

# A Zero Knowledge proof for Subset Selection from a Family of Sets with applications to Multiparty/Multicandidate Electronic Elections

---

Dimitris Foteinakis

[fotd@intracom.gr](mailto:fotd@intracom.gr)

and

Tassos Dimitriou

[tassos@ait.edu.gr](mailto:tassos@ait.edu.gr)

TCGOV 2005



## Agenda

---

- Goals
- Secure Electronic Voting System Requirements
- Cryptographic Tools Overview
- System Description
- System Evaluation

# Goals

---

- Implement an online voting system that:
  - satisfies all the properties of secure electronic elections
  - allows multiple parties and the selection of  $t$  candidates from *one and only one* of those parties

# Secure Voting Requirements

---

- *Completeness*: All valid votes must be counted correctly
- *Soundness*: The dishonest voter cannot interrupt the voting process
- *Privacy and Integrity*: All votes must remain secret and cannot be altered in transit
- *Anonymity*: Voters' right to secrecy of their voter
- *Unreusability and Eligibility*: No one can vote twice and only eligible voters are allowed to vote
- *Fairness*: Nothing (colluding authorities, malicious voters) must affect the voting process
- *Verifiability*: Any external party can verify the result

## Optional Requirements

- *Receipt Freeness*: The voter cannot construct any short of proof of how he voted
- *Non-Duplication*: No one can duplicate another voter's vote
- *Public Participation*: Everyone can see who voted

# Cryptographic Tools: High Level Overview

---

- Public Key Encryption
  - The Paillier Public Key cryptosystem
- Homomorphic Property
- Threshold Cryptosystems
- Zero Knowledge proofs

---

## The Paillier Cryptosystem

---

- Proposed by Paillier in 1999
- a *Homomorphic* Probabilistic Public Key cryptosystem
- $N$  an RSA modulus:  $N = p \cdot q$  ( $p, q$  primes)
- $g$ :  $g = \text{gcd}(L(g^{\lambda(N)} \bmod N^2), N) = 1$ 
  - where: 
$$L(u) = \frac{u-1}{N}$$
  - Simplification:  $g = N + 1$
- Public Key:  $(N, g)$
- Secret Key:  $\lambda(N) = \text{lcm}[(p-1)(q-1)]$

# Encryption - Decryption

- To *encrypt* a message  $M \in \mathbb{Z}_N$  choose a random  $r \in \mathbb{Z}_N^*$  and compute the ciphertext:

$$c = g^M r^N \bmod N^2$$

- To *decrypt* a ciphertext  $c$ :

$$M = \frac{L(c^{\lambda(N)} \bmod N^2)}{L(g^{\lambda(N)} \bmod N^2)} \bmod N$$

- We use the Paillier Cryptosystem in order to encrypt the voter's selections

# Homomorphic Property

- Algebraic property: Allows computations with encrypted values

- $\forall m_1, m_2 \in \mathbb{Z}_N$ :

$$\begin{aligned} D(E(m_1) \cdot E(m_2) \bmod N^2) &= D(g^{m_1} r_1^N \bmod N^2 \cdot g^{m_2} r_2^N \bmod N^2) = \\ &= D(g^{m_1+m_2} r^N \bmod N^2) = m_1 + m_2 \bmod N \end{aligned}$$

- e.g. :  $E(3) \cdot E(2) = (g^3 r_1^N \bmod N^2) \cdot (g^2 r_2^N \bmod N^2) =$   
 $= g^3 g^2 r_1^N r_2^N \bmod N^2 = g^5 r^N = E(5)$

- The homomorphic property is utilized to compute the final tally

# Threshold Cryptosystems

---

- A  $(t, n)$  threshold scheme does not reveal a secret  $S$  unless  $t$  or more participants work together
- The Secret Key is shared between  $n$  authorities (share  $SK_i$ )
- All authorities publish their verification keys  $VK_i$
- To decrypt a ciphertext:
  - Each authority produces a partial decryption and a proof of validity
  - If more than  $t$  are found valid, the plaintext is recovered

# Zero Knowledge Proofs

---

- Protocols that allow a Prover (Peggy) to prove to a Verifier (Victor) of the validity of a statement, without revealing any information other than its correctness
- Interactive protocols:
  - Victor asks Peggy a series of questions
  - If Peggy knows the secret, she can answer correctly
  - If not, she has some chance to guess
  - After 10 or more correct answers, Victor can be convinced, that Peggy knows the secret

# Ballot Construction

- The purpose of the vote is to select  $t$  candidates from a single party
- Suppose there are  $K$  parties participating in the election and let each party has  $L$  candidates
- We will hold  $K \times L$  parallel “yes/no” votes
  - represent “yes” with “1” and “no” with “0”
  - the encrypted selection for candidate  $i$  will be:
    - $E_i = g^1 \times r^N$  for “yes”
    - $E_i = g^0 \times r^N$  for “no”

Candidate	Selection	Encryption
1	NO	$g^0 \times r_1^N$
2	NO	$g^0 \times r_2^N$
3	YES	$g^1 \times r_3^N$
4	YES	$g^1 \times r_4^N$
5	NO	$g^0 \times r_5^N$
6	YES	$g^1 \times r_6^N$
7	NO	$g^0 \times r_7^N$
8	NO	$g^0 \times r_8^N$
...	...	...
13	NO	$g^0 \times r_{13}^N$
...	...	...
C	NO	$g^0 \times r_C^N$
C+1	NO	$g^0 \times r_{C+1}^N$
...	...	...
C+5	NO	$g^0 \times r_{C+5}^N$

## Using the Homomorphic Property

- The homomorphic property of the cryptosystem is utilized to compute the encrypted tally
  - Example:  $N$  candidates, 4 Votes

	Candidate 1	Candidate 2	...	Candidate N
Ballot 1	$g^0 \times r_{11}^N$	$g^1 \times r_{12}^N$	...	$g^1 \times r_{1N}^N$
Ballot 2	$g^1 \times r_{21}^N$	$g^1 \times r_{22}^N$	...	$g^0 \times r_{2N}^N$
Ballot 3	$g^1 \times r_{31}^N$	$g^0 \times r_{32}^N$	...	$g^0 \times r_{3N}^N$
Ballot 4	$g^0 \times r_{41}^N$	$g^1 \times r_{42}^N$	...	$g^1 \times r_{4N}^N$
Ballot Product	$g^2 \times r_1^N$	$g^3 \times r_2^N$	...	$g^2 \times r_N^N$

- The ballot product holds the encryption of the total selections for each candidate

# Election Initialization Phase

---

- ❑ Parties, Candidates, number of selections are specified and written in the applet and server code
- ❑ Key generation
  - Keys are generated and distributed to the voters
  - Keys are generated for the servers and added in the applet and server code
  - Tallying authorities jointly generate the public key, and store individual key shares
- ❑ Voting applet is compiled and signed by a trusted source
- ❑ Instead, dedicated voting machines can be used

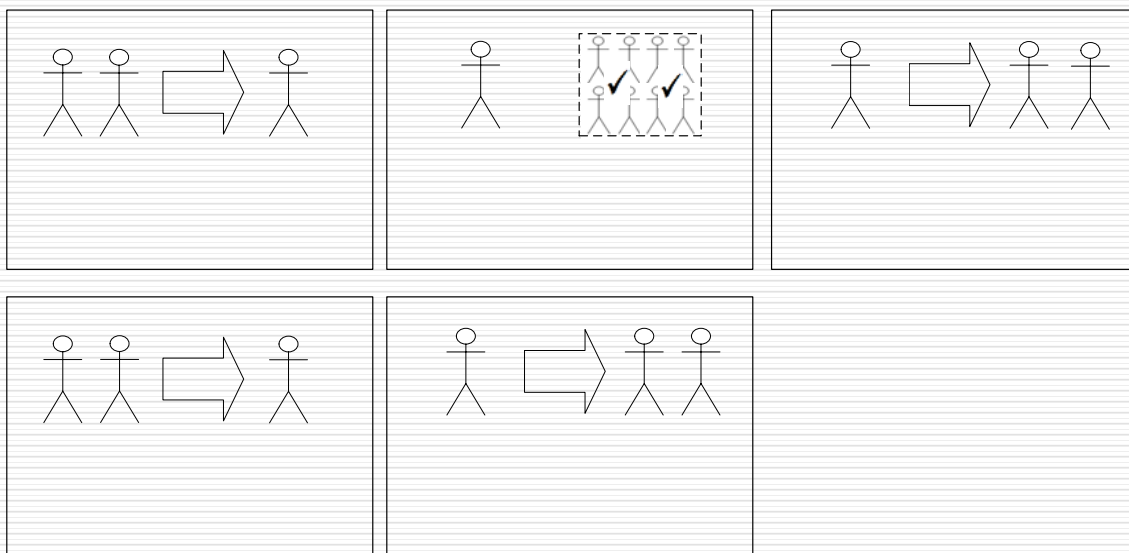
# Voting Phase

---

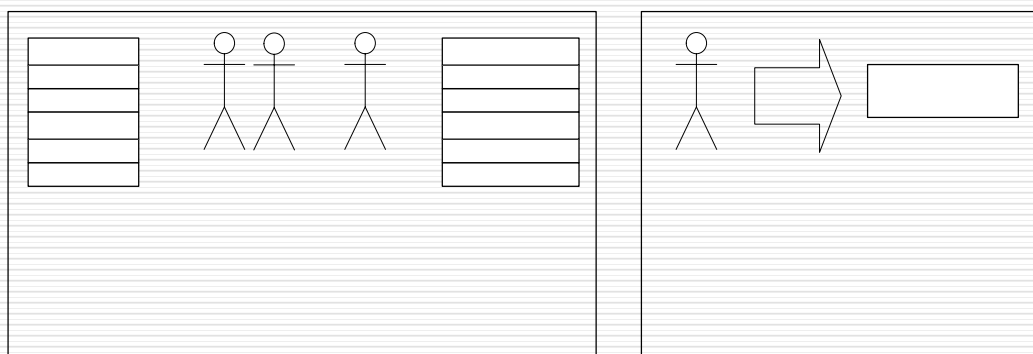
- ❑ A Voter downloads the Voting applet
  - Signature verifies integrity and origin
- ❑ The Voter provides his username and points to the key file containing his private key
- ❑ The voter selects the desired candidates
- ❑ The voting applet creates proofs of validity and encrypts selections
- ❑ The voting applet uses the user's private key to sign the ballot

# Voting Phase

## Sending the Ballot to the Authorities



# Computation of the result



# Important Issues

---

- Since individual ballots are not decrypted, we need a way to test the validity
  - If invalid, they could alter or damage the final tally
- Voter provides proof of validity in the form of Zero Knowledge Proofs

## Zero Knowledge Proofs of Validity

---

- Voter provides proofs:
  - That the vote is either an encryption of 0 or 1
  - That he has either selected 0 or  $t$  candidates from each party
  - That he has selected  $t$  candidates in total
- Essentially proves that he has selected *exactly*  $t$  candidates from a *single* party

# The Ballot Structure

---

- The ballot which will be submitted to the Storage Server will have the following form

			...			...
--	--	--	-----	--	--	-----

...		...		...			
-----	--	-----	--	-----	--	--	--

## Satisfying the Necessary Properties

---

- Privacy and Anonymity
  - The contents of each ballot remain encrypted (use of Homomorphic property)
  - Privacy is guaranteed unless more than  $t$  authorities collude (use of Threshold Cryptosystem)
  - Zero Knowledge proofs do not reveal any information about the actual content
- Public Participation
  - Everyone can see who voted and who did not
  - Ballots are posted in the Bulletin board which is accessible by anyone
  - This property is required in Greek elections, where participation is compulsory

# Satisfying the Necessary Properties

---

## □ Public Verifiability

- Ballots are posted and an external party can:
  - View the ballots
  - Verify their correctness
  - Compute the ballot product
- One can verify the partial decryption results posted by the tallying authorities
- One can use the partial decryption results and compute the decrypted tally

# Satisfying the Necessary Properties

---

## □ Fairness and Soundness

- No malicious party, voter or authority, can affect the voting process, or alter its results
- Invalid ballots are discarded by the AS since they will not have correct proofs
- If the ballot is sent directly to the SS, it will not have a valid signature by the AS
- A malicious AS cannot send its own ballots, since they will not have a valid voter signature
- Up to  $t$  malicious or faulty tallying authorities can be tolerated

# Satisfying the Necessary Properties

## □ Eligibility and Unreusability

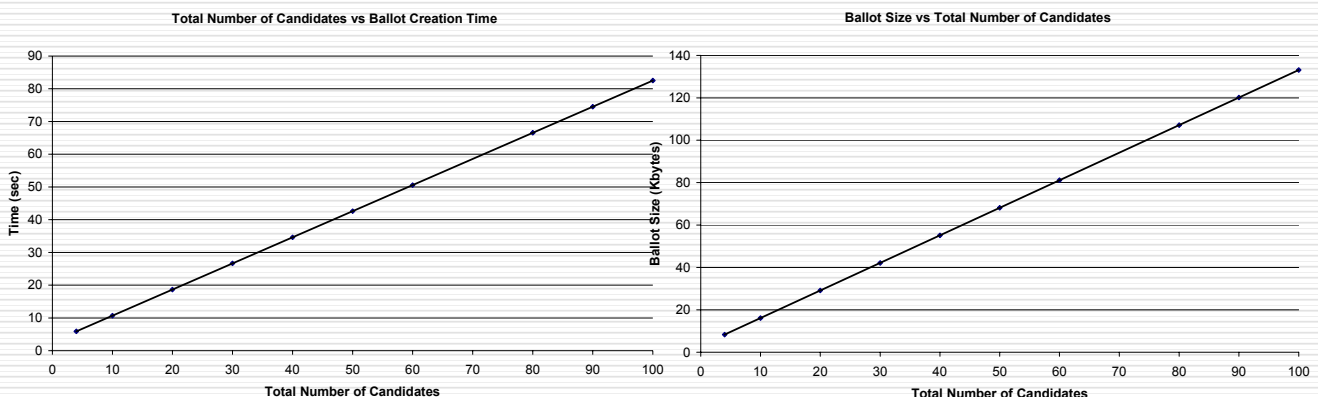
- Our system guarantees that only eligible voters are allowed to cast a vote
- The AS authenticates each voter and allows one vote per voter, by maintaining a list of who has voted

## □ Vote Duplication

- No one can duplicate the vote of another voter since:
  - Ballot consists of the username, and the AS's signature includes this field
  - During the construction of the Zero Knowledge proofs, voter's credentials are hashed along with other data to provide unique challenges

# System Evaluation

- Our system behaves linearly with respect to the total number of candidates



# Ballot Size

---

- The size of a ballot is:

$$(5|N^2| + 2|H|) \cdot C + (4|N^2| + 2|H|) \cdot (K + 1)$$

$C$  is the number of candidates,  $K$  the number of parties,  $|H|$  the size of the hashed commitments and  $|N^2|$  the size of the modulus

- Ballot length is  $O(C+K)$  and  $C \gg K$ , thus  $O(C)$
- In practical applications  $|H| = 80$  and  $|N^2| = 2048$
- For 20 candidates in 3 parties the size will be about 30Kbytes

# Conclusions & Future Research

---

- We have presented a methodology for proving in Zero Knowledge the validity of a selection of  $t$  elements from one out of  $k$  sets of  $n$  elements each
- We constructed an election that provides complex ballot options, based on the above principle
- Computational and Communication complexity is linear with respect to the total number of candidates, thus making a real system feasible to implement
- We are currently exploring the possibility to reduce the complexity to being proportional to the total number of parties plus the number of selections a voter makes