CybozuLabs

投票の検証可能性

2020/11/5

光成滋生

概要

- 電子投票の機密性と真正性
- ブラインド署名
- 準同型暗号
- Mix-Net
- 検証可能性の考察

電子投票EVS(Electronic voting system)

- 目標
  - 情報提供者を特定することなく情報提供者の情報を集約する
- 考慮すべき事柄
  - 機密性
  - 真正性
  - 認証
  - 検証可能性

機密性と真正性

- 機密性
  - 投票者の秘密は守らなければならない
- 真正性
  - 受理した投票は投票者のものである
  - 有効な投票しか数えない
  - だれも投票内容を変更できない

認証と検証可能性

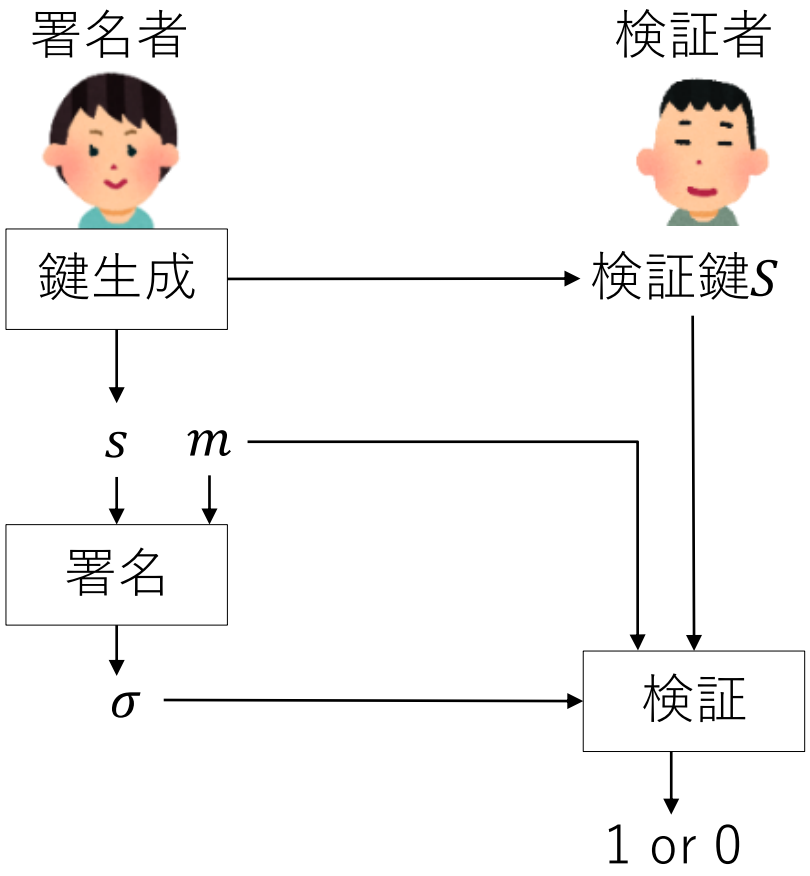
- 認証
  - 投票者が投票の権利を持っていることを確認する
  - 一人一票の原則（二重投票の禁止）
- 検証可能性
  - 個人の検証可能性
    - 投票者は自分の投票がちゃんと集計されたことを検証できる
  - 全体の検証可能性
    - 投票後に誰もが正しく集計されたことを検証できる

検証可能性2

- クレーム内容の検証可能性
  - (正当な)投票者が「自分の票が集計されていない」と発言
    - その発言が真であることを検証できる
  - (偽の)投票者が「自分の票が集計されていない」と発言
    - その発言が偽であることを検証できる
  - 可能なら正当な投票者の票の秘密を守ったまま

署名の復習

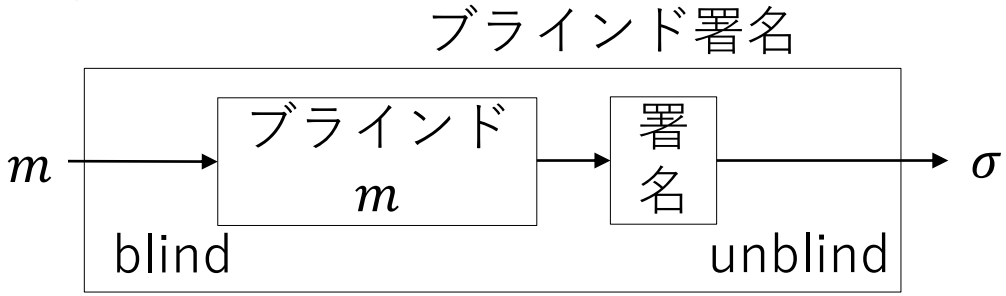
- 鍵生成
  - 署名鍵（秘密鍵）sと検証鍵（公開鍵）Sの生成
  - 検証鍵Sは公開する
- メッセージmに対する署名
  - $\sigma = \mathrm{Sign}(s, m)$
- メッセージmと署名σの検証
  - $\mathrm{Verify}(S, m, \sigma) = 1$(受理), 0(却下)

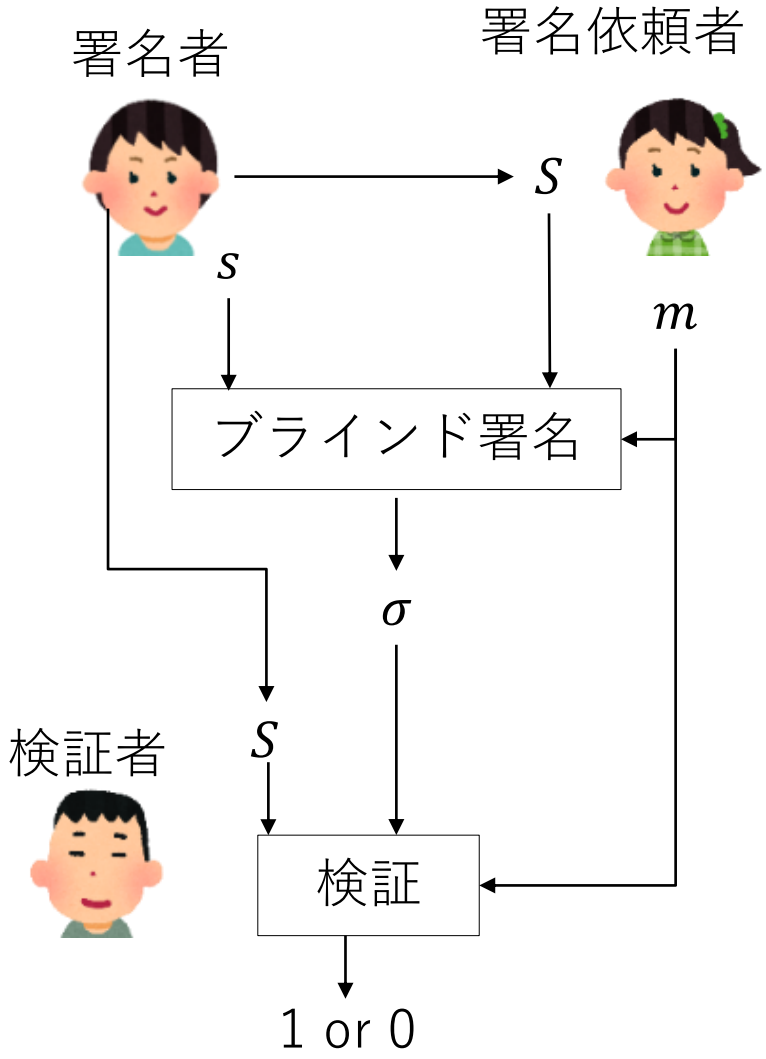


ブラインド署名

CybozuLabs

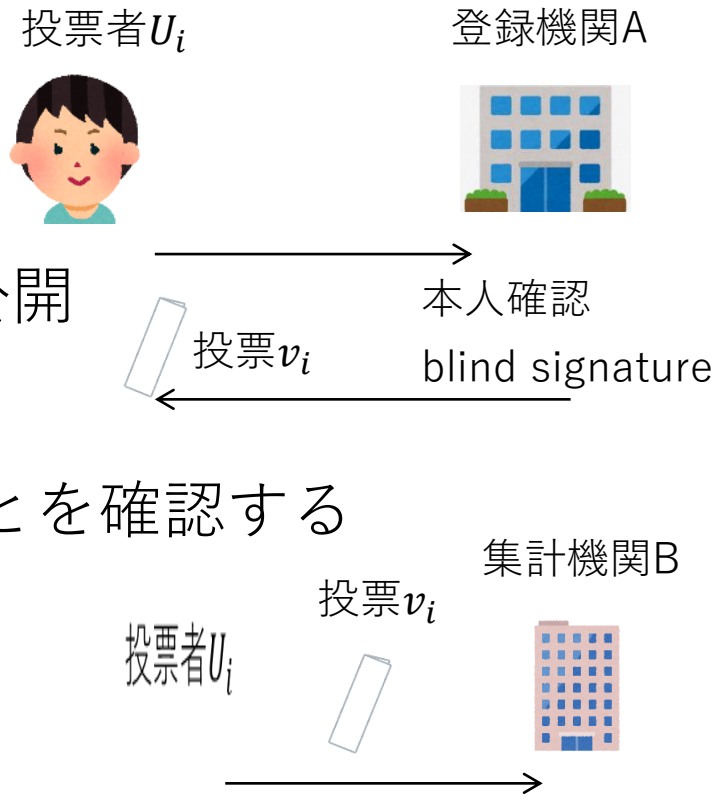
- 署名者はmを知らずに署名する
- 鍵生成
  - 署名鍵sと検証鍵Sの生成
- メッセージmに対する署名
  - $\sigma = \mathrm{Sign}(s, m)$
  - 署名者はmを知らない
- メッセージmと署名σの検証
  - $\mathrm{Verify}(S, m, \sigma) = 1$(受理), 0(却下)
- イメージ





ブラインド署名によるEVS

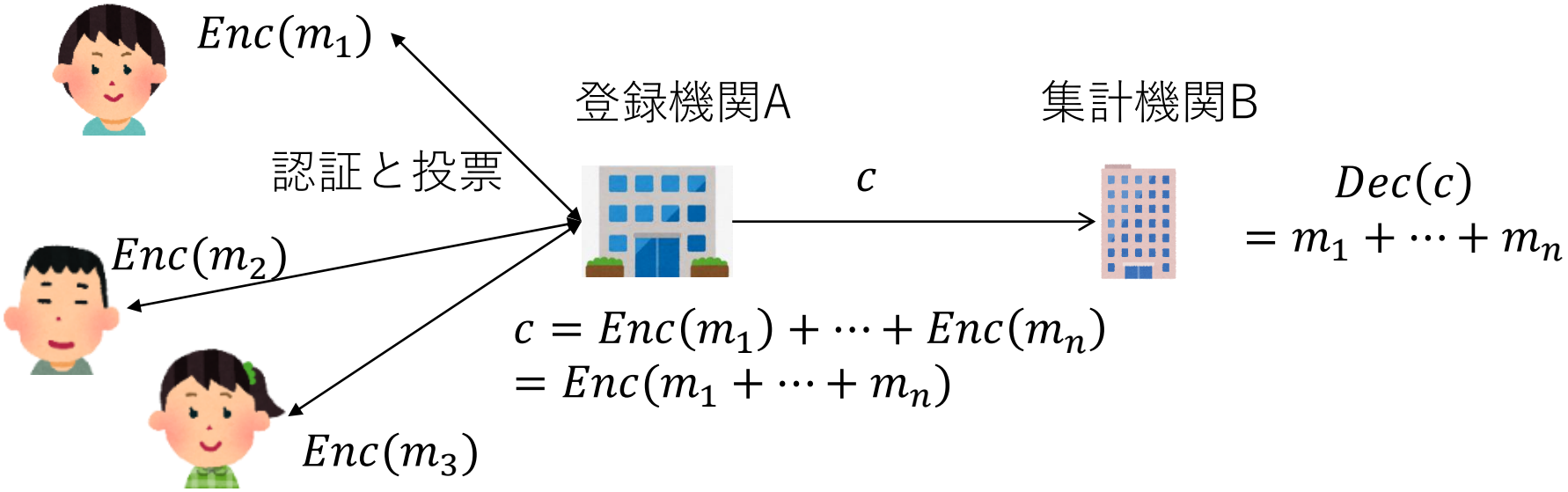
- 登場人物
  - 登録機関A, 投票者U_i, 集計機関B
- 準備
  - Aの署名鍵sと検証鍵Sを生成しSを公開
- 投票
  - AはU_iの本人確認と未投票であることを確認する
  - U_iは投票内容m_iについて
    ブラインド署名してもらう
  - $v_i = (m_i, \sigma_i = Sign(m_i, s))$
- 集計
  - U_iは匿名でBにv_iを送信し、Bは署名を検証して受理
  - 投票完了後Bは全投票を公開しm_iを集計する



準同型暗号による投票

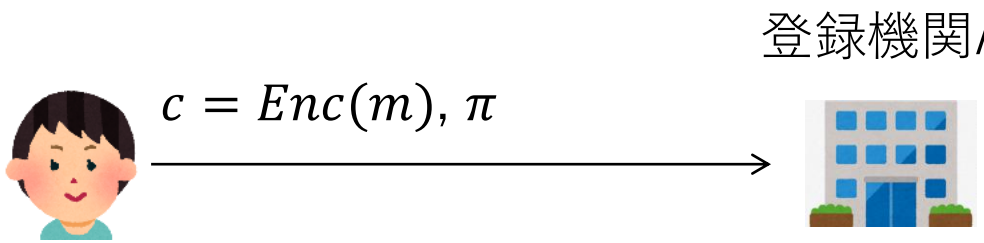
CybozuLabs

- 準同型暗号
  - 暗号文同士の演算ができる暗号
  - $Enc(x) + Enc(y) = Enc(x + y)$
- 集計機関Bが公開鍵Sと秘密鍵sを準備し公開鍵を公開
  - 賛成1、反対0の集計をする



ゼロ知識証明

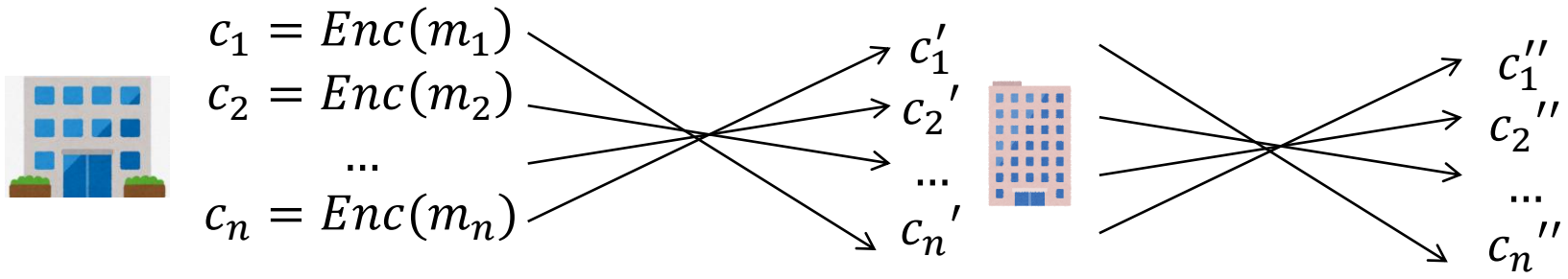
- $Enc(m)$のmが100なら一人で100票
  - 登録機関Aは$c = Enc(m_i)$のm_iが0か1であることを確認したい
  - ZKP（Zero-knowledge proof）を使う



  - 投票者は暗号文cと対応する証明πを送る
  - 登録機関は(c, π)から$m \in \{0,1\}$であることを確認する
    - $m = 0$か$m = 1$のどちらかは分からない
  - WebAssemblyでの実装例
    - https://github.com/herumi/she-wasm

Mix-Net

- 暗号文同士を混ぜる



  - $\{c_1, \ldots, c_n\} = \{c_1', \ldots, c_n'\}$
- 置換とZKP
  - 入れ換えであることは分かる
  - どれがどれに移動したかは分からない
  - Mix-Netを繰り返す
- $Dec(c) = m$のZKP
  - 秘密鍵を公開せずに正しく暗号文を復号していることを示す

安全性

- 仮定
  - 登録機関と集計機関は信頼できる
    - 不正はしない
    - 結託しない
- 機密性
  - 登録機関
    - ブラインド署名により投票内容は分からない
    - 「投票権の確認」と「投票内容の署名」を分離
  - 集計機関
    - 投票者が投票するときの接続情報を収集しない
      - ブロックチェーンでもOK?
      - 必要ならTorなどを使う?

ブラインド署名についての検証可能性

CybozuLabs

- 個人
  - 公開された全投票から自分の投票を探す
- 全体
  - だれもが投票の署名の正しさを確認できる
  - 集計結果の正しさも確認できる
- 投票に対するクレーム
  - 投票者が「自分の票が集計されていない」と発言
    - その投票者にその票を開示してもらう例
      - m_iの代わりに乱数r_iとハッシュ関数Hを用いて$H(r_i)||m_i$に署名する
      - 正当な投票者ならr_iを開示できる
      - 偽の投票者なら開示できない

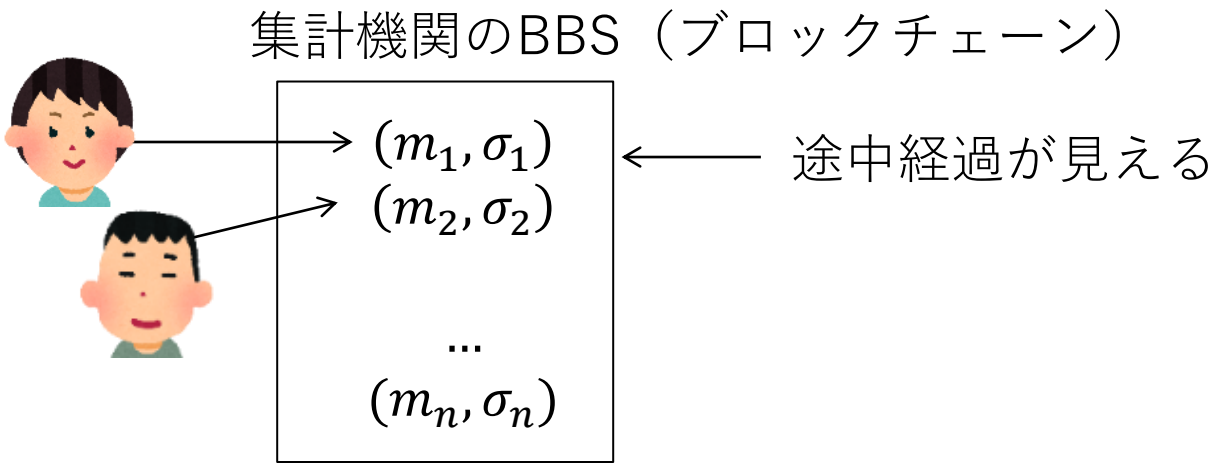
登録機関の不正の検討

CybozuLabs

- 登録機関は勝手に正しい投票用紙を捏造できる
- 一つの対策案
  - 登録機関の分散化$A_1, \ldots, A_n$
    - 全登録機関が結託しない限り安全
    - 投票者の$\overline{m_i} = H(r_i)||m_i, r_i$；乱数
    - 各A_jに$\overline{m_i}$をブラインド署名してもらう$(\sigma_{i1}, \ldots, \sigma_{in})$
    - 集計機関
      - $\sigma_{i1}, \ldots, \sigma_{in}$が$\overline{m_i}$に対する正しい署名であることを検証

集計機関の不正の検討

- 投票の操作
  - 基本的に集計やデータの改竄はできない
  - 投票者の情報収集の可能性
- 投票完了前に集計状況を見える可能性



いくつかの対策案

CybozuLabs

- コミットメントを使う
  - $c = Commit(m)$; mの中身は分からない
  - $Open(c)$; mを開示する
- 各投票者はm_iの代わりに$c_i = Com(m_i)$を投票に記す
- 全投票$\{c_i, \sigma_i\}$が公開される
- そのあとコミットメントを$Open$して全員が$\{m_i\}$を取得
- Cons : Openしない人の存在 / 投票者は開票までデータ保持