Scytl Secure Electronic Voting

Mathematics in the Elections: beyond the simple counting of votes

November 2011

Jordi Puiggali
VP R&D
Jordi.Puiggali@scytl.com

scytl
Secure Electronic Voting

Index

- Scytl's R&D introduction
- Electronic voting
- Internet voting cryptography – Privacy
- Internet voting cryptography – Verifiability

Index

- Scytl's R&D introduction
- Electronic voting
- Internet voting cryptography – Privacy
- Internet voting cryptography – Verifiability

Origen de Scytl

1994

2001

2011

1994: Se inicia en la *UAB* la investigación en criptografía aplicada al voto electrónico

1996: 1ª Tesis doctoral: "*Design and Development of a Cryptographic Scheme to Perform Secure Eleccións over a LAN*"

1997: 1ª elección vinculante: Presidencia del capítulo Español de las *TIC* del *IEEE*

1999: 2ª Tesis doctoral "*Design of Implementable Solutions for Large Scale Electronic Voting Schemes*"

2000: Problema durante les elecciones presidenciales *EEUU* en Florida. Comienza a gestarse la idea de Scytl.

2001: Creación de *Scytl* como una spin-off del grupo de criptografía de la *UAB* mediante la *Fundación Empresa y Ciència* de dicha universidad. Capital inicial aportado por *FFF*. Inicio del desarrollo del producto y presentación de la primera patente.

2002: Entrada del fondo de capital riesgo (Riva y Garcia). Consolidación del equipo. Primera versión del producto utilizada en unas elecciones de voto remoto (UB).

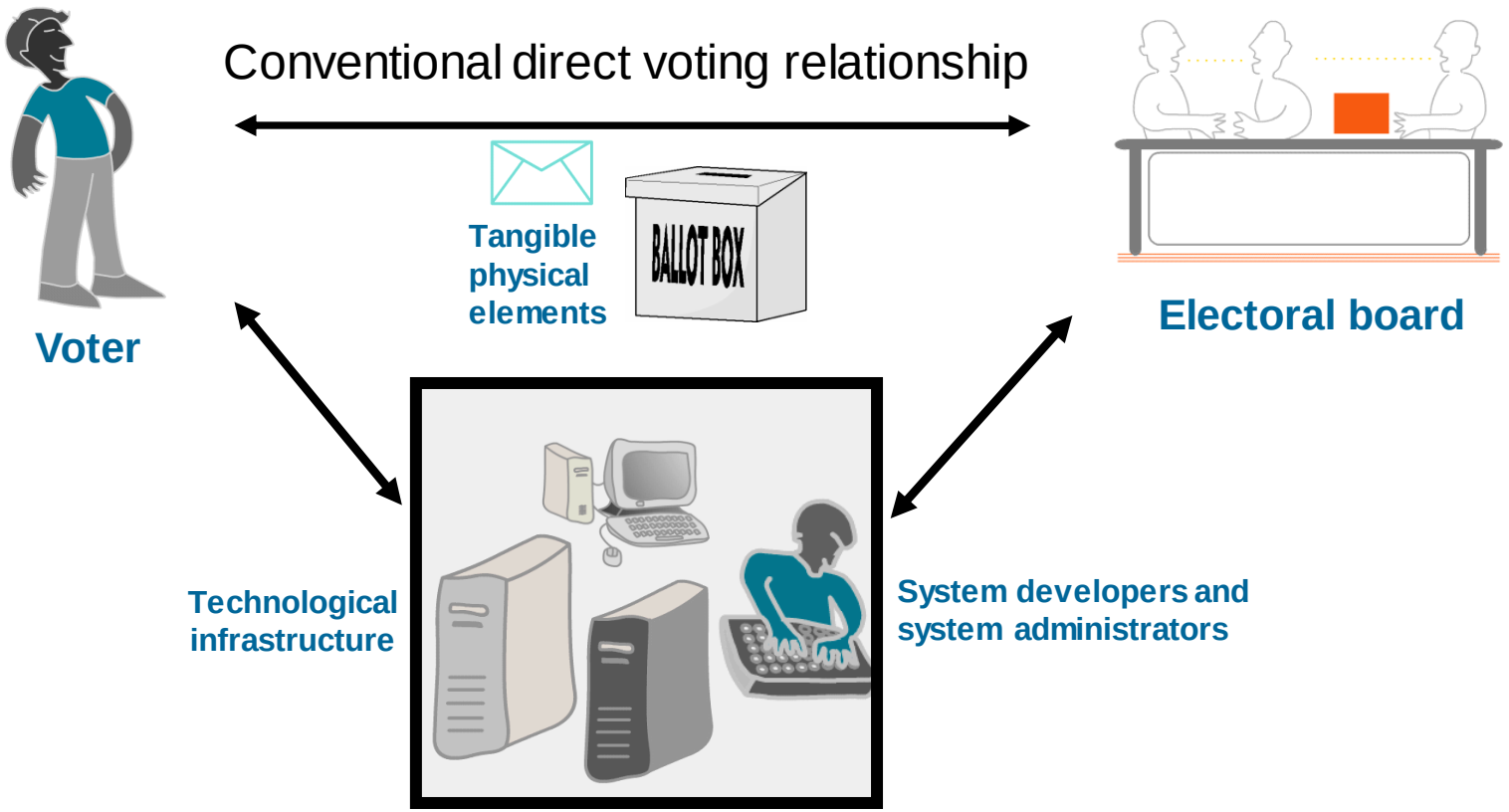
...

Actualidad: El equipo de investigación dispone de más de 30 publicaciones científicas internacionales en el campo de la seguridad en el voto electrónico. 3ª Tesis Doctoral en Scytl sobre seguridad en el voto electrónico y dos más en proceso. 24 patentes concedidas y más de 15 en proceso. Empresa con más referencias internacionales de voto electrónico remoto para procesos gubernamentales (más de 14 países).

Index

- Scytl's R&D introduction
- Electronic voting
- Internet voting cryptography – Privacy
- Internet voting cryptography – Verifiability

Electronic Voting: Change of voting paradigm



Electronic voting creates a new indirect voting relationship that brings **new security risks** that reduce the trustworthiness of the electoral process.

Electronic Voting: Security Requirements

- Specific security requirements that must be fulfilled by any secure voting protocol:
  - Strong authentication of voters
  - Voters privacy
  - Accuracy of election results
  - Secrecy of intermediate results
  - Verifiability
  - Prevention of coercion and vote-selling

scytl Secure Electronic Voting

Electronic Voting: Cryptographic Research

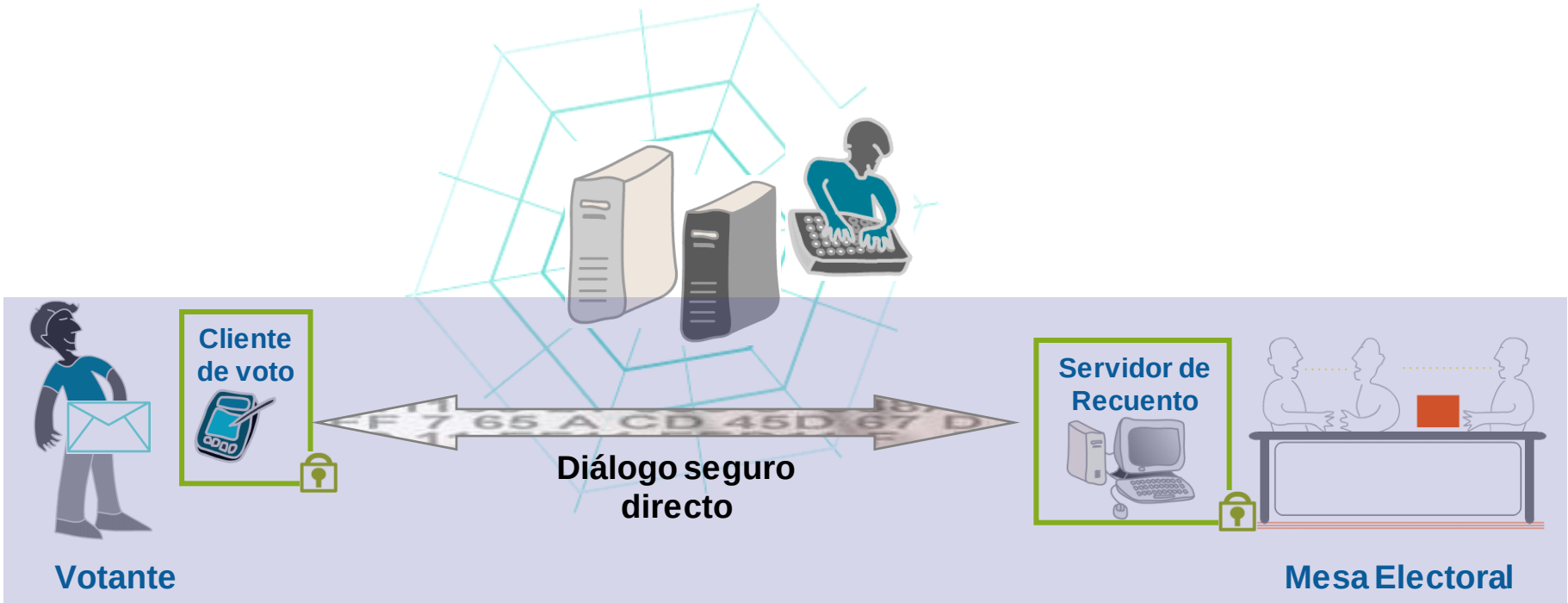
- Cryptography research is focused in two areas
  - Protect Voter Privacy
    - How to prevent any disclosure of the voting options selected by the voter
    - How to prevent any correlation between clear-text votes and voters
    - How to prevent coercion and vote buying
  - Enable End-to-end verifiability
    - How to allow voters to verify the proper recording and storage of their votes
    - How to allow observers and independent auditors to verify the proper behavior of the voting system

scytl
Secure Electronic Voting

Index

- Scytl's R&D introduction
- Electronic voting
- Internet voting cryptography – Privacy
- Internet voting cryptography – Verifiability

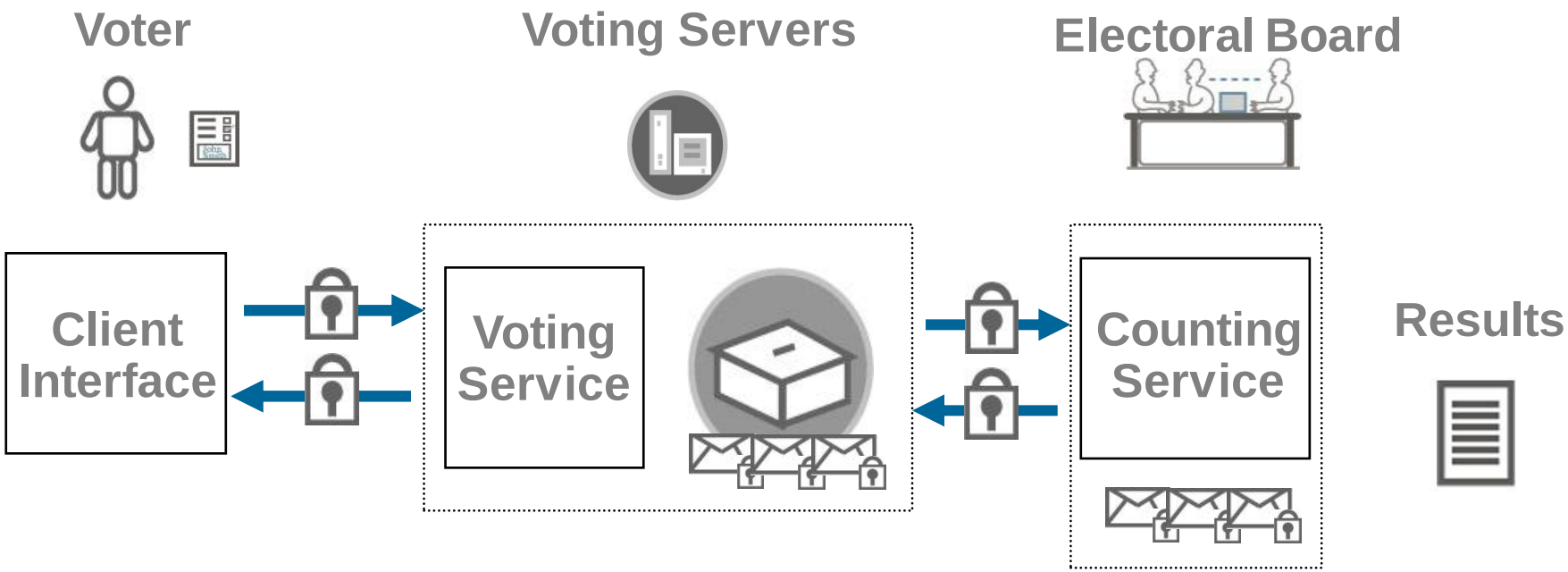
End-to-end security



The main objective of a secure architecture is to allow a **secure direct dialog** between voter and Electoral Board, protecting them from attacks comming from the IT infrastructe between them.

scytl
Secure Electronic Voting

Double envelope approach



Double Envelope is based in the concept used for postal voting:

- Inner envelope to protect vote privacy: vote encryption
- Outer envelope to identify voter: digital signature of encrypted vote

Encryption is based on digital Envelope:

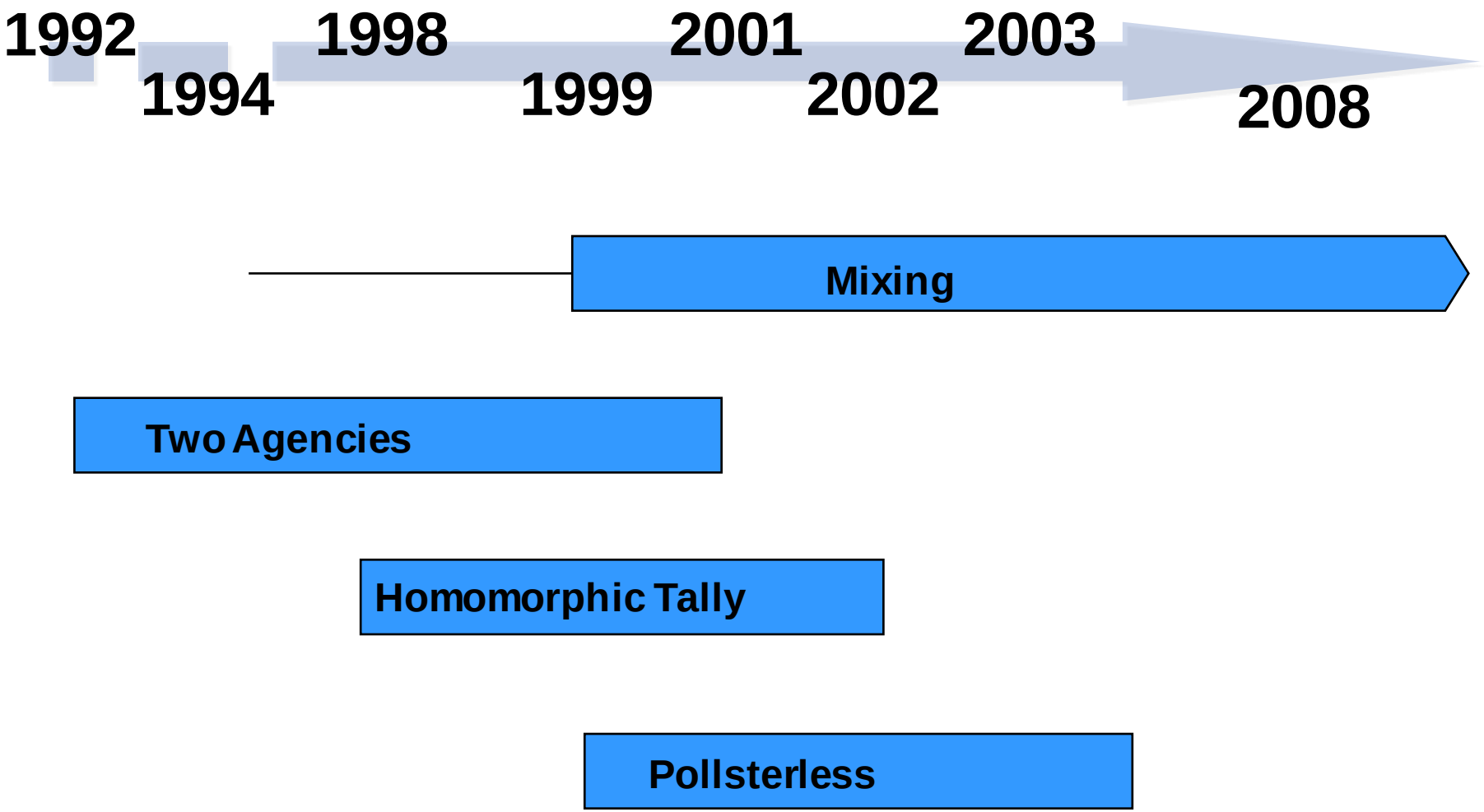
$E_k(vote), P_{EB}(k)$

Electronic Voting: Non-cryptographic protocol

- Strong authentication of voters ✔
  - Digital certificates
- Voters privacy ✘
  - Correlation between decrypted votes and voters
- Accuracy of election results !
  - It is possible to verify if votes have been cast by valid voters
  - Decryption process can provide different contents
- Secrecy of intermediate results ✔
  - Votes only can be decrypted by the Electoral Board
- Verifiability ✘
  - No means to verify if the votes is properly processed
- Prevention of coercion and vote-selling ✔
  - It does not provide information of the selected voting options

scytl Secure Electronic Voting

I-voting cryptographic schemes SoA



scytl Secure Electronic Voting

Blind signature

- Concept introduced by Chaum in 1982
- Allows a message to be digitally signed without knowing the message contents
- RSA reminder:
  - Public key: (e,n) – Private key: (d)
  - Signature *s* of message *m*: $s = m^d \bmod n$
  - Signature *s* verification: $s^e \bmod n = m$

scytl
Secure Electronic Voting

Blind signature (ii)

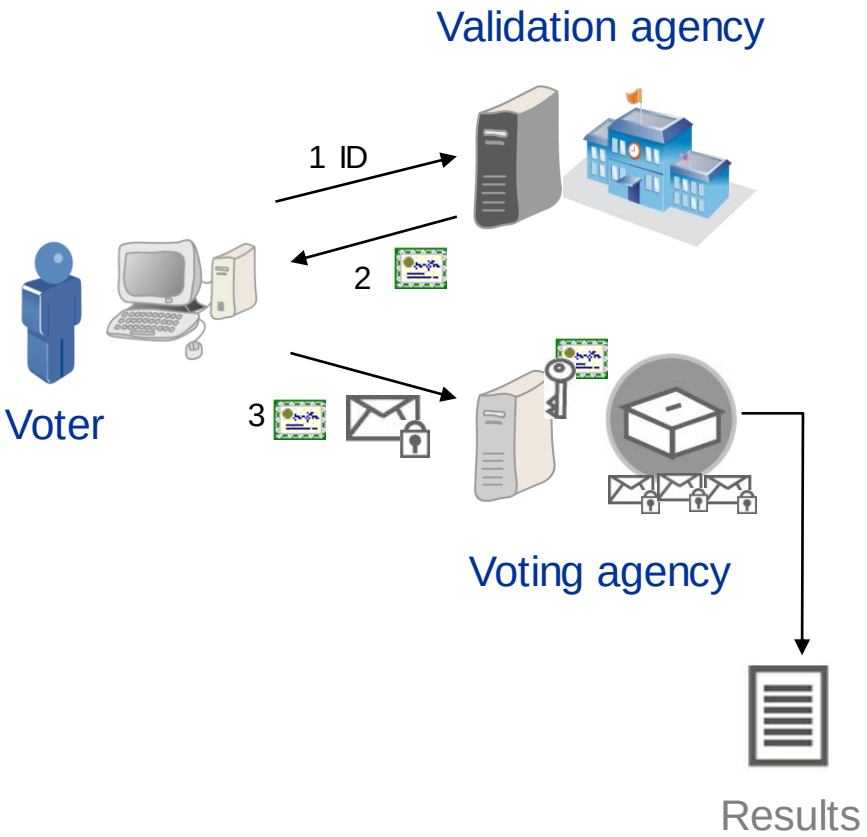
- RSA blind signature:
  - V implements the following steps:
    - Generates a random number r: $r \bmod n$
    - Blinds the message m:
      - $m' = m \cdot r^e \bmod n$
  - S receives m', and signs it:
    - $s' = (m')^d \bmod n = m^d (r^e)^d \bmod n = m^d \cdot r \bmod n$
  - V receives s' and retrieves the digital signature:
    - $s = s'/r = (m^d \cdot r)/r \bmod n = m^d \bmod n$

scytl Secure Electronic Voting

- **Two agencies features**

  - Split the voting process in two independent agencies
    - Validation agency: Authenticates the voter and issues an anonymous token for casting a vote
    - Voting agency: accepts encrypted votes if anonymous token is valid, digitally signs the vote and stores it

  - Encrypted votes are stored digitally signed by one of the agencies (depending on the implementation approach)



Two-agencies model - analysis

- Strong authentication of voters ✔
  - Digital certificates
- Privacy of voters !
  - Votes have no info of the votes
  - Assumes no collusion between the two agencies
- Accuracy of election results !
  - A malicious Validation Agency can cast valid votes
- Secrecy of intermediate results ✔
  - Votes only can be decrypted by the Electoral Board
- Verifiability ✘
  - No means to verify if the votes are properly processed
- Prevention of coercion and vote-selling ✔
  - It does not provide information of the selected voting options

scytl Secure Electronic Voting

Cryptographic voting protocols: Homomorphic tally model

- Objective: To obtain the final results of the election without decrypting the votes

- These models are based on the additive property of the homomorphic algorithms such as:
  - ElGamal
  - Paillier

- Can only be used with ballots that can be represented in a numerical form (e.g., the selection of a candidate is represented with a 1 and candidates not selected as 0s). It cannot be used in elections which support write-in responses.

scytl
Secure Electronic Voting

Additive properties of homomorphic algorithms

- The concept is based on the property that with these algorithms the results of operating two encrypted messages is the encryption of the result of operating these messages:
  - $P(m_1) \cdot P(m_2) = P(m_1 \, o \, m_2)$
- In the case of ElGamal the addition of two encrypted votes yields an encryption of the sum of the votes:

$$E(v_1) \oplus E(v_2) \equiv E(v_1 + v_2)$$

- Therefore, the addition of the encrypted votes (if these votes are represented in a numerical format) returns the encryption of the sum of the votes of each candidate (i.e., the encryption of the result)

scytl
Secure Electronic Voting

Using ElGamal as reference, we have the following components:

g generator of Z_p^*	p large prime p=2q+1
private key: x	x random number p
clave publica: (h, g, p)	$h = g^x \bmod p$
mensaje: m	

Encrypted message: $c = (a,b) = (m.h^w , g^w)$ *w* is a random number

Addition of encrypted votes

To encrypt a vote, these schemes assume that each voting option has a binary value *v* equal to 1 if the option has been selected or 0 if it hasn't. Following this rationale, votes are encrypted before raising a pre-defined root *g* to the value v. Assuming a vote with one option, the encryption will be done as:

Encrypted vote: $c = (g^{v}.h^{w} , g^{w})$ $v=\{1,0\}$

If two votes c' and c'' encrypted with the same public key are multiplied:

$c' = (a', b') = (g^{v'}.h^{w'}, g^{w'})$

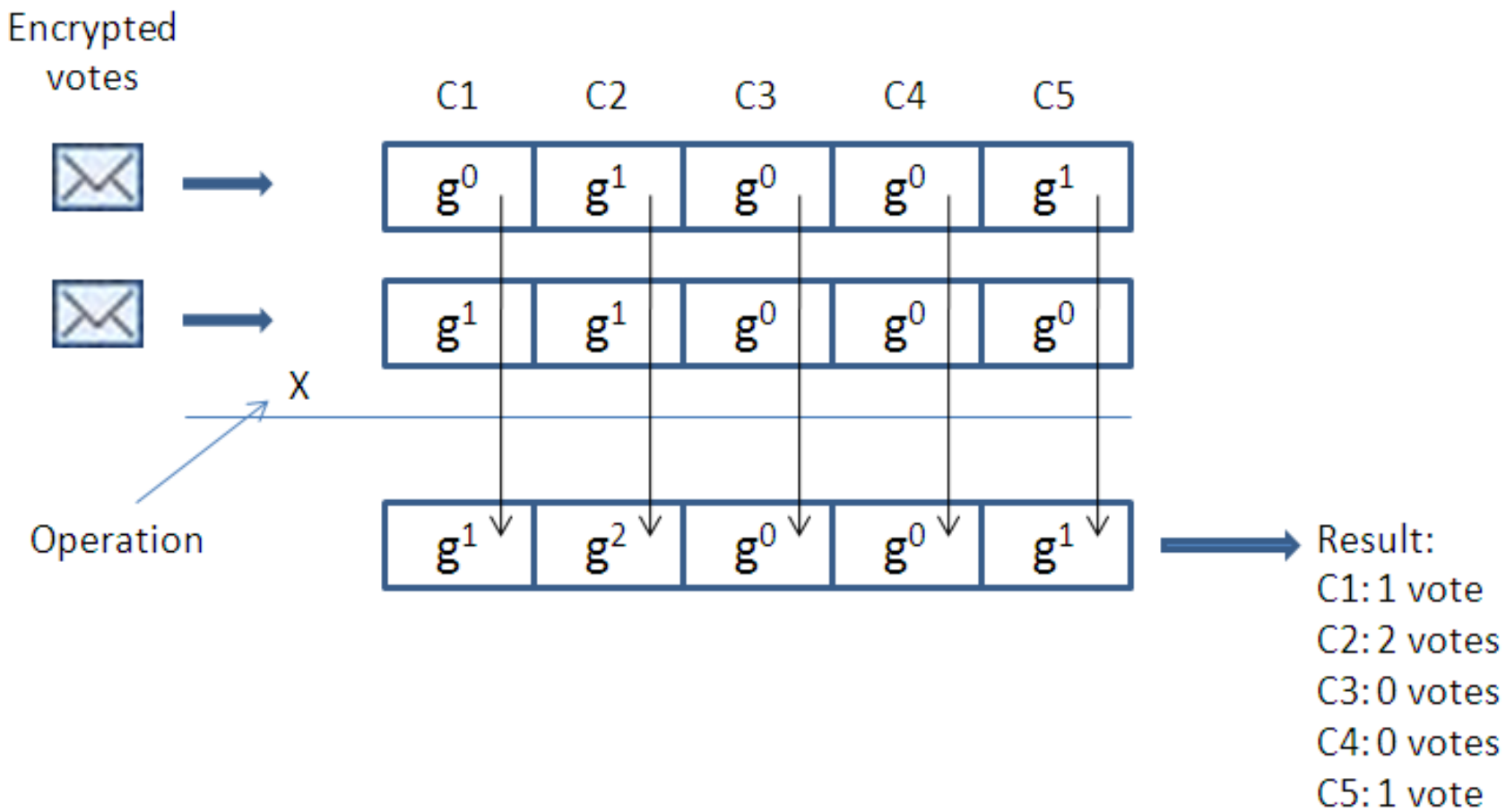
$c'' = (a'', b'') = (g^{v''}.h^{w''}, g^{w''})$

$c = c'. c'' = (a', b') . (a'',b'') = (g^{v'}.h^{w'}, g^{w'}). (g^{v''}.h^{w''} , g^{w''}) = (g^{v'+v''}.h^{w'+w''}, g^{w'+w''})$

Therefore, when the product is decrypted we obtain $g^{v'+v''}$ and making a logarithmic operation we retrieve $v'+v''$. I.e., the number of times a voting option has been selected.

scytl Secure Electronic Voting

Homomorphic Tally



scytl Secure Electronic Voting

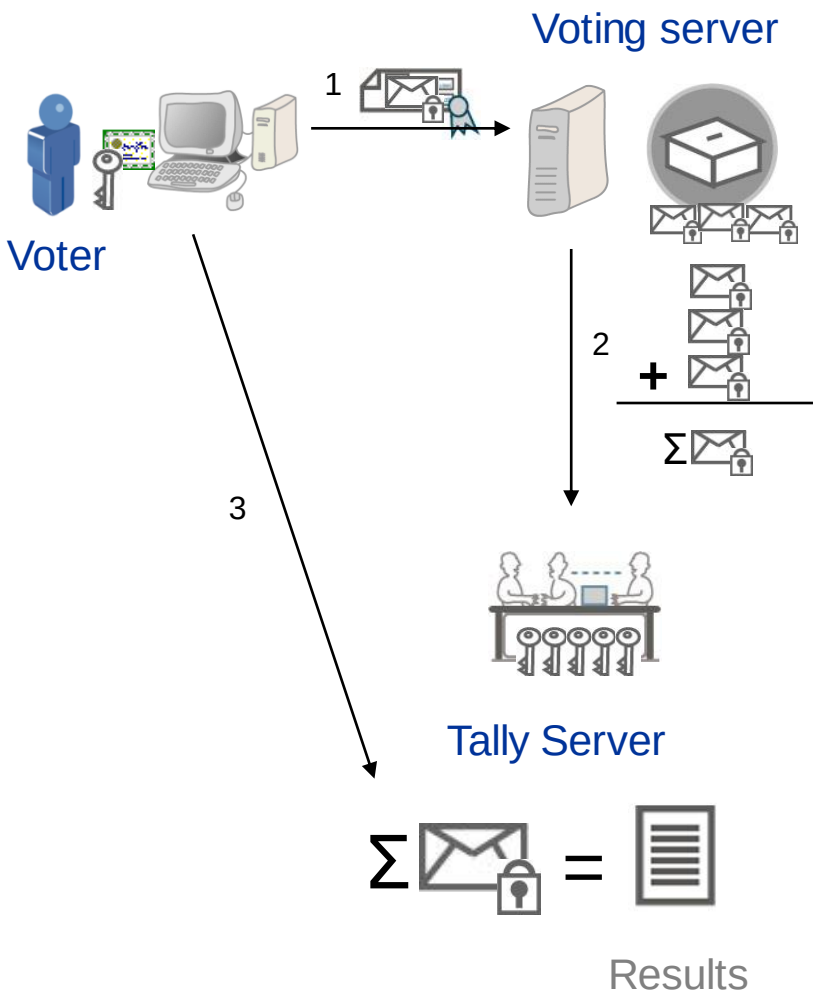
- **Homomorphic features**

  - Votes are encrypted by voters using a cryptographic algorithm with homomorphic properties (e.g., El Gammal)

  - Votes are digitally signed by voters before being cast

  - Encrypted votes are operated to obtain a encrypted result

  - To obtain the election result, only the encrypted result of the vote operation is decrypted instead the individual votes



Homomorphic tally model - analysis

- Strong authentication of voters ✔
  - Digital certificates
- Privacy of voters ✔
  - Vote are not individually decrypted
- Accuracy of election results !
  - Voters can use exponents different from 0 or 1
  - Can be solve using ZKP of content
- Secrecy of intermediate results ✔
  - Votes only can be decrypted by the Electoral Board
- Verifiability !
  - No means to verify if the votes are properly processed
  - Can be solved with ZKP of correct decryption
- Prevention of coercion and vote-selling ✔
  - It does not provide information of the selected voting options

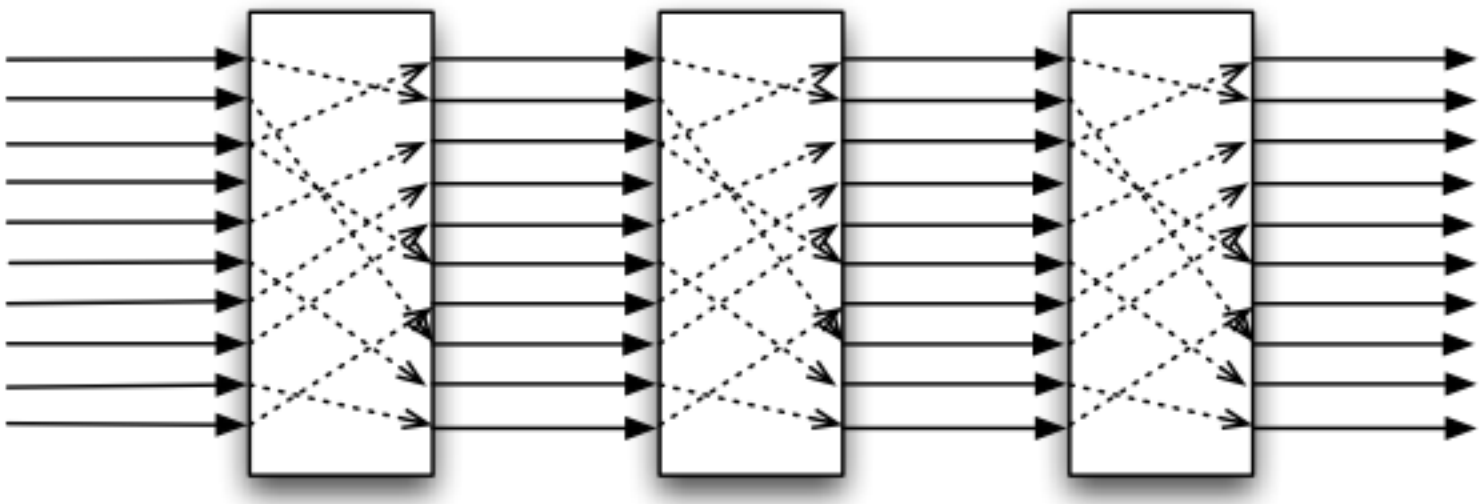
scytl Secure Electronic Voting

Cryptographic voting protocols: Mixing model

- Objective: Break the correlation between the encrypted votes (i.e., voters) and the decrypted votes
- This protocols can use voting receipts to facilitate the voter verification of the results
- Two different types of Mixing
  - Decryption Mixing
  - Re-encryption Mixing

Mixing concept

- This concept is based on encrypting/decrypting the votes and shuffling them at the same time.



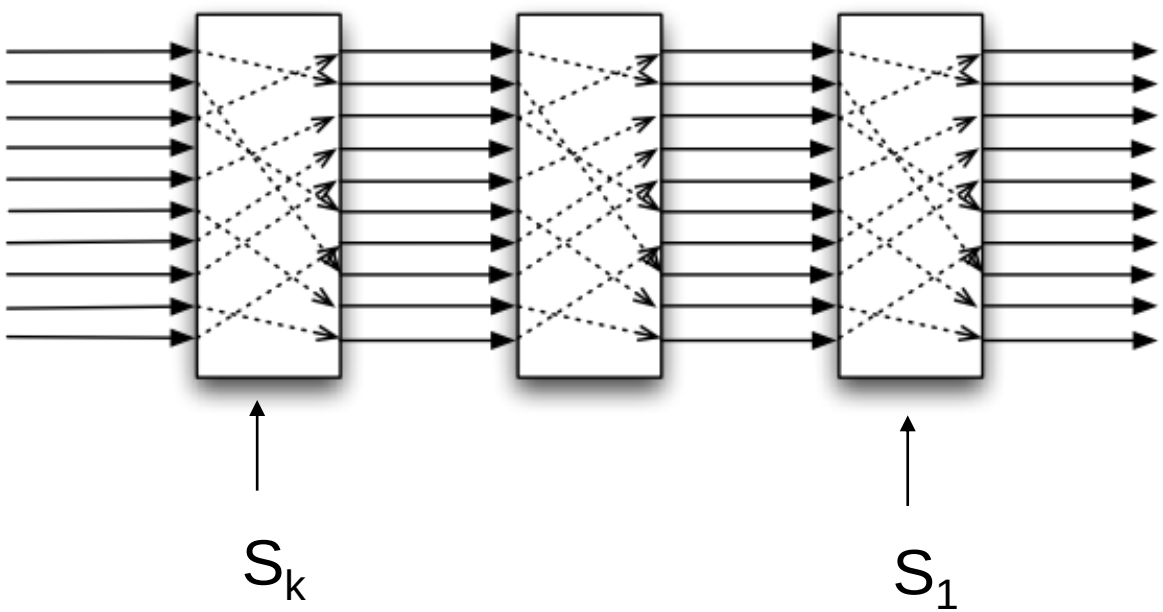
- This process can be implemented in different nodes (Mix-net)
  - Decryption Mixing:
    - Each node partially decrypt the vote during shuffling
  - Re-encryption Mixing:
    - Each Mix-net node re-encrypts and shuffles the encrypted votes using the same Electoral Board private key
    - At the end of the Mix-net, the Electoral Board decrypts the votes using the private key only once

Decryption Mix-net

- Decryption Mixing:
  - Votes are nested encrypted by voters using the public key of each of the Mix-net nodes, following the inverse order of the Mix-net.

$$V_i^k = P_k(r_k \mid P_{k-1}(r_{k-1} \mid .....P_1(r_1 \mid V_i))...)$$

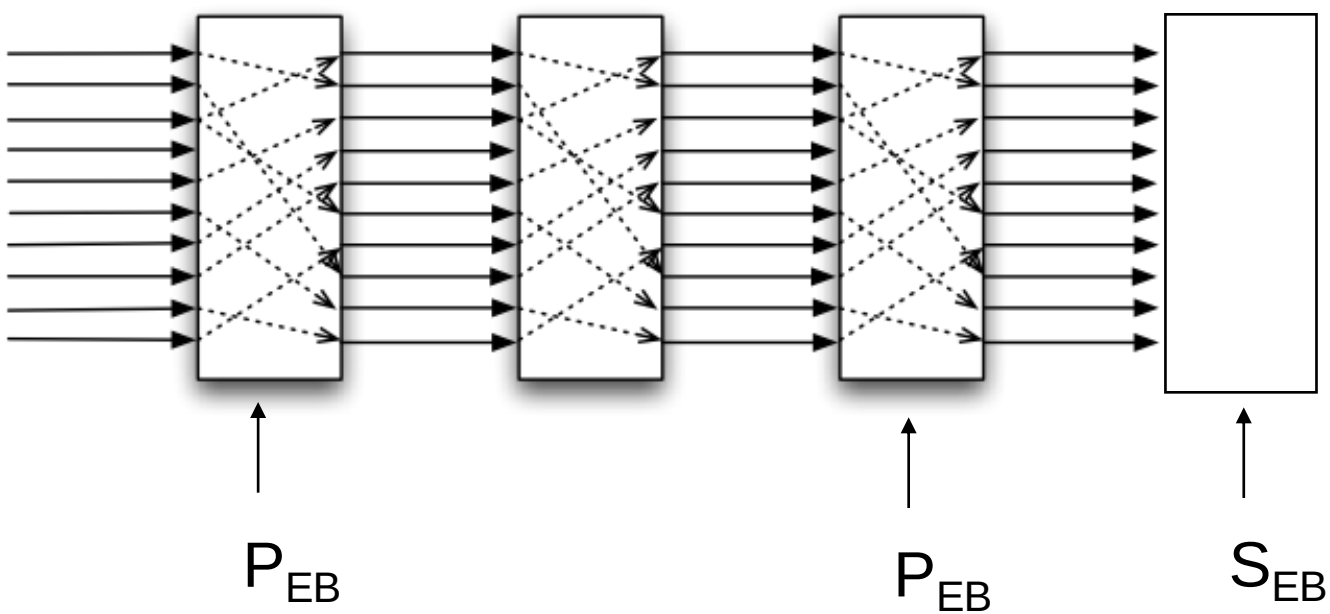
  - Each Mix-net node decrypts and shuffles the votes.



Re-encryption Mix-net

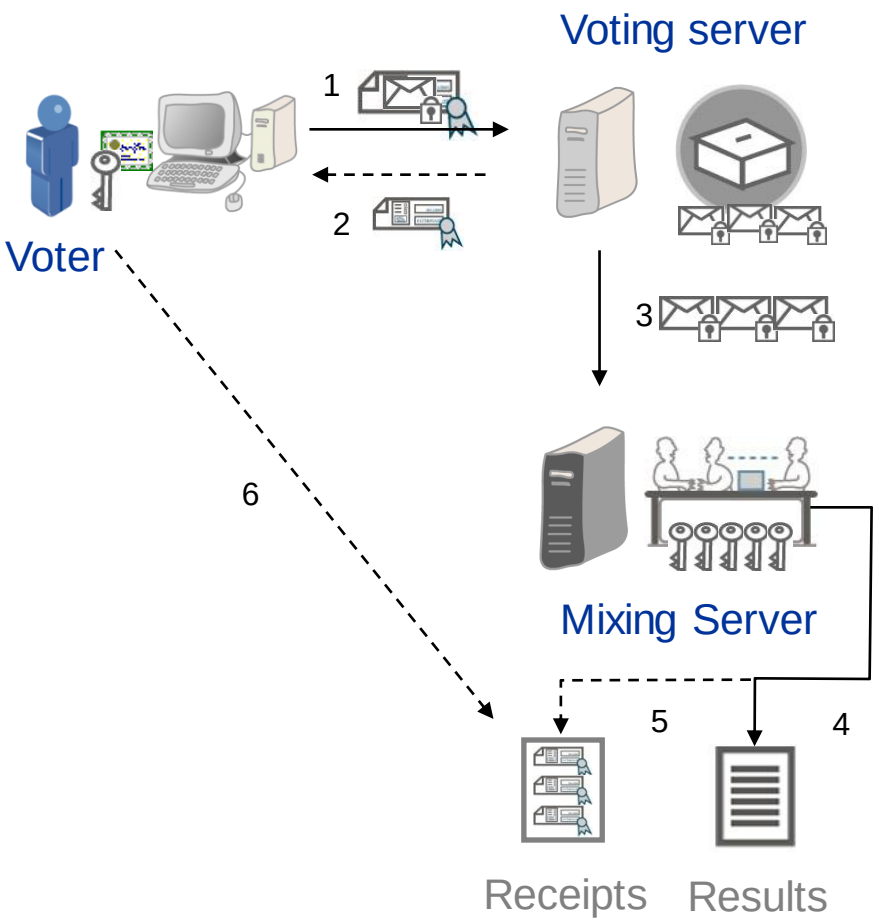
- Re-encryption Mixing:
  - It uses the properties of homomorphic algorithms: re-encrypting one vote with the same public key does not require multiple decryptions of the vote with the private key, only one.

    $c = c'.(1.h^{w''}, g^{w''}) = (m'.h^{w'}, g^{w'}).(1.h^{w''}, g^{w''}) = (m'.h^{w'+w''}, g^{w'+w''})$
  - Votes are initially encrypted by voters using the Electoral Board public key
  - Each Mix-net node re-encrypts and shuffles the encrypted votes using the same Electoral Board private key
  - At the end of the Mix-net, the Electoral Board decrypts the votes using the private key only once



scytl Secure Electronic Voting

- **Mixing features**
  - Encrypted votes are digitally signed by voter's digital certificate
  - Voting server accepts votes digitally signed by an eligible voter
  - Mixing server shuffles and decrypts the votes in an isolated environment



Mixing model - analysis

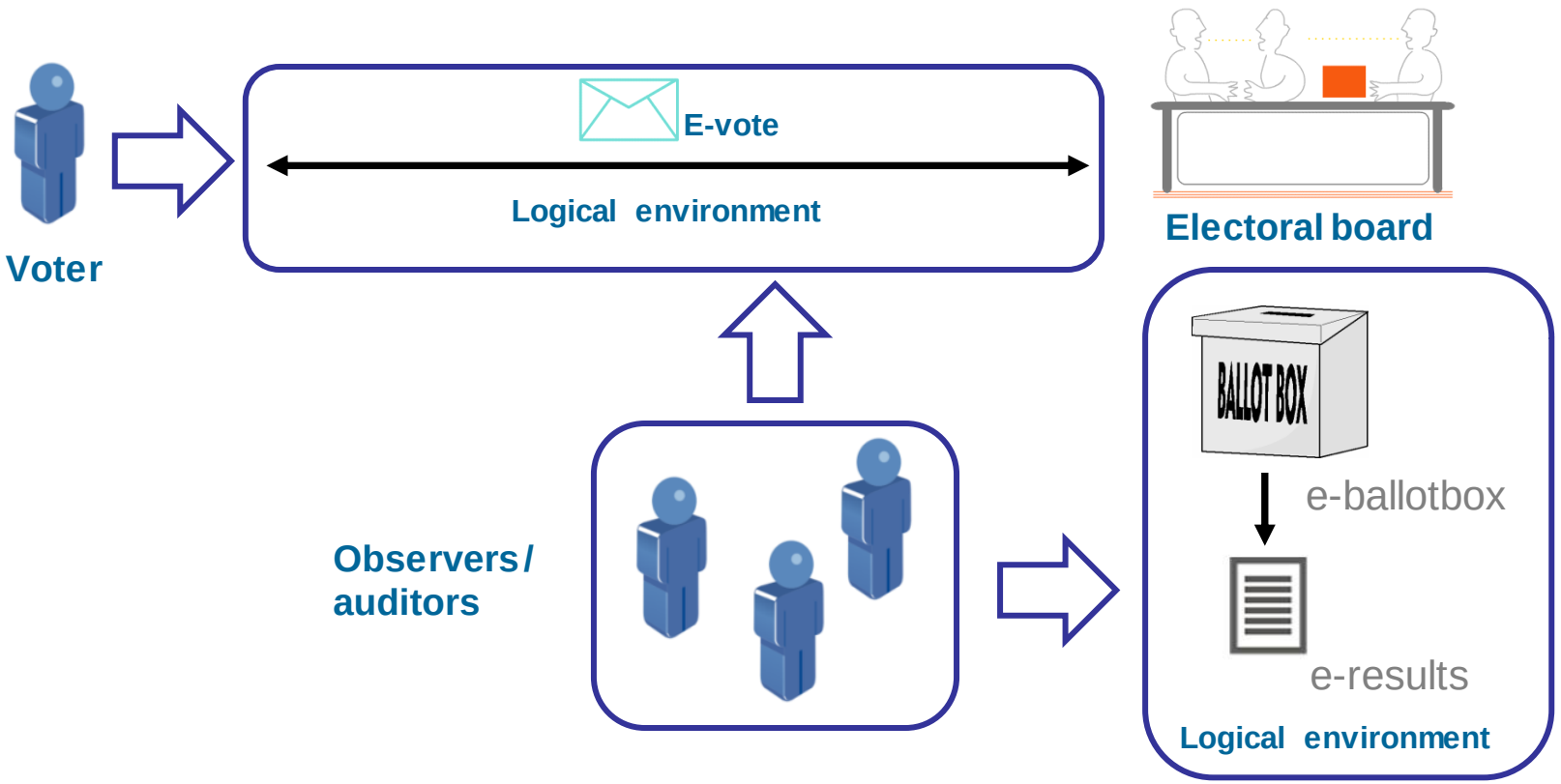
- Strong authentication of voters ✔
  - Digital certificates
- Privacy of voters ✔
  - Assumes at least one node is honest
- Accuracy of election results !
  - If one node is malicious, it can change votes
  - Can be solved using ZKP
- Secrecy of intermediate results ✔
  - Votes only can be decrypted by the Electoral Board
- Verifiability !
  - No means to verify if the votes are properly processed
  - Can be solved with ZKP
- Prevention of coercion and vote-selling ✔
  - It does not provide information of the selected voting options

scytl Secure Electronic Voting

Index

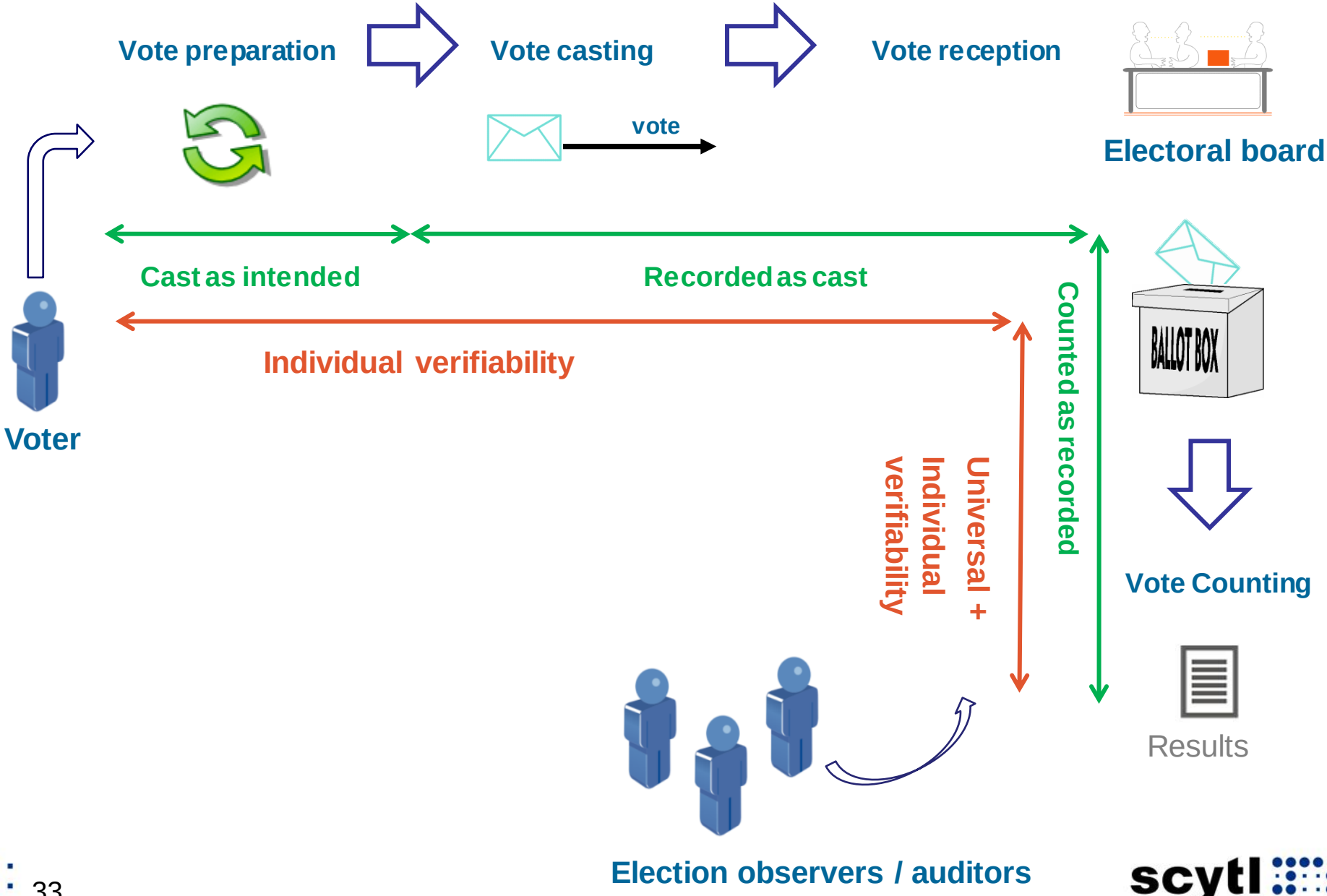
- Scytl's R&D introduction
- Electronic voting
- Internet voting cryptography – Privacy
- Internet voting cryptography – Verifiability

Auditability in e-voting



- Votes and processes are happening in a logical dimension:
  - Audit cannot be done by human means.
  - Difficult to monitor the behavior of other observers.

Verifiability and election processes

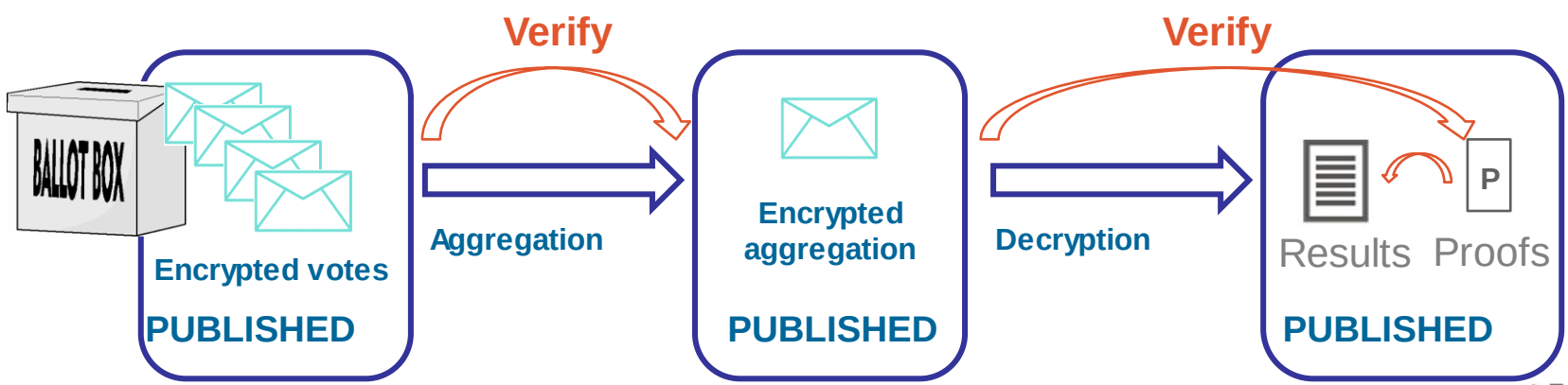


scytl Secure Electronic Voting

Homomorphic tally

- **Zero Knowledge Proof of correct decryption**, based on the equality of discrete logarithms:
  - Remember $c = (a, b) = (m \cdot h^w, g^w)$. Decryption recovers m using the private key x.
  - Given a tuple (g, b, h, v), where v: encryption factor $h^w = a / m$.
  - The prover can prove that he knows the secret value w satisfying $w = \log_g h = \log_b v$, without giving this value w.

- **Verification:**
  - Anyone can calculate the result of the operation using the encrypted votes.
  - The process generates proofs of correct decryption of the result that can be verified by anyone.



Homomorphic tally model - analysis

- Strong authentication of voters ✔
  - Digital certificates
- Privacy of voters ✔
  - Vote are not individually decrypted
- Accuracy of election results ✔
  - Solved using ZKP of correct decryption and ZKP of correct contents
- Secrecy of intermediate results ✔
  - Votes only can be decrypted by the Electoral Board
- Verifiability ✔
  - Solved using ZKP of correct decryption and ZKP of correct contents
- Prevention of coercion and vote-selling ✔
  - It does not provide information of the selected voting options

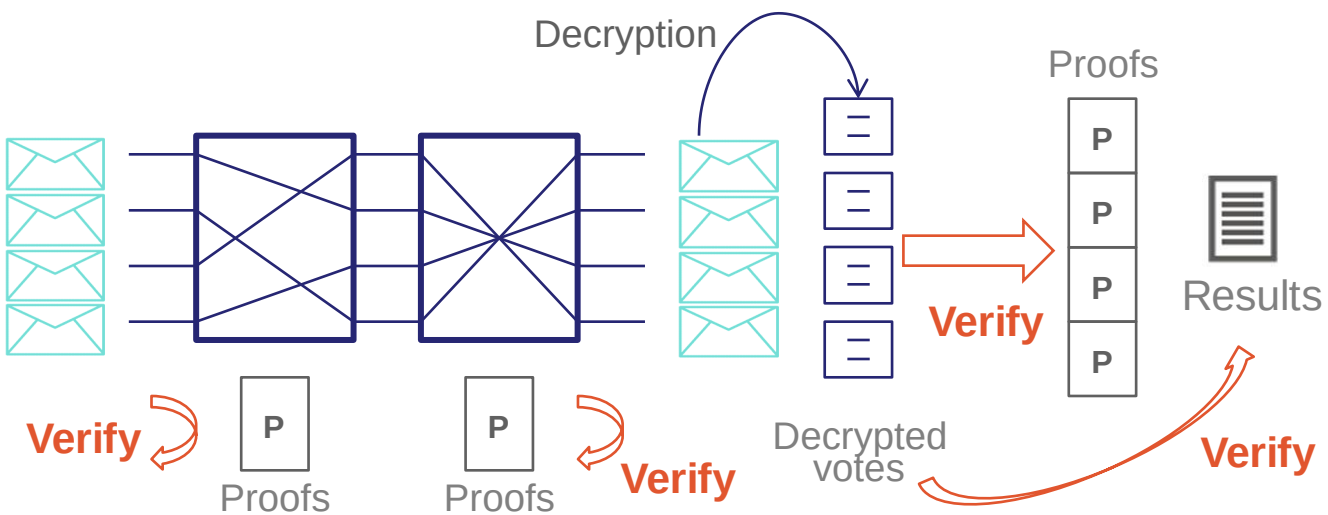
scytl Secure Electronic Voting

Universal verifiable Mix-nets (1/2)

- **Zero Knowledge Proof of plaintext equivalence** to demonstrate the correct re-encryption, based on the equality of discrete logarithms:
  - At one node, input is $c = (a, b) = (m \cdot h^w, g^w)$. Output is $c' = (a', b') = (m \cdot h^{w+w'}, g^{w+w'})$.
  - Given a tuple (g, u, h, v), where $u = b'/b = g^{w'}$, and $v = a'/a = h^{w'}$.
  - The prover can prove that he knows the secret value w' satisfying $w' = \log_g u = \log_h v$, without giving this value w'.
- **Zero Knowledge Proof of correct decryption**, based on the equality of discrete logarithms:
  - Remember $c = (a, b) = (m \cdot h^w, g^w)$. Decryption recovers m using the private key x.
  - Given a tuple (g, b, h, v), where v: encryption factor $h^w = a / m$.
  - The prover can prove that he knows the secret value w satisfying $w = \log_g h = \log_b v$, without giving this value w.

Universal verifiable Mix-nets (2/2)

- **Verification:**
  - Each mix-node calculates proofs of correct shuffling and correct re-encryption / decryption.
  - All the proofs are verifiable by anyone to detect that the input and output votes are based on the same original plaintexts (i.e., have not been changed).



Mixing model - analysis

- Strong authentication of voters ✔
  - Digital certificates
- Privacy of voters ✔
  - Assumes at least one node is honest
- Accuracy of election results ✔
  - Solved using ZKP of correct decryption and ZKP of correct re-encryption (Scytl's approach)
- Secrecy of intermediate results ✔
  - Votes only can be decrypted by the Electoral Board
- Verifiability ✔
  - Solved using ZKP of correct decryption and ZKP of correct re-encryption (Scytl's approach)
- Prevention of coercion and vote-selling ✔
  - It does not provide information of the selected voting options

scytl Secure Electronic Voting

www.scytl.com