MSBD5017 HW1

XiaoLi_20925438

27 Sep 2022

Question1

Write a program or go to the reference website to hash the phrase: "Hello, world!*" with a number of appended to generate 4 leading "0"s. (e.g. Hello, world!0, Hello, world!1)

```
import hashlib


def do_cal(string):
    md5 = hashlib.md5()
    md5.update(string.encode("utf-8"))
    return md5.hexdigest()


a = "Hello, world!"
b = ""
number = 1
while b[:4] != "0000":
    c = a + str(number)
    b = do_cal(c)
    number = number + 1
print("The string is: " + c + "\n" + "The hash value of string is: " + b)
```

```
## The string is: Hello, world!59794
## The hash value of string is: 00006b9729f7ac9427c04f6b8080930c
```

Question2

Show the correctness of the verification model of ECDSA signature scheme.

① We define
- large prime number p
- finite field $F_p = \{0, \cdots, p-1\}$
- $a, b \in F_p$, which are parameter of Elliptic Curve
- base point $G_1(x_G, y_G) \in E(F_p)$ — all points on Elliptic Curve

② secret key : $sk \in F_p$

③ public key : $pk = sk \cdot G_1 \in E(F_p)$

④ encrypt message with hash : $H(m)$ ← may take part info from $H(m)$

⑤ choose randomly k compute : $(x, y) = k \cdot G_1$

⑥ $r = x \pmod p$

⑦ $s = k^{-1}(z + r \cdot sk) \pmod p$

⑧ (r, s) is the signature of m by ECDSA

Verify:

① $r, s \in F_p = \{0, \cdots, 1-p\}$

② $z = hash(m)$

③ $u_1 = z \cdot s^{-1} \pmod p = z \cdot k(z + rd)^{-1}$

$u_2 = r \cdot s^{-1} \pmod p$

④ $(x', y') = u_1 G_1 + u_2 \cdot sk \cdot G_1 \pmod p$ (base point; public key (known))

⑤ If $r = x'$ $\Rightarrow$ the signature is verified successfully

Here we show how ⑤ works:

$$(x', y') = \mu_1 G + \mu_2 \, sk \cdot G \pmod p$$

$$= z \cdot s^{-1} G + r \cdot s^{-1} \, sk \cdot G \pmod p$$

$$= (z + r \, sk) \, s^{-1} \cdot G \pmod p$$

$$= (z + r \, sk) \left[k^{-1} (z + r \cdot sk) \right]^{-1} G \pmod p$$

$$= (z + r \, sk) \, k \cdot (z + r \cdot sk)^{-1} \cdot G \pmod p$$

$$= k \cdot G \pmod p$$

$$= (x, y) \pmod p$$

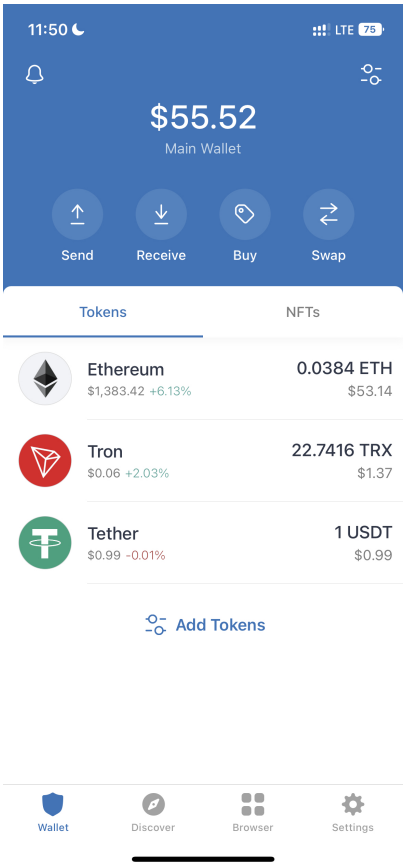
$\therefore r = x \pmod p$

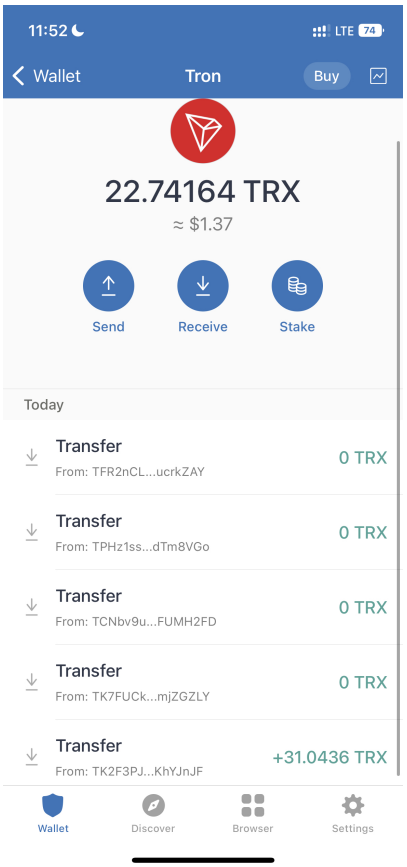
$\therefore$ When $r = x'$ $\Rightarrow$ verify successfully

Question3

Experience Blockchain:

- **(1) Install a Crypto Wallet on your PC and/or mobile phone;**
- **(2) Make some transactions (e.g. purchase something or exchange with a friend), and show the transaction record as proof;**
- **(3) List three areas of improvement for the wallet software that you use**





Improvement for application named trust:

- do not support USDT(TRC20) swap for other cryptos, when you want to transfer USDT(TRC20) you need to buy TRX to pay the transaction fee.
- browser interface embedded is very simple, needs more fuctions
- no chrome extension