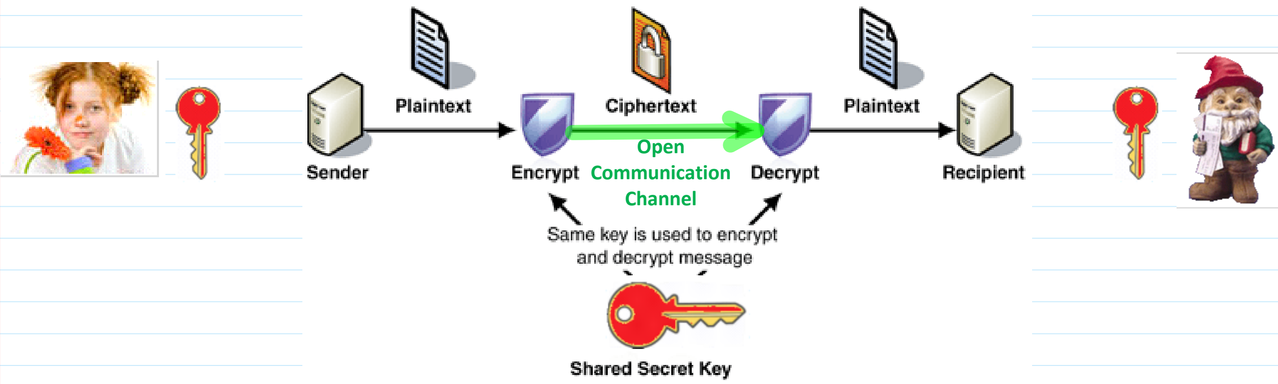
Symmetric Cryptography -------------------- Asymmetric Cryptography
Secret Key Cryptography Public Key Cryptography

Symmetric encryption
H-functions, Message digest
HMAC H-Message Authentication Code

Asymmetric encryption
E-signature - Public Key Infrastructure - PKI
E-money, Blockchain
E-voting
Digital Rights Management - DRM (Marlin)
Etc.



Public Key Cryptography - PKC

Principles of Public Key Cryptography

Instead of using single symmetric key shared in advance by the parties for realization of symmetric cryptography, asymmetric cryptography uses two ***mathematically*** related keys named as private key and public key we denote by **PrK** and **PuK** respectively.
PrK is a secret key owned ***personally*** by every user of cryptosystem and must be kept secretly. Due to the great importance of **PrK** secrecy for information security we labeled it in **red** color. **PuK** is a non-secret ***personal*** key and it is known for every user of cryptosystem and therefore we labeled it by **green** color. The loss of **PrK** causes a dramatic consequences comparable with those as losing password or pin code. This means that cryptographic identity of the user is lost. Then, for example, if user has no copy of **PrK** he get no access to his bank account. Moreover, his cryptocurrencies are lost forever. If **PrK** is got into the wrong hands, e.g. into adversary hands, then it reveals a way to impersonate the user. Since user's **PuK** is known for everybody then adversary knows his key pair (**PrK**, **Puk**) and can forge his Digital Signature, decrypt messages, get access to the data available to the user (bank account or cryptocurrency account) and etc.
Let function relating key pair (**PrK**, **Puk**) be ***F***. Then in most cases of our study (if not declared opposite) this relation is expressed in the following way:

$$\mathbf{PuK}=F(\mathbf{PrK}).$$

In open cryptography according to Kerchoff principle function ***F*** must be known to all users of cryptosystem while security is achieved by secrecy of cryptographic keys. To be more precise to compute **PuK** using function ***F*** it must be defined using some parameters named as public parameters we denote by **PP** and color in blue that should be defined at the first step of cryptosystem creation. Since we will start from the cryptosystems based on discrete exponent function then these public parameters are

PP = (p, g).

Notice that relation represents very important cause and consequence relation we name as the direct relation: when given **PrK** we compute **PuK**.

Let us imagine that for given F we can find the inverse relation to compute **PrK** when **PuK** is given. Abstractly this relation can be represented by the inverse function F^{-1}. Then

PrK=F^{-1}(**PuK**).

In this case the secrecy of **PrK** is lost with all negative consequences above. To avoid these undesirable consequences function F must be **one-way function** – OWF. In this case informally OWF is defined in the following way:

1. The computation of its direct value **PuK** when **PrK** and F in are given is effective.
2. The computation of its inverse value **PrK** when **PuK** and F are given is infeasible, meaning that to find F^{-1} is infeasible.

The one-wayness of F allow us to relate person with his/her **PrK** through the **PuK**. If F is 1-to-1, then the pair (**PrK**, **Puk**) is unique. So **PrK** could be reckoned as a unique secret parameter associated with certain person. This person can declare the possession or **PrK** by sharing his/her **PuK** as his public parameter related with **PrK** and and at the same time not revealing **PrK**.

So, every user in asymmetric cryptography possesses key pair (**PrK**, **PuK**). Therefore, cryptosystems based on asymmetric cryptography are named as **Public Key CryptoSystems** (**PKCS**).

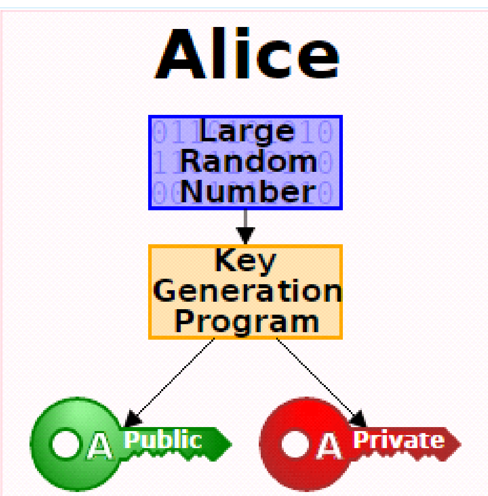
We will consider the same two traditional (canonical) actors in our study, namely Alice and Bob.

Everybody is having the corresponding key pair (**PrK**$_A$, **PuK**$_A$) and (**PrK**$_B$, **PuK**$_B$) and are exchanging with their public keys using open communication channel as indicated in figure below.

Animaction: Key generation

https://imimsociety.net/en/14-cryptography

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



PrK and **PuK** are related

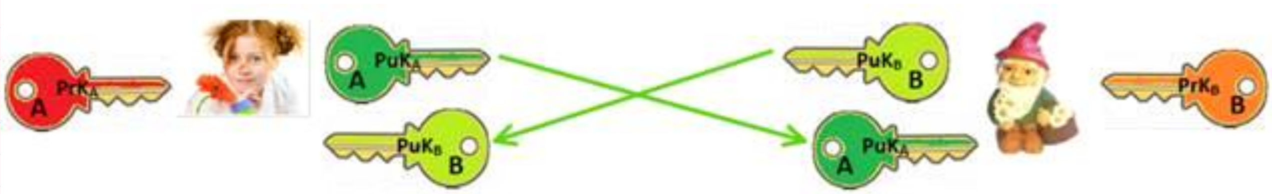
PuK = **F**(**PrK**)

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**).



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

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

>> p = genstrongprime (28)

PrK = *x* <-- randi ==> PuK = *a* = g^x mod p

$|PrK| = 2048$ bits $\quad [1, 2^{2048}]$

$|PuK| = 2048$ bits

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

>> a = mod_exp(g, x, p) $\quad |p| = 28$ 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 $\quad [1, 2^{2048}]$

$|PuK| = 2048$ bits

Asymmetric Encryption - Decryption

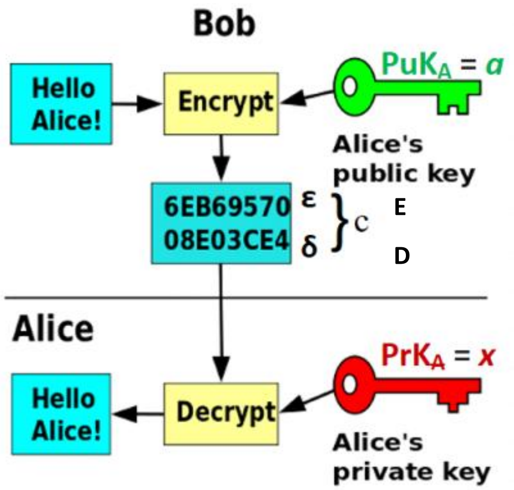
c=Enc(PuK_A, m) = (ε, δ) =(E, D)

m=Dec(PrK_A, c)

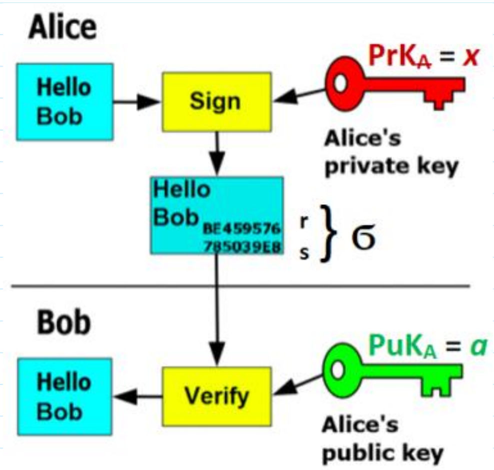
Asymmetric Signing - Verification

σ=Sign(PrK_A, m) = (r, s)

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



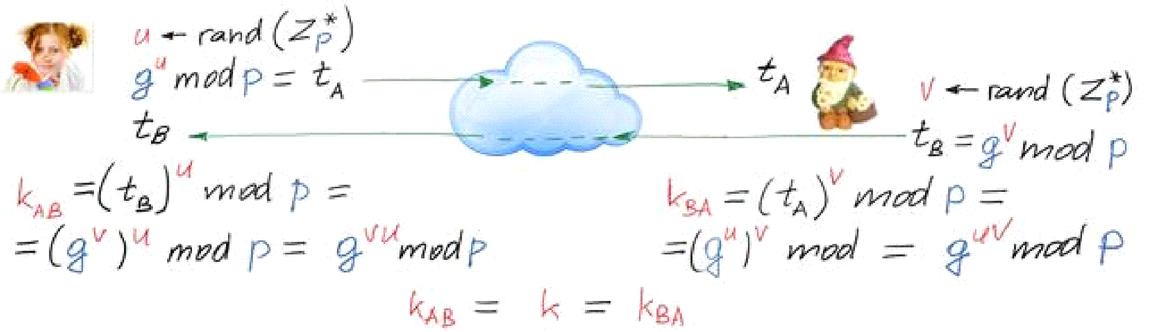
Confidentiality



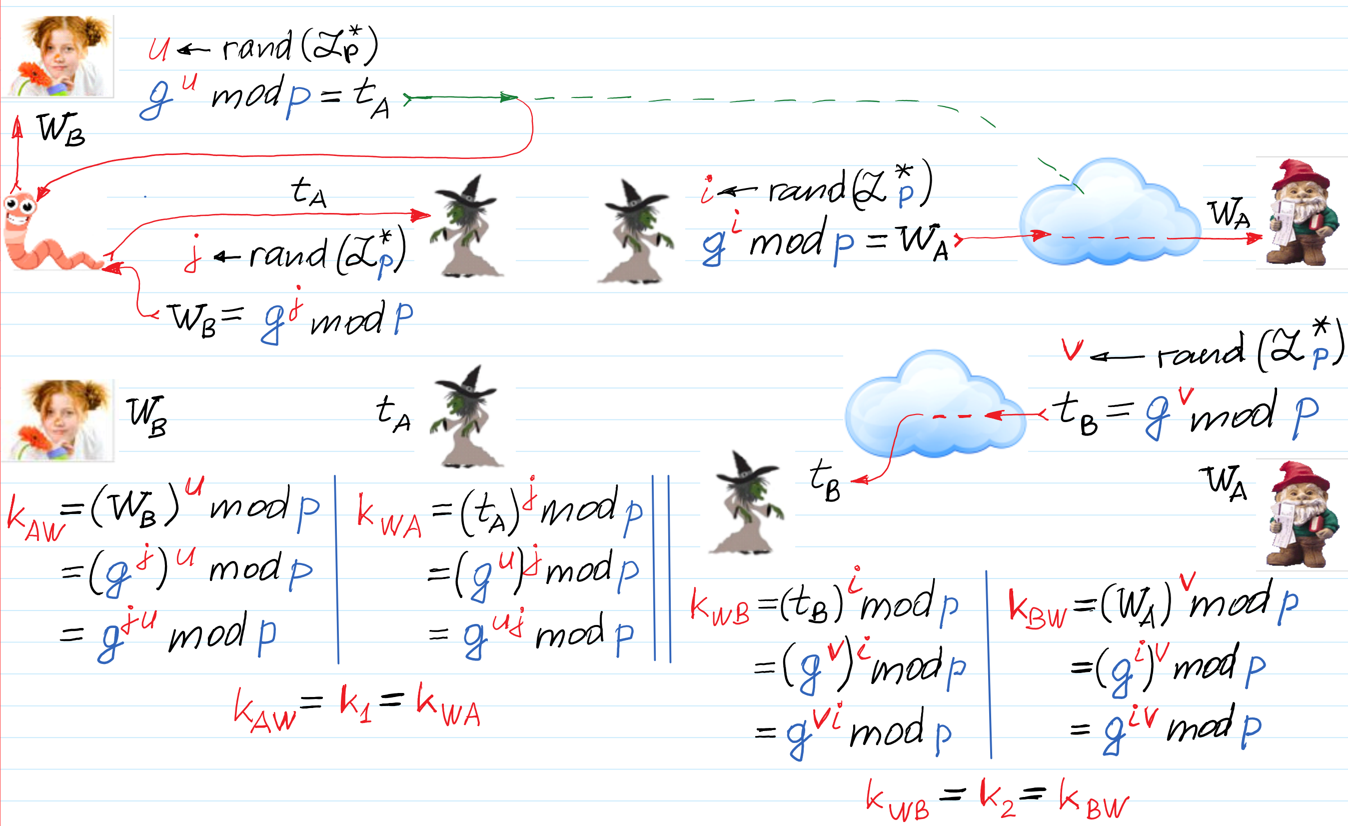
Authenticity

Diffie-Hellman Key Agreement Protocol (DH KAP)

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



Man in the Middle Attack - MiMA - Impersonation Attack



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 $u \leftarrow \mathrm{rand}(Z_p^*)$, 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 $j \leftarrow \mathrm{rand}(Z_p^*)$, 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 €

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

$Enc(k_1, M) = C$

$Dec(k_1, C) = M$

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

$M' \neq M;$
$Enc(k_2, M') = C'$

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

$M' \leftarrow 10\,000$ €

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

Authenticated Key Agreement Protocol - AKAP

A: $PrK_A = x$; $PuK_A = a$.
$PuK_B = b$.

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

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

$t_A = g^u \bmod p$

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

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

← t_B, σ_B

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

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

Lo : $PrK = z$; $PuK = d$. $\xrightarrow{d}$ B :

ačiū

E-signature - Public Key Infrastructure - PKI

CA - Certification Authority

RA - Registration Authority

Certicom

A: $PrK_A = x$; $PuK_A = a$.

$\xrightarrow{a}$

$\xleftarrow{Cert_{CA}}$

CA : $PrK = u$; $PuK_{CA} = c$. TTP

$Cert_{CA}$: $Sign(u, a) = \sigma_{CA}$

$Cert_{CA}$: $\{ \sigma_{CA} \,||\, Data \}$.