Payword

- Schema de microplata electronica bazata pe lanturi hash https://people.csail.mit.edu/rivest/pubs/RS96a.prepub.pdf
- utilizata pentru realizarea de cumparaturi ce au valori mici (1 cent, 1$)
- destinata furnizarii repetate a unei cantitati mici de informatii:
  - plata pentru paginile web vizitate
  - plata pentru melodii, filme descarcate
  - plata pentru știri, articole de jurnal
- scopul constructiei mecanismului de plata: reducerea aplicarii operatiilor cu chei publice prin inlocuirea acestora cu operatii hash

Descrierea Schemei

- notatie: $\mathbf{sig_B(m)}$ inseamna concatenarea $\mathbf{m, sig_B(h(m))}$
  unde
  h - functia hash SHA-1(https://en.wikipedia.org/wiki/Secure_Hash_Algorithm),
  sig - semnatura digitala RSA
  (http://profs.info.uaic.ro/~cbirjoveanu/RSA Signature.pdf
  https://en.wikipedia.org/wiki/RSA_cryptosystem#Signing_messages)
- participanti:
  - user/cumparator (U)
  - vanzator (V)
  - broker (B)
- doua faze:

1. Inregistrarea U la B

U furnizeaza informatii personale (identitatea sa, $\mathbf{K_U}$ - cheia sa publica RSA, etc) la B pe un canal privat autentic.
B trimite la U un certificat payword **C(U)**

- **B → U: $C(U) = sig_B(B, U, K_B, K_U, exp, info)$**
  unde:
  B/U - identitate Broker/User
  K_B/K_U - cheia publica RSA a lui B/U
  exp - data de expirare
  info - serial, limita creditare, etc

- U verifica C(U) prin verificarea semnaturii lui B (utilizand cheia publica RSA a lui B)
- Prin C(U), B autorizeaza U sa realizeze plati catre vanzatori utilizand acest certificat, adica sa construiasca lanturi hash

2. <u>Plata</u>

Daca U doreste realizarea unei plati la V pentru prima data in acea zi, atunci U genereaza un nou lant hash

$\mathbf{c_n}$
$\mathbf{c_{n-1} = h(c_n)}$
$\mathbf{c_{n-2} = h(c_{n-1})}$
:
$\mathbf{c_1 = h(c_2)}$
$\mathbf{c_0 = h(c_1)}$

unde

n - ales de U convenabil pentru a permite succesiunea de plati
$\mathbf{c_n}$ - secret generat aleator de U
toate payword-urile $\mathbf{c_i}$ au aceeasi valoare (de exemplu de 1 cent)

U calculeaza un angajament **commit(U) = $\mathbf{sig_U}$ (V, C(U), $\mathbf{c_0}$, d, info)**
unde

V - identitate vanzator
C(U) - certificat payword al lui U
$\mathbf{c_0}$ **-** radacina lantului hash
d - data curenta
info - lunginea lantului (n), etc

U trimite angajamentul la V

- **U → V: commit(U)**
  - V verifica semnatura lui U pe commit(U) si semnatura lui B pe C(U), datele de expirare
  - Prin commit(U), V este asigurat ca U este autorizat sa realizeze plati, si toate payword-urile $\mathbf{c_i}$ pe care V le va receptiona de la U (pana la data **d**) le va rascumpara de la B

U trimite prima plata la V:

- **U → V: c_1**, 1
  - V verifica autenticitatea platii: daca **$h(c_1) = c_0$** accepta plata,altfel nu

...

In aceeasi maniera, U trimite a i-a plata la V:

- **U → V: c_i**, i

La sfarsitul zilei, V va rascumpara de la B payword-urile receptionate de la U

- **V → B: commit(U), c_l**, l

unde

c_l, l - ultima plata receptionata de V de la U

- B verifica commit(U), ultima plata (prin l aplicari ale functiei h)
- daca plata e autentica si nu a mai fost rascumparata intr-o sesiune anterioara, atunci B transfera l centi din contul lui U in contul lui V

Exercitiu:

Implementati schema de microplata Payword descrisa mai sus utilizand comunicatii client/server pentru a simula transmiterea mesajelor intre participanti. Implementarea trebuie sa permita plati pentru produse de valori diferite si sa verifice cazurile in care U incearca sa utilizeze aceleasi payword-uri pentru a cumpara mai multe produse de la V.