# RECSI 2010

## XI REUNIÓN ESPAÑOLA SOBRE CRIPTOLOGÍA Y SEGURIDAD DE LA INFORMACIÓN

# Eficiencia y Privacidad en una Mixnet Universalmente Verificable

Septiembre 2010

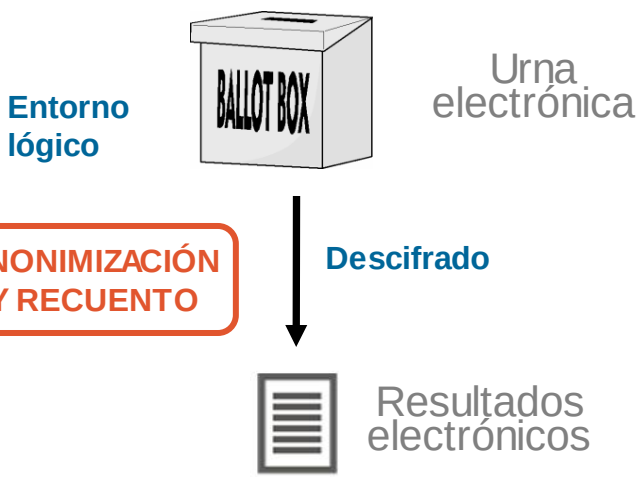
Sandra Guasch
Researcher
Sandra.Guasch@scytl.com

scytl
Secure Electronic Voting

# Índice

# Verificación en el voto electrónico

Procesos a verificar:


- Características de los procesos de verificación de la fase de recuento:
  - Corrección/precisión: alta probabilidad de detectar un mal comportamiento.
  - Privacidad de los votantes: prevenir la correlación entre los votos cifrados y los descifrados.
  - Eficiencia: reducir la cantidad de operaciones necesarias para generar y verificar las pruebas criptográficas.

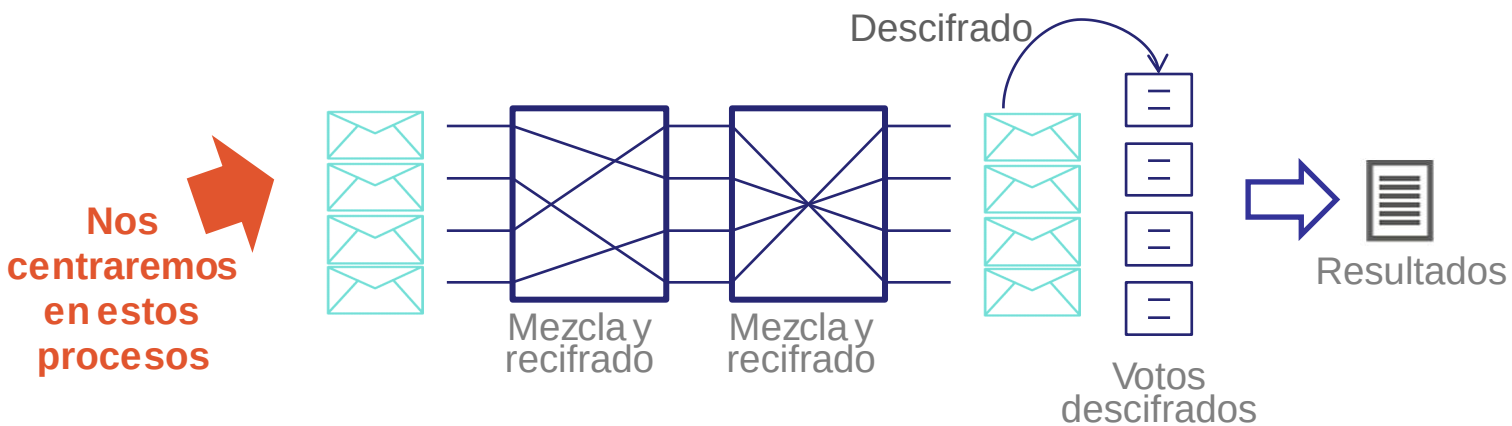
# Anonimización del voto (I)

- Recuento homomórfico:
  - Los votos cifrados se operan. El resultado es el cifrado de la agregación de los votos (cifrado con propiedades homomórficas).
  - El resultado de la agregación se descifra para obtener los resultados de la elección.


  - Problemas de flexibilidad:
    - Soporta formatos de voto muy específicos (e.g., no acepta preguntas abiertas).
  - Problemas de escalabilidad:
    - El número de operaciones criptográficas es proporcional al número de candidatos.

# Anonimización del voto (II)

- Mixnets:
  - Compuestas por varios nodos. Cada nodo mezcla y transforma (recifrado o descifrado) los votos de entrada.
  - Los votos se han separado previamente de sus firmas digitales.
  - No podemos asegurar, examinando las entradas y las salidas de un nodo, que no se ha modificado el valor de algún voto.




  - Proceso de verificación más complejo que en el recuento homomórfico, pero mayor flexibilidad en el formato del voto y más escalabilidad.

  - La verificación del proceso debe ser **universal**:
    - Enfocada al público, auditores y votantes en general.

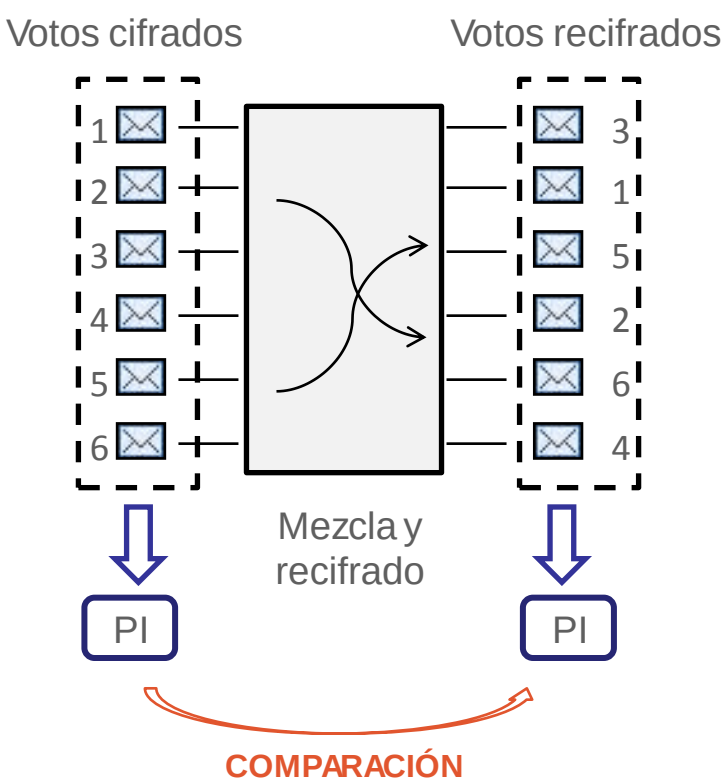
# Índice

- Introducción
- Mixnets universalmente verificables
- Descripción de la propuesta
- Propiedades
- Cifrado eficiente de los votos
- Conclusiones

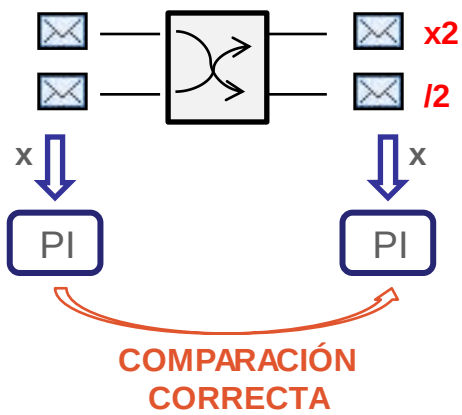
# Mixnets universalmente verificables

- Propuestas actuales:
  - Sako y Kilian [SK95], Furukawa y Sako [FS01], y Neff [Ne01]:
    - Buena precisión, preservan la privacidad de los votantes.
    - Poca eficiencia en elecciones con muchos participantes.
  - Random Partial Checking [JJR02]
    - Sacrifica principalmente privacidad, y también precisión, para aumentar la eficiencia del proceso.
  - Optimistic Mixing [Go02]
    - Preserva la privacidad de los votantes a costa de empeorar la precisión, para aumentar la eficiencia del proceso.
  - Almost entirely correct mixing [BG02]
    - Reduce privacidad y precisión para aumentar la eficiencia del proceso.
    - Resultados buenos con grandes cantidades de votos.
- Nuestra propuesta intenta resolver las limitaciones de las propuestas anteriores.

- Caso concreto: *Optimistic Mixing*
  - Realiza pruebas de integridad sobre los votos cifrados a la entrada y a la salida de cada nodo, para comprobar que los contenidos no se han modificado.




- Problemática:
  - Hay manipulaciones que no se detectan:




- Proponemos realizar las PI por grupos para reducir la probabilidad de no detección.



scytl Secure Electronic Voting

# Índice

- Introducción
- Mixnets universalmente verificables
- Descripción de la propuesta
- Propiedades
- Cifrado eficiente de los votos
- Conclusiones

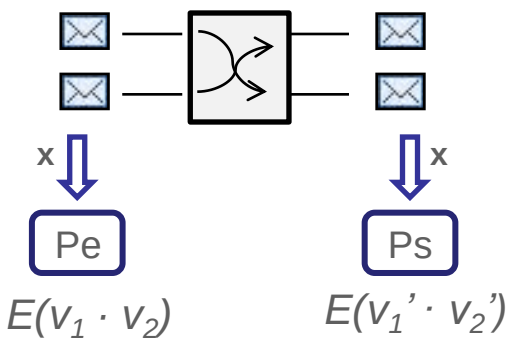
## Primitivas criptográficas (I)

- Criptosistema ElGamal para cifrar los votos:
  - Clave pública: $h = g^x \bmod p$, $x$ = clave privada, $g$ = generador elementos $Z_p^*$.
  - Cifrado: $c = (c_1, c_2) = (v \cdot h^w \bmod p, g^w \bmod p)$.
  - Descifrado: $v = c_1 \cdot c_2^{-x} \bmod p$.

- Mixnet de recifrado:
  - Recifrado ElGamal:
    - Voto cifrado a la entrada de un nodo: $c = (v \cdot h^w, g^w)$
    - Voto recifrado a la salida del nodo: $c' = c \cdot (1 \cdot h^{w'}, g^{w'}) = (v \cdot h^w, g^w) \cdot (1 \cdot h^{w'}, g^{w'}) = (v \cdot h^{w+w'}, g^{w+w'})$
  - No se necesitan operaciones adicionales de descifrado.

## Primitivas criptográficas (II)

- Operación homomórfica de los votos:
  - Permite generar pruebas de integridad sobre grupos de votos cifrados sin conocer sus contenidos.
  - Propiedad homomórfica: $E(v_1) \Phi E(v_2) = E(v_1 \ \theta \ v_2)$




- Prueba de conocimiento nulo de recifrado:
  - Permite verificar que el resultado de la operación de recifrado preserva el valor de los votos cifrados originalmente, sin conocer este valor.
  - Prueba de Chaum-Pedersen o Protocolo de Identidad de Schnorr.
  - Prueba que: $E(E(v_1 \cdot v_2)) = E(v_1' \cdot v_2')$

## Pasos del proceso de mixing y verificación

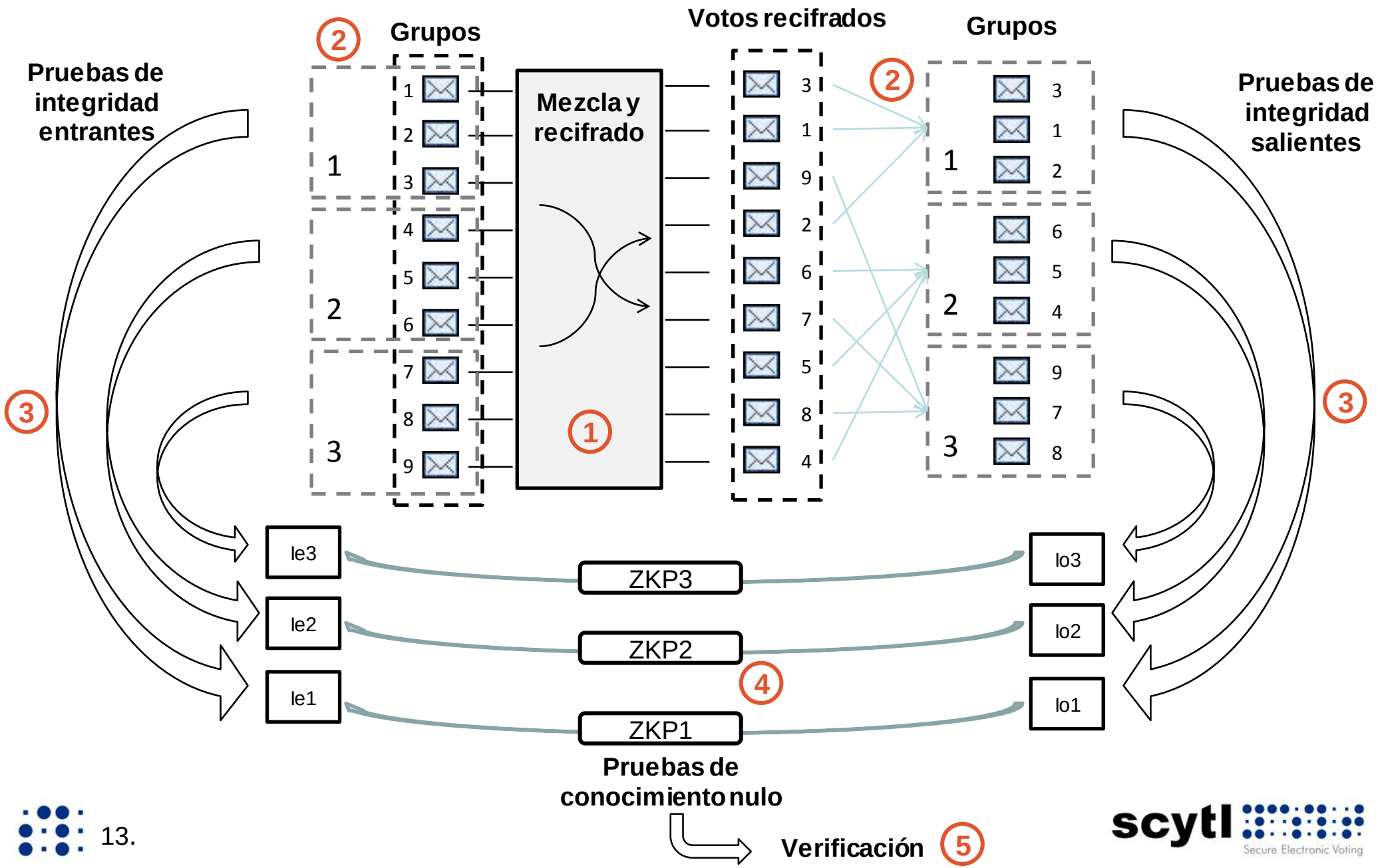
- Mixing de los votos: (1)
  - Los votos cifrados se recifran y mezclan utilitzando una Mixnet de recifrado.

- Petición de pruebas a la Mixnet:
  - El auditor define grupos sobre los votos de entrada de la Mixnet. (2)
  - Los nodos de la Mixnet generan pruebas de integridad de entrada y salida sobre los grupos de votos. (3)
  - Cada nodo genera pruebas de conocimiento nulo para demostrar que las pruebas de integridad de salida son el recifrado de las pruebas de integridad de entrada. (4)

- Verificación de las pruebas:
  - Las pruebas de integridad y las pruebas de conocimiento nulo se verifican antes de proceder al descifrado de los votos. (5)

- La información proporcionada por los nodos de la Mixnet (entradas, salidas, pruebas de integridad y pruebas de conocimiento nulo) puede ser almacenada para realizar verificaciones posteriores.

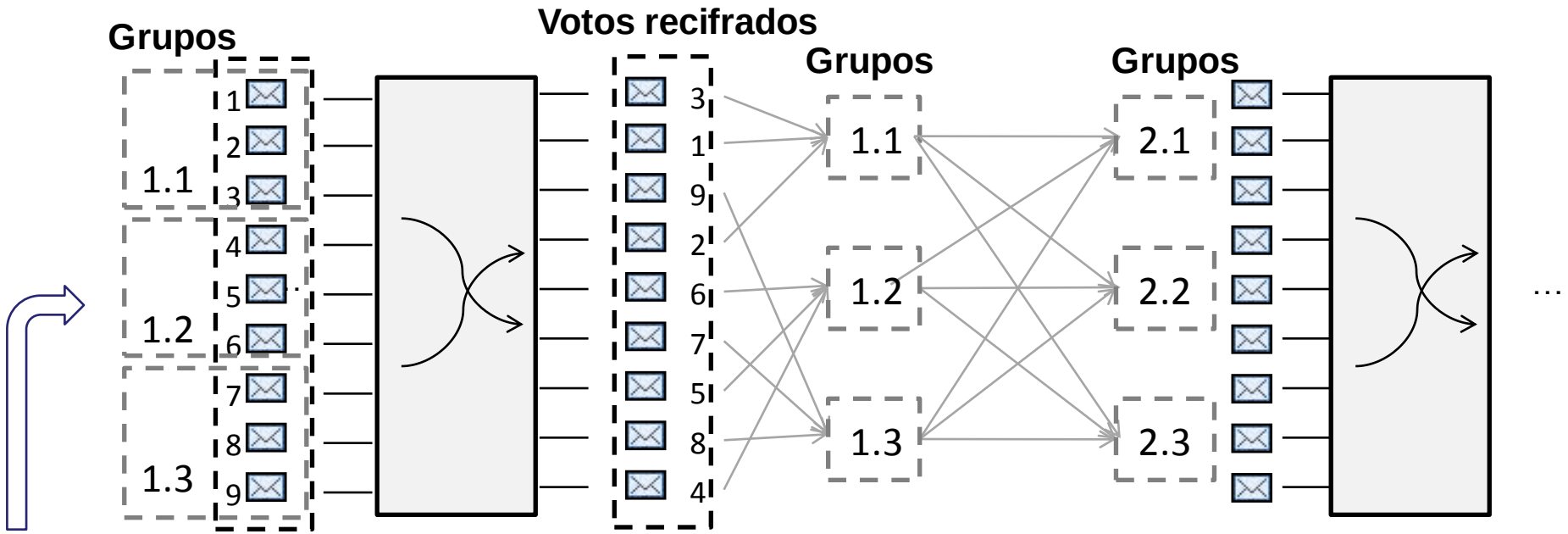
# Funcionamiento del proceso de verificación






scytl Secure Electronic Voting

# Definición de los grupos




**Auditor**

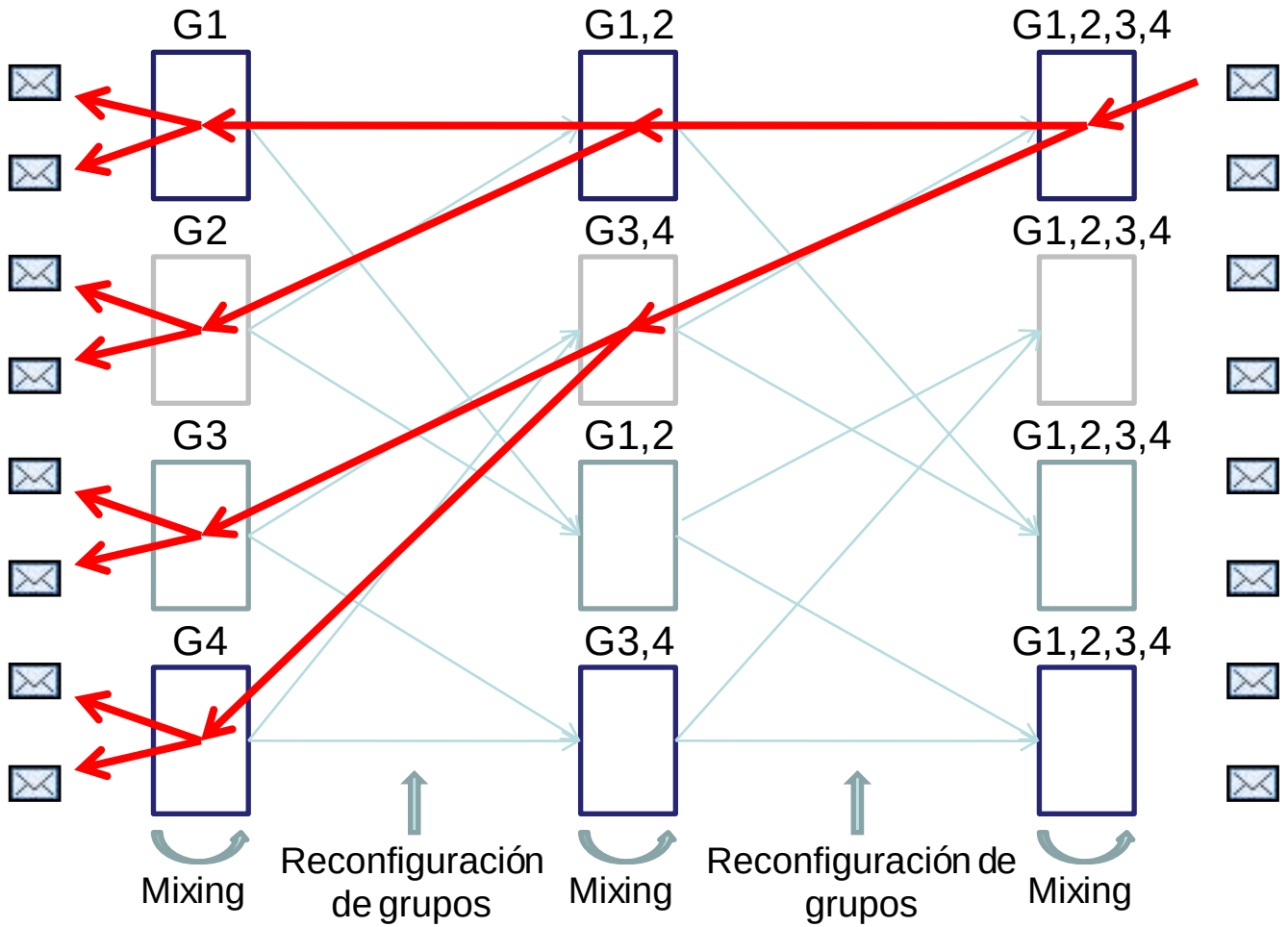
- Para mantener la privacidad, los grupos en el último nodo deben estar formados por al menos un voto de cada uno de los grupos en el primer nodo.
- El tamaño mínimo de los grupos debe ser:

$$n = \sqrt[t]{m}$$

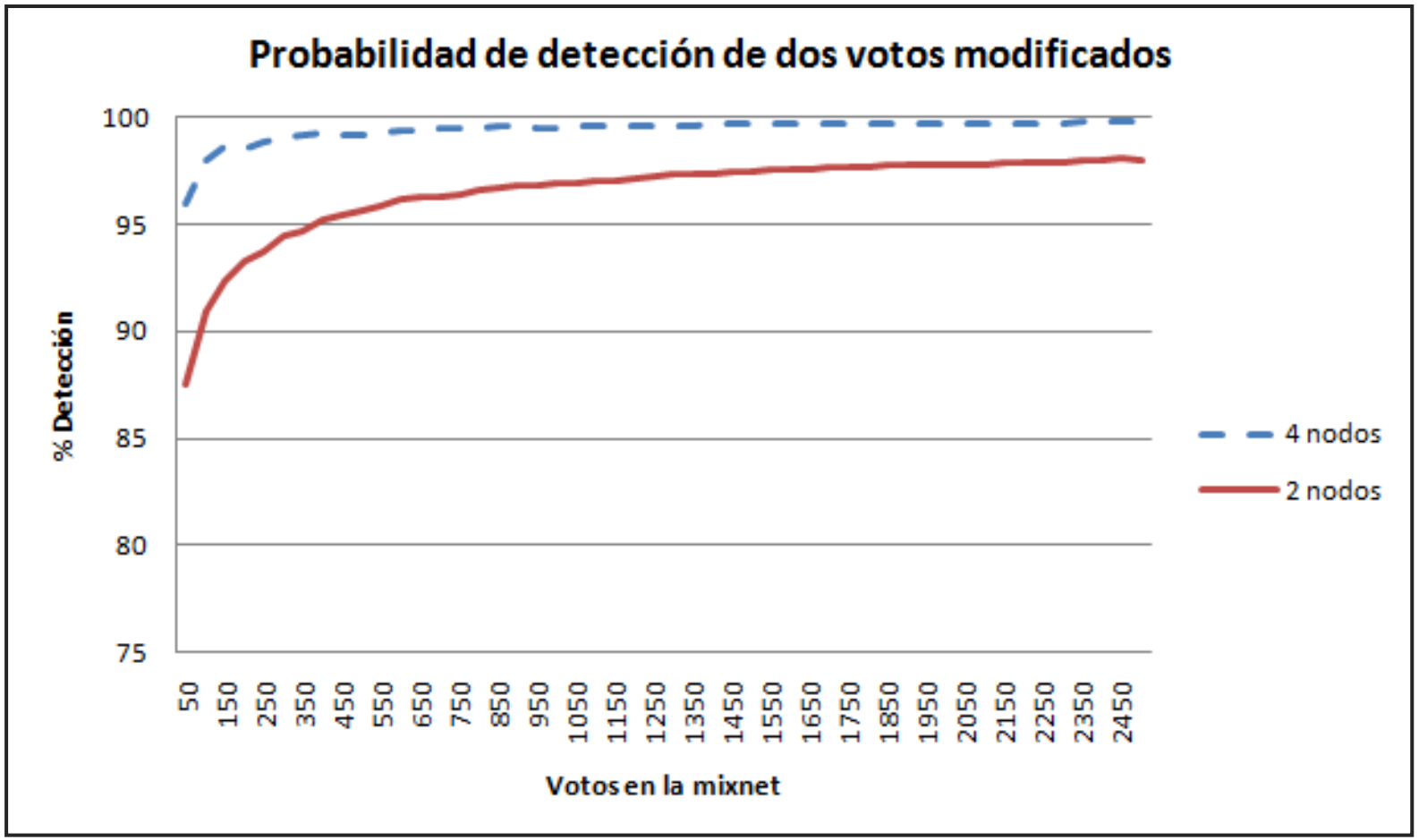
t es el número de nodos de la Mixnet, y m es el número total de votos

# Índice

- Introducción
- Mixnets universalmente verificables
- Descripción de la propuesta
- Propiedades
- Cifrado eficiente de los votos
- Conclusiones


G1
G1,2
G1,2,3,4
G2
G3,4
G1,2,3,4
G3
G1,2
G1,2,3,4
G4
G3,4
G1,2,3,4
Mixing
Reconfiguración de grupos
Mixing
Reconfiguración de grupos
Mixing

# Probabilidad de detección de modificaciones



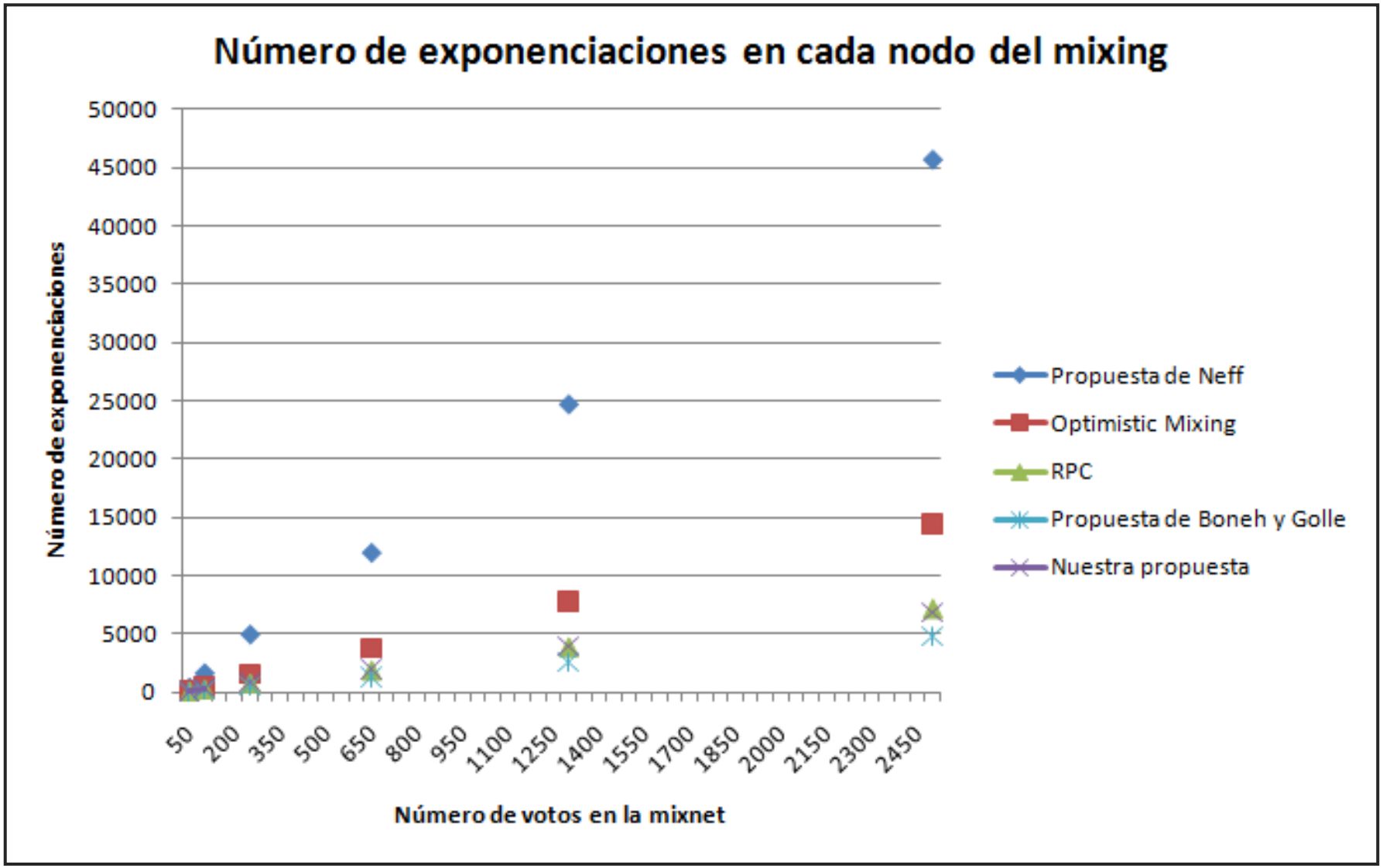

Con grupos más pequeños la probabilidad de detección aumenta.

$$P_{success} = 1 - \frac{n-1}{m-1}$$

m es el total de votos, n es el número de votos en cada grupo



scytl Secure Electronic Voting


Número de exponenciaciones en cada nodo del mixing
Número de exponenciaciones
50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
50
200
350
500
650
800
950
1100
1250
1400
1550
1700
1850
2000
2150
2300
2450
Número de votos en la mixnet
Propuesta de Neff
Optimistic Mixing
RPC
Propuesta de Boneh y Golle
Nuestra propuesta

# Índice

- Auditabilidad en el voto electrónico
- Mixnets universalmente verificables
- Descripción de la propuesta
- Propiedades
- Cifrado eficiente de los votos
- Conclusiones

# Índice

- Auditabilidad en el voto electrónico
- Mixnets universalmente verificables
- Descripción de la propuesta
- Propiedades
- Cifrado eficiente de los votos
- Conclusiones

# Conclusions

- From the point of view of efficiency, the computation cost of our proposal is close to the fastest one (Boneh and Golle [BG02]) the fastest one, and faster than Random Partial Checking [JJR02] for medium amounts of votes (more than 1500)

- In terms of privacy, it does not pose any privacy concerns as the other efficient methods.

- In terms of accuracy, our proposal achieves a high level of cheating detection compared to the other most efficient methods. This probability is closer to 100% when the number of votes is near 300 votes (99%). The other methods, except [Ne01], have similar or lower accuracy levels.

- In summary, compared with the current verification methods, our solution is the most well-balanced in terms of efficiency, privacy, and accuracy, while providing universal verification properties.

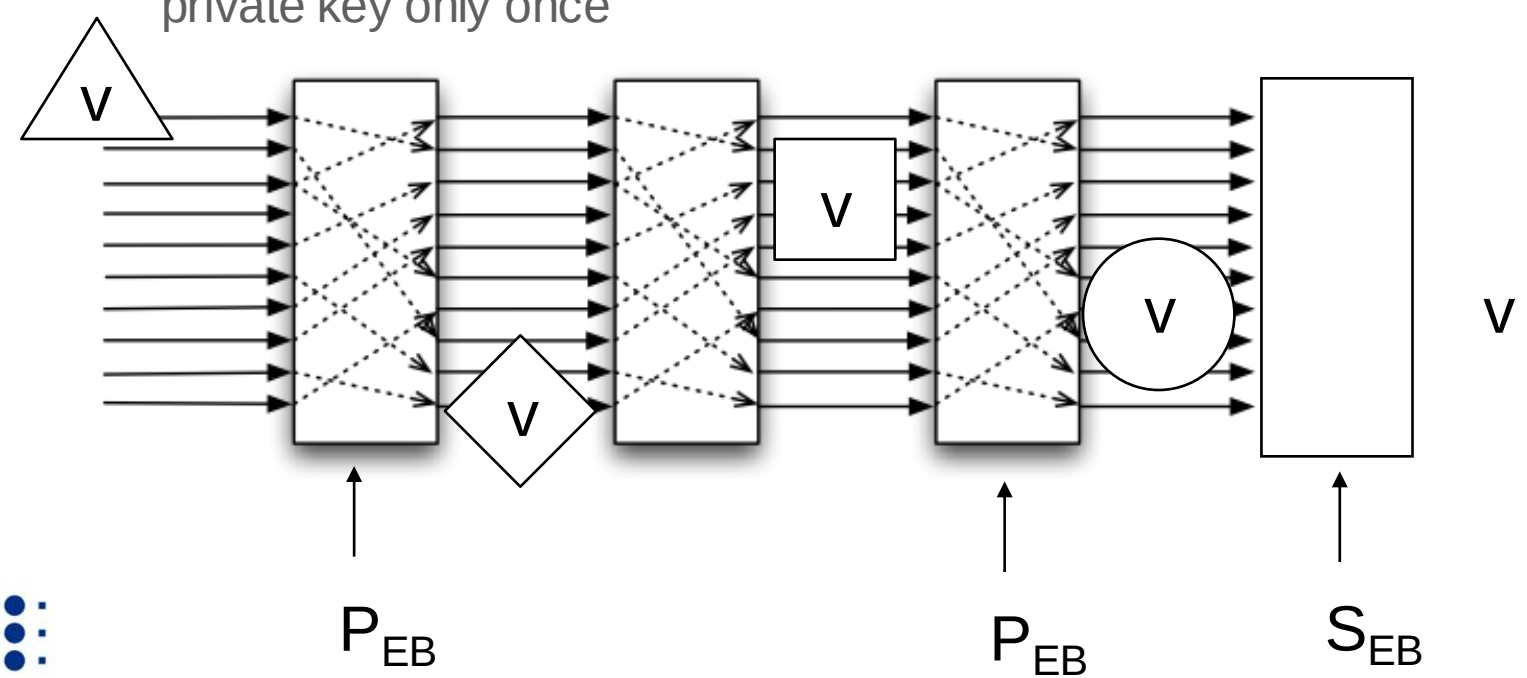
¿Alguna pregunta?

www.scytl.com

# Re-encryption Mix-net

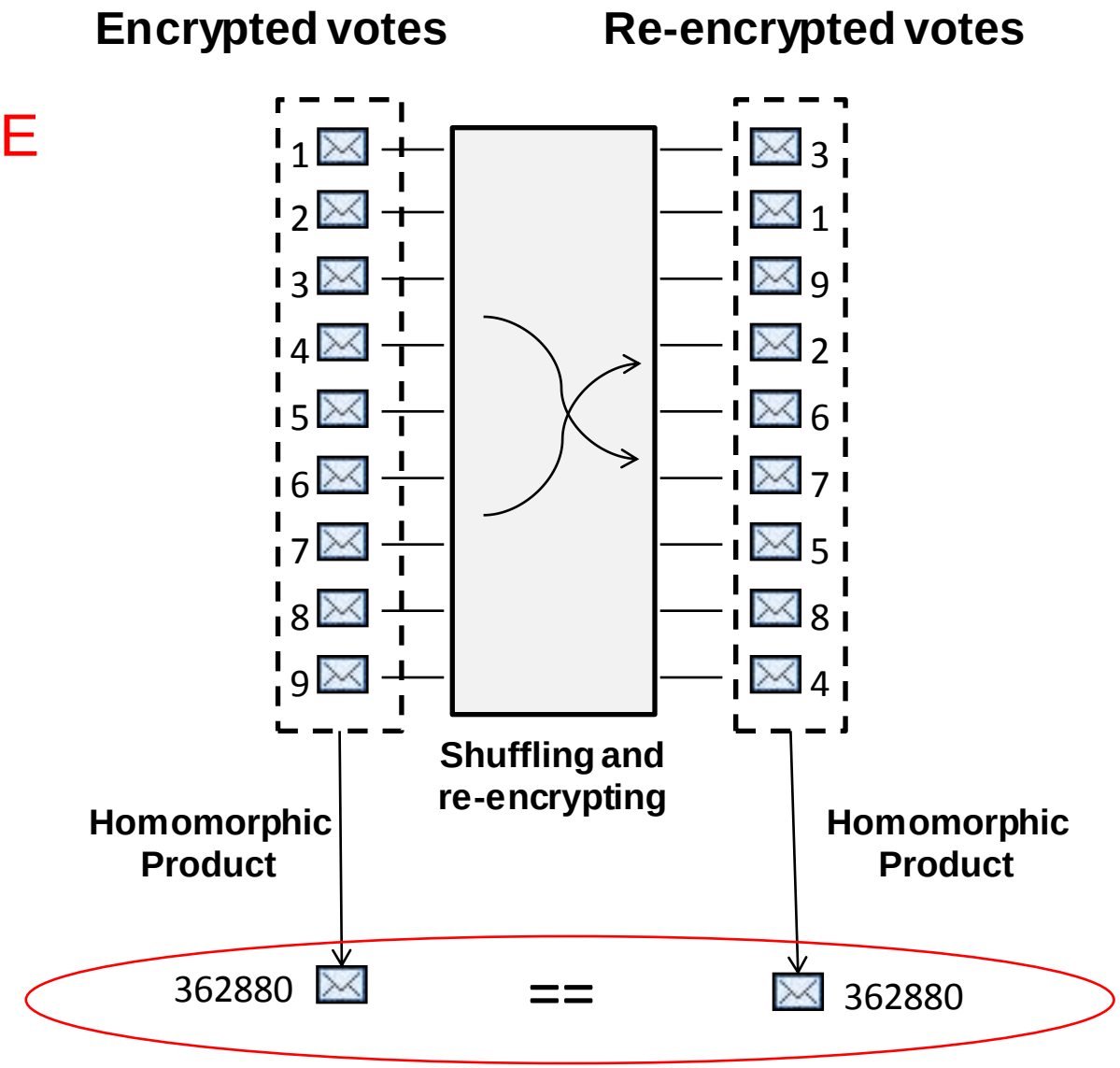
- Re-encryption Mixing : DELETE
  - It uses the homomorphic and probabilistic properties of some algorithms: re-encrypting one vote with the same public key does not require multiple decryptions of the vote with the private key, only one.

    $c = c'.(1.h^{w''}, g^{w''}) = (m'.h^{w'}, g^{w'}).(1.h^{w''}, g^{w''}) = (m'.h^{w'+w''}, g^{w'+w''})$
  - 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



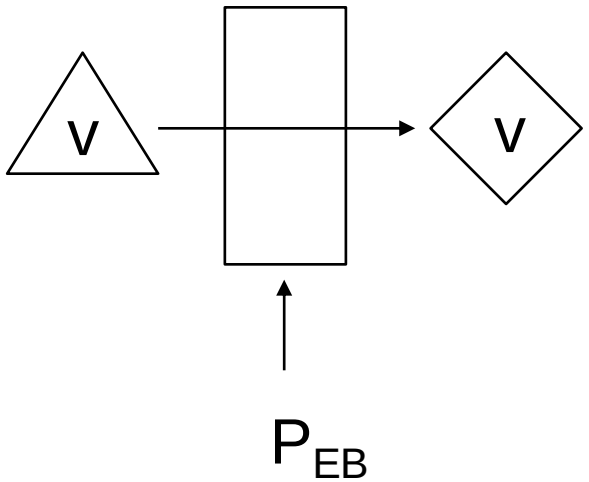
# Homomorphic Operation

DELETE

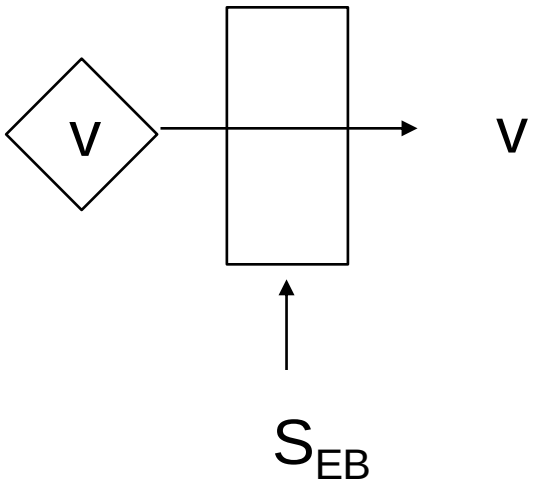


DELETE

- Zero Knowledge proof of re-encryption
  - Proofs that the Mix-node knows the re-encryption factor without disclosing it.
  - Based on Schnorr Identification Protocol



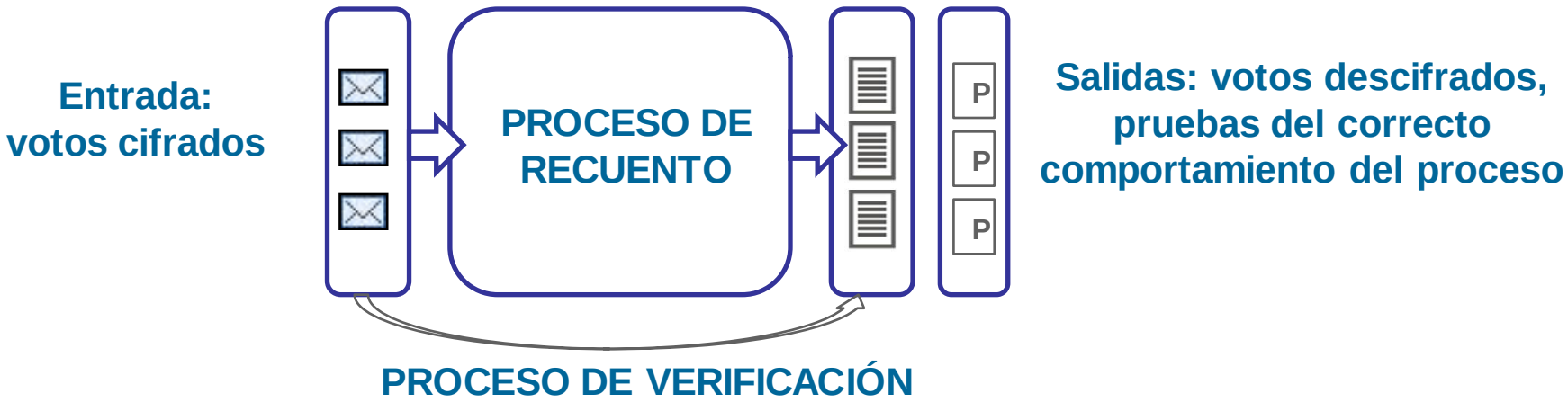

- Zero Knowledge proof of correct decryption
  - Proofs that the Mix-node knows the decryption factor without disclosing this factor or the private key.
  - Based on Schnorr Identification Protocol



- Auditability in e-voting
- Universal verifiable Mix-nets
- Building blocks
- Proposal description
- Properties
- Conclusions

DELETE

# Protocolos criptográficos de verificación universal




- El protocolo de verificación debe preservar la privacidad de los votantes y la integridad de la elección.

- Protocolos universalmente verificables en la fase de recuento de los votos:
  - **Recuento homomórfico.**
    - Problemas de flexibilidad: no se soportan todos los formatos de voto.
    - Problemas de escalabilidad: la cantidad de operaciones depende del número de candidatos.
  - **Mixnets universalmente verificables.**
    - No existen limitaciones en el formato del voto.
    - Más eficiente para listas largas de candidatos.

# Índice

- Auditabilidad en el voto electrónico
- Mixnets universalmente verificables
- Descripción de la propuesta
- Propiedades
- Cifrado eficiente de los votos
- Conclusiones

# Índice

- Auditabilidad en el voto electrónico
- Mixnets universalmente verificables
- Descripción de la propuesta
- Propiedades
- Cifrado eficiente de los votos
- Conclusiones

# ¿Cómo podemos verificar los procesos lógicos?

- **Revisión/certificación del código.**
  - Procesos de auditoría hechos por adelantado.
  - No permite verificar qué esta pasando en el momento de ejecución del software de voto.
- **Logs de auditoría.**
  - Permiten trazar lo que está ocurriendo durante la ejecución del software de voto.
  - Pueden ser modificados por el software de voto durante su ejecución.
- **Procesos de monitorización.**
  - Permite monitorizar el comportamiento de la plataforma de voto.
  - Intrusivos: ¿Quién monitoriza estos procesos? (¿Quién vigila al vigilante?).

¿Existe algún modo de verificar el comportamiento del software de voto sin utilizar ningún método intrusivo?

*Solución: Técnicas criptográficas de verificación.*