Blind Signatures

GIANLUCA DINI
Dept. of Ingegneria dell'Informazione
University of Pisa
email: gianluca.dini@unipi.it
Version: 07/04/2025

1

Digital signatures | Blind signatures

INTRODUCTORY CONCEPTS

2

Blind signatures

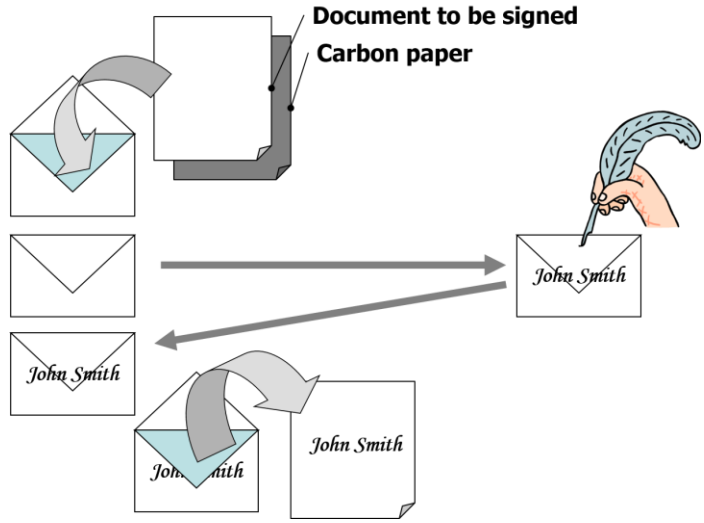
- Intuition
  - In a blind signature scheme, the signer can't see what it is signing
- Unlinkabiliy
  - The signer is not able to link the signature to the act of signing

Signer cannot repudiate signature but cannot say "I signed this at that exact moment"

Apr-25 Digital signatures 3

3

The metaphor



Apr-25 Digital signatures 4

4

Digital signatures | Blind signatures

RSA-BASED BLIND SIGNATURES

Trick is using homomorphic property.

5

Blind signatures →

- The protocol
  1. Alice
     a) Randomly chooses b s.t. gcd(b, n) = 1
     b) Computes $x' \equiv x \cdot b^e \pmod n$
     c) Sends x' to Bob (signer)
  2. Bob
     a) Receive x'
     b) Compute $s' \equiv (x')^d \pmod n$
     c) Returns s' Alice

- X = message to be signed
  X' = postcard containing message and carbon copy.
  e : public key of signer

6

Blind signatures →

- The protocol
  3. Alice
     a) Receive s'
     b) Compute s, the digital signature of x, $s \equiv s' \cdot b^{-1} \pmod{n}$
- Proof
  - $s' \cdot b^{-1} \equiv (x')^d \cdot b^{-1} \equiv (x \cdot b^e)^d \cdot b^{-1} \equiv x^d \cdot b^{ed} \cdot b^{-1} \equiv$
    $\equiv x^d \cdot b \cdot b^{-1} \equiv x^d \equiv s \bmod n$ QED

You can do something similar w/ El Gamal

Apr-25 Digital signatures 7

7

Applications

- Privacy related applications
  - Digital cash
    - Chaum, David. "Blind Signatures for Untraceable Payments. Advances in Cryptology. 1983.
  - Electronic voting
    - G. Dini. A secure and available electronic voting service for a large-scale distributed system. Future Generation Computer Systems. Volume 19, Issue 1, January 2003.
  - Pseudonims

Apr-25 Digital signatures 8

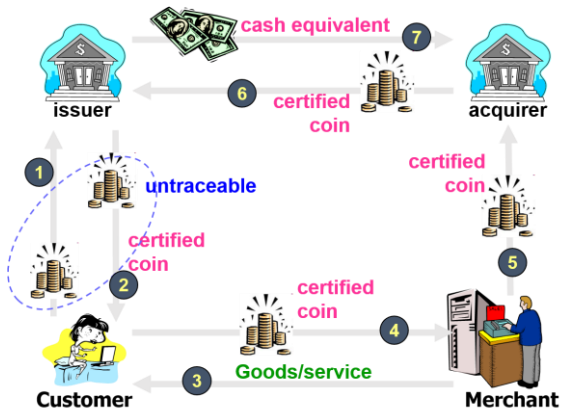
8

Blind Digital Signatures | Applications

DIGITAL CASH

9

Digital cash



- coin: a random number
- coin·b^e: blinded coin
- coin, $coin^d$: certified coin
- $d_{10€}$: a 10€ worth bank's private key

Customer vs Merchant. Issuer: bank of customer. Acquirer: bank of merchant.
Protocol: merchant obtains some proof of payment, goes to acquirer, that then requires money from issuer.

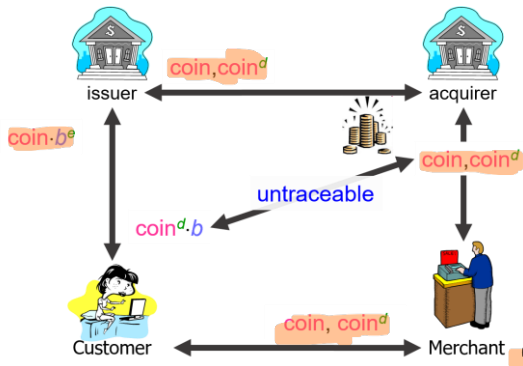
10

1. Customer sends bits to issuer (COIN money) and issuer gives value to those bits. Issuer digitally signs bits corresponding to amount you need (and then debits your bank account by 10 euros). I obtain bits whose value is 10 euros because of digital signature. I send them to merchant that provides goods and sed certified coins to the acquirer, that sends this to issuer which replies with real money.

Sequence of bits in (1) must be unique. Plus, if issuer didn't use blind signature, it would know signed bits. If I later spend the bank can see where I spent them.

Key gives a certain value to the coin signed.

Digital cash



- coin: a random number
- coin·b^e: blinded coin
- coin, $coin^d$: certified coin
- $d_{10€}$: a 10€ worth bank's private key

collects coin, forwards it to acquirer, acquirer gives coin to issuer and gets real money.

11

Double spending →

- The protocol does not prevent double spending
  - the customer can spend the digital coin multiple times
  - The merchant can deposit the digital coin multiple times
- Partial countermeasure
  - The issuer maintains the list of spent digital coins
    - Protect the bank from frauds
    - Don't allow issuer to identify the fraudster

Issuer knows coin has been already spent. But we don't know whether customer or merchant is guilty.

12

Double spending solution ↓

- Purely criptographic solutions based on
  - Secret splitting
  - Bit commitment
  - Cut-and-choose
  - Inefficient but great impulse to cryptography ①

Advanced mechanisms

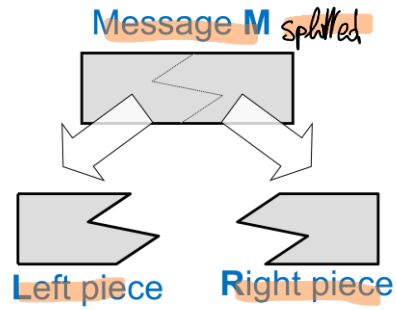
- Hardware solutions → using smart cards: coin is not on user hard disk but stored on card which is property of bank; HW blocks double spending
  - The Mondex smart card e-cash system
    - 90's technology; never left the experimental phase
- Bitcoin and blockchain ↳ customers refused

① Gave a nice impulse to cryptography and HW security

13

Secret splitting [→]

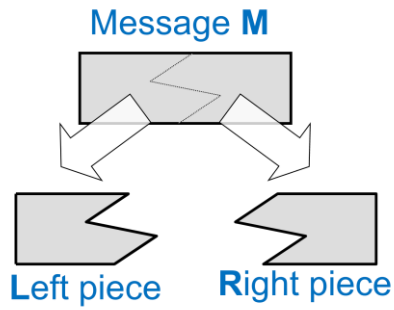
- Each piece alone gives no information on the message
- Both pieces make it possible to reconstruct the message



14

Secret splitting

- EXAMPLE
- Creating L and R
  - Message **M**
  - **R** ← random()
  - **L = M ⊕ R** ①
- Message reconstruction
  - M = L ⊕ R



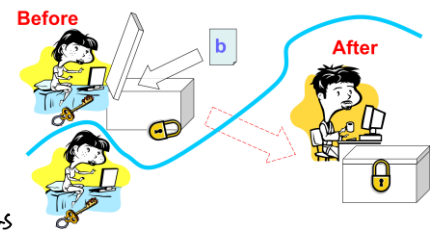
① if R is random, XOR gives back a random message. L, R are random pieces.

15

Bit commitment [→]

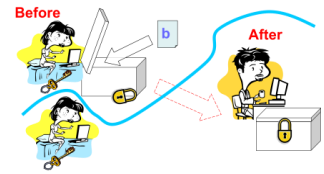
- Alice thinks of a number and Bob has to guess it.
- Alice thinks about the number but doesn't want to reveal it.
- Bob guesses the number but wants to be sure Alice doesn't change it.

Idea: Alice may put # in a box, close and lock the box. You give the locked box to Bob and Alice retains key. Bob makes guess and then you open box.



16

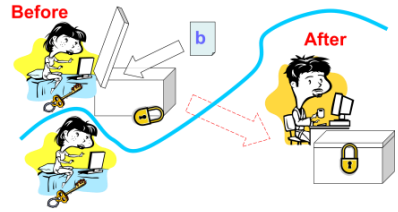
Bit Commitment [→]



- Perfectly binding
  - It is theoretically impossible for Alice to alter her commitment after she makes it
- Perfectly concealing
  - It is theoretically impossible for Bob to find commitment without Alice revealing it
- THM There exists no commitment scheme which is both perfectly binding and perfectly hiding

17

Bit Commitment: toy example

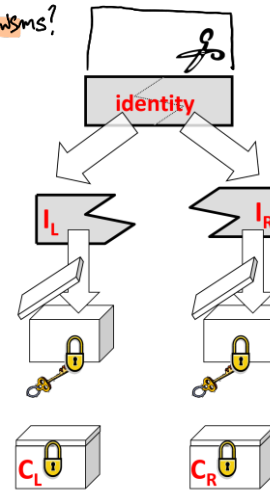


- Example (Perfectly binding)
  - Parameters
    - p: large prime
    - g: a generator
  - Commitment phase
    - Alice randomly selects b in [0, p – 1]
    - Alice computes commitment $c = g^b \bmod p$
    - Alice publishes c
  - Reveal Phase
    - Alice publishes b
    - Bob checks whether $c == g^b \bmod p$
  - Not perfectly concealing as $\leq_p$ DLP.

18

On solving double spending

How to use those mechanisms?



unique identifier

Secret splitting of the two parts

Left and right commitments: I put them in a box.

19

On solving double spending

↳ Represents identity of Alice. In case of double spending we know problem.

- Coin = [coin, identity string, h(coin, identity string)^d]
- Uniqueness bit string: coin ← random() To identify each coin
- Identity bit strings
  - ① $I_i \rightarrow \langle I_{iL}, I_{iR} \rangle$ Split identity string
  - ② ($C1_L$, C_{1R}), (C_{2L}, C_{2R}), …, (C_{100L}, C_{100R}) COMMITMENT
  - Pairs are different from each other

Bank obtains commitments, from commitments gets the identities

- Setup (money order)
  - Alice prepares 100 blank coin

① I split it in 100 different ways
② We save IS splits

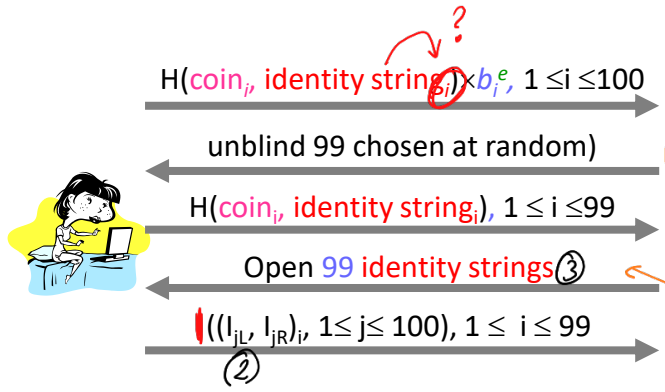
20

WHY 100 for each coin? Isn't 1 enough? To have $1-(\frac{1}{2})^{100}$ prob.

I generate a number of these coins and for each coin I repeat this split + commitment.

Alice sends all the coins in blinded form.

On solving double spending: cut-and-choose

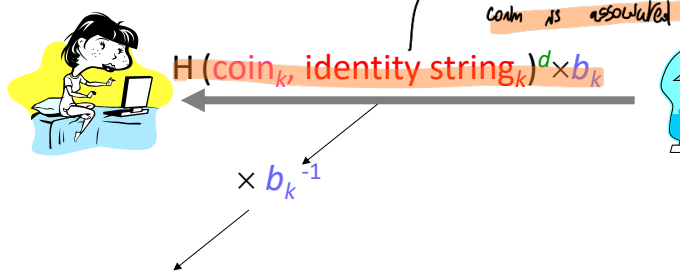


At the end of the protocol, the bank is 99% convinced that the undisclosed commitment contains Alice's identity

Apr-25 Digital signatures 21

21

On solving double spending: withdraw



$H(coin_k, identity\ string_k)^d$ Authenticated coin

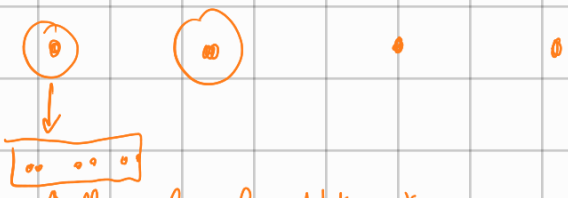
The bank "signs" the "blank" coin that is left over (e.g., the k-th)

Apr-25 Digital signatures 22

22

Give bank 100 coins, bank selects 99 of them to check and see if they are okay: I want to sign blindly but I want to be sure I sign the right stuff.

1. I select 99 random coins. For each of the coins I want the couples. (For each coin I split identity in a certain number of parts.



↑ all couples of identity streams

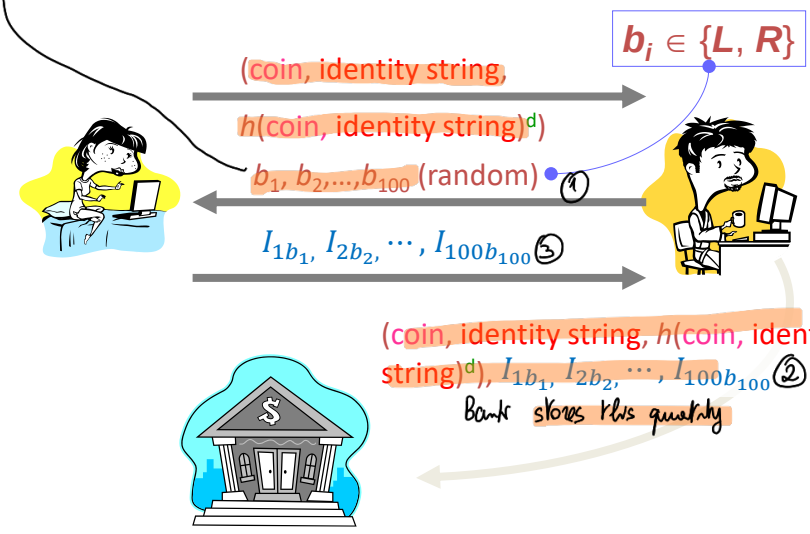
N coins, and L parts for identity stream

- Bank takes N-1 coins and asks to see L couples for identity number.

I suppose only Alice and bank know it

Bob requires me to open either left or right at random (Identity is secret).
③ Those might be left piece, right piece etc. so Bob cannot reconstruct Alice identity.

On solving double spending: spend



Apr-25 Digital signatures 23

① You open either the left or the right box. Alice returns left or right component of identity string. Bob then stores this quantity ②
I have 100 identities, Bob for each of them requires either left or right.

23

On solving double spending: bank's controls

1. The bank verifies the digital signature
2. If the coin has not yet been spent How? Bank stores coin
   1. the bank credits an amount equal to the denomination to Bob
3. Otherwise (double spending) Bank already stores something like Ⓐ
   1. if the identity strings are the same
      1. then the fraudster is the merchant Bob;
   2. otherwise
      1. the fraudster is Alice

Alice spent money with another merchant who did another selection of left and right.

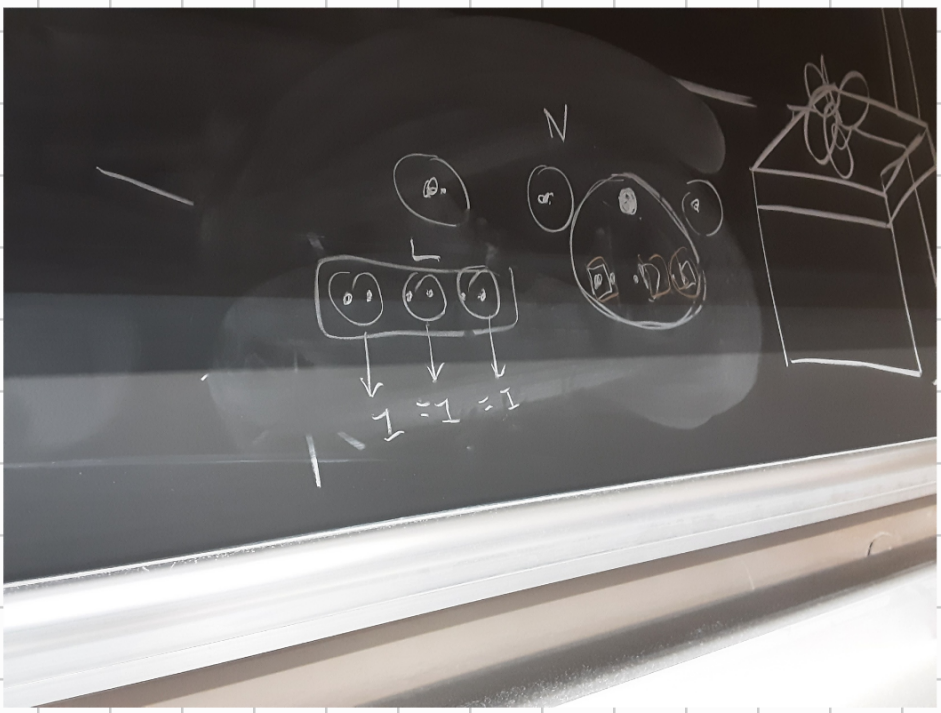
Apr-25 Digital signatures 24

24

Merchant selects casually left or right of the chose coin.

Banca vede 99 valido e ha 99% di probabilità di avere coin valido: Vuole sapere cosa sta firmando.

- 1st part: Bank wants to have 99% probability of knowing that it is signing a coin, but wants to still maintain secrecy for blind signature.
- 2nd part: prevent double spending! Merchants unlikely choose same value.
- Why do we need Contaminants?



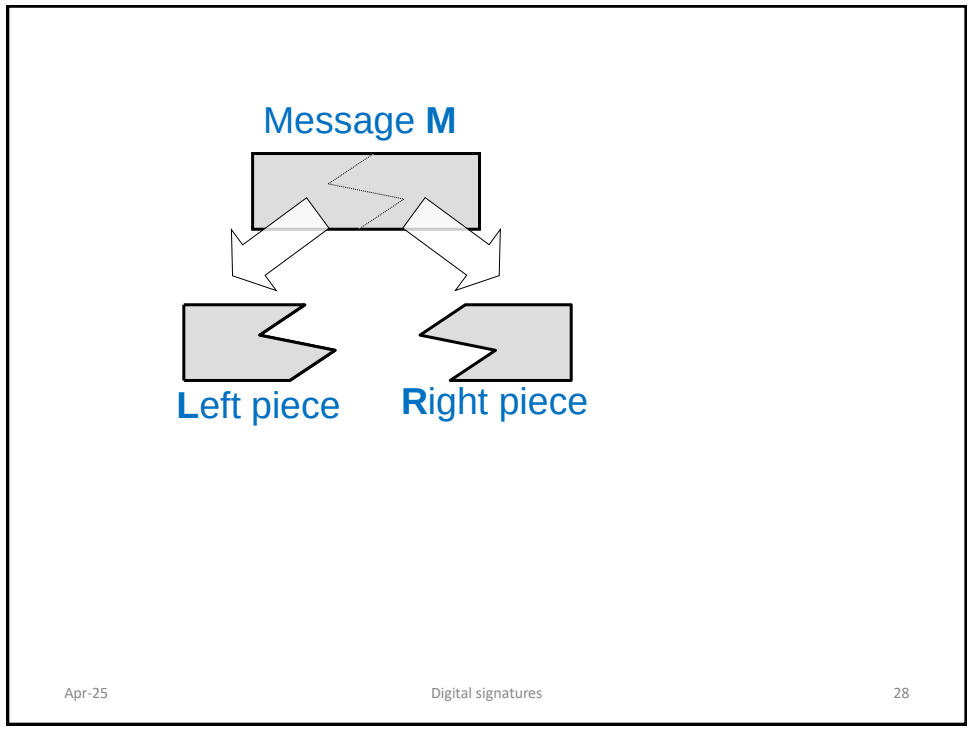
On solving double spending: fraudster detection

- In case the coin has already been spent
- If the identity strings are the same, then the fraudster is Bob, otherwise
- If the identity strings are different, then the fraudster is Alice
  - The bank finds a position in the identity string where Alice has revealed the right and left pieces of her identity with probability $1 - (\frac{1}{2})^{100}$
  - From the two pieces the bank determines Alice's identity

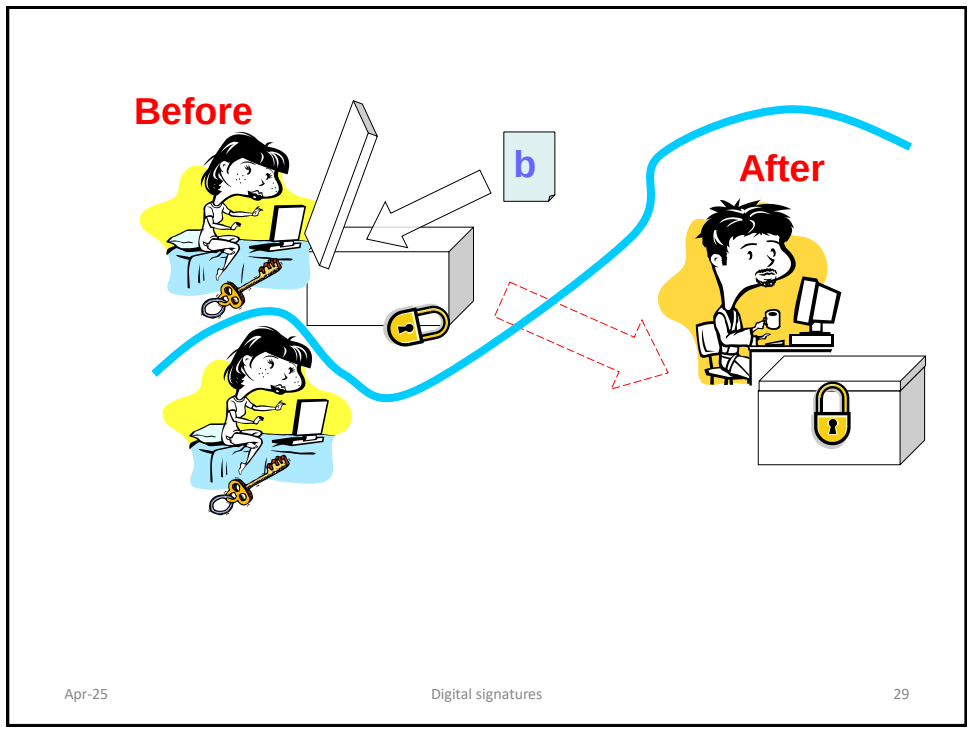
25

If Alice double speds, she does same for two merchants:
1st might require LRRL pieces etc., the second RLLR etc.
↑
Requires Alice to open first commitment (open the left box of the first pair and obtain left piece of box).

27


Message M
Left piece
Right piece
Apr-25
Digital signatures
28

28


Before
b
After
Apr-25
Digital signatures
29

29