



# New Techniques for Electronic Voting

August 11, 2015

**Alan Szepieniec** and Bart Preneel

`firstname.lastname@esat.kuleuven.be`

KU Leuven, ESAT/COSIC and

iMinds, Belgium



# Outline

0. UC Voting with Universal Verifiability
1. Tally-Hiding Vote
2. Self-Tallying Vote
3. Authenticated Voting Credentials

|     | UV | UC |
|-----|----|----|
| THV | ✓  | ✓  |
| STV | ✓  | ✓  |
| AVC | ✓  | ✓  |

# 0. Universally Composable Voting

# Voting System

## Definition

Let  $\mathbf{O}$  be a set of *options* and  $\mathbf{PO}$  be the set of permutations of this set. Let  $f : (\mathbf{PO})^n \rightarrow \{0, 1\}^*$  be a *tallying function*.

A *voting system* is an interactive protocol between voters  $V_1, \dots, V_n$ , who each hold a vote  $v_i \in \mathbf{PO} \forall i \in \{1, \dots, n\}$ , and *authorities*  $A_1, \dots, A_k$ , if it computes the tally  $t = f(v_1, \dots, v_n)$ .

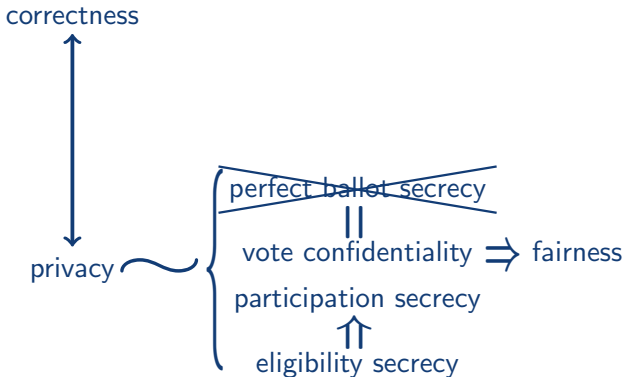
# Properties of Voting Systems

correctness



privacy

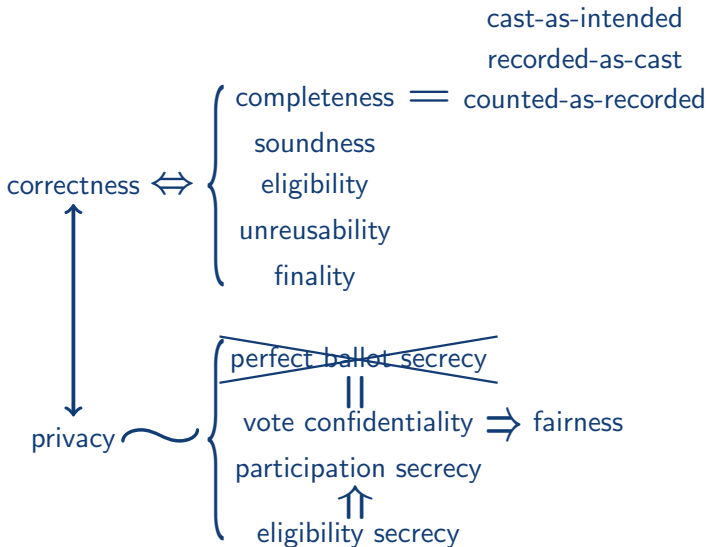
# Properties of Voting Systems



# Properties of Voting Systems



# Properties of Voting Systems

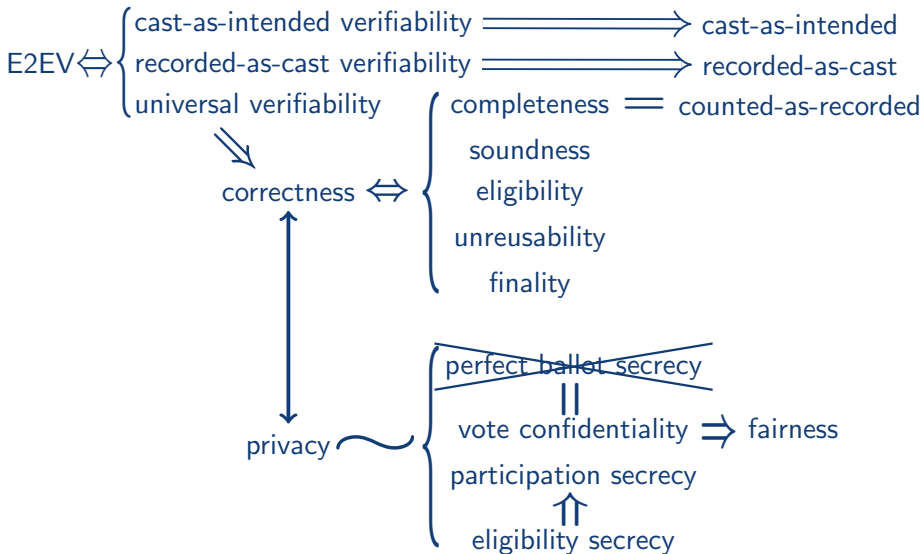




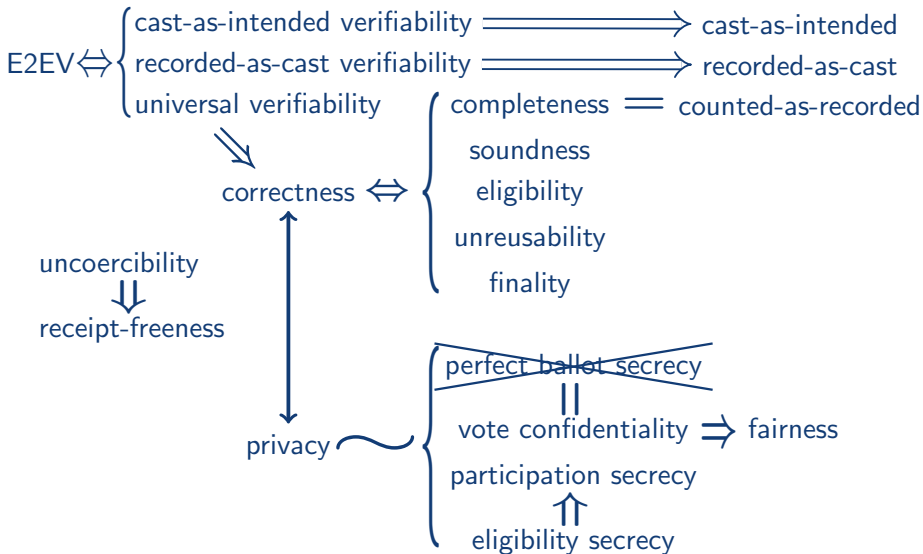
# Properties of Voting Systems



# Properties of Voting Systems



# Properties of Voting Systems

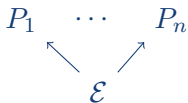


# Universal Composability

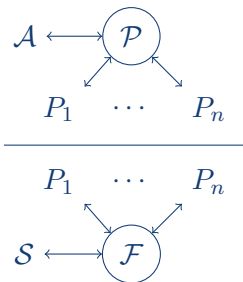
- standard framework for provable security of protocols
- composability  $\Rightarrow$  allows modular protocol design
- ideal functionality  $\mathcal{F}$ : abstract description
- protocol  $\mathcal{P}$ : concrete instantiation of  $\mathcal{F}$
- an experiment is conducted in one of two worlds:
  - *real world*: an adversary  $\mathcal{A}$  attacks  $\mathcal{P}$
  - *ideal world*: a simulator  $\mathcal{S}$  attacks  $\mathcal{F}$
- the *environment machine*  $\mathcal{E}$ :
  - chooses players' inputs beforehand;
  - reads players' outputs afterwards;
  - decides in which world the experiment took place

# Universal Composability

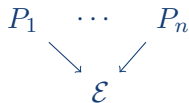
1.



2.



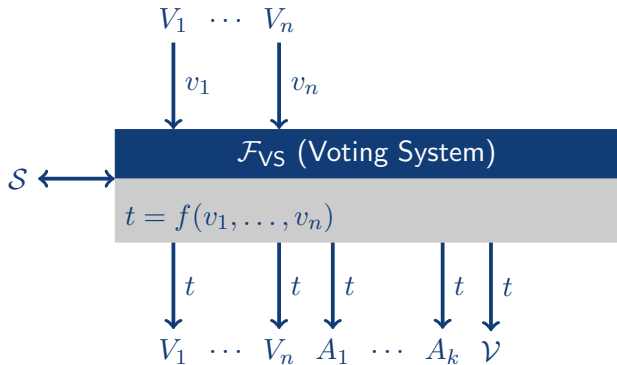
3.



## Definition

Protocol  $\mathcal{P}$  is a *UC-secure realization* of ideal functionality  $\mathcal{F}$  if for all adversaries  $\mathcal{A}$  attacking  $\mathcal{P}$ , there exists an adversary-simulator  $\mathcal{S}$  attacking  $\mathcal{F}$  such that no environment  $\mathcal{E}$  can tell the difference.

## Ideal Functionality: Voting System



- $S$  can block votes
- $\mathcal{F}$  computes  $t$  when the authorities say so
- $\mathcal{V}$  receives  $t$  also

# UC Voting System

$\mathcal{P}_{VS}$  (Voting System)

$\mathcal{F}_{BB}$

Bulletin Board:  
anonymous,  
public access  
append-only list  
of messages

$\mathcal{F}_{PKG}$

Participant Key  
Gen: generates and  
distributes keypairs  
for each participant

$\mathcal{F}_{SKG}$

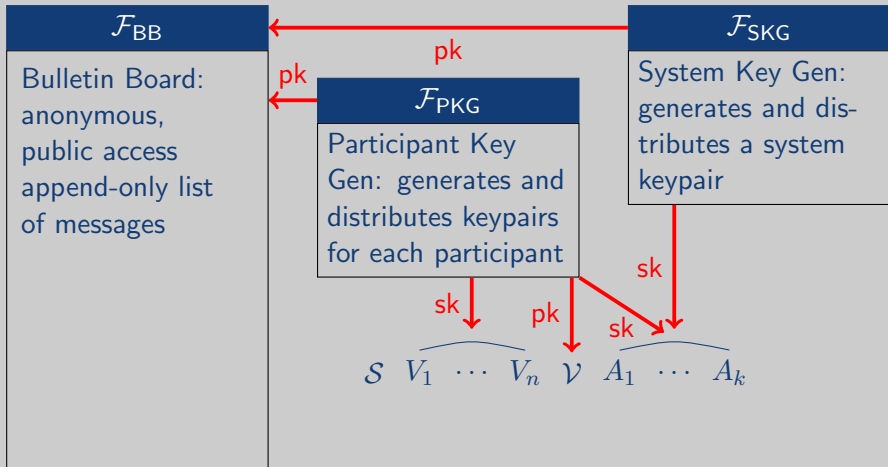
System Key Gen:  
generates and dis-  
tributes a system  
keypair

$\mathcal{S} \quad \overbrace{V_1 \cdots V_n} \quad \mathcal{V} \quad \overbrace{A_1 \cdots A_k}$

# UC Voting System

1

$\mathcal{P}_{VS}$  (Voting System)



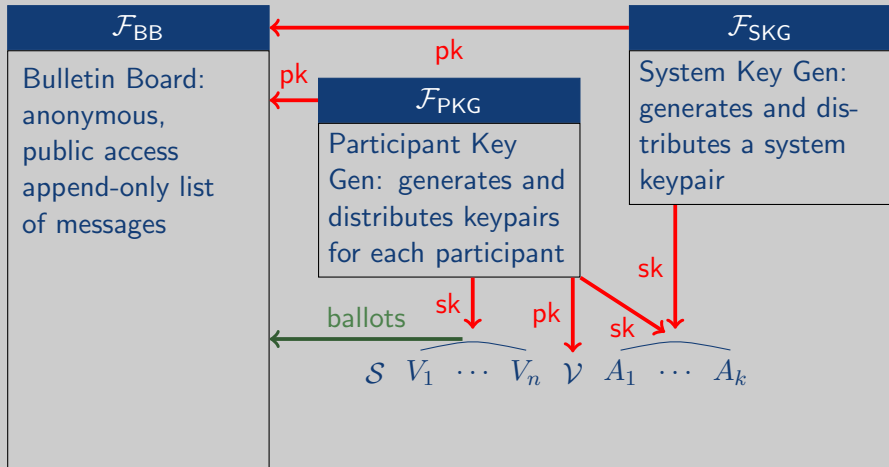


# UC Voting System

1

2

$\mathcal{P}_{VS}$  (Voting System)

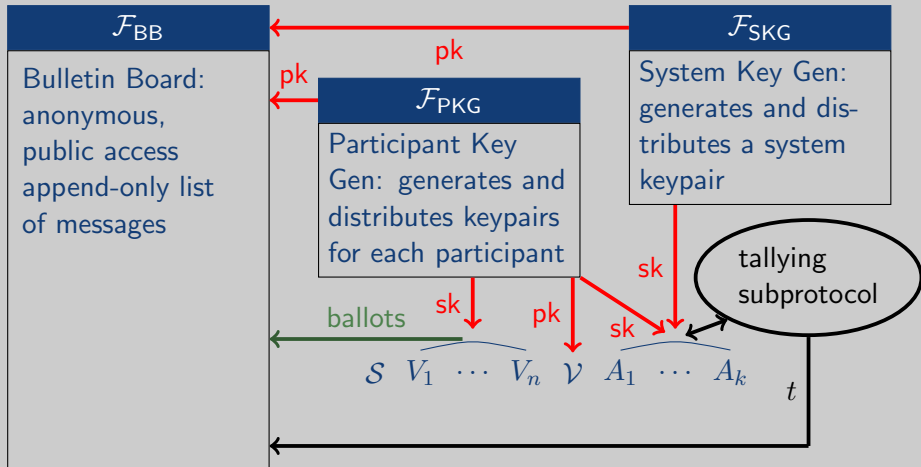


# UC Voting System

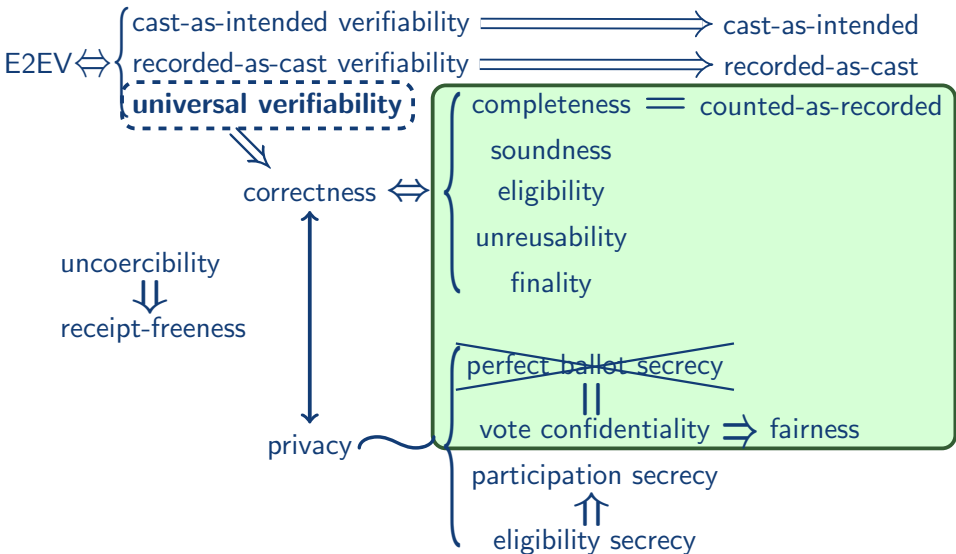
1

2

3

 $\mathcal{P}_{VS}$  (Voting System)

# Properties of UC-Secure Voting Systems



## Universal Verifiability

- $\textcircled{\mathcal{P}} \leftrightarrow \mathcal{A} \quad \mathcal{V} \quad | \quad \textcircled{\mathcal{P}} \rightarrow \mathcal{A} \quad \mathcal{V}$   
 $b = 1 \quad \quad \quad b = 0$
- $\textcircled{\mathcal{P}} \quad \mathcal{A} \xrightarrow{T} \mathcal{V}$
- $\textcircled{\mathcal{P}} \quad \mathcal{A} \quad \mathcal{V} \xrightarrow{\hat{b}}$

### Definition

Protocol  $\mathcal{P}$  is *universally verifiable* if there exists a verifier  $\mathcal{V}$  who retains, for all adversaries  $\mathcal{A}$  attacking  $\mathcal{P}$ , significant distinguishing power:

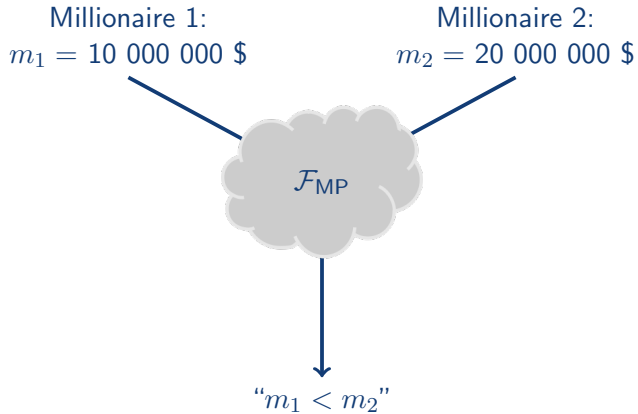
$$|\Pr[b = \hat{b}] - \Pr[b \neq \hat{b}]| \geq \frac{1}{2} .$$

# 1. Tally-Hiding Vote

## Tally-Hiding Vote: Idea

- vote counts leak unnecessary information
- vote counts remain hidden
- tally identifies the winning option
- better name: *vote count hiding*
- preferential votes don't need vote counts

# Millionaire Problem



# UC-Secure Tally-Hiding Vote

two options: A and B

1. voters cast ballots:  $\forall i : V_i \xrightarrow{E(v_{i,A}), E(v_{i,B})} \mathcal{F}_{BB}$
2. homomorphic aggregation:  $E(c_A) = E(\sum_i v_{i,A})$  and  $E(c_B)$
3. millionaire problem:  $t = \mathcal{F}_{MP}(E(c_A), E(c_B))$



# Paillier Cryptosystem

KeyGen( $1^\kappa$ ):

$p, q \xleftarrow{\$}$  random primes

$n \leftarrow pq$  (public key)

$d = 1 \pmod n$  and

$d = 0 \pmod{\varphi(n)}$  (private key)

Encrypt( $m$ ):

$r \xleftarrow{\$} \mathbb{Z}_{n^2}$

$E(m) = (1 + n)^m r^n \pmod{n^2}$

Decrypt( $c$ ):

$\ell \leftarrow c^d \pmod{n^2}$

$m \leftarrow \frac{\ell - 1}{n}$

Homomorphic Add( $c_1, c_2$ ):

$c \leftarrow c_1 c_2 \pmod{n^2}$

## Millionaire Problem Protocol for Paillier

Damgård-Jurik cryptosystem:

$$E_2(m) = (1 + n)^m r^n \pmod{n^3} \quad m \in \mathbb{Z}_{n^2}$$

Black-box lifting procedure Lift maps a Paillier ciphertext ( $m \in \mathbb{Z}_n$ ) to a Damgård-Jurik ciphertext ( $m \in \mathbb{Z}_{n^2}$ ).

$$\text{Lift} : \mathbb{Z}_{n^3} \rightarrow \mathbb{Z}_{n^2}$$

Millionaire Problem ( $c_1, c_2$ ):

$$B \leftarrow \text{Lift}(c_1) \ominus \text{Lift}(c_2)$$

$$A \leftarrow \text{Lift}(c_1 \ominus c_2)$$

$$D(B \ominus A) = 0 \Rightarrow \text{no overflow} \Rightarrow c_1 \geq c_2$$

$$D(B \ominus A) \neq 0 \Rightarrow \text{overflow!} \Rightarrow c_1 < c_2$$

## Ciphertext Lifting

secret key is distributed among authorities  $A_1, \dots, A_k$  s.t.

$$(1+n)^{4\Delta^2 m} = 1 + 4\Delta^2 mn = \prod_i c^{4\Delta^2 s_i \prod_{j \neq i} \frac{-j}{i-j}} \pmod{n^2} .$$

Lift( $c$ ) :

- $1 + 4\Delta^2 \underline{m}n = \prod_i \underline{c}_i \pmod{n^2}$
- $\underline{c}_i^{(4\Delta^2)^{-1} \pmod{n}} = a_i + n\underline{b}_i \pmod{n^2}$  with  $a_i < n$
- $\underline{m} = \left[ \prod_i a_i \right]_2 + \sum_i \underline{b}_i \prod_{j \neq i} a_j \pmod{n^*}$
- $E_2(\underline{m}) =$   
 $E_2\left(\left[\prod_i a_i\right]_2\right) \oplus E_2\left(\frac{b_1}{j \neq 1} \prod a_i\right) \oplus E_2\left(\frac{b_2}{j \neq 2} \prod a_i\right) \oplus \dots$

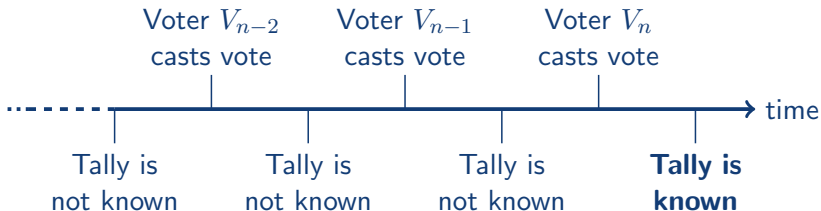
## 2. Self-Tallying Vote

## Self-Tallying Vote: Idea



- cut out the expensive tallying procedure
- tally is known as soon as last vote is cast (but not before)

## Control Voter



- $V_n$  knows the tally *before casting his vote*
- violates *fairness*
- cannot be UC-secure
- solution:  $V_n$  *cannot be corrupted*

## Self-Tallying Vote with Paillier

- $\mathcal{F}_{\text{SKG}}$  distributes  $x_i$  among voters s.t.  $\sum_i x_i = 0$
- common randomizer  $r \in \mathbb{Z}_{n^2}$  (from timestamp or hash)
- voters encrypt votes as  $c_i = (1 + n)^{v_i r^{x_i n}} \pmod{n^2}$
- homomorphic aggregation:

$$\begin{aligned}1 + nt &= c_1 \oplus c_2 \oplus \cdots \oplus c_n \\ &= \prod_i c_i \pmod{n^2} \\ &= \prod_i (1 + n)^{v_i r^{n x_i}} \pmod{n^2} \\ &= \left( \prod_i (1 + n)^{v_i} \right) (r^n)^{\sum_i x_i} \pmod{n^2} \\ &= \prod_i (1 + n)^{v_i} \pmod{n^2} \\ &= 1 + n \sum_i v_i \pmod{n^2}\end{aligned}$$

# 3. Authenticated Voting Credentials



## Voting Credentials: Idea

initialization:

$$\{A_1, \dots, A_k\} \xrightarrow{\circ} V_i$$

voting:

$$\mathcal{F}_{\text{BB}} \xleftarrow{\circ, v_i} V_{\blacksquare}$$

tallying:

$$t = \sum_i v_i$$

- *anonymous access* to  $\mathcal{F}_{\text{BB}}$
- fairness  $\Rightarrow \mathcal{A}$  cannot read  $\mathcal{F}_{\text{BB}}$  during voting
- invalid credential  $\Rightarrow v_i$  not counted
- duplicate credential  $\Rightarrow v_i$  not counted

# Authenticated Voting Credentials

adversarial model:



authenticated voting credential:



- $\mathcal{A}$ 's vote does not match credential  $\Rightarrow$  invalid ballot
- the credential is authenticated by the vote
- *the credential cannot be re-purposed*

# Ferguson Credential Withdrawal

Public knowledge:  $n, v, g, h$ .

Private knowledge for  $\mathcal{B}$ :  $1/v = v^{-1} \pmod{\varphi(n)}$ .

$\mathcal{A}$

$$a_1, \gamma, \sigma \xleftarrow{\$} \mathbb{Z}_n^*$$

$$b \leftarrow \gamma^v a_1 g^\sigma \pmod n$$

$$a_2 \leftarrow H(b)$$

$$a \leftarrow a_1 a_2 \pmod n$$

$$c \leftarrow f(h^a) - \sigma$$

$\mathcal{B}$

$$\xrightarrow{b, c}$$

$$a_2 \leftarrow H(b)$$

$$\bar{A} \leftarrow (b a_2 g^c)^{1/v} \pmod n$$

$$\xleftarrow{\bar{A}}$$

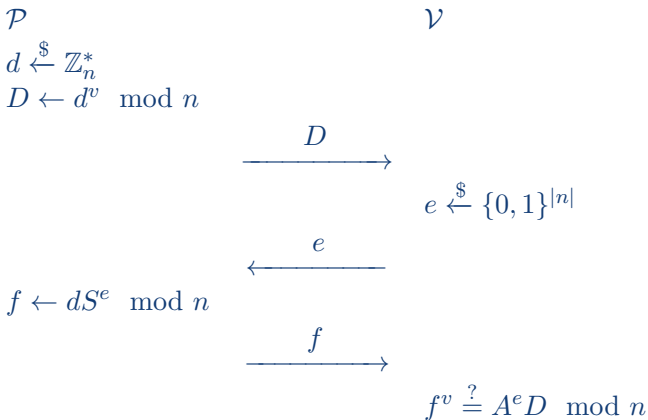
$$S \leftarrow \bar{A} \gamma^{-1} \pmod n$$

- credential:  $(S, a)$  such that  $S^v = a g^{f(h^a)} \pmod n$
- $\mathcal{B}$  learns no information on  $S$  or  $a$

## Guillou-Quisquater Proof

Public knowledge:  $n, v, A$ .

Private knowledge for  $\mathcal{P}$ :  $S$  s.t.  $S^v = A \pmod n$ .



- $S$  is kept secret
- spent credential:  $(a, D, e, f)$
- where  $e = \text{H}(A \parallel D \parallel v_i)$

## Conclusion

- voting formalism
- universal composability + voting
- formalism of universal verifiability
- Tally-Hiding Vote
  - Millionaire Problem
  - Ciphertext Lifting
- Self-Tallying Vote
- Authenticated Voting Credentials

## Conclusion

- voting formalism
- universal composability + voting
- formalism of universal verifiability
- Tally-Hiding Vote
  - Millionaire Problem
  - Ciphertext Lifting
- Self-Tallying Vote
- Authenticated Voting Credentials

|     | UV | UC |
|-----|----|----|
| THV | ✓  | ✓  |
| MP  | ✓  | ✓  |
| CL  | ✓  | ?  |
| STV | ✓  | ✓  |
| AVC | ✓  | ✓  |

## Conclusion

- voting formalism
- universal composability + voting
- formalism of universal verifiability
- Tally-Hiding Vote
  - Millionaire Problem
  - Ciphertext Lifting
- Self-Tallying Vote
- Authenticated Voting Credentials

|     | UV | UC |
|-----|----|----|
| THV | ✓  | ✓  |
| MP  | ✓  | ✓  |
| CL  | ✓  | ?  |
| STV | ✓  | ✓  |
| AVC | ✓  | ✓  |

sort of ...

## Conclusion

- voting formalism
- universal composability + voting
- formalism of universal verifiability
- Tally-Hiding Vote
  - Millionaire Problem
  - Ciphertext Lifting
- Self-Tallying Vote
- Authenticated Voting Credentials

} future work:  
cover all  
properties

|     | UV | UC |
|-----|----|----|
| THV | ✓  | ✓  |
| MP  | ✓  | ✓  |
| CL  | ✓  | ?  |
| STV | ✓  | ✓  |
| AVC | ✓  | ✓  |

sort of ...