Micropayments and PKI

Vít Bukač

Outline

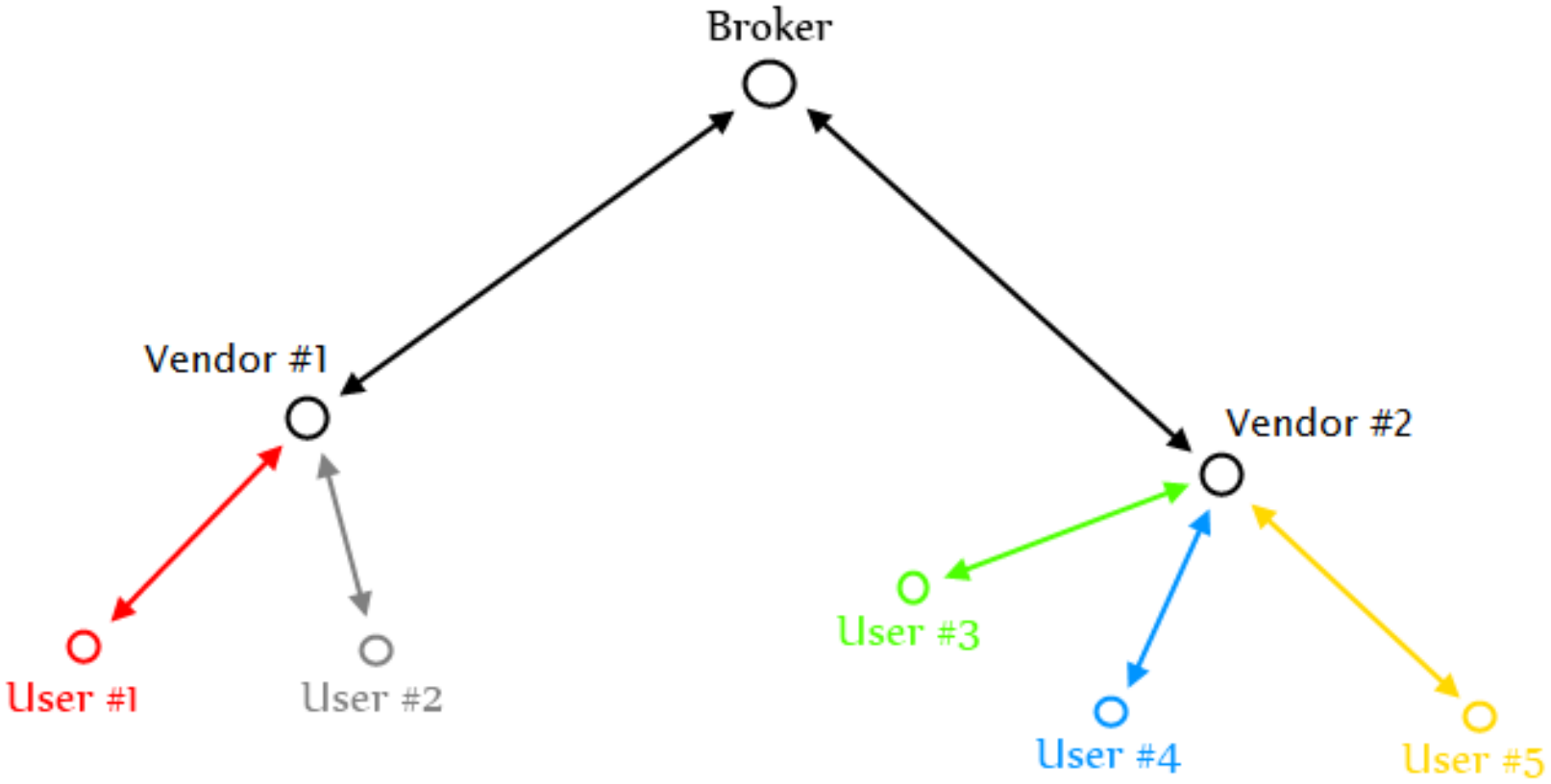
- Micropayments overview
- Blacklisting performance
- PKI collaboration
- Scheme modification(?)

Micropayments

- Electronic payment scheme suited for payments with a low nominal value

- Cheap
- Fast
- Anonymous
- Online/Offline

Micropayments scheme



Blacklisting

- We require maximum freshness of certificate status while keeping low cost (cost ~= bandwidth)
- Issuing of certificate revocation list (CRL) is a potential bottleneck
- In micropayments scheme vendor verifies user certificates but user does NOT verify vendors' certificates
  - Bytes transferred: (vendor_count)*(CRL_size)
- Each vendor downloads CRL only once in the validity period

Blacklisting

- user_count >> vendor_count
  - Vendors download CRL very soon after it was issued
  - <15 minutes by our estimates
- Classical CRL
  - Freshness achieved by keeping short validity period
- Delta CRL
  - High freshness
  - High user_count is a disadvantage
- Over-issued CRL
  - Balances load during validity period
  - Lower freshness but good bandwidth

PKI usage

- Public Key Infrastructure can hardly replace micropayments
  - Slower processing, twice the size of certificates
- PKI may be able to supplement micropayments with additional functionality
- It is standardized and verified by extensive use
  - “Why invent the wheel?”

PKI Usage

- Broker – user
  - User certificate blacklisting
  - Customer objections
- Broker – vendor
  - Secured transport of broker's public key to vendor
  - CRL authentication
- User registration (User possess a qualified certificate either on a chip card or as a file in his computer)

Key pair algorithm identifier

- User certificate (274 B)
  - ID_{User} (4 B), ID_{Broker} (4 B)
  - $PublicK_{User}$ (128 B)
  - issued_on (4 B), expires_on (4 B)
  - Limit (2 B)
  - CertSignature (128 B)
- Disadvantages
  - Card reader cannot recognize used key pair algorithm
  - When the algorithm is considered broken all certificates have to be revoked simultaneously
- Add algorithm identifier (0.5 – 1 B) to the certificate?
  - Certificates with the new algorithm can be issued while the old algorithm is still in use

Version identifier

- User certificate (274 B)
  - ID_{User} (4 B), ID_{Broker} (4 B)
  - $PublicK_{User}$ (128 B)
  - issued_on (4 B), expires_on (4 B)
  - Limit (2 B)
  - CertSignature (128 B)
- Disadvantages
  - Any modification to current system must be deployed globally and simultaneously
- Add version identifier (0.5 - 1 B)?
  - Version identifier can be tied with key pair algorithm identifier (new algorithm/different key length represent a new version)

?

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Thank you for your time!