Registration to https://imimsociety.net/en/
ZKP-Tale
Topics of Course Work

7 Taher S170M123

7 Alhajjeh Taher
Bobojonov Ravshan
Hamzeh Zeid
Iqbal Mussa
Majidiniya Arezoo
Medina Paz Maria Fernanda
Momodu Abdulmalik
Olowogunle James Sunday
Stephen Nikitha
Ukonu Chibuzor Mike

>> p = genstrongprime(28)
>> q = (p-1)/2
>> isprime(q)

What are prime numbers: 3, 4, 5, 6, 7
$p=2q+1$ is srong prime if q is prime.
What are strong prime numbers: 7, 9, 11, 13

$Z_p^* = \{1, 2, 3, \ldots, p\text{-}1\}$ multiplication *mod p

Fact C.23. Say $p=2q+1$ is srong prime where q is prime. Then g in Z_p^* is a generator of Z_p^* iff (if and only if - tada ir tik tada) $g^q \neq 1 \bmod p$ and $g^2 \neq 1 \bmod p$.

p = 178096967
g = 20

PP = (p, g) - public parameters
$p \sim 2^{2048}$; $|p| = 2048$ bits

PrK = x = 29224923
PuK = a = 55545202

>> x = randi(2^28)
>> a = mod_exp(g, x, p)

p=268435019; g=2.

$a = g^x \bmod p$

Parties: **Alice - A** and **Bank - B**

Registration phase: Bank generates **PrK = x** and **PuK = a** to Alice
And hands over this data in smart card or other crypto chip in Alice's smart phone
Or in software for Smart ID.

$PrK_B = y \leftarrow$ randi(28)
$1 < x < p-1$
$PuK_B = b = a^y \bmod p$

$PrK_B = y \leftarrow randi(28)$

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

$1 < x < p-1$

$x \leftarrow randi(2^{\wedge}28)$

$a = g^x \bmod p$

$\mathcal{A}:$ $\xleftarrow{p, g, x, a, b}$ $\mathcal{B}:$

Schnorr Id Scenario: Alice wants to prove Bank that she knows her Private Key - **PrK** which corresponds to her Public Key - **PuK** not revealing **PrK**.
Protocol execution between Alice and Bank has time limit.
Alice's computation resources has a limit --> protocol must be computationally effective.

Zero knowledge Proof - ZKP

$\mathcal{A}$ - is a prover; $\mathcal{B}$ - is a verifier

Proof procedure is performed by the conversation between $\mathcal{A}$ and $\mathcal{B}$.

$\mathcal{A}$: $u \leftarrow rand(\mathbb{Z}_p^*)$; $1 < u < p-1$.

$t = g^u \bmod p$

t - commitment (1) $\xrightarrow{t,\ a}$ $\mathcal{B}$: $h \leftarrow rand(\mathbb{Z}_p^*)$ challenge

r - response $\xleftarrow{h}$ (2)

$r = u + xh \bmod (p-1)$

(3) $\xrightarrow{r}$

$\mathcal{I}o$: $r, h, a \to$ find x or u

$u = r - xh \bmod (p-1)$

$u, x \sim 2^{112} \sim 10^{40}$

brute force, total scan attack

$x = (r - u)\, h^{-1} \bmod (p-1)$

$g^r \bmod p =$

$= g^{u+xh} \bmod p = g^u \cdot g^{xh} \bmod p =$

$= t \cdot (g^x)^h \bmod p = t \cdot a^h \bmod p$

$\mathcal{A}$: computation resources are small $\Rightarrow$

$\Rightarrow$ arithm. operations should be effective.

Most expensive operation is $\boxed{t = g^u \bmod p}$

Arithm. operations should be effective.

Most expensive operation is $\boxed{t = g^u \bmod p}$

1) Time slot of Id is restricted

2) t is sent before the h is received.

Signature

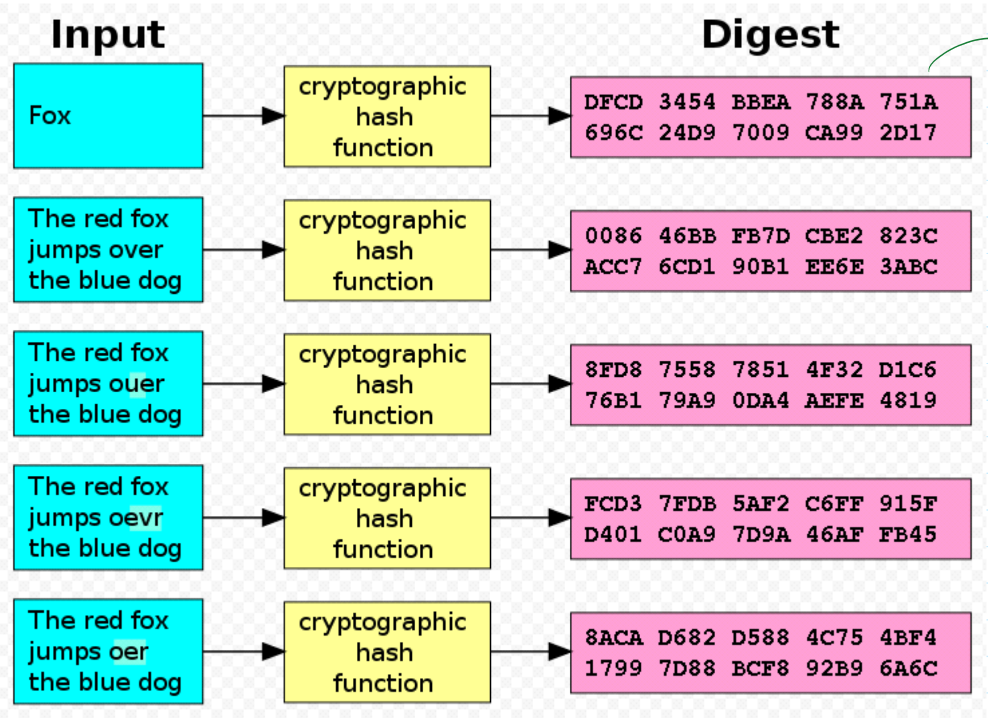
H-Functions are working horses in cryptography [Bruce Schneier].

A **cryptographic hash function** is a special class of hash function that has certain properties which make it suitable for use in cryptography.
It is a mathematical algorithm that maps data of arbitrary finite size to a bit string of a fixed size (a hash function) which is designed to also be a one-way function, that is, a function which is infeasible to invert.
The only way to recreate the input data from an ideal cryptographic hash function's output is to attempt a brute-force search of possible inputs to see if they produce a match.
The input data is often called the ***message***, and the output (the ***hash value*** or ***hash***) is often called the ***message digest*** or simply the ***digest***.

M – message to be signed (big message ~ 1 GB → 8 G bits)

$|p| \sim 2048$ bits

$H(M) = h$; $|h| \sim 256$ bits



$SHA_{160}(Fox) =$

awelange effect

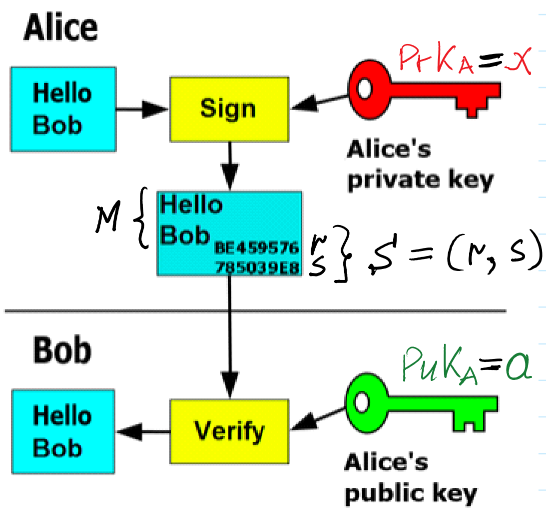
Public Parameters $PP = (p, g)$

M - be a message to be signed

Asymmetric Signing - Verification

S=Sig(PrK$_A$, h)=(r, s)

V=Ver(PuK$_A$, S, h), V$\in \{\text{True}, \text{False}\} \equiv \{1, 0\}$



Signing:

$u \leftarrow randi(p-1)$

$r = g^u \bmod p$; r - "commitment"

$h = H(M \| r)$

$s = u + x h \bmod (p-1)$

$S = (r, s)$

Verifying

$$g^s \bmod p = g^{u+xh} \bmod p =$$

$$= g^u \cdot g^{xh} \bmod p = r \cdot (g^x)^h \bmod p =$$

$$= r \cdot a^h \bmod p.$$