Stacked Countersignatures Attributes Implementation
A new protocol to secure P2P transactions
And use for WEBpayments, PISP & AISP

Introduction to the SCAI protocol

Achieving the Digital Market is considered as one of the major goals of most countries in the world. One of the main issues to build the Digital Market is to secure transactions. So far, securing transactions is mainly done on a Master-slave approach with Web servers, belonging to corporate, and web client software. This approach brings results for Business to consumer interactions.

On the other hand, this approach is quite difficult in the Peer2Peer flows, corporate or retail. This document proposes an innovative concept to bring trust and confidence to Peer 2 Peer transactions. This concept could be used also to improve trust in Master-Slave transactions;

P2P Paradigm

This paradigm is: “how Alice and Bob could trust a transaction between them”.

First we consider a « transaction » as a some interactions between 2 parties (Alice and Bob) that are meaningful in a business perspective, for example:

- Asking for a form, signing the form, sending back and receiving the acknowledgement is ONE transaction
- Sending a payment request, validating it with a payment report and receiving it, is ONE transaction.

We could conceptualize a transaction as a request and a report. This model could be easily generalised when solved.

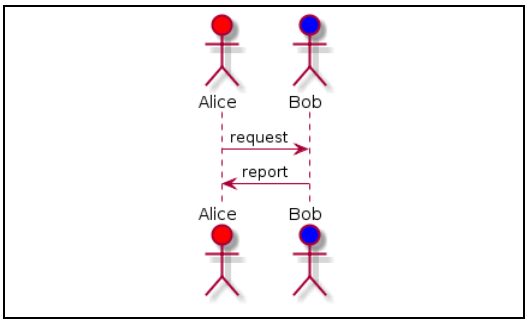


Image: CYV_SCAILIGHT-IMG-001.png

The existing web solution: “a shared PKI”

Description of “a shared PKI” solution

The solution today is a shared PKI between Bob and Alice:

- Bob and Alice should possess cryptographic keys (secret and public keys)
- The public keys should be certified by a unique PKI (Public Key Infrastructure)
- They could use existing protocols (TLS or SMIME) to send cerified request and certified report

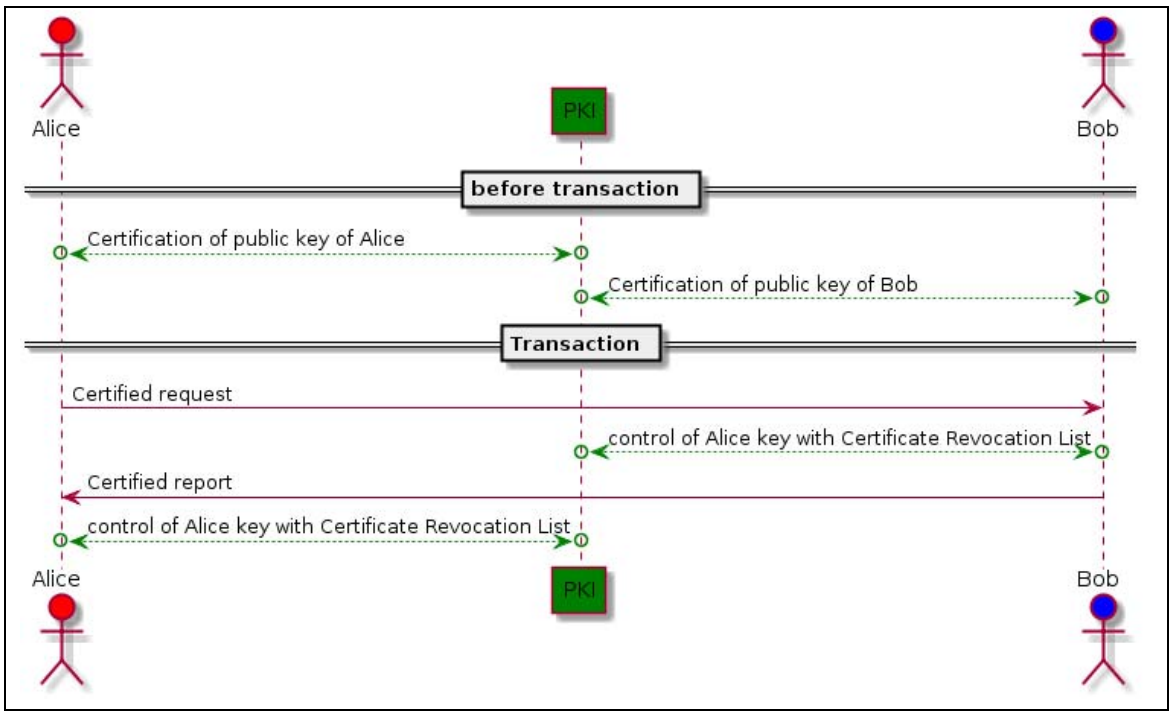


Image: CYV_SCAILIGHT-IMG-002.png

Advantages of “a shared PKI” solution

- Use of Public key of the counterparty enables to encipher the request (or the report) in order to provide also confidentiality
- All the protocols exist to perform this type of solution
- A lot of software (proprietary or open source) exist for corporate or for end users
- A lot of communities are deployed mainly at country level or corporate community level around a business (lawyers, banks, notaries, Telcos, electronic trust parties, government[1],...)

[1] E-IDAS is a directive to organise this type of network between European nations

Weakness of this solution

- ❑ This solution often needs that the PKI is dedicated to a business process in order to have a detailed PKI policy:
  - PKI for signing contracts
  - PKI for exchanging money
- ❑ As a matter of fact, deploying such PKI on a worldwide level has not been feasible so far. And almost all businesses on the web are worldwide.
- ❑ Protocols do not provide links between the request and the report. Access to log of both Bob and Alice are necessary in case of disputes
- ❑ Bob and Alice should controls the CRL of the PKI for each transaction
- ❑ As the certification process is quite cumbersome, the key should be manage carefully in a long term view

New complementary approach: SCAI

The proposed solution leverage on:

- ❑ Existing actors that are organised with a shared PKI,
- ❑ Easy creation of cryptographic keys at the Peers level
- ❑ A new protocol that define:
  - A workflow to secure keys of the Peers
  - A stack of messages to intricate the request and the report
- ❑ A clear separation between the transaction and the authentication processes needed during the transaction

Each of this capabilities are described in the following chapters.

Existing actors shared a PKI

A PKI between “parent” actors should exist. This PKI forms a network of trust”.

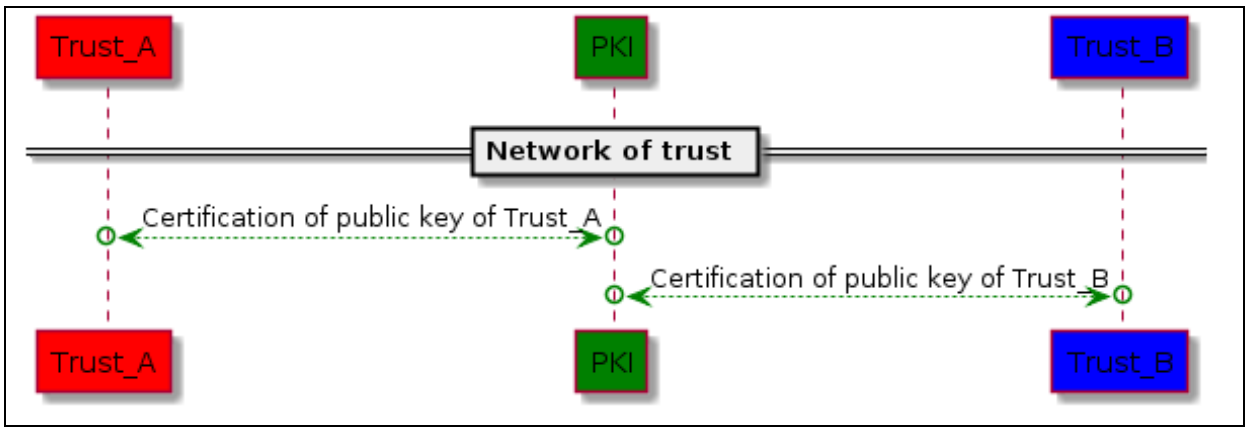


Image: CYV_SCAILIGHT-IMG-003.png

Cryptographic keys at Peers level

Using existing components, each “Peer” actors should create Public and Private Keys. They do not need to be certified (that is creating a certificate under a PKI with the Public Key). Nevertheless, it could work if a Peer is using a certified key.

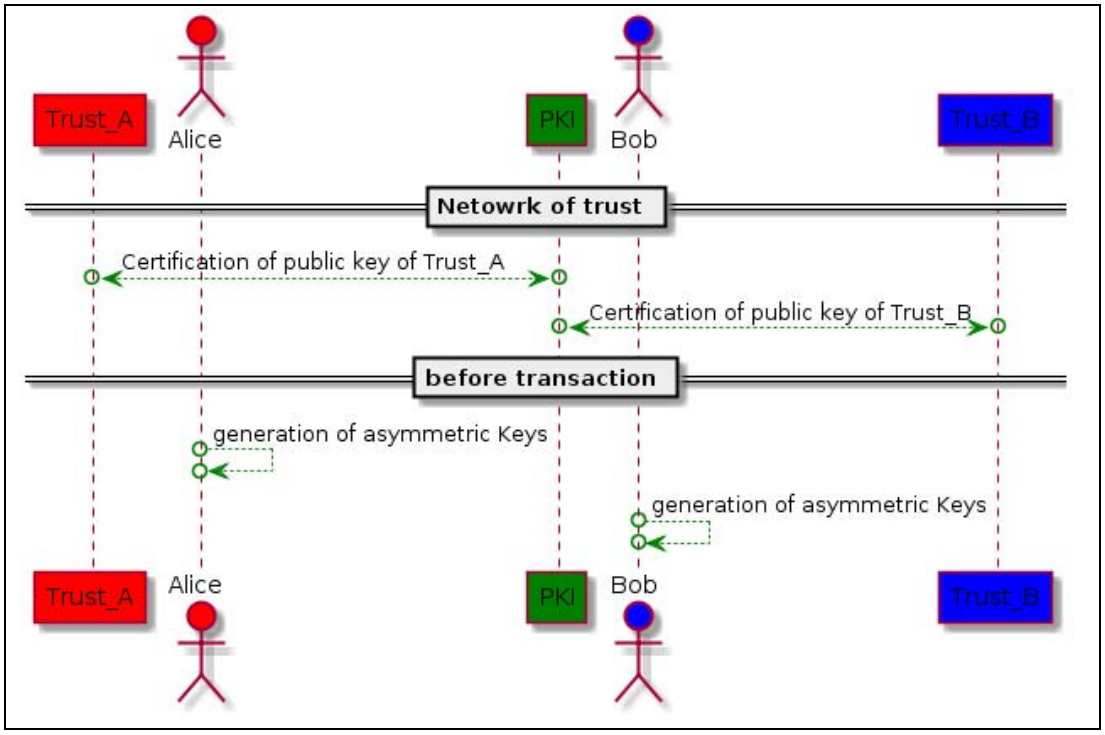


Image: CYV_SCAILIGHT-IMG-004.png

A workflow

The workflow of SCAI is designed to incrementally secure the request and the report in order to have a full secure transaction:

- The flow begins at Alice side
- The flow ends at Alice side

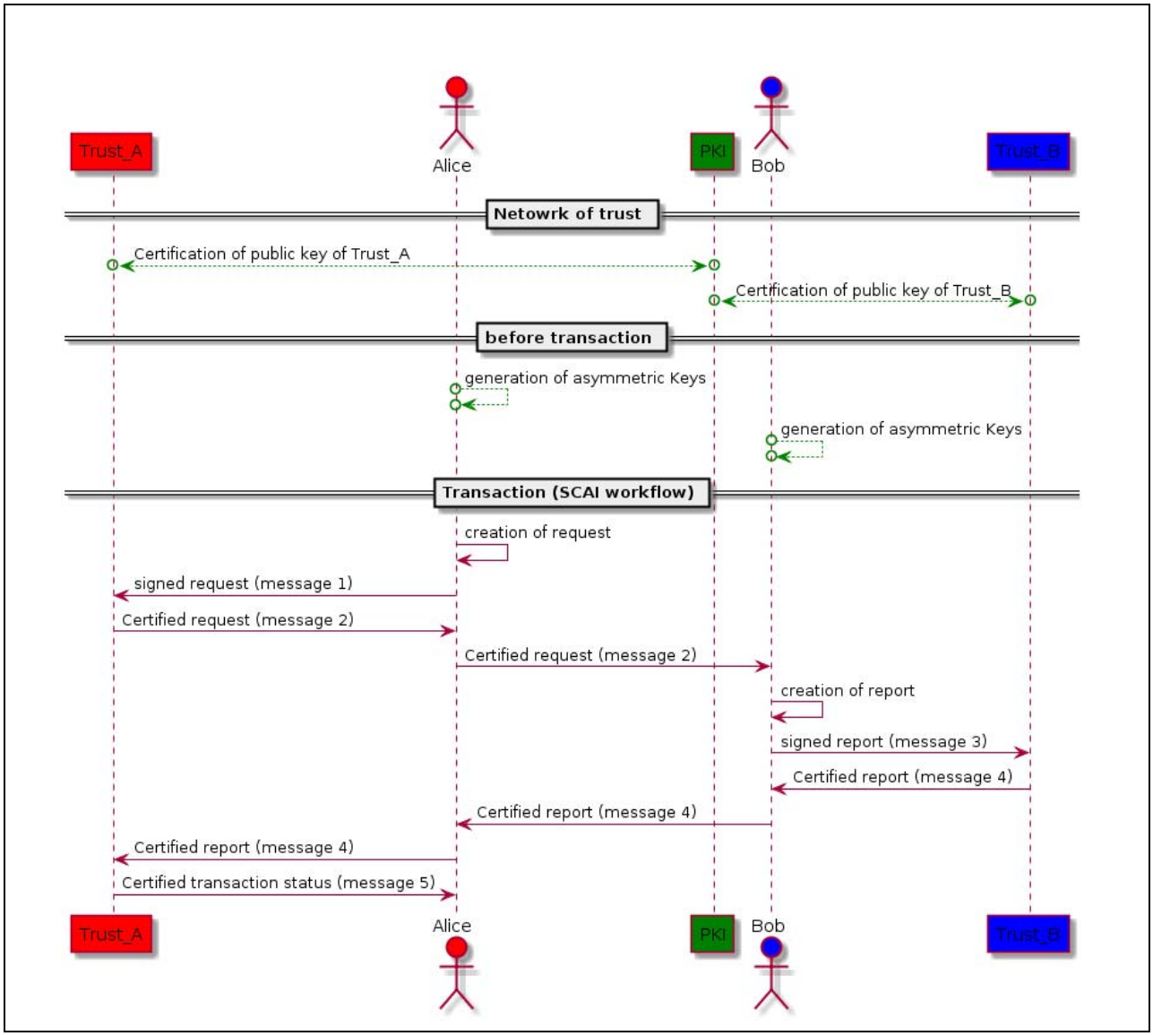


Image: CYV_SCAILIGHT-IMG-005.png

- ❑ Message 1 is the signed request
  - "Signed" because Alice cannot certify the request. She had only a simple cryptographic key
  - Signed in order to formalise to consent from Alice to deliver this request
- ❑ Message 2 is the certified request
  - The Trust Party of Alice will certified the request based on existing relationship with Alice
  - The label "certify" comes from the Key used by the Trust Party which belongs to a PKI
  - Message 2 is mainly a certification of message 1 but, depending on the business, could put additional data
- ❑ Message 3: signed report
  - Bob is receiving message 2 and should prepare his answer : the report based on the data of the request
  - At this stage Bob is not sure that nor Alice, nor Trust_A are ok

- Message 4: certified report
  - When receiving the Message 3, Trust Party of Bob should verify the correctness ot the Trust_A with the PKI
  - Then Trust Party (Trus_B) of Bob could add extra data if necessary and certified the report
  - Bob is now sure of Alice
- Message 5: the certified transaction status
  - When receiving the message 4, Trust_A will controls the PKI to be sure of Bob by transitivity
  - Alice is now sure of Bob

The SCAI line

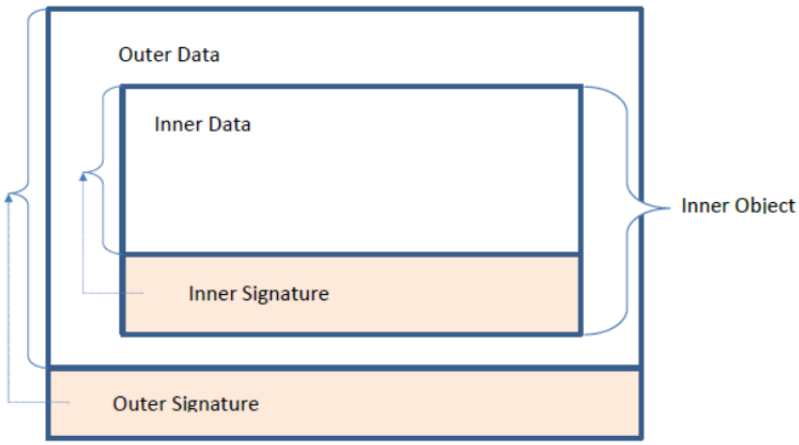
A Stack of messages

In order to be able to secure the full transaction (and not only the messages), the SCAI protocol oblige that all messages should be stacked:

```
{
"Certified Transaction Status (message 5)": {
  "data of message 5": "data",
        “Certified Report (message 4)”:{
              “data of message 4”: “data”,
              “Signed Report (message 3)”:{
                    “data of message 3”:”data”
                    “certified request (message 2)”: {
                          “data of message 2”:”data”,
                          “Signed request (message 1)”: {
                                “data of message 1”: “data”,
                                “Cryptographic signature of message
                          1”:”HEX”}
                          “Cryptographic signature of message 2”:
                    “HEX”}
                    “Cryptographic signature of message 3” : “HEX”}
               “Cryptographic signature of message 4”: “HEX”,}
  “Cryptographic signature of message 5”: “HEX”,}
}
```

- Using this format, the full transaction is build step by step and could not be tampered during the flows.
- Furthermore, at each step of the transaction only new data should be add, without needing to copy all previous data.

The signature process of the stack



- Inner Data represents data used in a phase of a multi-step message/transaction.
- Inner Signature is a Public Key signature for Inner Data.
- Outer Data represents specific data for a succeeding phase of the multi-step message/transaction.
- Outer Signature is a Public Key signature for Outer Data + Inner Object. That is, Inner Object is effectively countersigned.
- Possible additional phases would follow the same pattern.
- The net result is a self-contained and secured message/transaction object.

Note: The exact format of the signature is yet to be determined.

A clear separation with authentication

- In order to change as needed authentication methods, those are done outside of the SCAI protocol.
- Nevertheless, the level of the authentication method (weak, substantial, strong, for example) should be given as a data in the message in order for Alice and Bob to adapt their behaviour to the quality of authentication.

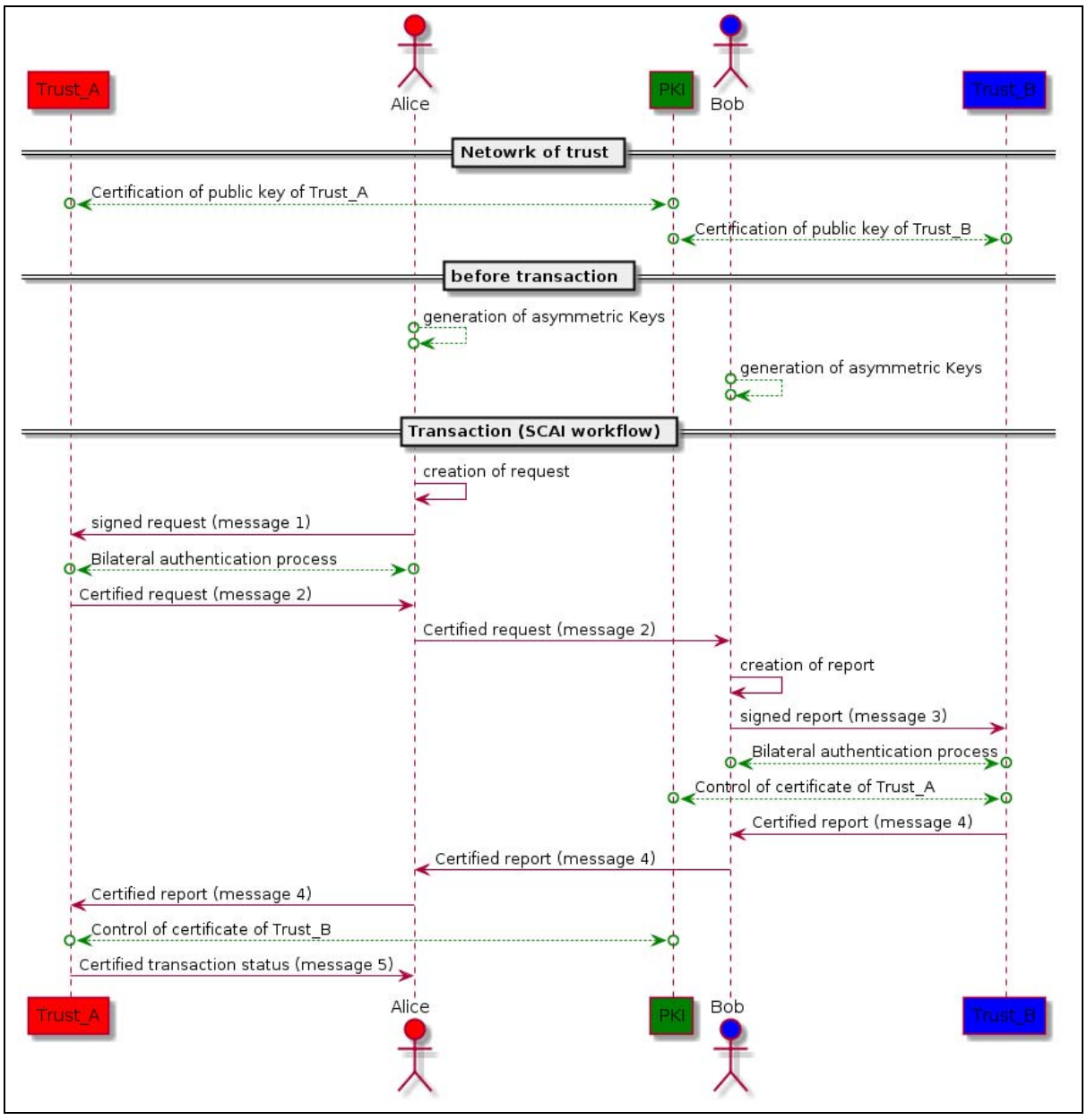


Image: CYV_SCAILIGHT-IMG-006.png

Tunnelling with SCAI

- Another benefit of SCAI is the possibility to send encrypted data through the transaction.

```
{
"Certified Transaction Status (message 5)": {
  "data of message 5": "data",
          "Certified Report (message 4)":{
                  "data of message 4": "data",
                  "Signed Report (message 3)":{
                          "data of message 3":"data"
                          "encrypted data with Public Key of Trust_A": "HEX"
                          "certified request (message 2)": {
                                  "data of message 2":"data",
                                  "Signed request (message 1)": {
                                          "data of message 1": "data",
                                          "Cryptographic signature of message 1":"HEX"}
                                  "Cryptographic signature of message 2": "HEX"}
                          "Cryptographic signature of message 3" : "HEX"}
                  "Cryptographic signature of message 4": "HEX",}
  "Cryptographic signature of message 5": "HEX",}
}
```

- Leveraging the public key used by Trust-A to sign message 2, Trust-B could encrypt data with this key. Only Trust_A will be able to decipher the data.
- This is a simple way of tokenisation. It will be used for web payment implementation of SCAI.

Follow-up or connecting to other business process

During the transaction, the Key of Alice (resp. the Key of Bob) has been controlled by the Trust-Company.
Alice could use this key to sign a new request and depending of bilateral conditions, this signature could be used as an easy authentication method of Alice.

Confidentiality

SCAI protocol does not aim to provide confidentiality of exchange. Existing protocol like TLS should be used bilaterally, that is point to point, in order to provide confidentiality on top of SCAI.

Advantages of using SCAI protocol

- Foster secure transaction without requisite a PKI to be deployed to ALL actors
- Integrity of transaction that could not be tampered
- Simplify transaction handling when disputes (no need to ask for logs)
- Could be extended to larger transaction (more than 5 messages)
- Easy tokenisation of sensible data
- Explicit audit trail of successive messages
- Transparency for all actors, except for some secret data (tunnelling)
- SCAI could be extended to more than four actors
- SCAI could comply with privacy needs by offering tunnelling capabilities and using session key at peer level
- SCAI is neutral versus the data transported and so could be used in a wide range of areas (payment, initiation, purchase, information)
- 2 peers using the same Trust_Compagny could use SCAI to be sure of equal treatment
- Key of Peer, used during the transaction could be also used later for authentication purpose in order to get updates of the transaction or to link with other business process.

Applying the SCAI trust model to payment related use cases

The SCAI protocol could help any type of business. In this chapter we will detailed 3 uses cases which could benefits from the SCAI trust model:

- **Web payment** in order to secure the check-out phase and to bring trust to:
  - the “consent to deliver” given by the Merchant to the Buyer
  - the “consent to pay” givent by the Buyer to the Merchant
- **PIPS**, Payment Intiation Service Providers, which provide service for Merchants and Buyer using Credit Transfer payment. SCAI will enable PUSH payment initiate by a PISP with the same architecture as proposed for the Web Payment.
- **AISP**, account information Service provider which provide aggregation services on account data. SCAI could propose a safe method to:
  - Formalise the « consent to access to account » from the customer to the AISP
  - Provide securely unique cryptographic credential that will be use later by the AISP to connect to the Bank API

Leveraging SCAI for the WEB Payment

Actually, SCAI will be used for the check-out phase, that is when the client is validating the e-cart.

In the proposed process:

- The generic “request” becomes a “Payment Request”
- The generic report becomes a “Payment Report”
- The generic Transaction Status becomes a “Payment Status”

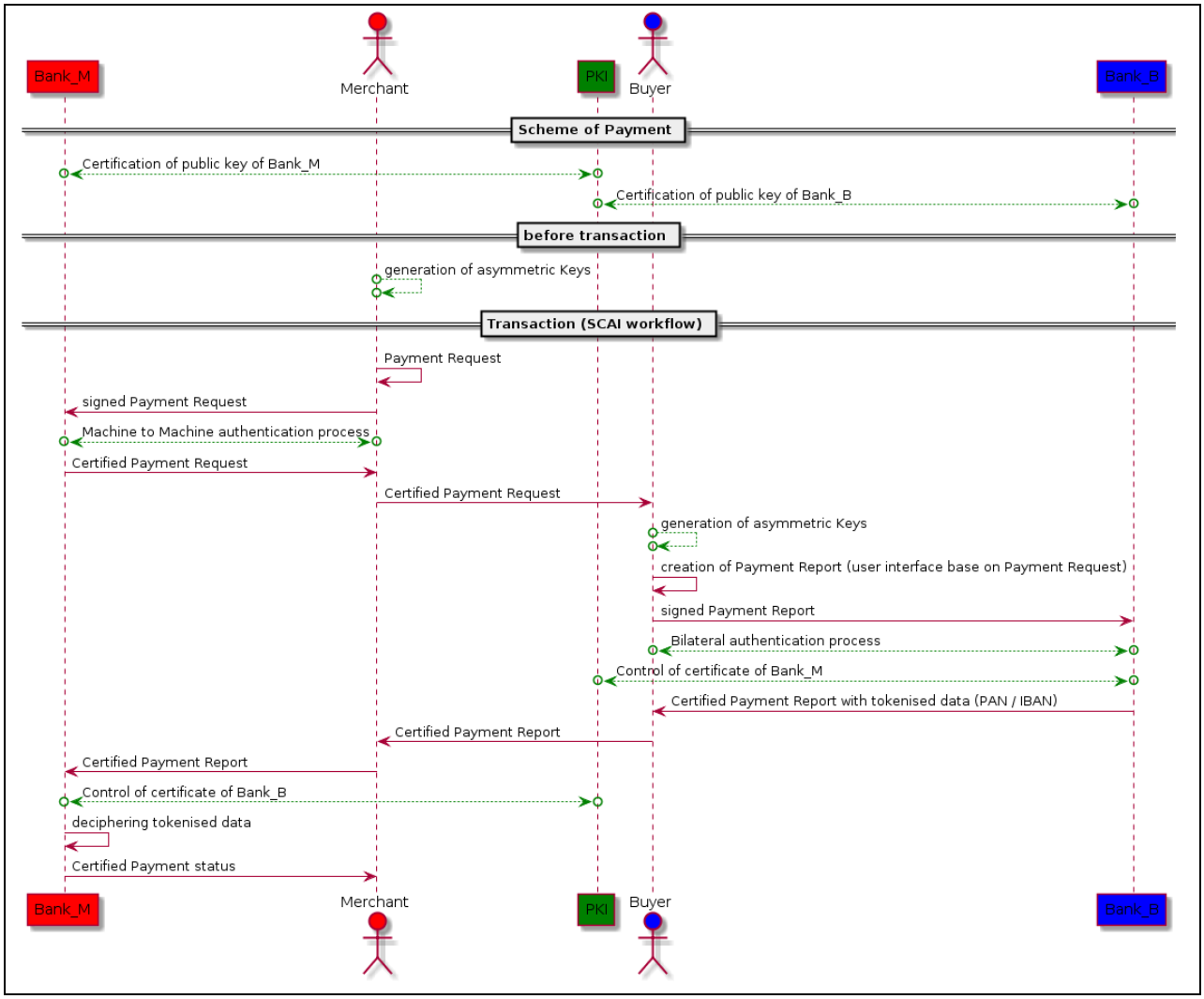


Image: CYV_SCAILIGHT-IMG-007.png

Advantages

- ❑ Only PKI through Banks is requisite
- ❑ Use of SCAI enables to secure transaction for card with
  - A formal consent of the user (certified signature of Payment report)
  - A tokenised PAN (through tunnelling) that could be only deciphered by Bank_M
- ❑ Use of SCAI enables to secure transaction for Direct Debit payment with
  - A formal consent of the user (certified signature of Payment report) which constitute an e-mandate certified by the Bank of the debtor
  - A tokenised IBAN (through tunnelling) that could be only deciphered by Bank_M in order to respect some countries requirements on the IBAN
- ❑ Use of SCAI enables to secure PUSH transaction (credit transfer) with:
  - A formal consent of the user (certified signature of Payment report)
  - A formal indication of the Bank of the Buyer of the status of the credit transfer

Leveraging SCAI for PSD2 new actors

Use of SCAI for a secure operational model for PISP

The use of SCAI for the PISP is exactly the same as for PUSH web payment. The PISP replace the Bank of the merchant. This one remains in the field when receiving funds. As the PISP receives a certified payment report, it could bring value to the merchant side.

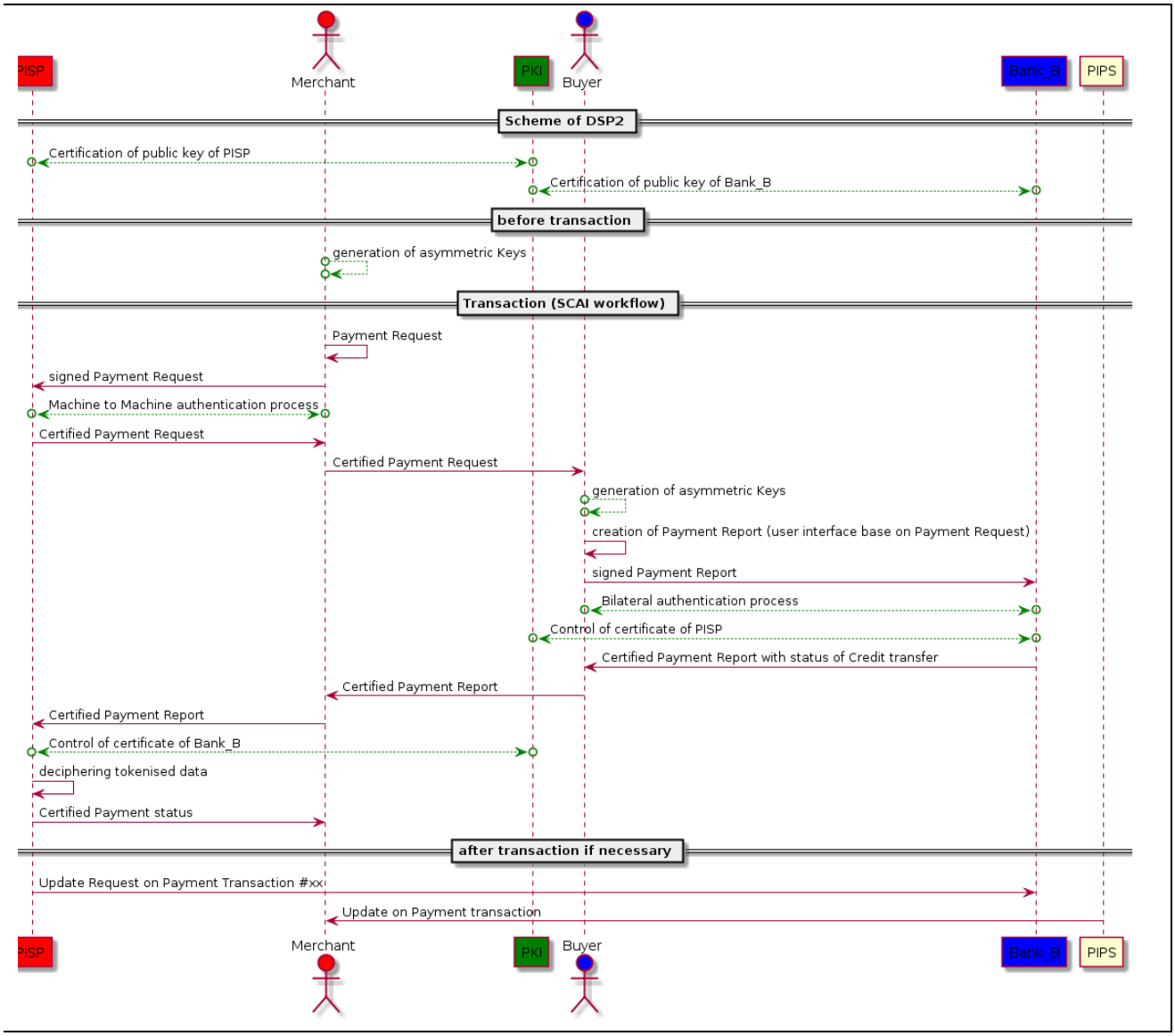


Image: CYV_SCAILIGHT-IMG-008.png

Furthermore, the SCAI protocol enables also to easily manage exemption, as for example:

- ❑ No Buyer authentication if the Key of buyer already known and amount low
- ❑ Low level of risks

Those exemptions (required or not by laws) are managed at the Bank of the Buyer level, where the risk is at its maximum.

Use of SCAI to secure transmits unique, tokenised credentials to AISP

Use of SCAI should provide standard procedure for:

- Validating the consent of the customer to give an access on his account
- Transmitting unique credentials to the AISP. As all the exchange are based on cryptographic standards, we believe that credentials should be Private and Public Key.

- ❑ For the SCAI perspective, the AISP will behave as a Trust Party (registered in the PKI) and the initiator of the "access to account request".
- ❑ The Certified "access request" will be followed by a user interface enabling to type-in all the account (IBAN) involved
- ❑ Thus, as many reports will be generated by forking the certified request
  - At the end of the process, as many SCAI line as IABN will exists
- ❑ The Signed reports will be certified by the different banks using their own authentication method
- ❑ The Certified report WILL mandatory contain a private/public keys generated specifically for the AISP to connect later to get account information. Those keys will be tokenised by tunnelling to the AISP
- ❑ Use of cyrpotographic credentials enables to guarantee no disputes in further authentication by the AISP during the life cycle: changing of AISP, stopping the services, …

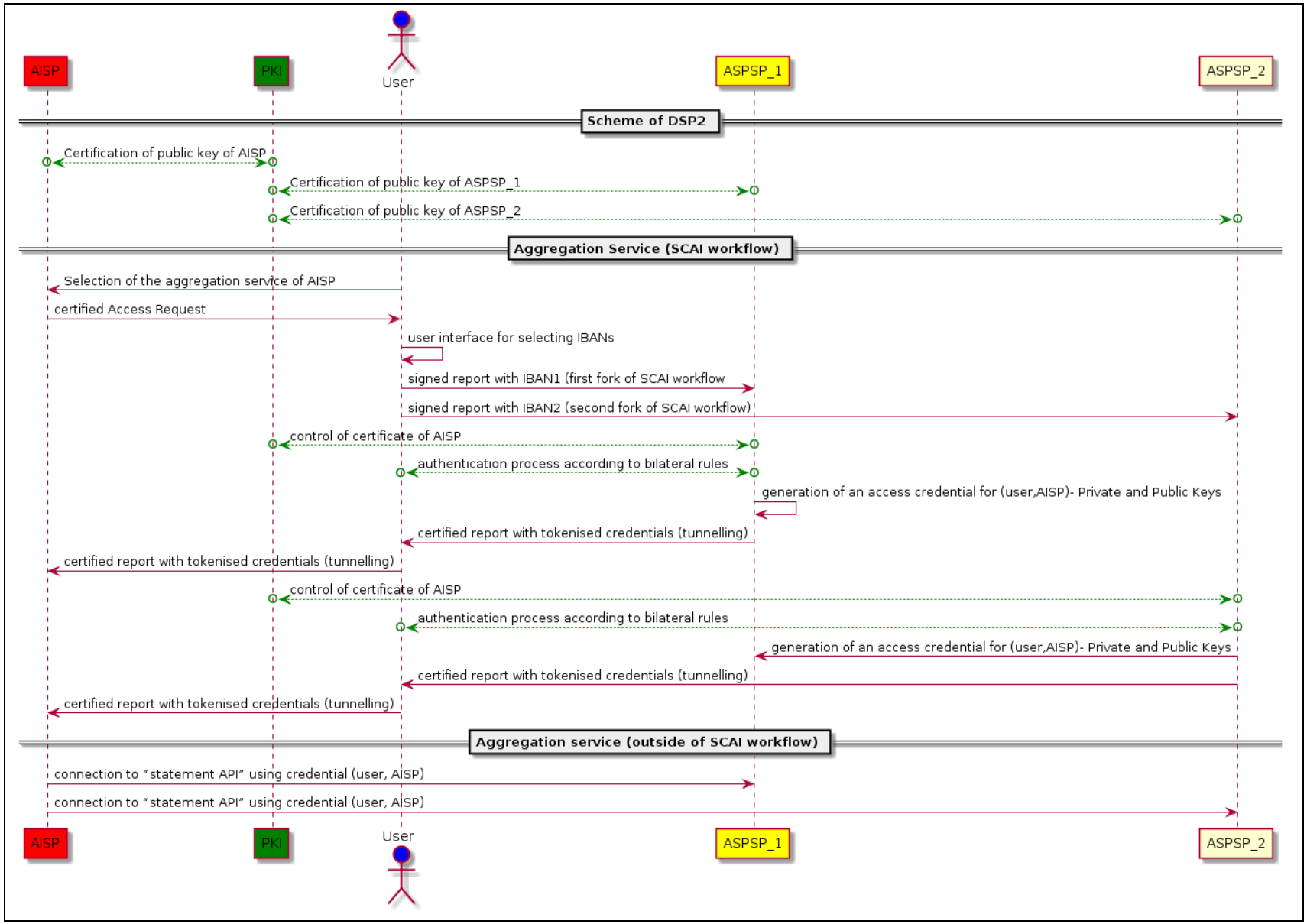


Image: CYV_SCAILIGHT-IMG-009.png