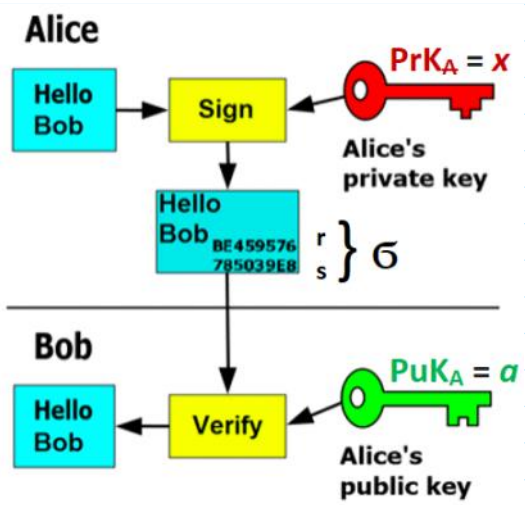
Paskaita Gegužės 18 d., 17:30. Egz. Repeticija.
KD gynimas Gegužės 25 d., 17:30, 140 a. Sharding, Smart Contracts, Private ang Public Ethereum, Products value chain, Roll-ups, ….. 15 slides, 8 min.
Egzaminas Birželio 15 d., 17:30, 103 f.

Subliminal Channel - Steganography using ElGamal Signature

>> p = 268 435 019; % 2^28-1 --> >> int64(2^28-1) % ans = **268 435 455**
>> g=2; % >> dec2bin(2^28-1)
% ans = 1111 1111 1111 1111 1111 1111 1111

```
>> p=268435019;
>> g=2;
```

ElGamal Signature: Sign(PrK,h)=σ=(r,s)



Subliminal Channel - Pasąmonės Kanalas: Bob was put in jail.
Paslėptas Kanalas

A: B:

```
>> z=randi(p-1)
z =  100497451                    z =  100497451
>> c=mod_exp(g,z,p)
c = 91968695                      c = 91968695
```

Let **M** is a message to be signed which can be arbitrary chosen.

To sign **M** the **h**-value of **M** must be computed: **h=H(M)**.
The secret message to be sent is a number **n=k** satisfying the following conditions:
1. Number **k<p**.
2. gcd(**k**, **p**-1)=1.

1.***Signature generation on* h :** $k \leftarrow randi(p-1)$ & $k=n$.

- Compute **k^{-1} mod (p-1): k^{-1} mod (p — 1)** exists if **gcd(k, p — 1) = 1**, **k** and **p-1** are relatively prime
k^{-1} can be found using either Extended Euclidean algorithmtor Euler theorem

```
>> k_m1=mulinv(k,p-1)   % k^-1 mod (p-1) computation.
```

- Compute **$r=g^k$ mod p**
- Compute **$s=(h-z*r)*k^{-1}$ mod (p-1)** --> **$h=z*r+s*k$ mod (p-1),**

Signature σ**=(r,s)**
For subliminal channel creation s^{-1} mod (p-1) must exists, such that **$s*s^{-1}$= 1 mod (p-1).**

$gcd(s, p-1) = 1$

A: $M, \sigma = (r, s)$ → B: $h = H(M)$

2.**Signature on h verification**
A signature (r,**s**) on message **h** is verified as follows.
1. 1**<r<**p**-1** and 1**<s<**p**-1.**
2. **V1=$c^r r^s$ mod p, V2=g^h mod p** and **V1=V2.**
The verifier accepts a signature if all conditions are satisfied and rejects it otherwise.

2.**Secret message n=k recovery from the equation**
$s=(h-z*r)*k^{-1}$ mod (p-1) /* k $mod(p-1)$
s must satisfy the condition that s^{-1} mod (p-1) exists, such that **$s*s^{-1}$= 1 mod (p-1).**

$k*s = h - z*r \mod (p-1) \;/\cdot s^{-1}$ % $gcd(s, p-1) = 1$
$k = (h - z*r) * s^{-1} \mod (p-1)$; $k = n$.

Subliminal Channel - Steganography
Using Schnorr Signature

M - masking message to be sign.
$h' = k$ - secret message to be sent.
$r = g^k \mod p$

$n' = k$ - secret message to be sent.

$r = g^k \bmod p$

$h = H(M \| r)$

$s = h + z * k \bmod (p-1)$

$M, \sigma = (r, s) \rightarrow$ B: $h = H(M \| r)$

$V1 = g^s \bmod p$

$V2 = r * c^h \bmod p$

$V1 \stackrel{?}{=} V2$

$s - h = z * k \quad / * z^{-1}$

$k = (s-h) * z^{-1} \bmod (p-1) = n$

Bit Commitment using H-function

B: Should I must sell my bitcoins?

A: Don't hurry, I know the price for next month.

B: Then tell me please.

A: I'll tell you next month, but if you want to know immediatly give me 3 BTC.

B: How it is to know me that you are not cheating?

A: We can use Bit commitment scheme.

Bit Commitment using HMAC

BTC-NM

$h = H(\text{BTC-NM} \| \text{rand})$

$h \rightarrow$ B:

After 1 month

1 BTC = 27 000 $

Send me your guess

BTC-NM || rand $\rightarrow$

$h' = H(\text{BTC-NM} \| \text{rand})$

$h \stackrel{?}{=} h'$

$\mathcal{A}$: $k = lsb_{256}(z)$ symmetric key

$\mathcal{B}$: $k = lsb_{256}(z)$

$HMAC(k, BTC\text{-}NM \| rand) = h$ —h→ After 1 month

← Send me your guess

$Enc(k, BTC\text{-}NM \| rand) = c$ —c→ $Dec(k, c) = BTC\text{-}NM \| rand$

$h' = HMAC(BTC\text{-}NM \| rand)$

If $h = h' \Rightarrow \mathcal{A}$ guessed BTC-NM.

Bit Commitment using RSA

Public Parameter $PP = (p)$ p - may be strong prime

$\mathcal{A}$: $K_A = (e_A, d_A)$ $\mathcal{B}$: $K_B = (e_B, d_B)$ % private keys

$e_A \cdot d_A = 1 \bmod (p-1)$ $e_B \cdot d_B = 1 \bmod (p-1)$

$e_A \leftarrow rand\,i : gcd(e_A, p-1) = 1 \Rightarrow \exists!\ d_A = e_A^{-1} \bmod (p-1)$

M - message : Bitcoin price next month

$\mathcal{A}$: Encrypts M with encryption function E and sends ciphertext c_1 to $\mathcal{B}$. $M < p$.

$Enc(e_A, M) = M^{e_A} \bmod p = c_1$

c_1 → $\mathcal{B}$: $Enc(e_B, c_1) = c_1^{e_B} \bmod p = c_2$

← c_2

After 1 month

$A: c_3 = c_2^{d_A} \bmod p \xrightarrow{c_3} B: c_4 = c_3^{d_B} = \dots$

$$= \left(c_2^{d_A}\right)^{d_B} = \left(\left(c_1^{e_B}\right)^{d_A}\right)^{d_B} =$$

$$= \left(\left(\left(M^{e_A}\right)^{e_B}\right)^{d_A}\right)^{d_B} \bmod p =$$

$$= M^{(e_A d_A)(e_B d_B) \bmod (p-1)} \bmod p =$$

$e_A d_A \bmod (p-1) = 1$ & $e_B d_B \bmod (p-1) = 1$

$$= M^{1 \cdot 1} \bmod p \overset{M<p}{=\!=} M$$