Voting verifiability

2020/11/5

Mitsunari Shigeo

Summary

- Confidentiality and authenticity of electronic voting
- Blind signature
- Homomorphic Encryption
- Mix-Net
- Consideration about verifiability

EVS (Electronic voting system)

- Objective
  - To collect information regarding informants while leaving their identifies unidentified
- Matters to consider
  - Confidentiality
  - Authenticity
  - Verification
  - Verifiability

Confidentiality and authenticity

- Confidentiality
  - Voter secrecy must be protected
- Authenticity
  - Accepted votes belong to voters
  - Only valid votes are counted
  - No one can change the vote

Verification and Verifiability

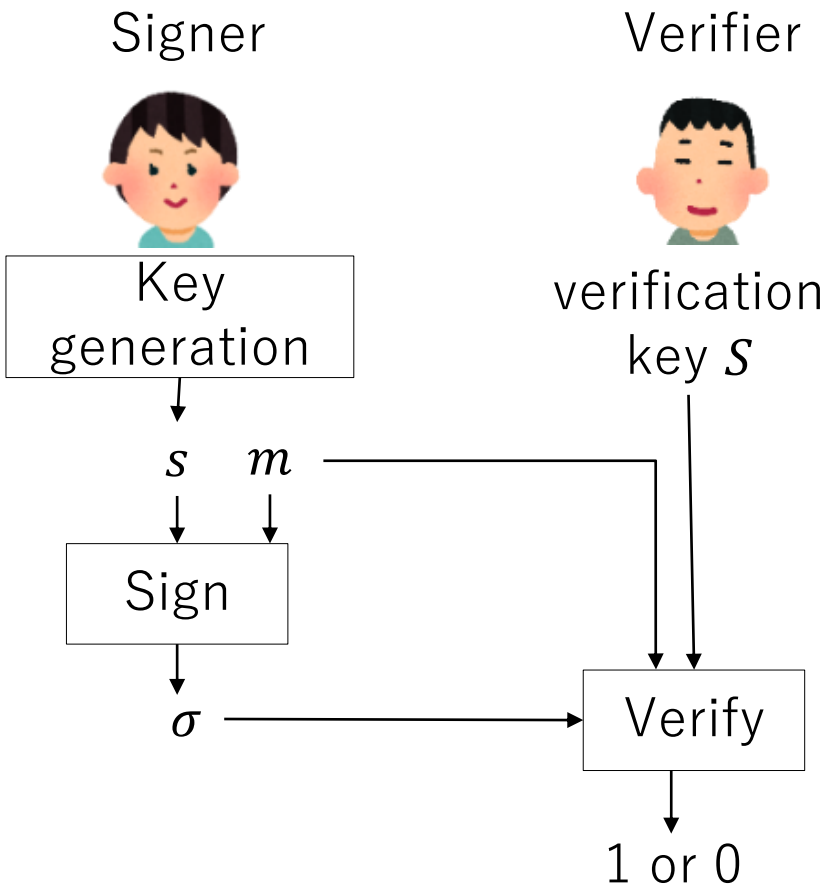
- Verification
  - To ensure that voters have the right to vote
  - The principle of "one person, one vote (prohibition of double voting) "
- Verifiability
  - Verifiability on an individual basis
    - Voters can verify that their votes were properly counted
  - Verifiability on a group basis
    - After the vote, everyone can verify that the tally was tallied correctly

Verifiability2

- Verifiability of complaints
  - A voter having right claims that his or her vote was not counted
    - Able to verify the claim to be true
  - A voter not having right claims that his or her vote was not counted
    - Able to verify the claim to be false
  - Keeping the votes of legitimate voters' secret if possible

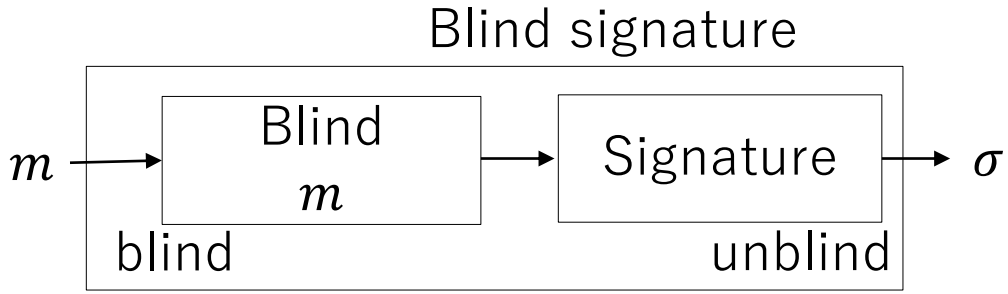
Signature : a review

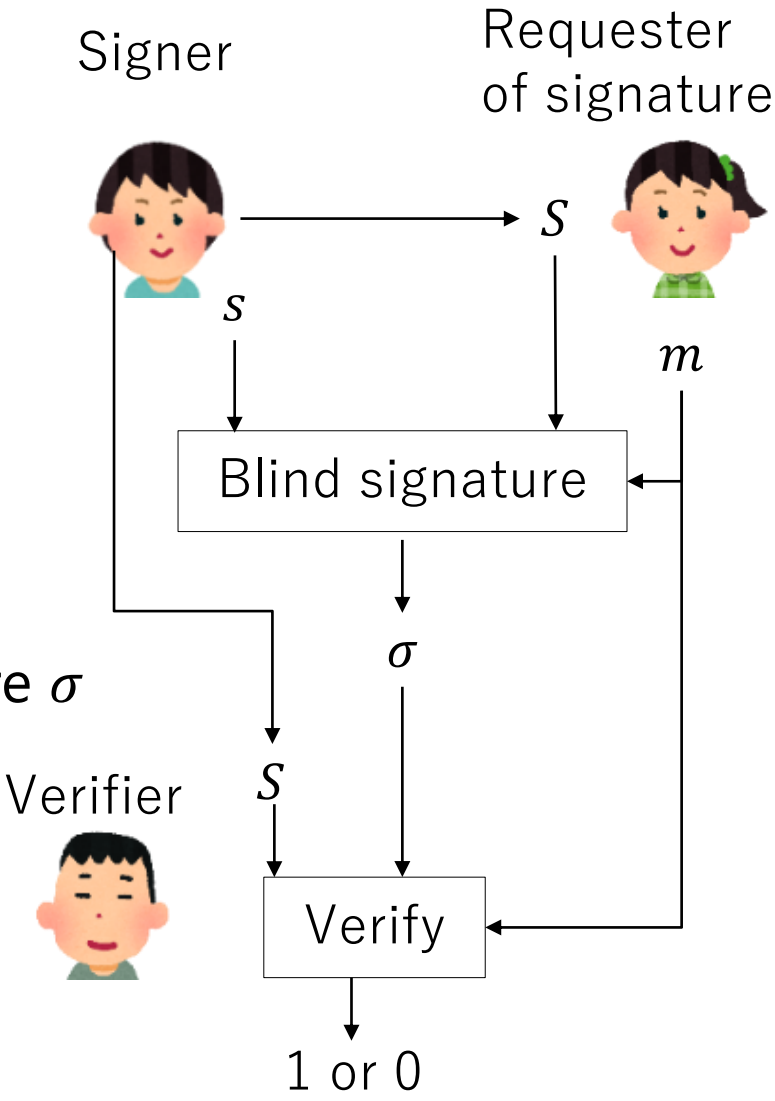
- Key generation
  - Generation of signature key (private key) and verification key (public key)
  - Disclose verification key
- Signature for message
  - σ=Sign(s,m)
- Verification for message and signature
  - Verify(S,m,σ)=1(accepted) , 0 (rejected)



Blind signature

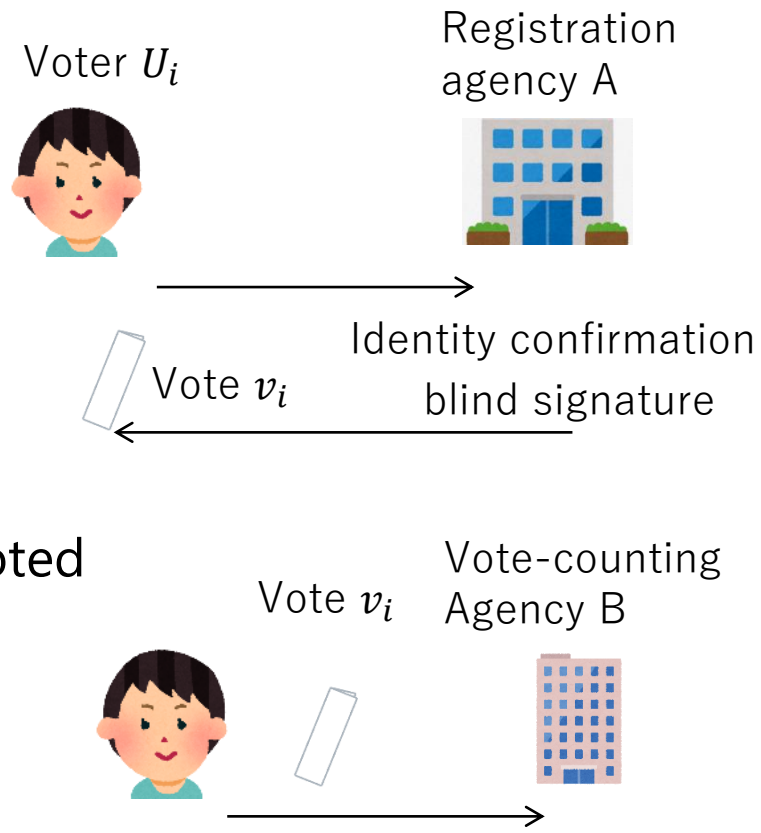
- Signer signs not knowing m
- Key generation
  - Generation of signature key$s$ and verification key S
- Signature for message
  - σ=Sign(s,m)
  - Signer doesn't know about m
- Verification for message m and signature σ
  - Verify(S,m,σ)=1 (accept) , 0 (reject)
- Illustration





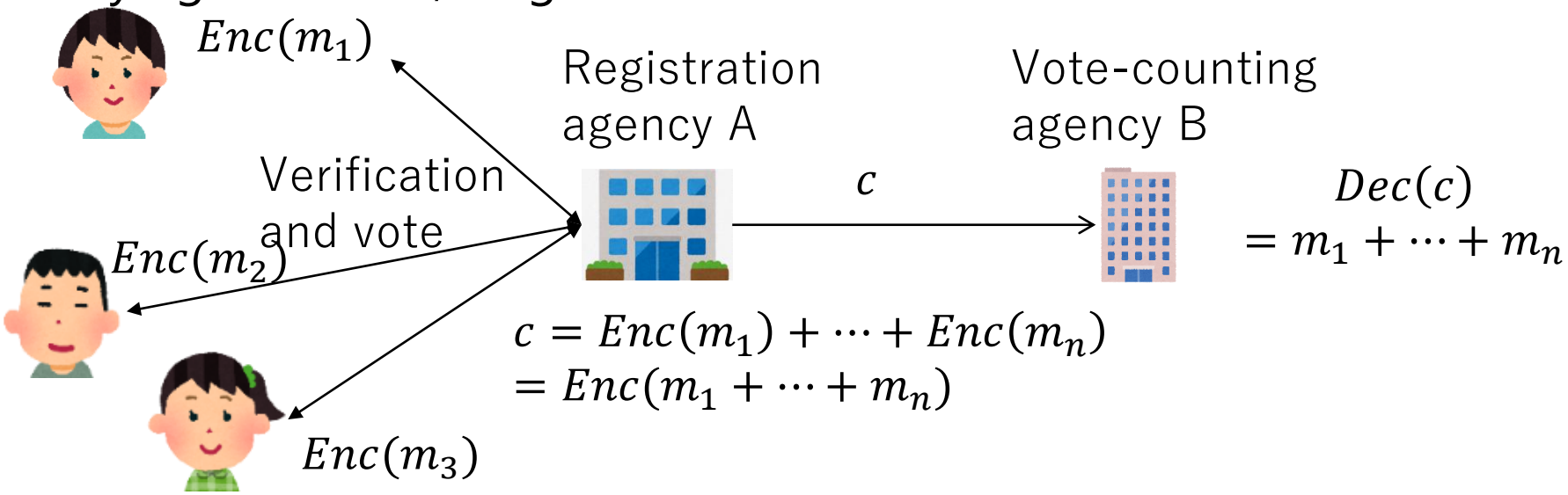
EVS by Blind signature

- Roles
  - Registration agency A, Voter U_i, Vote-counting Agency B
- Preparation
  - Generate A's signature key s and verification key S and disclose them
- Vote
  - A verifies identity and that he/ she hasn't voted
  - U_i does a blind signature for vote mi
  - v_i=(m_i,σ_i =$Sign(m_i,s)$)
- Vote-counting
  - U_i sends v_i anonymously to B. B then verifies the signature and accepts it
  - After the vote, B disclose all votes and tally



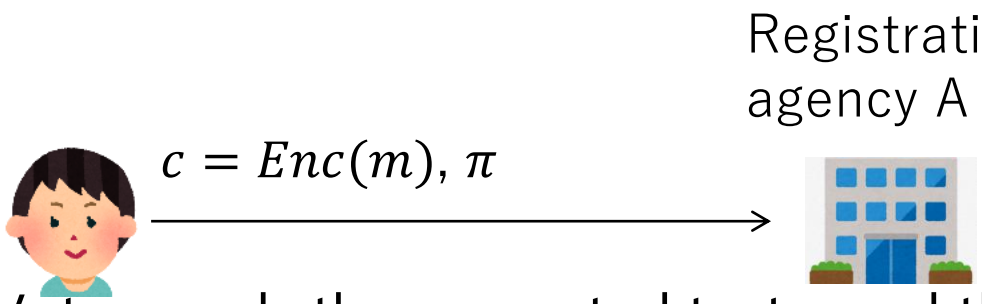
Vote by Homomorphic Encryption

- Homomorphic Encryption
  - An encryption that allows encrypted text-to-text calculations
  - $Enc(x)+Enc(y)=Enc(x+y)$
- Vote-counting agency B prepares public key S and private key and disclose public keys
  - Tallying 1 in favor, 0 against



Zero-knowledge proof

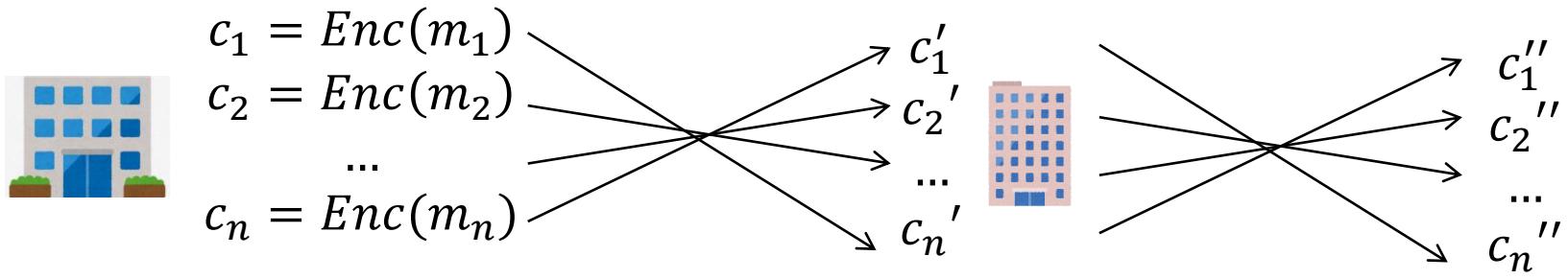
- If m = 100 in $Enc(m)$, then one person would have 100 votes
  - Registration agency A wants to verify whether 0 or 1 in mi
  - We use ZKP (Zero-knowledge proof)



  - Voter sends the encrypted text c and the corresponding certificate π
  - Registration agency confirms $m \in \{0,1\}$ Based on (c,π)
    - Doesn't know whether m=0 or m=1
- Implementation by using WebAssembly
  - https://github.com/herumi/she-wasm

Mix-Net

- Mix encrypted texts



  - $\{c_1, \ldots, c_n\} = \{c_1', \ldots, c_n'\}$
- Replacement and ZKP
  - Knows that texts have been replaced
  - Doesn't know which and to where it has been moved
  - Repeats Mix-Net
- ZKP at $Dec(c) = m$
  - Indicates that you have decrypted the encrypted text correctly without exposing the private key

Safety

- Suppose that…
  - We can all trust registration agency and vote-counting agency
    - There is no foul play
    - They don’t conspire
- Confidentiality
  - Registration agency
    - We don’t know which elector was voted by blind signature
    - Separate “confirmation of voting rights" and ”signature of vote contents"
  - Vote-counting agency
    - Do not collect information of connection when voters vote
      - Is blockchain available?
      - Use Tor when necessary?

Verifiability regarding blind signature

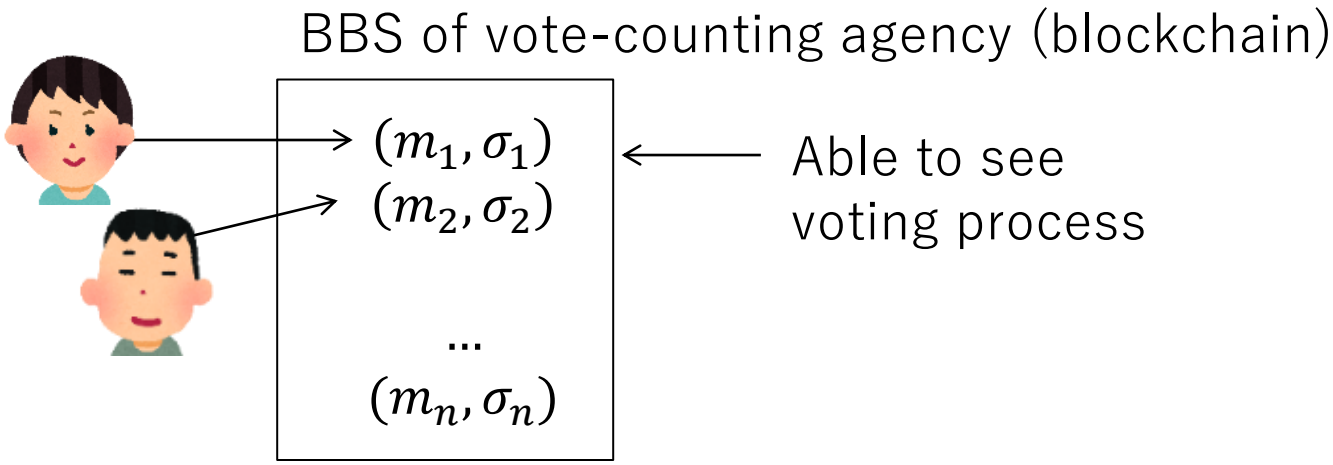
- Individual
  - Find your vote from all published votes
- Group
  - Everyone can check the authenticity of their vote signature
  - The correctness of the voting result also can be checked
- Complaints about vote
  - A voter claims that his or her vote was not counted
    - An example of getting that voter to disclose his/ her vote
      - Sign $H(r_i)||m_i$ by using random number r_i and hash function H insead of m_i
      - A voter with no foul play can disclose r_i
      - A voter with foul play can't

Examination of fraud by registration agency

- Registration agency can fabricate legitimate ballots at will
- One countermeasure
  - Decentralization of registration agency $A_1, \ldots, A_n$
    - Safe as long as registration agencies don't work together in conspiracy
    - Voter's $\overline{m_i} = H(r_i)||m_i$, r_i ; random number
    - Get each A_j to do a blind signature $\overline{m_i}$
    - Vote-counting Agency
      - $\sigma_{i_1}, \ldots, \sigma_{i_n}$ verify that it is the right signature to $\overline{m_i}$

Examination of fraud by vote-counting agency

- Vote manipulation
  - In principle, it's not possible to manipulate counting and data
  - Risk of exploitation of voters' information
- Possibility of seeing tally status before voting completes



Countermeasures

- Use commitment
  - No one knows what is in $c = Commit(m)$; m
  - Disclose $Open(c)$; m
- Each voter writes $c_i = Com(m_i)$ in vote instead of m_i
- All votes $\{c_i, \sigma_i\}$ are published
- Afterwards, $Open$ commitment and everyone gets $\{m_i\}$
- Cons : The presence of people who don't open/ data of voters are kept until opening of vote