Votare su Internet è possibile?

uno studio dei protocolli classici

Relazione per il corso di

Metodi Formali 2

Prof. Roberto Gorrieri

Scritta nel febbraio 2001

da

Fabrizio Bisi, bisi@cs.unibo.it

Requisiti di un buon sistema di voto elettronico

- **Invulnerabilità (democracy)**
  (1) solo agli aventi diritto possono votare e nessun altro
  (2) ogni elettore legittimato non può votare più di una volta
- **ACCURACY**
  (1) un voto non può essere alterato
  (2) ogni voto legittimo deve essere conteggiato nello spoglio finale
  (3) nessun voto non valido sarà conteggiato nello spoglio finale
- **Privacy**
  (1) nessuno può associare il voto all'elettore che lo ha espresso
  (2) nessun elettore può provare di aver votato in un certo modo
- **Verificabilità**
  - **forte:** chiunque può verificare che tutti i voti sono stati conteggiati
  - **debole:** ciascuno può controllare il proprio voto
- **Fairness (equità)**
  Nessuno è in grado di vedere il risultato parziale delle elezioni prima del termine

Requisiti pratici di un buon sistema di voto elettronico

- **Convenience** (**User Friendliness**)

  Un sistema è conveniente se è facile da usare, ossia se permette agli elettori di votare velocemente e in una sola sessione

- **Flexibility**

  Un sistema è flessibile se è possibile utilizzarlo per elezioni di diversa natura

- **Mobility**

  Un sistema è tanto più mobile quante meno restrizioni pone all'elettore sulla località da cui deve votare. Un sistema di i-vote è più mobile di un sistema a seggi; un sistema con libera scelta di seggio è più mobile di un sistema a seggio fisso

PROBLEMI CON LA MOBILITY

- un sistema di i-vote non può impedire la compravendita dei voti o la coercizione
- nessun sistema di autenticazione elettronico è sicuro quanto il riconoscimento fisico

Tipi di protocolli di voto elettronico

Possiamo suddividere i protocolli di voto elettronico in 3 grandi famiglie

- **protocolli ad auto-arbitraggio** (self-adjudicating protocols)
  non esiste nessuna autorità centrale ma prevedono interazioni tra gli elettori stessi
- **protocolli con Central Tabultating Facility (CTF)**
  la CTF è responsabile sia della raccolta dei voti che del conteggio finale
- **protocolli con due entità istituzionali**:
  - **Validation Authority (VA)** raccoglie i voti
    conosce gli aventi diritto al voto e chi tra loro ha votato ma non può risalire al loro voto
  - **Tabulating Facility (TF)** esegue il conteggio finale dei voti
    conosce i voti ma non può sapere quali elettori li hanno espressi

I protocolli con CTF non sarebbero buoni candidati per elezioni a larga scala per due ordini di motivi:

- tendono a fare uso di tecniche crittografiche computazionalmente più pesanti rispetto a quelli con due entità istituzionali
- la CTF costituisce un singolo punto di rottura: compromessa quella, tutto il sistema è compromesso.

Primi esempi di protocolli

Non si può fare a meno della crittografia. Per esempio il seguente è un possibile protocollo di voto non crittografico

Protocollo ingenuo che non fa uso di crittografia

1. $V_i \rightarrow$ VA: ID_i, m_i
2. VA controlla se V_i ha diritto al voto e se non ha già votato
3. VA $\rightarrow$ TF: m_i

ALCUNI PROBLEMI

- V_i può provare a indovinare codici identificativi regolarmente assegnati ad altri elettori
- VA può leggere o addirittura modificare i voti senza essere scoperta
- TF, può scartare i voti che gli arrivano e sostituirli nel conteggio finale con voti di proprio gusto senza che nessuno se ne accorga

Proviamo ad aggiungere qualche tecnica crittografica di base

Protocollo ingenuo arricchito con tecniche crittografiche

1. $V_i \rightarrow$ VA: $SK_i(ID_i , PK_{TF}(m_i))$
2. VA controlla se V_i ha diritto al voto e se non ha già votato.
3. VA $\rightarrow$ TF: $PK_{TF}(m_i)$

COSA ABBIAMO OTTENUTO

- Rispetto della democracy
- A meno che TF e VA non colludano il rispetto del primo punto della privacy

Protocollo a due enti di Nurmi, Salomaa e Santean (1991)

Prima fase di voto

1. $V_i \rightarrow$ VA: richiesta di codice ID_i
2. VA controlla se V_i ha diritto al voto e se non ha già votato
3. VA $\rightarrow V_i$: ID_i

Al termine della prima fase

4. VA $\rightarrow$ TF: $\forall i\ ID_i$

Seconda fase di voto

5. $(V_i) \rightarrow$ TF: $ID_i, E_{k_i}(ID_i, m_i)$
6. TF $\rightarrow$ *: $E_{k_i}(ID_i, m_i)$
7. $(V_i) \rightarrow$ TF: ID_i, k_i

Al termine della seconda fase

8. TF $\rightarrow$ *: $\forall i\ E_{k_i}(ID_i, m_i), m_i$

Fase di verifica (terza fase di voto)

9. se $E_{k_i}(ID_i, m_i)$ non è stato pubblicato V_i esegue 10
10. $(V_i) \rightarrow$ TF: $ID_i, E_{k_i}(ID_i, m_i), k_i$

COSA ABBIAMO OTTENUTO

- Rispetto della democracy
- A meno che TF e VA non colludano il rispetto del primo punto della privacy
- Se nessuno si astiene rispetto dell'accuracy
- Verificabilità debole

Protocollo a un ente di Salomaa (1991)

Il problema più grave del protocollo di prima è che se VA e TF colludono, la privacy può ancora essere violata. Un modo per risolvere questo problema è considerare un solo ente invece di due.
La prima fase di voto, quella dell'assegnamento dei codici, viene risolta usando un protocollo ANDOS (all-or-nothing disclosure of secrets).
Tutte le altre fasi restano invariate

COSA ABBIAMO OTTENUTO

- Ora la privacy viene rispettata sempre

PROBLEMI INSORTI

- I protocolli ANDOS sono computazionalmente inefficienti quando il numero di partecipanti è maggiore di poche decine

PROBLEMI COMUNI AI DUE PROTOCOLLI

- Il secondo punto della privacy non viene rispettato
- Se alcuni elettori si astengono VA può votare per loro

Un pratico schema di voto segreto per elezioni a larga scala (Fujioka Okamoto e Ohta, 1992)

Protocollo con due enti. Scopi:

- Tutela la privacy anche qualora gli enti colludano
- Permette a un elettore di dimostrare di aver votato in un certo modo senza obbligarlo a rivelare il voto
- Soddisfa il requisito di equità (fairness)
- Utilizza tecnologie che non richiedono grossi costi computazionali

Le tecniche crittografiche utilizzate sono tre:

- **un sistema di commitment** ($C_k(\ldots)$)
  consente di inoltrare un messaggio senza che il destinatario possa leggerlo ma anche senza che il mittente possa modificarlo
- **un sistema di firme digitali** (**digital signatures**) (SK(…))
- **un sistema di firme cieche** (**blind signatures**) (SK(b(…)))
  permettono di far firmare un documento da qualcuno senza consentirgli di leggerne il contenuto. La tecnica consiste nell'usare un blind factor b con cui moltiplicare il messaggio

Protocollo

Prima fase

1. $V_i \rightarrow VA$: V_i, $SK_i(b(C_{k_i}(m_i)))$, $b(C_{k_i}(m_i))$
2. VA controlla se Vi ha diritto al voto e se non ha già votato.
3. VA controlla la validità della firma
4. $VA \rightarrow V_i$: $SK_{VA}(b(C_{k_i}(m_i)))$
5. $VA \rightarrow *$: V_i, $SK_i(b(C_{k_i}(m_i)))$, $b(C_{k_i}(m_i))$
6. V_i toglie il blind factor, ottenendo: $SK_{VA}(C_{k_i}(m_i))$
7. V_i verifica che la chiave di VA sia corretta.
8. $(Vi) \rightarrow TF$: $SK_{VA}(C_{k_i}(m_i))$, $C_{k_i}(m_i)$

Pubblicazione dei voti (al termine della prima fase)

9. $TF \rightarrow *$: $\forall i\ SK_{VA}(C_{k_i}(m_i))$, $C_{k_i}(m_i)$

Seconda fase

10. V_i controlla se il numero di voti conteggiati da TF è identico al numero di voti certificati da VA. Se va tutto bene procede
11. V_i controlla se il suo voto compare nella lista preparata da TF e in che posizione (pos).
12. $(V_i) \rightarrow TF$: pos, k_i

Conteggio (al termine della seconda fase)

13. $TF \rightarrow *$: $\forall i\ SK_{VA}(C_{k_i}(m_i))$, $C_{k_i}(m_i)$, k_i, m_i

Verifica (opzionale)

14. Se V_i riscontra un'irregolarità esegue 15
15. $(V_i) \rightarrow TF$: $SK_{VA}(C_{k_i}(m_i))$, $C_{k_i}(m_i)$, pos, k_i

COSA ABBIAMO OTTENUTO

- Nessun problema con la democracy
- Nessun problema col primo punto della privacy, neppure se le autorità colludono
- Soddisfa la verificabilità debole
- Soddisfa la fairness
- Non è computazionalmente inefficiente
- Rispetta in buona misura il requisito di accuracy

PROBLEMI

- Se alcuni elettori si astengono VA può votare per loro
- Incapacità di distinguere tra V disonesto e TF disonesto
- Assumere un canale anonimo sicuro significa spostare l'onere della sua sicurezza sull'implementazione
- Non soddisfa il secondo punto della privacy

Riepilogo delle proprietà e dei protocolli visti

	Democracy	**Privacy, 1**	**Verificabilità**	**Accuracy**	**Fairness**	**Note**
1	-	-	-	-	-	
2	Ok	*	-	-	-	
3	Ok	*	Debole	(o)	-	
4	Ok	Ok	Debole	(o)	-	pesante
5	Ok	Ok	Debole	(o) (oo)	Ok	

(*) Solo se TF, VA non colludono

(o) Problema astensione

(oo) incapacità di distinguere tra V e TF disonesto

1. protocollo ingenuo senza crittografia
2. protocollo ingenuo con crittografia
3. protocollo a due enti di [NSS91]
4. protocollo a un ente di [Sal91]
5. [FOO92]

Sensus (1997)

Sensus è un'implementazione del protocollo appena visto.

Comunque, il protocollo originario è stato rivisto per permettere all'elettore di votare in una sola sessione.

Iin Sensus il programma che fa le veci dell'elettore effettua in una sola sessione le seguenti operazioni:

- dialoga con la VA spedendole il voto protetto da blind factor e ricevendolo firmato (proprio come in [FOO92])
- manda il certificato elettorale alla TF (come n [FOO92])
- ottiene dalla TF una ricevuta di voto
- se tutto va bene spedisce immediatamente a TF la chiave per aprire il voto

Si noti che in questo modo Sensus rinuncia alla fairness, poiché la TF conosce i risultati parziali delle elezioni, in favore della convenienza.

Altri piccoli dettagli per cui si differenziano i due protocolli sono:

- lo schema di commitment viene rimpiazzato da un semplice sistema a chiavi pubbliche (PK_s e SK_s)
- Per spedire messaggi si utilizza sempre la chiave pubblica del destinatario.
- Il formato della lista di VA è semplificato

Protocollo

Fase di voto

1. V_i prepara $c_i = b(PK_s(m_i))$
2. $V_i \rightarrow$ VA: $PK_{VA}(V_i, SK_i(c_i), c_i)$
3. VA controlla se Vi ha diritto al voto e se non ha già votato.
4. VA controlla la validità della firma
5. VA $\rightarrow V_i$: $PK_{V_i}(SK_{VA}(c_i))$
6. VA $\rightarrow$ *: V_i, PK_i, flag "has voted"
7. V_i toglie il blind factor, ottenendo il certificato di voto vero e proprio: $SK_{VA}(d_i)$ dove $d_i = PK_s(m_i)$
8. V_i verifica che la chiave di VA sia corretta.
9. (Vi) $\rightarrow$ TF: $PK_{TF}(SK_{VA}(d_i), d_i)$
10 TF verifica che la chiave di VA sia corretta.
11. TF $\rightarrow$ *: $SK_{TF}(d_i)$, d_i (in posizione pos)
12. TF $\rightarrow$ (V_i): $SK_{TF}(d_i)$, pos
13 V_i verifica che la chiave di TF sia corretta.
14. (V_i) $\rightarrow$ TF: pos, SK_s
15. TF $\rightarrow$ *: $SK_{TF}(d_i)$, d_i, SK_{TF}, m_i (in posizione pos)

Verifica (opzionale)

16. Se V_i riscontra un'irregolarità invia in maniera anonima: $SK_{VA}(d_i)$, d_i, pos, SK_s

ASSUNTI DI SENSUS

- L’utilizzatore mette a disposizione canali sicuri, al riparo da intercettazione e ritardi
- Viene fornito un canale anonimo sicuro (suggerisce l’uso di una catena di server WWW anonimizzatori)
- Il computer utilizzato dall’elettore è sicuro e non è possibile per nessuno vedere o loggare quello che sta facendo l’utente
- I messaggi degli elettori a TF e VA non arrivino nello stesso ordine,
- Tutti gli algoritmi crittografici reggono

Con questi assunti Sensus soddisfa tutti i requisiti teorici e pratici con l’eccezione del secondo requisito della privacy

E-Vox (1997)

E-Vox è un'altra implementazione di [FOO92], utilizzato per le elezioni dei rappresentanti degli studenti al MIT nel 1998.

Contrariamente a Sensus si sforza di gestire ogni aspetto implementativo, lasciando il minor onere possibile all'utilizzatore.

Il protocollo originario è stato rivisto per permettere all'elettore di votare in una sola sessione.

Per implementare il canale anonimo fa uso di un server anonimizzatore che raccoglie tutti i voti criptati spediti dagli elettori durante la fase di voto. Al termine delle elezioni spedisce tutti i voti a TF.

La fairness è preservata ma se anonimizzatore e TF colludono possono violare la privacy degli elettori.

L'autore cita a propria difesa uno studio di Rivest secondo cui le tecniche note per implementare un canale anonimo sono tutte passibili dello stesso tipo di timing attack

PECULIARITÀ

- **Autenticazione basata su password**

  Gli elettori non hanno una coppia di chiavi asimmetriche ma userID e password

- **Distribuzione delle chiavi**

  Le chiavi dei server devono essere diffuse prima dell'avvio dell'elezione

- **Canali sicuri**

  Si utilizza un protocollo crittografico per rendere sicura ogni comunicazione

  A → B: $S(m, MAC), PK_B(S)$

  E per l'eventuale risposta:

  B → A: $S(m'), PK_B(S)$

- **Canale anonimo**

  La soluzione adottata è quella di un server anonymizer (AN).
  AN si pone tra l'elettore V e la TF. L'elettore invia il certificato elettorale a AN. AN raccoglie tutti i voti e al termine delle elezioni li spedisce a TF in ordine sparso

  V → AN: $S'(S(\text{certificato elettorale},\ldots), PK_{TF}(S),\ldots), PK_{AN}(S')$

  AN → TF: $S(\text{certificato elettorale},\ldots), PK_{TF}(S)$

- **Gestione degli errori**

  Viene aggiunto un server Commissioner a cui elettori e server mandano i messaggi di errore.

Protocollo

Prima fase: voto

1. $V_i \rightarrow$ VA: userID, password, $b(C_{k_i}(m_i))$
2. VA controlla se Vi ha diritto al voto e se non ha già votato.
3. VA controlla la validità della password. Se va tutto bene procede
4. VA $\rightarrow V_i$: $SK_{VA}(b(C_{k_i}(m_i)))$
5. VA $\rightarrow$ *: userID, $SK_{VA}(b(C_{k_i}(m_i)))$, $b(C_{k_i}(m_i))$
6. V_i toglie il blind factor, ottenendo il certificato di voto vero e proprio: $SK_{VA}(C_{k_i}(m_i))$
7. V_i verifica che la chiave di VA sia corretta. Se va tutto bene procede
8. Vi $\rightarrow$ AN: [$V_i \rightarrow$ TF: $SK_{VA}(C_{k_i}(m_i))$, $C_{k_i}(m_i)$, m_i, k_i]

Seconda fase: raccolta e conteggio voti

9. AN $\rightarrow$ TF: invio di tutti i messaggi in ordine sparso
10. TF $\rightarrow$ *: $\forall i$ $SK_{VA}(C_{k_i}(m_i))$, $C_{k_i}(m_i)$, m_i, k_i

Verifica (opzionale)

11. Se V_i riscontra inoltra un reclamo segnalando:

$SK_{VA}(C_{k_i}(m_i))$, $C_{k_i}(m_i)$, pos, k_i

COSA ABBIAMO OTTENUTO

- Nessun problema con la democracy
- Nessun problema col primo punto della privacy, a meno che TF e AN colludano
- Soddisfa la verificabilità debole
- Soddisfa la fairness
- Non è computazionalmente inefficiente
- Rispetta in buona misura il requisito di accuracy
- E' resistente agli attacchi di DoS, poiché ogni server mantiene una lista nera di indirizzi IP
- Soddisfa i 3 requisiti pratici (mobilità, flessibilità, convenienza)
  è progettato per funzionare via Internet e tutto quello che gli serve è un browser Web che supporti applet Java

PROBLEMI

- Se alcuni elettori si astengono VA può votare per loro
- Incapacità di distinguere tra V disonesto e TF disonesto
- Non soddisfa il secondo punto della privacy

Migliorie a E-Vox

E-VOX CON AMMINISTRATORI MULTIPLI (1999)

Problema: se alcuni elettori non votano, l'amministratore (VA) può votare per loro.

Soluzione: creare tante unità amministrative che non hanno interesse a colludere (per esempio una per ogni candidato), ciascuna delle quali dovrà firmare il voto. Perché TF accetti un voto dovrà avere un numero minimo t di firme valide su n totali. Naturalmente occorre che sia $t > n/2$, altrimenti lo stesso elettore potrebbe preparare due voti validi facendoli firmare da due gruppi diversi di amministratori.

Questa modifica è stata effettivamente operata in [DuR99] ed ora fa parte del protocollo. Oltre a migliorare l'accuracy questa modifica rende il sistema più scalabile, poiché il carico di lavoro può essere suddiviso tra i vari server amministratori.

ANONYMIZER RIDONDANTI

Problema 1: l'anonymizer può scartare voti validi. La cosa non può essere rilevata prima della conclusione delle elezioni

Soluzione: diversi anonymizer in parallelo: se il voto arriva anche da uno solo di essi verrà correttamente conteggiato.

Problema 2: l'anonymizer può colludere con TF e violare la privacy

Soluzione: una catena di anonymizer

Nuovi sviluppi

Sebbene [FOO92] abbia avuto un largo successo e moltissime implementazioni lungo tutto l'arco degli anni '90 (per esempio anche il sistema di voto sviluppato dal Cineca si basa su questo), l'approccio dei più recenti protocolli di voto è mutato.

Invece di "nascondere" il voto l'idea è quella di dividerlo in pezzi (share), dando ciascun pezzo a un'autorità di raccolta dei voti differente. Ogni autorità di raccolta (TF) conteggia i voti indipendentemente e invia i risultati parziali a un'autorità centrale (CA), che li mette insieme per ottenere il risultato finale. Mentre le TF non conoscono il voto perché ne hanno solo un pezzo, anche la CA non conosce nulla dei voti, avendo ricevuto dalle TF solo risultati aggregati. In più si fa in modo che per ricomporre il voto originario servano t di n share (in modo da supportare la verificabilità senza grossi rischi per la privacy).

Per maggiori approfondimenti:

- *http://VoteHere.net*
- *www.SafeVote.com*