$k_{BW} = k_2 = k_{BW}$

$Dec(k_2, C') = M'$

off shore account
say in Panama.
„Panama Papers"

Imagine that Bob represents Bank and Alice is a customer of this Bank. Let Alice has a password to connect to the Bank which is compromised by the Worm infecting its computer It can be done by scanning Alice's keyboard when she is entering a password.

Let Alice wants to transfer a sum of money to her friend Bob2. Then she connects to the Bank and executes KAP described above. But the Alice do not suspect that her computer is infected by the Worm Zoe which realizes MiM Attack. So this Worm is in the role of Witch. When Alice composes the money transfer document ***M*** to Bob2, she encrypts it by the agreed secret key $\boldsymbol{k_1}$ using for example AES-128 symmetric encryption scheme by obtaining the following ciphertext

$\boldsymbol{C}$ = AES_Enc($\boldsymbol{k_1}$, ***M***).

Then she sends (she expects that she is sending) ***C*** to the Bank. Ciphertext ***C*** is intercepted by Zoe and sent to its computer. Then Zoe decrypts ***C*** and obtains ***M***

M = AES_Dec($\boldsymbol{k_1}$, ***C***),

and saw the transferring sum and Bob2 account. Then Zoe changes the money transfer account to her account creating a new message ***M`*** and encrypts it with key $\boldsymbol{k_2}$

C`*** = AES_Enc($\boldsymbol{k_2}$, ***M`).

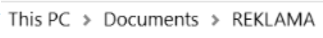
Zoe sends ***C`*** to the Bank.

Bank decrypts ***C`***

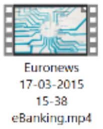
M`*** = AES_Dec($\boldsymbol{k_2}$, ***C`),

and transfers the indicated sum the Zoe account indicated in ***M`***.

http://crypto.fmf.ktu.lt/xdownload/



This PC › Documents › REKLAMA



Euronews 17-03-2015 15-38 eBanking.mp4

Smart Id --> GPRS

- Euronews 17-03-2015 15-38 CET _150316_HTSU_121B0-172837_E.mp4

http://www.euronews.com/2015/03/17/internet-banking-a-hacker-s-ideal-target/

Like Swiss Emmental cheese, the ways your online banking accounts are protected might be full of holes. According to internet security software developer Kaspersky, the number of cyberthreats reached record levels in 2014. One in three computers or mobile devices were subjected to at least one web attack over the year. Particular targets are companies or individuals using internet banking.

In January, a Swiss firm lost an estimated one **million** euros in an online financial transaction that was hacked.
The victim, an accountant at the company, was unaware of what was going on.
It started when he opened an email containing an attachment infected with a virus. Once they had taken control of his computer, all the hackers had to do was wait for him to connect online with his bank.
"When he tried to connect to his bank online, he activated the "Trojan horse". A message appeared asking him to hold. For **20 or 30 minutes**, he wasn't able to use his computer at all. During that time, the pirates took control of the computer and carried out several money transfers onto foreign accounts," says Frederic Marchon, spokesman for the Fribourg Police.
Plenty of viruses allowing that kind of illegal activity are available on the internet. The most updated versions are available for just over **1,000** euros on the **darknet**.
The hacker gets a warning as soon as someone connects with their bank online using an infected computer.
This IT expert explains how it works: "I can monitor all the computers I have successfully hacked, and I can see precisely, among them, how many are currently banking online and therefore vulnerable. So here, there are two which are currently connected," says IT expert Cedric Enzler.
Faced with a growing number of cyber attacks on companies, Switzerland has set up an emergency centre to track the attacks and analyse them. But the nature of the centre means they cannot provide with any names or figures.
"It's a really big problem. You've got to realise that anyone who wants to do harm and wants to make money that way will automatically turn to e-banking," says IT security expert Max Klaus.
For this professor at the Bern University of Applied Sciences, there's another big problem with this kind of cyber attack: most of the tools we use for internet banking like calculators or smartphone applications designed to read cryptograms are vulnerable to hacking.
"From an electronic point of vue, internet banking is safe. We use secure channels using SSL encryption. **The problem comes from the client's computer**, its use no longer guarantees a secure connection. Whether it's a computer or a smartphone, hackers can take and security is compromised," says Professor Reto Koenig.
None of the banks contacted agreed to answer to our questions on camera.
Swiss banks warn their clients about security problems linked to the use of internet in their general conditions – a warning which often comes with a clause clearing the bank of any responsibility in the event of an attack.
"The client is a victim twice over. First, he's the victim of a crook, and then he has hardly any chance to defend himself because of the general conditions in his contract. Sometimes, there are agreements between banks and clients but unfortunately, most of the time, these agreements are kept secret, they are confidential, so it's hard to find out what the procedure is, which is of course detrimental to the client," says Mathieu Fleury, of the Swiss consumer's rights association.
A coordinated cyber security taskforce and response scheme, aimed at providing cyber security services for small and medium enterprises in Europe, is to begin pilot deployments in 2015, starting in the UK, the Netherlands and Belgium.
EU authorities are concerned about the vulnerability of SMEs because they employ two-thirds of Europe's workforce.
More about:

- Banking
- Internet
- Security
- Switzerland

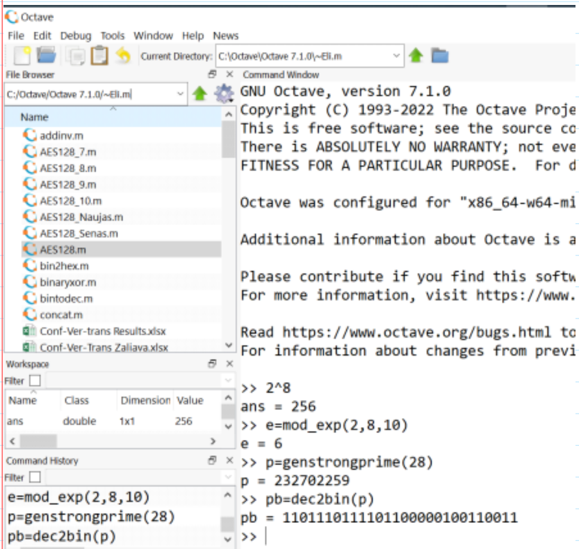
In this report it is pointed out that user, e.g. Alice had a ***weak identification*** at this time based only on Bank's passwords submitted to her. While Banks usually have a strong identification based on their public keys certification recognizable by users browsers. The material concerning Public Key Certificates (PKC) we will present later.
Since that one partial improvement was made by introducing two channel identification based on Smart Id protocol where user must confirm his/her identity using its smart phone and entering pin code.
To provide a strong identification it is required to use cryptographic identification methods together with something like Smart Id and biometrics. Smart Id
Therefore we start now from cryptographic identification methods and DS schemes. GPRS

https://imimsociety.net/en/

https://imimsociety.net/en/

http://crypto.fmf.ktu.lt/xdownload/

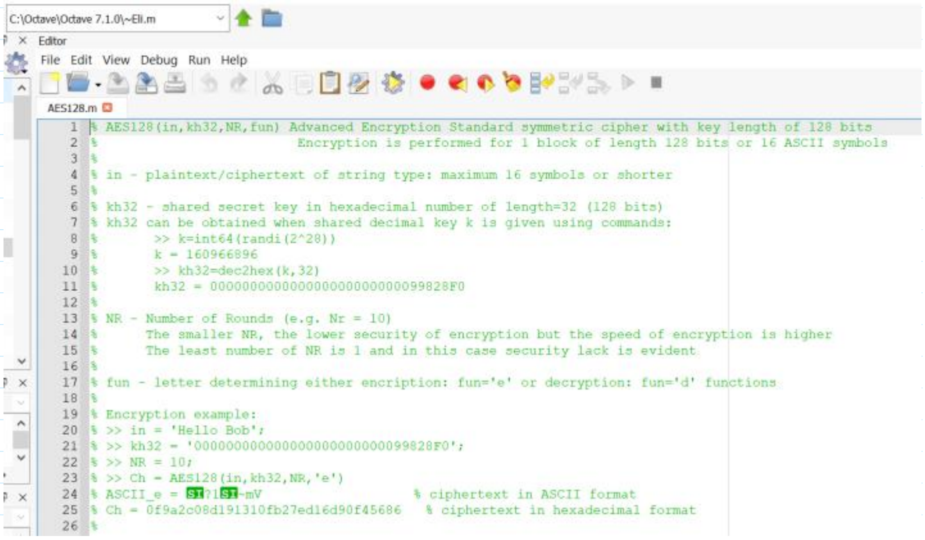
- octave-7.3.0-w64-installer.exe
- octave.Stud.7z



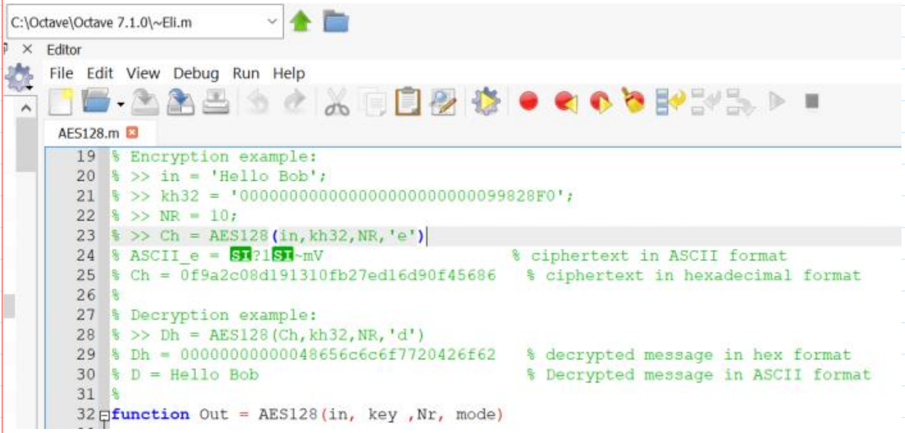
Name	Class	Dimension	Value
ans	double	1x1	256

```
e=mod_exp(2,8,10)
p=genstrongprime(28)
pb=dec2bin(p)
```

```
GNU Octave, version 7.1.0
Copyright (C) 1993-2022 The Octave Proje
This is free software; see the source co
There is ABSOLUTELY NO WARRANTY; not eve
FITNESS FOR A PARTICULAR PURPOSE.  For d

Octave was configured for "x86_64-w64-mi

Additional information about Octave is a

Please contribute if you find this softw
For more information, visit https://www.

Read https://www.octave.org/bugs.html to
For information about changes from previ

>> 2^8
ans = 256
>> e=mod_exp(2,8,10)
e = 6
>> p=genstrongprime(28)
p = 232702259
>> pb=dec2bin(p)
pb = 1101110111011000000100110011
>> 
```



```
% 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 = 000000000000000000000000099828F0
%
% 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 = '000000000000000000000000099828F0';
% >> NR = 10;
% >> Ch = AES128(in,kh32,NR,'e')
% ASCII_e = SI?1SI-mV                     % ciphertext in ASCII format
% Ch = 0f9a2c08d191310fb27ed16d90f45686   % ciphertext in hexadecimal format
%
```



```
% Encryption example:
% >> in = 'Hello Bob';
% >> kh32 = '000000000000000000000000099828F0';
% >> NR = 10;
% >> Ch = AES128(in,kh32,NR,'e')
% ASCII_e = SI?1SI-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
%
function Out = AES128(in, key ,Nr, mode)
```

For example:

Authenticated Key Agreement Protocol - AKAP

Smart Id.

Identification: Go Trust (Taiwan) → μSD

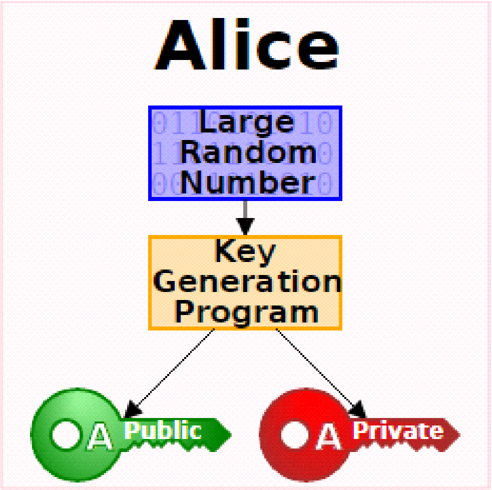
A: $PrK = x$; $PuK = a = g^x \bmod p$: it is infeasible to find x when p, g, a are given.

Strong prime p: $p \sim 2^{2048} \approx 10^{700}$; $|p| = 2048$ b.

In our simulation we will deal with $p \sim 2^{28}$ $|p| = 28$ b.

```
>> p = genstrongprime(28)
```

Digital signature: to sign a message $M \equiv t_A$ for KAP.



PrK and PuK are related

PuK = F(PrK)

F is one-way function - OWF:

1.It is easy to compute PuK when F and PrK are given.

Kerchoff principe.

2.Having PuK and F, it is infeasible to find PrK = F^{-1}(PuK).

Public Parameters PP = (p, g) $p \sim 2^{2048} \approx 10^{700}$; $|p| = 2048$ b,

= 700 dec. digits

We will use $|p| = 28$ bits.

To generate PrK and PuK we need to generate PP = (p, g)

PrK_A = x <-- randi ==> PuK_A = a = g^x mod p

F is a modular exponent function; $F_{p,g}(x) = g^x \bmod p = a$.

```
>> a = mod_exp(g, x, p)
```
$|p| = 28$ bit length

1. It is easy to compute PuK = a when p, g and x are given.

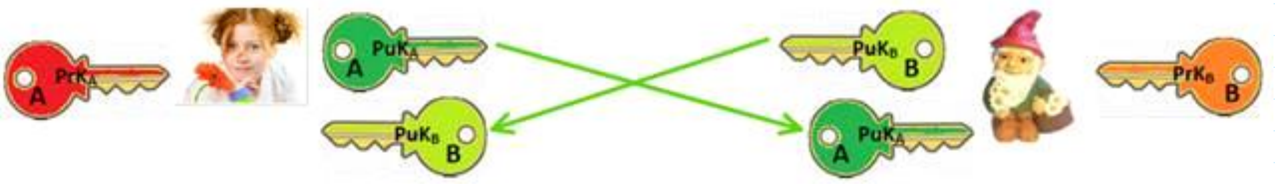
2. It is infeasible to find PrK = x when p, g and a are given.

Open SSL software
Python
Go

$|PrK| = 2048$ bits

$|PuK| = 2048$ bits

$[1, 2^{2048}]$



Asymmetric Encryption - Decryption

c=Enc(PuK_A, m) = (ε, δ) =(E, D)

m=Dec(PrK_A, c)

Asymmetric Signing - Verification

σ=Sign(PrK_A, m) = (r, s)

V=Ver(PuK_A, m, σ), V∈{True, False} = {1, 0}

Asymmetric Encryption - Decryption

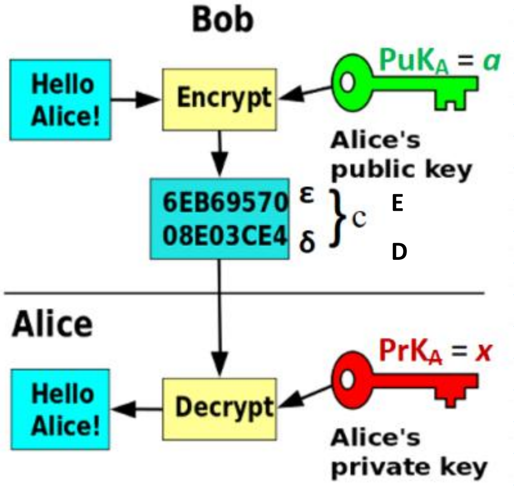
c=Enc(PuK_A, m) = (ε, δ) =(E, D)

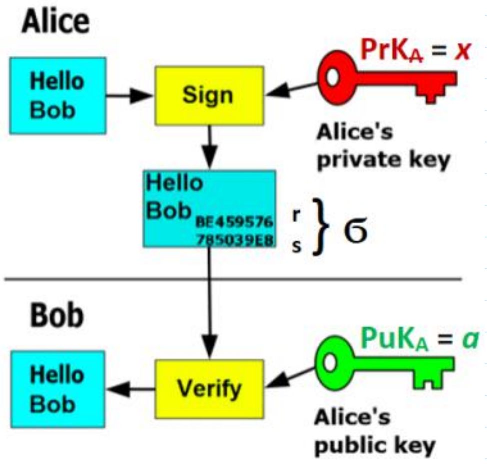
m=Dec(PrK_A, c)

Asymmetric Signing - Verification

σ=Sign(PrK_A, m) = (r, s)

V=Ver(PuK_A, m, σ), V∈{True, False} ≡ {1, 0}





Confidentiality

Authenticity

A: $PrK_A = x$ · $PuK_A = a$.

$PuK_B = b$.

$u \leftarrow randi(\mathbb{Z}_p^*)$

$t_A = g^u \bmod p$

$Sign(x, t_A) = \sigma_A = (r_A, s_A)$

$\xrightarrow{t_A,\ \sigma_A}$

B: $PrK_B = y$; $PuK_B = b$.

$PuK_A = a$.

1. Verification of σ_A on the t_A

$Ver(a, \sigma_A, t_A) \Rightarrow \{0, 1\} = \{F, T\}$

2. $v \leftarrow randi(\mathbb{Z}_p^*)$

$t_B = g^v \bmod p$

3. $Sign(y, t_B) = \sigma_B = (r_B, s_B)$

$\xleftarrow{t_B,\ \sigma_B}$

$Ver(b, \sigma_B, t_B) \Rightarrow T$

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