A new Payment Protocol over the Internet

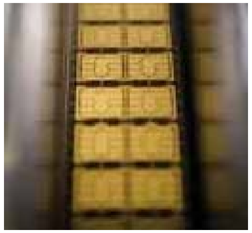
SSD Team

A join work with Aude Plateaux (University of Limoges),
Pierre Girard & Karine Villegas (Gemalto)

Jean-Louis Lanet

Jean-louis.lanet@unilim.fr

Université de Limoges | FACULTÉ DES SCIENCES ET TECHNIQUES



Agenda

- Problem description,
- General architecture,
- Security of the protocol,
- Smart Card implementation,
- Conclusion.

Current payment architecture

- Existing architecture:
  - The merchant transfers the customer to the bank for the payment,
  - The bank asks for the cardholder name, card number and expiration date, and the transaction date,
  - The bank informs the merchant that the operation succeeded and the merchant can deliver the product,
  - If later the cardholder repudiates the transaction, due to the fact it has not been PIN protected the transaction is cancelled.
- Such a protocol is widely accepted, simple and efficient, protected with SSL for integrity and confidentiality,
- There are no guarantee that the customer is the owner of the card.

Alternative solutions

- Modifying the protocol,
  - Secure Electronic Transaction (1996), uses certificates and signatures for authentication, integrity and confidentiality. Not a success, too complex,
  - 3D Secure, (2001) provides mutual authentication currently low acceptance,
    - Weak authentication with birthday date,
    - Using OTP, some banks propose dongle that generates One Time Password,
- Expected properties
  - Integrity, Authentication and non-repudiation.
  - Confidentiality (to protect card number and cardholder name)

The idea

- Using the same architecture (use of *https* protocol) provides to the bank the customer authentication,
  - Provide a proof of transaction authenticity by a smartcard,
  - A PIN code is used to authenticate the *user* to the smartcard.
- The bank and the card share a secret key stored during personalisation phase in the card,
- We send to the bank the same information as previously,
- We split the transaction in several sub-transaction.

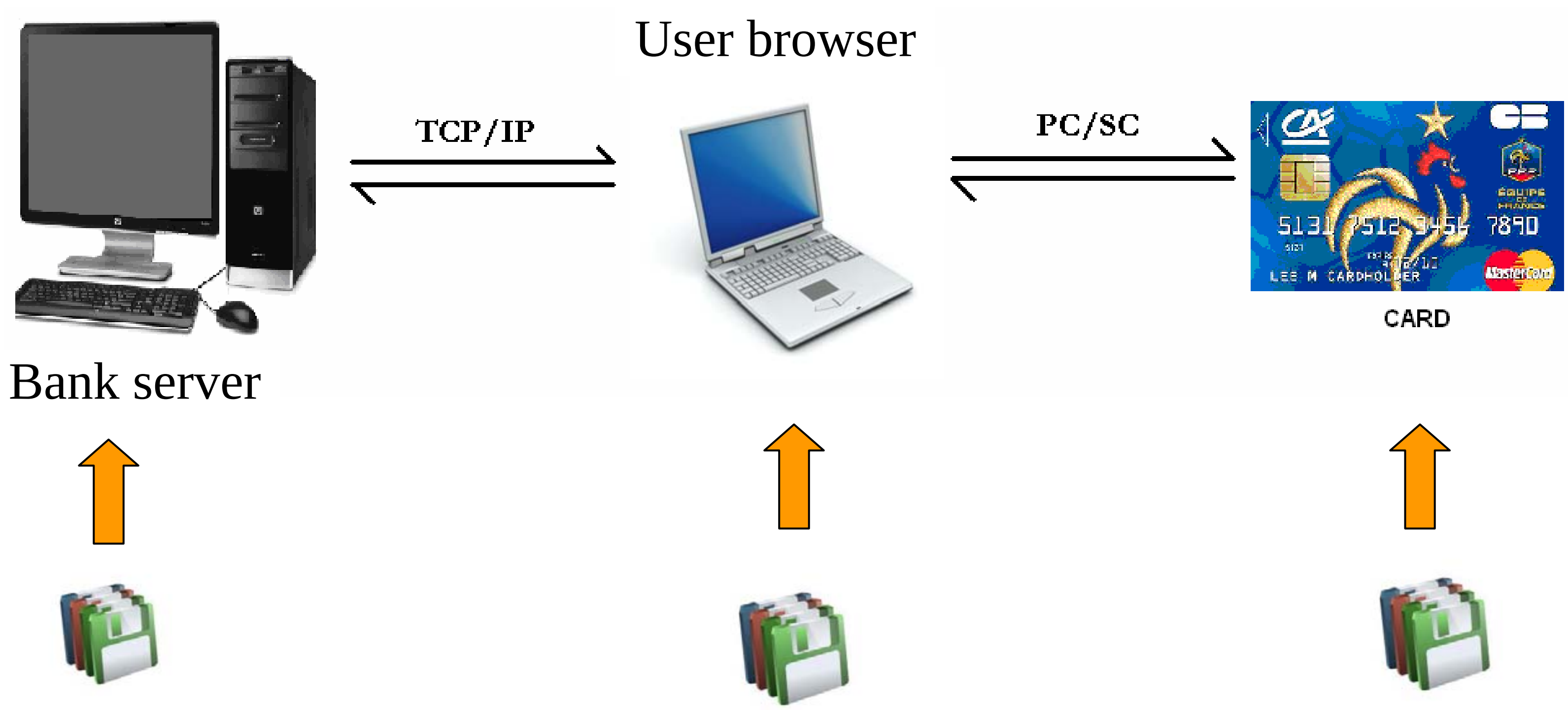
Protocol Design

- **Idea:** divide the amount into several sub-amounts,
- **Authenticity:** sub-amounts are function of a secret shared with *bank* (which verifies the correct cutting),
- When the bank receives all the sub-amounts, only the card that belongs to the cardholder name sent, can generates this split,
- Number of sub-amounts is adjusted to the optimal security-performance compromise.

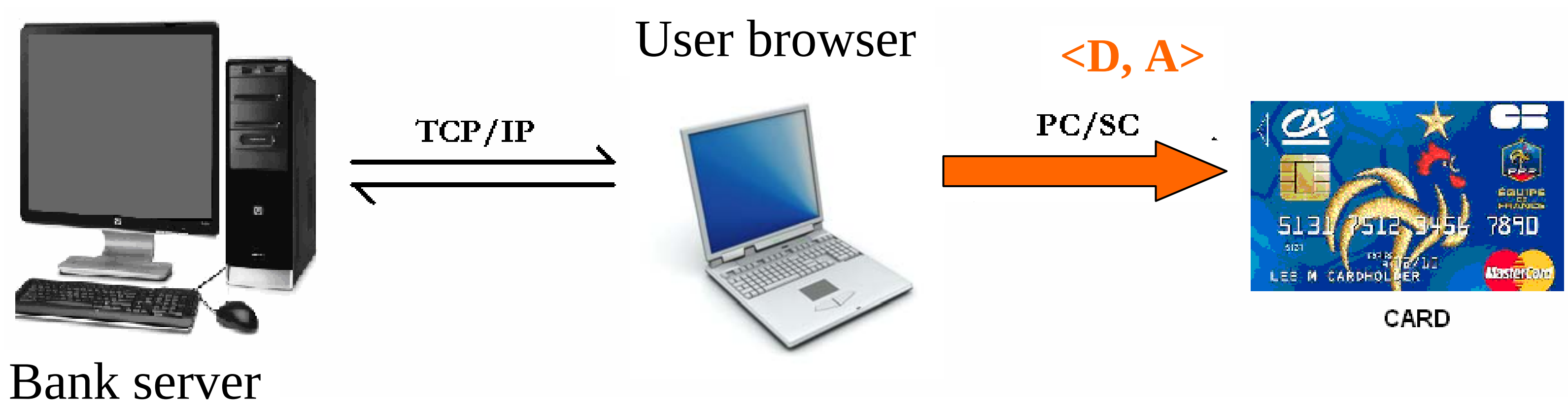
Example

- The *user* wants to pay ***100,00€*** $\Rightarrow$ amount is sent
- The *terminal* returns date and time of transaction
- The algorithm gives 3 sub-amounts ***20,45€* | *42,42€* | *37,13€***
- The *terminal* transmits information to the *bank*
- The *bank* is able to verify the right cutting of ***100,00€*** thanks to:
  - shared secret key
  - 3 sub-amounts
  - the date and time
- The transaction is **authenticate**

The architecture



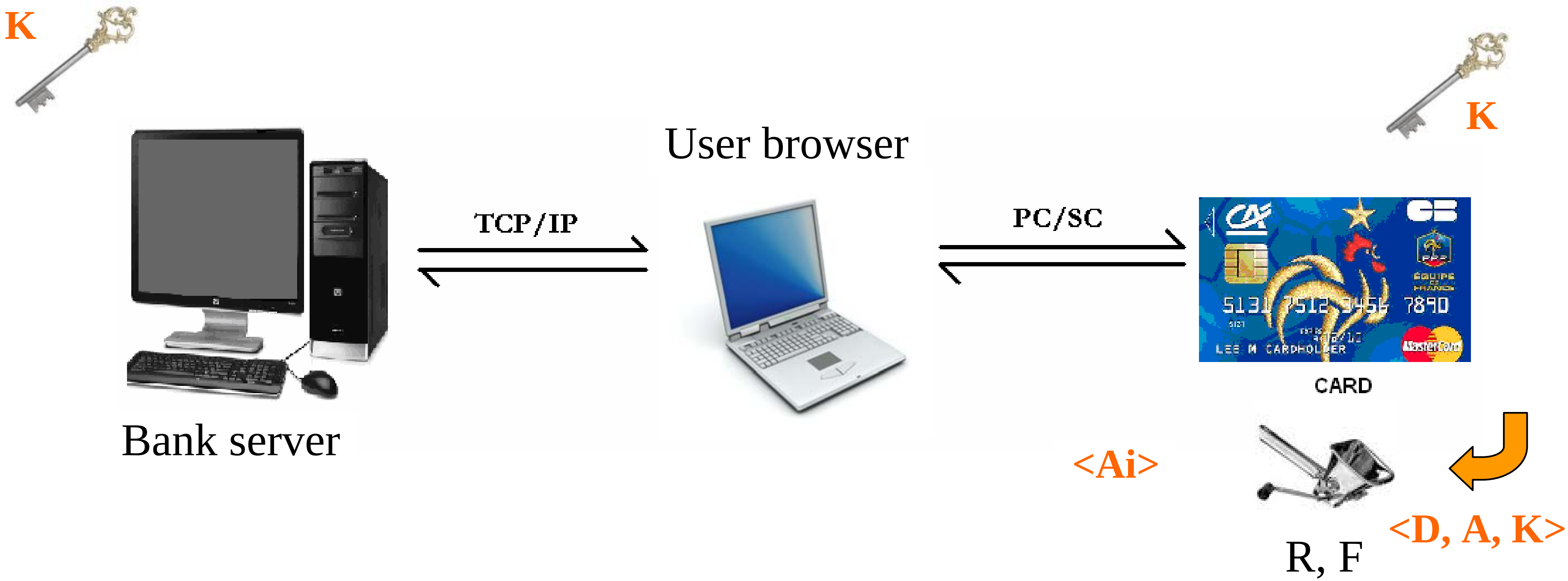
The architecture



- The merchant send to the user the amount A to be paid,
- The user sends to the card the date and time (denoted *D*) and the transaction amount (denoted *A*).

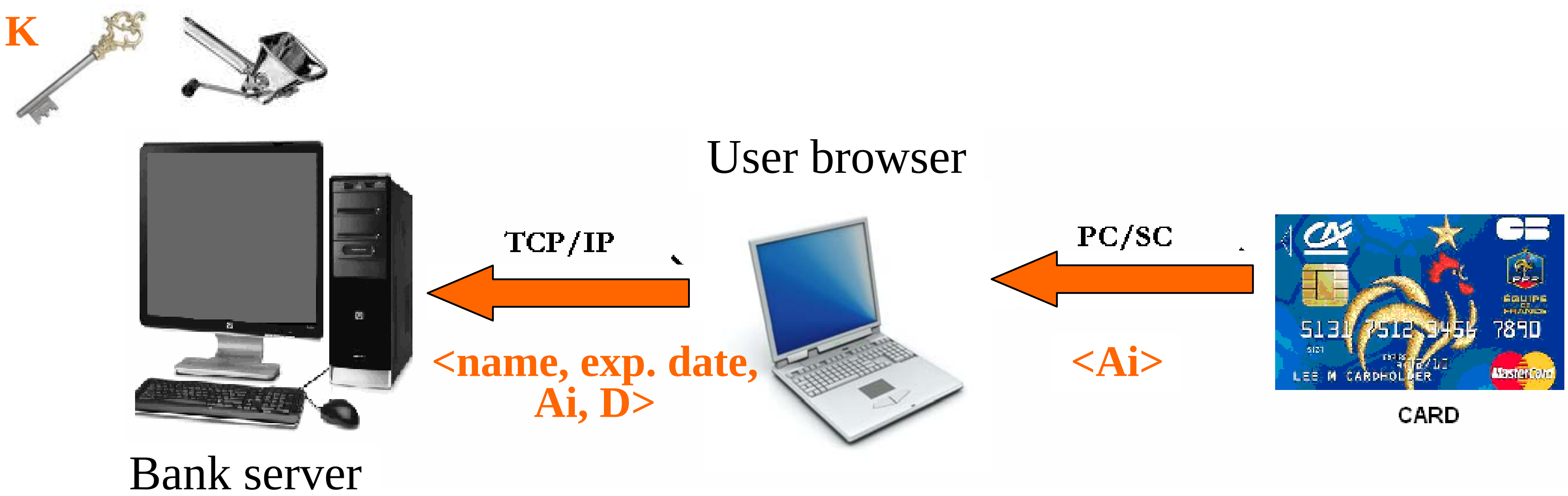
Université de Limoges | FACULTÉ DES SCIENCES ET TECHNIQUES

The architecture



- The card stores a secret key ***K***, shared with the bank.
- The card uses a function *R* (taking in argument *D*, *A* and ***K***) to obtain a pseudo-random bit stream *S*.
- The card uses a function *F* (taking *S* and *A* as arguments) to produce *n* sub-amounts *Ai*.

The architecture



- The n transactions of Ai amounts with the same date and time are sent to the payment system,
- For a given card, the bank groups the n transactions with the same date and time D and computes the sum A of the different amounts Ai.
- The bank retrieves the secret key **K** of the card and computes $S = R(D, K, A)$, *and* computes $A'i = F(S, A)$.
- The bank verifies that for each i, $Ai = A'i$, if yes the n transactions are authenticated

The R function

- Type: Hash function
- Objective: Return a pseudo-random bit stream S for computing A_i
- Principe: S $\leftarrow$ HMAC-SHA2 (D||A, K)
  - In this case we obtain up to 512 bits of pseudo-random source S_0,
  - We can generate more random sources S_i by computing $S_i = R(S_{i-1}, K)$.

The F function

- Objective:
  - Compute n sub-amounts Ai,
  - Send one sub-amount over as the card produces one (no storage !).
- Principe: Find its high-order and low-order bit in S

  $\rightarrow$ Collect the result of successive divisions and modular reductions of *S*.
- Useful parameters: S, A, n and k

  where k the smallest integer such as : $2^k = \frac{A}{n} + \varepsilon$

The F function

- Objective:
  - Compute *n* sub-amounts A_i,
  - Send one sub-amount over as the card produces one (no storage !).

$$F:(S,A)\mapsto\begin{cases} A_1 = S \bmod 2^k, \\ A_2 = \left\lfloor \frac{S}{2^k} \right\rfloor \bmod 2^k \\ \cdots \\ A_{n-1} = \left\lfloor \frac{S}{2^{(n-2)k}} \right\rfloor \bmod 2^k \\ A_n = A - \sum_{1}^{n-1} A_i \text{, such that } A_n \in \left[\frac{A}{n} - (n-1).\varepsilon, A \right] \end{cases}$$

Example

- The user is willing to pay 100,00 €.
- We will take A = 10 000 and n = 10
- Compute k :
  - A/n = $2^k + \varepsilon$ = 1000
  - 2^{10} = 1024 so k=10 and ε = 24
- Compute $A_1..A_9$
  - $\forall i, i \in [1,9], A_i \in [0;1024], A_i = \left[\frac{S}{2^{10(i-1)}}\right] \bmod 2^{10}$
- The last sub amount: $A_{10} \in [784;10\ 000]$

Université de Limoges | FACULTÉ DES SCIENCES ET TECHNIQUES

Algorithm’s security

- The challenge for the attacker is to find a sub amount collection that could be generated with a genuine card.
  - the probability *P* for finding the right splitting is

$$\frac{1}{\left(\frac{A}{n}+\varepsilon+1\right)^{n-1}}$$

  - Then for an amount A = 100,00 €
  - The number of sub amounts n=10.
  - The probability is then 2^{-90}

Algorithm’s security

- The split for a given amount is unique for a given date which is part of the hash function data.
- Hence the date and time should be precise up to the second.
- In case of replay:
  - The replay occurs while the bank has not finished to check the transaction, the transaction is cancelled,
  - The replay occurs after the validation of the transaction, the original one is validated, the new one is rejected;

Implementation consideration

- The smart card applet has been developed on two different platforms: a dot net card and a Java Card
  - The first step was the user authentication with a PIN code,
  - the HMAC SHA 2 was not present on the Java Card 2.2 (need to use a more recent card), we simulate with

$$\mathrm{HMAC}_K(m) = h\Big((K \oplus \mathrm{opad}) \| h\big((K \oplus \mathrm{ipad}) \| m\big)\Big)$$

  - SHA 2 is part of the API on a dot net card, applet was simple to develop
- The complete prototype including the bank side has been developed.

Conclusion

- We solved the Authentication problem within an internet transaction, using a smart card we add non repudiation,
- New payment protocol splits the transaction into several sub-transactions,
- Verification of the authenticity of its transactions, and correctness of the protocol.
- Java Card code and report are available at : **`http://www.msi.unilim.fr/~lanet/`**

Any question ?