Large-Scale Electronic Voting Protocols

Mike Carpenter

Introduction

What is meant by large-scale electronic voting protocol:

- Primarily Internet-based
- Users voting from their own devices (such as home PC/laptop)
- Aimed toward actual country-wide election (e.g. USA Presidential elections)

Where are we?

- **Currently very few countries actually use e-voting systems.**
- **Obstacles include:**
  - Contradictory or internally-inconsistent legal requirements
  - Anonymity vs. auditability
  - Client-side security (malware potentially hijacking votes)
  - NISTIR 7770 lists four primary areas of concern:
    - Confidentiality
    - Integrity
    - Availability
    - Identification and Authentication

Existing Technologies

For the purposes of this presentation I'll be dividing existing voting protocols into the following categories:

- **Blind Signature**
- **Mix-Networks**
- **Homomorphic encryption**
  - Additive homomorphism
  - Multiplicative homomorphism

Blind Signature Scheme

- **The concept of a “blind signature” was invented by David Chaum in 1983 and is primarily used in election protocols and cryptocurrencies.**
  - A blind signature obscures the contents of a message before the signing authority can sign.
  - In this case, an automated election authority would authenticate a user, and subsequently blind-sign their submitted vote.

Based on [3]

Blind Signature Scheme

- **Steps for a blind signature protocol:**
  - Preparation
    - Voter fills out ballot, blinds, signs, and forwards to Administrator.
  - Administration
    - Administrator checks voter credentials; if valid, returns certificate to voter.
  - Voting
    - Upon receipt of the certificate, voter checks validity and submits vote through anonymous channel.

Based on [3]

Blind Signature Scheme

- **Steps for a blind signature protocol (cont'd):**
  - Collecting
    - Counter checks all votes, adds to a list, and publishes list.
  - Opening
    - Publicly verified by voters that the number of votes published in the list is equivalent to the number of votes cast.
  - Counting
    - Vote list is committed and tallied.

Based on [3]

Mix-Networks

Mix-Network voting schemes use multiple encryptions and decryptions to "shuffle" votes in such a way that the source of each vote is indeterminable.

- **Exceedingly popular method**
- **Seen many applications since first proposed by David Chaum in 1981 (e.g. onion routing)**
- **Potentially very expensive, especially with large number of voters (such as in a national election)**
  - But they are still the best choice for elections with a large number of candidates or for preferential voting.
- **Because individual votes are decrypted, vote validity checks are unnecessary**

Mix-Network Scheme

Basic step-by-step:

1. “n” votes are passed through a mixer

1. Mixer randomly determines some permutation of 1..n to determine reordering.
2. Individual votes are encrypted and returned in the order determined.

2. Mixer passes on to next mixer, who repeats the process.

- Votes are encrypted like a Matryoshka doll

3. After all mixers complete, they cooperate to decrypt the final permutation.

Homomorphic Schemes

Voting schemes taking advantage of homomorphic encryption fall into two categories:

- **Additively homomorphic**
  - Calculates the sum of all votes before decrypting, thereby only decrypting the result and not any individual votes
  - Common cryptosystems include Paillier and modified ElGamal
  - Much more common, but slower and more basic

Homomorphic Schemes

Voting schemes taking advantage of homomorphic encryption fall into two categories:

- **Multiplicatively homomorphic**
  - Assigns each candidate a prime number, calculates the product of votes, then factors decrypted result
  - Uses the standard ElGamal cryptosystem
  - Relatively obscure
  - More efficient and flexible than additive systems, but faces its own set of problems

Additive Homomorphic Scheme

I'll be outlining the system by Hirt and Sako, which uses a modified ElGamal encryption scheme:

- Private key a is split amongst t authorities such that (t - 1) colluding authorities cannot determine the private key.
- Rather than encrypting message m, one encrypts γ^m, where γ is a generator in group G (in this case, independent from the generator used to generate the public key, g).
- Encryption: $E(m) = (g^k, \gamma^m h^k) = (y_1, y_2)$
  - h is the public key, k is a random number.
- Decryption: $D(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p = \gamma^m$

Based on [2]

Additive Homomorphic Scheme

In this modified scheme, $D(E(m_1)*E(m_2)) = m_1+m_2$.

- In standard ElGamal, $D(E(m_1)*E(m_2)) = m_1m_2$.
- Because we are encrypting γ^m instead of m…
  - $\gamma^{m1}*\gamma^{m2} = \gamma^{m1+m2}$

  Must find the discrete logarithm, which in this context is supposedly computable in $O(\sqrt{(M)}^{L-1})$
    - M is the number of voters
    - L is the number of choices (in a yes/no election, L=2)

Based on [2]

Additive Homomorphic Scheme

In this scheme, the *sum* of votes is used to determine the results of the election based on the number of possible choices L. Where V is the set of possible votes:

- If $L = 2$, $V = \{1, -1\}$ (0 may be added for abstention)
- If $L > 2$, $V = \{1, M, M^2, \ldots, M^{L-1}\}$

Because only the sum of votes is decrypted, and not any individual vote, privacy is preserved for all voters.

Based on [2]

Another Additive Homomorphic Scheme

- **The previous example is by no means the only additive scheme to exist.**
- **Another example involves encrypting a separate vote {1, 0} for each candidate.**
  - This increases verification cost by a lot, because more than one vote must be proven valid.
- **Some schemes use Paillier encryption, which is additively homomorphic**
  - However, Paillier is more costly than the modified ElGamal system

Multiplicative Homomorphic Scheme

I'll be outlining the system by Peng, et al., which uses textbook ElGamal encryption.

- **Each of *m* candidates is assigned a small prime *q* such that *all* primes in $Q = \{q_1, q_2, \ldots, q_m\}$ are either quadratic residues or quadratic nonresidues modulo *p*.**
- **This is a similar concept to additive homomorphic e-voting, except the goal is the *product* of votes rather then the sum, which is then factored.**
  - With vote validation, we know all the factors ahead of time, so factoring is trivial.

Based on [4,5]

Multiplicative Homomorphic Scheme

- **This system is computationally more efficient than additive systems, but suffers from a huge drawback.**
- **If the product exceeds the modulus *p*, votes will be lost and decryption may fail.**
- **Thus, votes must be split into groups to be multiplied:**
  - Ideal group size is the largest integer *k* such that $Max(Q)^k < p$.
  - Privacy is inherently compromised here, as attributing a vote to a voter becomes much easier when one only needs to chose from among *k* votes in a group rather than the total number of votes cast overall.

Based on [4,5]

Multiplicative Homomorphic Scheme

- **The grouping privacy problem can be solved by borrowing from another voting scheme: mix-networks.**
  - Because the shuffling is done on groups rather than individual votes, there is *much* less to shuffle, and therefore is computationally cheaper.
    - Even combining mix-networks and homomorphic tallying, the multiplicative system is more efficient than both.
  - After shuffling it isn't known which group is which, and therefore it isn't known to which group an individual's vote was committed.
    - This gives the system equivalent vote privacy to additive homomorphic e-voting, with greater efficiency, albeit more conceptual complexity and more opportunity for implementation mistakes.

Based on [4,5]

Efficiency comparison

Table 2. Efficiency comparison

Scheme	Cost of a voter		Cost of a tallier in tallying[a]	Cost of verification of	
	encryption	vote validity proof		vote validity	tallying
shuffling based voting	2	unnecessary	$\geq 10n$ $= 100000$	unnecessary	$\geq 6tn$ $= 300000$
additive homomorphic voting	$2m$ $= 200$	$\geq 5m$ $= 500$	$\geq 3m$ $= 300$	$\geq 4nm$ $=4000000$	$\geq 4tm$ $= 2000$
[34]	2	$\geq 5m$ $= 500$	3β $= 300$	$\geq 4nm$ $= 4000000$	$\geq 4t\beta$ $= 2000$
new scheme	2	14	10β $= 1000$	$10n$ $= 100000$	$6t\beta$ $=3000$

[a] Including shuffling, proof of validity of shuffling, partial decryption and proof of validity of decryption

- *m*: number of candidate choices (*m* = 100)
- *t*: number of cooperating talliers needed (*t* = 5)
- β: number of vote groups in mult. Systems (β = 100)
- *n*: number of votes (*n* = 10000)

Table from [5]

Questions?

References

[1] N. Hastings, R. Peralta, S. Popoveniuc, A. Regenscheid. 2011. "(NISTIR 7770) Security Considerations for Remote Electronic UOCAVA Voting". Retrieved 15 May, 2015, from the National Institute of Standards and Technology: http://www.nist.gov/itl/vote/upload/NISTIR-7700-feb2011.pdf

[2] Hirt, M., Sako, K.: Efficient receipt-free voting based on homomorphic encryption. In: Preneel, B. (ed.) *EUROCRYPT 2000*. LNCS, vol. 1807, pp. 539–556. Springer, Heidelberg (2000)

[3] A. Fujioka, T. Okamoto, K. Ohta. 1992. "A Practical Secret Voting Scheme for Large Scale Elections". In *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques: Advances in Cryptology (ASIACRYPT '92)*, J. Seberry, Y. Zheng (Eds.). Springer-Verlag, London, UK, UK, 244-251

[4] K. Peng, R. Aditya, C. Boyd, E. Dawson, B. Lee. "Multiplicative homomorphic e-voting." In *Progress in Cryptology--INDOCRYPT 2004*, pp. 61-72. Springer Berlin Heidelberg, 2005.

[5] K. Peng, F. Bao. "Efficient multiplicative homomorphic e-voting." In *Information Security*, pp. 381-393. Springer Berlin Heidelberg, 2011.

[6] A. Trechsel, F. Mendex, R. Kies. 2003. "Remote voting via the Internet? The Canton of Geneva pilot project". In *Secure Electronic Voting*, Gritzalis, D.A. (Ed.). Kluwer Academic Publishers, Norwell, MA, USA, 181-194

[7] S. Kazue, J. Kilian. 1995. "Receipt-Free Mix-Type Voting Scheme". In *Advances in Cryptology - EUROCRYPT '95*, L. Guillou, J. Quisquater (Eds.), Springer Berlin Heidelberg, 393-403