Voting with Blind Signatures

Panagiotis Grontas

$\mu\Pi\lambda\forall$ - CoReLab Crypto Group

26/03/2014

Blind Signatures I

Definition

A set of algorithms $(Blind, Sign, Verify, Unblind)$:

1. The message m to be signed is blinded using the Blind Algorithm and some randomness. $b = Blind(m, r)$
2. Afterwards it is signed $S_b = Sign(b)$ in a way that the signature is transferred to the original message.
3. The Unblind function is used on the blind signature obtaining: $S_m = Unblind(S_b)$
4. The signature is verified by executing $Verify(S_m)$

Blind Signatures II

RSA BS

1. $b = Blind(m, r) = r^e H(m) \pmod n$
2. $S_b = Sign(b) = r^{ed \pmod{\phi(n)}} H(M)^d \pmod n = rH(m)^d$
3. $S_m = Unblind(S_b) = S_b \frac{1}{r} = H(m)^d \pmod n$
4. The signature is verified by executing $Verify(S_m) =$ if ${S_m}^e = H(m)$ then $True$ else $False$

Blind Signatures and Voting [Cha83] I

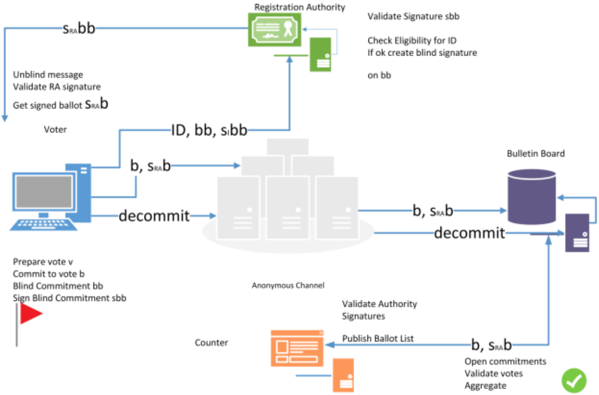
- The voter submits a blinded version of the ballot along with his registration information.
- The voting authority validates the voter data, and if the voter has the right to vote, signs the blinded ballot and returns it to the voter.
- The voter validates the signature of the authority and posts the signed ballot *anonymously*.
- The authority receives the signed ballots, validates its signature and posts them to a bulletin board for verification. Verification is done through a random pattern that the voter has embedded into the ballot, known only to him.

Blind Signatures and Voting [Cha83] II

Properties

- Anonymity
- Individual verifiability
- The authority knows the intermediate voting results
- Dispute Resolution: The voter must show its vote $\rightarrow$ Anonymity is lost

A Practical Secret Voting Scheme for Large Scale Elections [FOO93] I



Registration Authority

Validate Signature sbb

Check Eligibility for ID
If ok create blind signature

on bb

$s_{RA}bb$

Unblind message
Validate RA signature
Get signed ballot $s_{RA}b$

Voter

ID, bb, $s_i$bb

b, $s_{RA}b$

decommit

Bulletin Board

b, $s_{RA}b$

decommit

Prepare vote v
Commit to vote b
Blind Commitment bb
Sign Blind Commitment sbb

Anonymous Channel

Validate Authority
Signatures

Publish Ballot List

Counter

b, $s_{RA}b$

Open commitments
Validate votes
Aggregate

Figure : Voting with blind signatures [FOO93]

A Practical Secret Voting Scheme for Large Scale Elections [FOO93] II

Separate the functions of the authority

- The registration authority, that knows the voter's identity but not the actual vote
- The tallying authority, that knows the vote but not the identity

1. **Preparation**
    1. The RA has a public and private key pair (e_A, d_A)
    2. Each has a public and private key pair (e_I, d_I)
    3. The *i*-th voter prepares her vote v_i
    4. Based on the vote and using randomness rc_i, she creates a ballot $b_i = commit(v_i, rc_i) = g^{rc_i}h^{v_i}$. The bit commitment scheme ensures that the voter cannot behave differently in the preparation and generation phases. Moreover it substitutes the random pattern that the user must embed into her vote in order to verify it in the bulletin board.

A Practical Secret Voting Scheme for Large Scale Elections [FOO93] III

   5. Using randomness rb_i and the public key of the authority she creates the blinded ballot $bb_i = blind(b_i, rb_i) = b_i rb_i^{e_A}$
   6. Using her private key she signs the blinded ballot: $sbb_i^I = sign_{d_I}(bb_i)$
   7. She submits the message (i, bb_i, sbb_i^I) to the election authority, where the index i denotes identity information for the voter.

2. **Authorisation**
   1. Upon receipt, the authority validates voter's signature, voter's eligibility from the identity information and checks for double requests so as to defend against double voting.
   2. If all checks turn out ok, it signs the blinded ballot $sbb_i^A = sign_{d_A}(bb_i) = b_i^{d_A} rb_i$
   3. The authority sends sbb_i^A to voter i and announces the total number of voters

A Practical Secret Voting Scheme for Large Scale Elections [FOO93] IV

3. **Voting**
    1. Upon receipt, the voter unblinds the signed blind ballot, obtaining the signature of the authority to the ballot. $sb_i^A = unblind(sbb_A^i) = b_i^{d_A}$
    2. The signature is validated with the public key of the authority. Authority cheating can be proved by showing b_i, sb_i^A, without revealing the vote, so that everybody can verify that the signature is invalid.
    3. If all checks turn out ok, then she **anonymously** sends b_i, sb_i^A to the counter. The use of an anonymous channel is required, in order to hide the network identity of the voter from the counter, in order not to be able to trace it back to the real identity.
4. **Collecting**
    1. The counter validates the sb_i^A with the public key of the authority
    2. If all checks turn out ok, it publishes an indexed list of all the ballots along with the signature $\{idx, b_i, sb_i^A\}$

A Practical Secret Voting Scheme for Large Scale Elections [FOO93] V

5. **Opening**
    1. Each voter validates that the voters published by the authority equal the number of ballots and that her ballot is included on the list, with index *idx*. A corrupt counter, that has not included or has altered the ballot, can be uncovered by presenting the valid ballot and the valid signature by the authority.
    2. If all checks turn out ok, she sends anonymously the decommitment value idx, rc_i.
    3. A corrupt voter might send an invalid opening value in order to cancel the vote. This is a weakness of the current scheme.
    4. A corrupt counter might claim that it received invalid values and cannot open the commitment. The offended voter might object, but since the opening values are sent anonymously the dispute cannot be resolved anonymously. A solution would be to use a verifiable mixnet in this stage, but this will hamper the performance.

A Practical Secret Voting Scheme for Large Scale Elections [FOO93] VI

6. **Counting**
   1. Now all the ballots can be made public, by opening the commitments. Since the voting phase has ended nobody can benefit from knowledge of a premature and partial result. This guarantees fairness. In addition the ballots to be opened are anonymous, since they were transported through an anonymous channel and lack any other identifying information.
   2. After opening, the votes are checked to conform to the voting scheme. Then they are counted and the result is announced.
   3. Everybody can verify the result, by computing it on their own.

Properties

- Simple
- Supports many social choice functions
- Anonymity

A Practical Secret Voting Scheme for Large Scale Elections [FOO93] VII

- Fairness
- Individual Verifiability
- Efficiency depends on the actual form of the anonymous channel
- Universal verifiability:
  - Everybody can check that the published decommitments indeed open the commitments and that the result corresponds to the opened values
  - BUT: A corrupt voter or a corrupt counter can nullify votes without being able to prove invalidity.
- A corrupt RA can inject votes for absentees
- Requires voter interaction in at least three stages NOT *vote and go*
- Not Receipt Free (commitment = receipt). The voter can sell his vote, and prove his offer by providing a v_i, rc_i that open the commitments

Reduce Voter interaction [OMA$^+$99] I

- Replace the commitment scheme with a threshold encryption scheme
- The counter gets his own key pair - the private key is split (subcounters)
- Instead of committing to her choice, the voter encrypts with the public key of the counter
- The encrypted vote is then blind-signed by the registration authority, and sent with the signature to the $\mathcal{BB}$ during the voting phase.
- Decrypt the ballot and write to the BB
- Proof of decryption? Individual Verifiability? Assumed because of the threshold scheme

Support for Vote Cancelling [HS98] I

- Reminder: Each voter is required to have a public, private key pair
- The authority blindly signs the voter public key
- The voter submits the public key along with the signature for validation
- The voting phase is split into 2 subphases:
  - The actual ballot sent during the voting phase is encrypted using a symmetric cipher, along with the public key
  - To decrypt it the voter must 'open'it, by sending the symmetric key
- All interactions take place through an anonymous channel
- If a voter changes her mind, she can omit to send the symmetric key, and/or resubmit the new in a similar manner.
- To prevent double voting, the counter can check that there is only one decrypted vote per public key or retrieve the last vote per public key.

Support for Vote Cancelling [HS98] II

What is needed: A *blind identifier* to group the votes. [CD07] The voter can create multiple identities, called *pseudo-voter identities (PVID)*. The registration authority blindly signs each one of them. Each PVID should be used with a single authority (either tallying, verifying etc.)

Anonymous One Time Login I

- The participants register with a registration authority using their real names and passwords.
- When voting is required the participants create a random credential m and commit to it using a hash function $h(m)$
- Then the credential is blinded, creating $b_m = h(m)k^e modn$
- To validate that they are indeed authorised to vote, they login with their real credentials and submit b_m
- The registration authority checks that the login credentials are valid and that the participant is authorised to vote. In that case, it signs the blinded credentials $bs_A = b_m^d = h(m)^d k^{ed} = h(m)^d k \pmod n$ and presents them to the participant
- The participants unblinds the signature and retrieves the signature on the committed credential: $bs_A k^{-1} = h(m)^d = s_A$

Anonymous One Time Login II

- As a result the participant has an anonymous username m signed by an authority that can validate his identity.
- Now when voting is due, the participant provides m and s_A to a login form. The voting system retrieves the public key of the authority and validates whether $s_A^e = h(m)$. If the signature is valid then the participant can vote.

Receipt Freeness [Oka96], [Oka98] I

- Replace the commitment
- Use a trapdoor bit commitment scheme
- Commitment will be made using the private key
- Opening will require the private key, allowing the voter to open it in multiple ways
- Variations:
  - Secret Sharing of the commitment private key *parameter registration committee - PRC*
  - Let α_i be the commitment key and $G_i = g^{\alpha_i}$
  - Split to n parts α_{ij} such that $\alpha_i = \sum_{j=1}^{N} \alpha_{ij} \pmod q$
  - Public $G_{ij} = g^{\alpha_{ij}}$
  - The registration authority blind signs G_i and G_{ij} along with the committed vote.
  - The PRC members can compute G_i and post to the *BB*
  - Use a (t, n) scheme in order to avoid blocking by a corrupted authority
- Not coercion resistant...

Coercion Resistance

- Registration yields a list of credentials - voter roll
- Each voter can create his own
- Voter submits credential for certification
- Authority can check if it is found in the voter roll
- Use a blind group signature: two groups valid invalid
- Transfer the credential list from authority to tallier

References I

Orhan Cetinkaya and Ali Doganaksoy.
Pseudo-voter identity (pvid) scheme for e-voting protocols.
In *ARES*, pages 1190–1196, 2007.

David Chaum.
Blind signatures for untraceable payments.
In D. Chaum, R.L. Rivest, and A.T. Sherman, editors, *Advances in Cryptology Proceedings of Crypto 82*, pages 199–203, 1983.

Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta.
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, pages 244–251, London, UK, UK, 1993. Springer-Verlag.

References II

Qi He and Zhongmin Su.

A new practical secure e-voting scheme.

In *IFIP SEC*, volume 98, pages 196–205, 1998.

Tatsuaki Okamoto.

An electronic voting scheme.

In Nobuyoshi Terashima and Edward Altman, editors, *Advanced IT Tools*, IFIP — The International Federation for Information Processing, pages 21–30. Springer US, 1996.

Tatsuaki Okamoto.

Receipt-free electronic voting schemes for large scale elections.

In *Security Protocols*, pages 25–35. Springer, 1998.

References III

Miyako Ohkubo, Fumiaki Miura, Masayuki Abe, Atsushi Fujioka, and Tatsuaki Okamoto.

An improvement on a practical secret voting scheme.

In *Information Security*, volume 1729 of *Lecture Notes in Computer Science*, pages 225–234. Springer Berlin Heidelberg, 1999.