

Aspetti di sicurezza nelle votazioni elettroniche

Progetto del corso di Sicurezza

Fabio Giuseppe Strozzi

<fstrozzi@cs.unibo.it>

Indice

1	Introduzione	1
1.1	Organizzazione	2
2	Receipt-freeness	2
2.1	Possibili attacchi	2
3	Strumenti	3
3.1	Sistema crittografico di ElGamal	3
3.2	Sistema crittografico di Paillier	4
3.3	Omomorfismo	5
3.4	Distribuzione della chiave segreta	5
3.5	Zero-knowledge proof	6
3.6	Mix-net	6
3.7	Untappable channels	7
3.8	Bulletin board	7
4	Schema di Hirt e Sako	7
4.1	Requisiti di base	8
4.2	Il protocollo	8
4.3	Implementazione	9
4.4	Sicurezza	10
4.4.1	Privacy	10
4.4.2	Correttezza e verificabilità	10
4.4.3	Robustezza	10
4.4.4	Receipt-freeness	10
4.5	Difetti	11
5	Schema di Acquisti	11
5.1	Panoramica	12
5.2	Fondamenta	12
5.3	Presupposti	13
5.4	Fasi preliminari	13
5.4.1	Creazione delle chiavi	13
5.4.2	Creazione delle quote delle scelte	14
5.5	Il protocollo	14
5.6	Sicurezza	16
5.6.1	Idoneità	16
5.6.2	Privacy	17
5.6.3	Correttezza	17
5.6.4	Verificabilità	17
5.6.5	Robustezza	18
5.6.6	Receipt-freeness	18
6	Conclusioni	19
	Riferimenti Bibliografici	22

1 Introduzione

Il sistema tradizionale di voto è caratterizzato dall'uso di schede di carta che vengono marcate dall'elettore in una apposita cabina; in questo sistema il conteggio è condotto manualmente all'interno dei seggi da parte di figure appositamente designate. Col termine *votazione elettronica* ci si riferisce a quei metodi di voto che differiscono dal tradizionale nel modo in cui viene effettuata la scelta da parte dell'elettore o nel meccanismo di conteggio: si va dalle schede perforate apparse negli anni '60, alle schede a lettura ottica degli anni '80 fino alle soluzioni di voto via internet proposte più di recente.

Sostituire i sistemi tradizionali con soluzioni informatiche può risultare vantaggioso se si riesce a garantire le seguenti proprietà (vedi [CGS97]):

Idoneità : solo gli elettori che hanno diritto al voto possono votare e possono farlo una sola volta.

Correttezza : il risultato dello scrutinio è coerente con le schede votate; solo le schede valide sono state conteggiate e solo quelle non valide sono state scartate.

Verificabilità : qualunque figura, compreso un osservatore passivo, può verificare che l'elezione è corretta.

Privacy : le schede elettorali devono essere tali che nessuna parte coinvolta nell'elezione possa associare un voto al rispettivo elettore. Questo deve valere anche quando più parti (escluso l'elettore stesso) si coalizzano.

Robustezza : il sistema elettorale deve garantire il funzionamento anche in caso di comportamenti erranei (maliziosi o meno) dei partecipanti (ovviamente entro una certa soglia prestabilita).

Scalabilità : le votazioni coinvolgono milioni di persone su scala nazionale, quindi è fondamentale che l'ordine dei votanti non influisca in maniera consistente sulle prestazioni del protocollo.

A fianco di queste fondamentali proprietà, il sistema elettorale elettronico permette di:

- migliorare i tempi dello scrutinio;
- facilitare l'accesso da parte di disabili;
- aumentare la partecipazione, ad esempio permettendo di votare anche da una città diversa da quella di appartenenza;
- ridurre i costi.

1.1 Organizzazione

Come i sistemi tradizionali, anche quelli elettronici possono essere vittima di attacchi. In questa ricerca verranno trattati in particolare gli aspetti di sicurezza legati al concetto di *receipt-freeness*. Nel prossimo capitolo verrà illustrato il problema e i possibili attacchi. Nel capitolo 3 verranno descritte le tecniche crittografiche necessarie per trattare l'argomento. I capitoli successivi analizzano due diverse soluzioni nell'ordine con cui sono state pubblicate. Il capitolo 4 analizza lo schema di Hirt e Sako [HS00] mentre il capitolo 5 presenta lo schema di Acquisti [Acq04].

2 Receipt-freeness

Nell'articolo [BT94], Benaloh e Tuinstra hanno per primi introdotto il concetto di *receipt* (ricevuta) per le votazioni elettroniche. Essi hanno mostrato come i protocolli di e-voting proposti fino ad allora presentavano tutti un difetto comune: l'elettore al termine dell'operazione possedeva una ricevuta che dimostrava la scelta effettuata. La ricevuta solleva un problema di sicurezza in quanto mina la segretezza del voto: se qualcuno è in grado di provare con assoluta certezza di aver votato in un certo modo, allora possono avvenire fenomeni quali la *vendita dei voti* e l'*estorsione dei voti*.

Di fronte a questo tipo di minacce, un sistema elettorale non solo dovrebbe assicurare che l'elettore *possa* tenere il proprio voto segreto, ma anche (e soprattutto) che *debba* tenerlo segreto.

La nozione di *receipt-freeness* è più forte di quella di privacy e stabilisce che l'elettore non deve essere in grado di dimostrare a terzi come ha votato; in sostanza, non deve poter ottenere o ricostruire alcuna prova che dimostri il contenuto della sua scheda. Come verrà mostrato in seguito, questa proprietà può essere soddisfatta contemporaneamente alla proprietà di verificabilità delle elezioni. In [JdV06] viene proposto un formalismo per catturare la nozione di receipt-freeness su cui si basano numerosi protocolli.

2.1 Possibili attacchi

Hirt e Sako hanno dimostrato in [HS00] che lo schema proposto da Benaloh e Tuinstra in realtà non è receipt-free perché un elettore è in grado di costruire una prova del proprio voto fornendo al sistema l'output di una funzione hash nota applicata ad una stringa arbitraria. Lo schema inoltre non era abbastanza robusto da tollerare la corruzione di alcune parti coinvolte nel protocollo.

Estorsione e vendita del voto diventano minacce ancora più serie e problematiche se il voto avviene via internet piuttosto che con i sistemi tradizionali. In [JCJ05], vengono elencati due possibili attacchi reali che possono essere condotti in mancanza di receipt-freeness:

Randomization : chi attacca può costringere l'elettore a fornire come scheda elettorale una stringa composta in maniera arbitraria. L'effetto dell'attacco è quello di rendere nulla la scheda con un'alta probabilità. Un attaccante che volesse favorire un certo partito potrebbe condurre questo attacco contro gli elettori di un seggio nel quale ha la maggioranza il partito avversario.

Forced-abstention : l'attaccante costringe l'elettore ad astenersi dal voto. Molti sistemi di e-voting sono suscettibili a questo attacco perché rendono pubblico l'elenco dei votanti (e quindi espongono l'elettore a possibili ritorsioni).

3 Strumenti

La crittografia è la chiave per risolvere i problemi di verificabilità e receipt-freeness. Di seguito verranno descritti gli "strumenti" con cui si costruiscono i protocolli analizzati in questa ricerca.

3.1 Sistema crittografico di ElGamal

ElGamal è un sistema crittografico a chiave pubblica. Mentre schemi come RSA si reggono sull'intrattabilità del problema della fattorizzazione in numeri primi, la sicurezza di ElGamal è legata all'assunzione decisionale di Diffie-Hellman e in particolare all'intrattabilità del problema del logaritmo discreto in un gruppo ciclico G . Si vedano [MvOV01] (capitolo 3.6) e [Mao03] (capitoli 5.2, 8.4 e 8.12) per un approfondimento sugli aspetti algebrici.

Generazione delle chiavi

Per generare una coppia di chiavi pubblica e privata bisogna seguire i seguenti passi:

1. Generare un numero primo grande p e un generatore g del gruppo moltiplicativo \mathbb{Z}_p^* degli interi modulo p .
2. Scegliere un intero random s , $1 \leq s \leq p - 2$ e calcolare $h = g^s \bmod p$.
3. La chiave pubblica sarà h e quella privata s .

Cifratura di un messaggio

ElGamal è un sistema di cifratura probabilistico: per un dato testo in chiaro possono esistere più crittogrammi prodotti con la stessa chiave pubblica. Lo schema, infatti, dipende dalla generazione di numeri casuali durante la cifratura. Dato un messaggio $m \in G$, si procede in questo modo:

1. Il mittente recupera la chiave pubblica h del destinatario.
2. Sceglie un numero casuale $\alpha, 1 \leq \alpha \leq p - 2$.
3. Calcola $\gamma = g^\alpha \bmod p$ e $\delta = m \cdot h^\alpha \bmod p$.
4. Spedisce il crittogramma (γ, δ) .
5. Il destinatario per decifrare (γ, δ) deve calcolare $m = \gamma/\delta^s$ usando la propria chiave privata s .

Si noti che il numero casuale α non deve essere noto a chi decifra il messaggio.

3.2 Sistema crittografico di Paillier

Anche il sistema crittografico di Paillier appartiene alla famiglia degli schemi a cifratura pubblica probabilistici. Per una trattazione approfondita si veda [Pai99, CGHGN01].

Generazione delle chiavi

La generazione delle chiavi avviene nel seguente modo:

1. Sia n il prodotto di due numeri interi primi p e q grandi e indipendenti l'uno dall'altro.
2. Calcolare $\lambda = \text{lcm}(p - 1, q - 1)$ dove lcm è il minimo comune multiplo.
3. Generare un numero casuale $g \in \mathbb{Z}_{n^2}^*$.
4. Calcolare l'inverso moltiplicativo

$$\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod n \text{ dove } L(u) = \frac{u - 1}{n}$$

5. La chiave pubblica è data dalla coppia (n, g) e quella privata dalla coppia (λ, μ) .

Cifratura di un messaggio

Per cifrare un messaggio $m \in \mathbb{Z}_n$:

- Il mittente sceglie un numero casuale $r \in \mathbb{Z}_{n^2}^*$.
- Spedisce il crittogramma $c = g^m r^n \bmod n^2$ usando le informazioni della chiave pubblica del destinatario.
- Quest'ultimo, decifra $c \in \mathbb{Z}_{n^2}^*$ calcolando

$$m = L(c^\lambda \bmod n^2) \cdot \mu \bmod n^2$$

3.3 Omomorfismo

Il sistema crittografico di Paillier viene utilizzato nei protocolli di e-voting perché ha l'importante caratteristica di essere un sistema omomorfico. Definiamo $E(m, r)$ l'operazione di cifratura di un messaggio in chiaro m usando il valore casuale r ; sia $D(c)$ l'operazione che decifra il crittogramma c . La proprietà di omomorfismo può essere espressa così:

$$D(E(m_1, r_1) \cdot E(m_2, r_2) \bmod n^2) = (m_1 + m_2) \bmod n^2$$

Ovvero, il prodotto di due crittogrammi viene decifrato come la somma dei relativi messaggi in chiaro.

Anche il sistema di ElGamal può essere omomorfico rispetto all'operazione di somma a patto di codificare il crittogramma non più con

$$(g^\alpha \bmod p, m \cdot h^\alpha \bmod p)$$

ma piuttosto come

$$(g^\alpha \bmod p, g^m \cdot h^\alpha \bmod p)$$

Sfortunatamente un tale sistema non gode di una trapdoor per calcolare m dato $g^m \bmod p$. Come vedremo, questo sarà uno dei difetti dello schema di Hirt e Sako (capitolo 4).

L'omomorfismo è una proprietà fondamentale per effettuare il conteggio dei voti senza "aprire" le schede. Infatti, dati c_1, \dots, c_n voti cifrati, il conteggio può essere fatto calcolando $c = c_1 \otimes \dots \otimes c_n$ e infine decifrando c . L'operatore \otimes rappresenta il prodotto tra crittogrammi nello schema di cifratura scelto (ElGamal o Paillier).

3.4 Distribuzione della chiave segreta

I sistemi ElGamal e Paillier possono essere adattati per far sì che una chiave segreta possa essere divisa in n parti dette *quote*. Ogni quota viene assegnata ad una particolare "autorità" (sarà chiaro più avanti); a quel punto, siccome nessuno possiede più la chiave segreta di partenza, un messaggio può essere decifrato solo se tutte le autorità interagiscono. Il senso di questa operazione è evitare che una singola autorità (corrotta) possa conoscere il contenuto di un voto cifrato e violare la privacy dell'elettore. Talvolta si ricorre ad una soglia di tolleranza $t \leq n$: in tal caso per decifrare un messaggio sono necessarie almeno t autorità su n . Questo meccanismo trova una analogia nelle tecniche di *key escrow*. Per un approfondimento si vedano [CGS97, HS00, BFP⁺01, Acq04].

Da ora in poi indicheremo con PK la chiave pubblica e con SK_i la quota di chiave segreta posseduta dalla i -esima autorità. Per cifrare un messaggio c un utente deve usare la chiave PK . Ogni autorità singolarmente ottiene solo una parziale decodifica c_i . Alla fine, se ci sono sufficienti parti deciptate, un algoritmo di combinazione recupera il testo in chiaro.

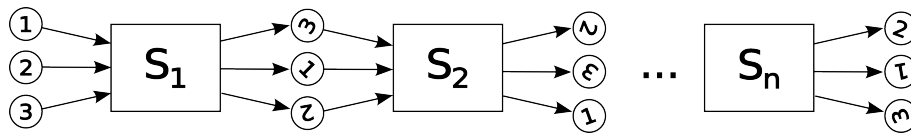


Figura 1: Mix-net con tre messaggi cifrati e n proxy server: i messaggi ruotati indicano che sono stati ri-cifrati; la permutazione è indicata dal diverso ordine con cui sono emessi in output.

3.5 Zero-knowledge proof

In crittografia, per *zero-knowledge proof* si intende un metodo interattivo in cui una parte (dimostratore) cerca di dimostrare ad un'altra (detta verificatore) che una certa sentenza (matematica) è vera senza rivelare null'altro che la veridicità della sentenza. Queste prove devono soddisfare tre requisiti:

Completezza : se la sentenza è vera allora un verificatore onesto sarà convinto di questo fatto da una prova corretta.

Correttezza : se la sentenza è falsa, allora la probabilità che qualcuno riesca a convincere del contrario un verificatore onesto è bassa.

Zero-knowledge : il verificatore non è in grado di apprendere dalla prova null'altro che la sua veridicità.

Prove fatte in questo modo servono a controllare che le parti interessate stiano seguendo il protocollo onestamente; allo stesso tempo la verifica è tale da non violare i segreti dei soggetti interessati. Nei capitoli 4 e 5 verranno mostrati i possibili usi. I dettagli sulla loro costruzione si trovano in [Mao03] capitolo 18 e [CGS97, HS00, BFP⁺01, JCJ05, Acq04].

3.6 Mix-net

Le mix-nets (o digital mixes) sono delle reti all'interno delle quali dei *proxy server* permutano e ri-cifrano i messaggi in arrivo in modo da eliminare ogni relazione tra i messaggi finali e i relativi mittenti. Esistono diverse implementazioni di queste reti, come quelle illustrate in [Cha81, HS00, BT94]. La figura 1 mostra un esempio con tre messaggi in ingresso. Ogni server ordina casualmente i messaggi in entrata e non rivela agli altri server la permutazione. È sufficiente che anche un solo server si comporti secondo questo protocollo perché venga persa ogni relazione tra il mittente ed il messaggio.

3.7 Untappable channels

Con questo termine si indicano dei canali di comunicazione che garantiscono la totale segretezza dei messaggi scambiati. Tali canali devono essere realizzati fisicamente e non possono pertanto basarsi sulla rete Internet [TR06].

3.8 Bulletin board

Il *bulletin board* può essere considerato un canale pubblico di comunicazione con memoria [CGS97]: tutte le comunicazioni possono essere lette da tutti i partecipanti all'elezione ma nessuno può cancellarle. Solo agli utenti autorizzati è concesso di accodare nuovi messaggi. Per garantire che il contenuto non venga alterato, le autorità delle elezioni possono periodicamente firmarlo.

4 Schema di Hirt e Sako

L'idea alla base di questo modello è che l'elettore effettua la propria scelta di voto partendo da un elenco di scelte che è stato precedentemente permutato casualmente e criptato grazie a una mix-net (si immaginino le possibili scelte come singoli messaggi). Siccome ad utenti diversi corrispondono permutazioni e ri-codifiche diverse, non sarà possibile risalire all'elettore una volta che ha effettuato la scelta.

Le entità coinvolte nel processo sono N autorità A_1, \dots, A_N (ognuna può corrispondere ad un server) e M votanti. Una soglia t indica il numero di autorità che bisogna corrompere per violare la sicurezza del protocollo. Le autorità svolgono un duplice ruolo:

- Costituiscono la mix-net che permuta e cifra le possibili scelte di voto.
- Svolgono l'operazione di scrutinio.

Lo schema funziona per schede le cui possibili scelte sono 2 (“si” o “no”), 3 (“si”, “no” o “bianca”) o maggiori (ad esempio una lista di nomi). Indichiamo con V l'insieme delle scelte valide (cioè quelle ammesse dalla scheda elettorale) e L la sua cardinalità.

Il protocollo si basa sulle seguenti assunzioni:

- I canali che collegano gli elettori alle autorità sono di tipo *one-way untappable*: nessuno, compreso l'elettore, può dimostrare cosa viene trasmesso attraverso i canali (in particolare nel verso che va dalle autorità all'utente).
- Esiste una *national key infrastructure*, cioè una infrastruttura a livello nazionale che rilascia e certifica le chiavi pubbliche e private di ogni elettore.

- Esiste un meccanismo di autenticazione che permetta solo agli elettori legittimi di accedere al voto.

4.1 Requisiti di base

Oltre alle proprietà di omomorfismo e alla distribuzione della chiave privata tra le autorità (già trattate nei capitoli 3.3 e 3.4), Hirt e Sako fanno uso di altre proprietà dei sistemi crittografici:

1. Verifiable decryption

Si richiede che, data la somma dei voti criptati $e \in E(T)$, le autorità siano in grado di fornire sia la somma dei voti T che la prova (di tipo zero-knowledge) che T corrisponde effettivamente alla decodifica di e .

2. Random re-encryption

Si richiede che dato il crittogramma $e \in E(v)$ del voto v (ignoto), esista un algoritmo probabilistico R che cifri nuovamente e ottenendo $e' \in E(v)$. Il valore random usato per la nuova codifica è chiamato *testimone*. Questa proprietà viene anche chiamata **self-blinding**.

3. 1 out of L re-encryption proof

Si assume l'esistenza di un meccanismo efficiente per generare la prova (di tipo zero-knowledge) che dimostri che, data una lista di crittogrammi e_1, \dots, e_L ed un valore e , esiste un e_i appartenente alla lista che ri-cifra e . Ciò dev'esser fatto senza rivelare i .

4. Designated-verifier re-encryption proof

Dato il crittogramma e , si assume che sia possibile costruire una prova zero-knowledge che dimostri, senza rivelare il *testimone* usato, che una ri-codifica $e' \in R(e)$ è corretta (cioè non ha alterato il testo in chiaro). La prova deve essere costruita in modo che un solo destinatario designato possa verificarne la correttezza.

4.2 Il protocollo

Il protocollo passa attraverso tre fasi: generazione della scheda, votazione e infine conteggio.

Generazione della scheda

Supponiamo che la scheda elettorale ammetta le possibili scelte $v_i \in V$ e sia $e_i^{(0)} \in E(v_i)$ la *cifatura standard* di v_i ottenuta usando la chiave pubblica PK delle autorità. Infine, sia $e_i^{(0)}, \dots, e_L^{(0)}$ una lista pubblica di tutte le scelte cifrate. A turno, per ogni autorità A_k (dove $k = 1, \dots, N$):

1. A_k riceve la lista $e_1^{(k-1)}, \dots, e_L^{(k-1)}$ dall'autorità che la precede e agisce come un proxy server di una mix-net: per ogni $i = 1, \dots, L$, usando la permutazione π_k , A_k ri-cifra $e_i^{(k-1)}$ in $e_{\pi_k(i)}^{(k)}$ (grazie alla proprietà 2).
2. A_k sfruttando la proprietà 3 crea una prova dell'esistenza di una nuova codifica di $e_i^{(k-1)}$ tra i suoi $e_1^{(k)}, \dots, e_L^{(k)}$ per ogni $i = 1, \dots, L$. Successivamente pubblica le permutazioni e le L prove su un bulletin board.
3. Usando il canale *untappable*, rivela privatamente all'utente la permutazione operata: per ogni $i = 1, \dots, L$, crea la prova che la scelta $e_{\pi_k(i)}^{(k)}$ è la ri-codifica del crittogramma $e_i^{(k-1)}$ (proprietà 4).
4. L'elettore riceve la permutazione π_k e la relativa prova di correttezza: se dalla verifica riscontra dei problemi può segnalarli in modo tale che la permutazione dell'autorità A_k venga ignorata. L'elettore può reclamare al più contro $N - t$ autorità.

Votazione

Al termine del procedimento, l'elettore ha ricevuto tutte le permutazioni delle scelte della scheda e quindi è in grado di risalire all'indice i del voto cifrato $e_i^{(N)}$ che rappresenta la propria scelta. Quindi annuncia pubblicamente i sul bulletin board.

Scrutinio

I crittogrammi dei voti degli utenti vengono sommati usando la proprietà di omomorfismo descritta nel capitolo 3.3: si ottiene il crittogramma $E(T)$. Grazie al meccanismo della distribuzione della chiave privata (capitolo 3.4), servono almeno t autorità per ricavare T . Una volta ottenuto questo valore le autorità generano una prova della correttezza dello scrutinio (proprietà 1).

4.3 Implementazione

Il sistema crittografico che gli autori hanno scelto per una possibile realizzazione del protocollo è quello di ElGamal opportunamente modificato per essere omomorfo rispetto all'addizione e per gestire la distribuzione della chiave privata (capitoli 3.1, 3.3 e 3.4). Con questo sistema hanno dimostrato che è possibile soddisfare le proprietà descritte nel capitolo 4.1. Si rimanda all'articolo [HS00] per i dettagli sulle dimostrazioni.

Un aspetto importante è legato alla codifica delle scelte. Come è stato detto all'inizio del capitolo 4, questo protocollo supporta schede elettorali con più di 2 scelte. Ma come si è visto poc'anzi, la somma dei voti non fa

distinzione tra una scelta e l'altra (per non violare la segretezza). D'altra parte, nelle elezioni è indispensabile distinguere il numero di preferenze. Per recuperare le somme parziali dal conteggio totale, gli autori propongono di esprimere le scelte delle schede come $V = \{1, M, M^2, \dots, M^{L-1}\}$ dove M è il numero di elettori.

4.4 Sicurezza

Diversi articoli [JCJ05, TR06, Acq04, BFP⁺01] hanno messo in luce i difetti del protocollo di Hirt e Sako. Di seguito verranno elencate le proprietà di sicurezza che non sono state ancora confutate. I difetti veri e propri sono trattati nel capitolo 4.5.

4.4.1 Privacy

L'elettore ha la garanzia che il suo voto non verrà decriptato da una persona esterna o da un gruppo di autorità in numero inferiore a t .

Inoltre, anche se le permutazioni delle scelte della scheda vengono pubblicate sul bulletin board, non è possibile risalire a quale scelta della lista originale $(e_i^{(0)}, \dots, e_L^{(0)})$ corrisponde il voto $e_i^{(N)}$ di un elettore (per succedere devono collaborare almeno t autorità).

4.4.2 Correttezza e verificabilità

Il risultato dello scrutinio T , il suo valore criptato $e \in E(T)$ e la prova che che T è la decodifica di e sono resi pubblici: chiunque può confermare (o confutare) la correttezza della prova.

Secondo il protocollo, tutte le liste di scelte permutate e ri-cifrate da parte delle autorità della mix-net, insieme alle prove di correttezza della permutazione e all'indice i della scelta $e_i^{(N)}$ effettuata dall'elettore devono essere pubblicate nel bulletin board (passo 2 della generazione della scheda). Quindi chiunque può verificare che:

1. il proprio voto compaia nell'elenco;
2. ogni permutazione sia accompagnata da una prova valida;
3. il totale dei crittogrammi scelti sommati tra loro corrisponda a e .

4.4.3 Robustezza

Il protocollo è robusto nel senso che è in grado di tollerare la corruzione (o il guasto) di al più t autorità, anche nel caso in cui queste collaborino tra di loro.

4.4.4 Receipt-freeness

L'elettore non è in grado di provare a nessun altro di aver votato in un certo modo o di non aver votato. Infatti, qualunque prova egli utilizzi, questa non è più credibile di una prova costruita ad-hoc. Anche le prove del tipo zero-knowledge che soddisfano la proprietà 4 e che le autorità forniscono all'elettore non sono trasferibili ad una terza persona. Si rimanda all'articolo [JSI96] per capire come si ottiene questo tipo di prove. Inoltre, il bulletin board è assolutamente anonimo: il voto dell'elettore è associato esclusivamente alle scelte cifrate e permutate $e_i^{(N)}$ e in nessun caso a dati sensibili che possono identificarlo. Questo permette di evitare attacchi di tipo *forced-abstention*.

Un problema può sorgere se le autorità collaborano con un estorsore o un compratore di voti. In tal caso la receipt-freeness è garantita fintanto che l'utente conosce almeno una autorità onesta A_k che gli permette di mentire sulla permutazione π_k .

4.5 Difetti

Come si è detto nel capitolo 3.3 il sistema crittografico ElGamal modificato per essere omomorfico rispetto all'operazione di somma è privo di trapdoor per calcolare efficientemente T dato $g^T \bmod p$ (dove T è il conteggio finale). La soluzione banale sarebbe quella di provare tutti gli esponenti fino ad ottenere g^T . Come proposto da altri autori [CGS97] però, si può ricorrere ad un algoritmo quale il "baby-step, giant-step" [Wik] e ottenere la soluzione in tempo $O(\sqrt{M}^{L-1})$. Ciò nonostante, il costo computazionale del protocollo nei termini in cui è stato proposto dagli autori, è molto elevato soprattutto quando le scelte possibili sulla scheda sono molte e si vogliono ottenere anche i risultati parziali. Gli autori stessi ammettono che il protocollo può esser reso efficiente solo nel caso di schede con due sole scelte possibili.

Baudron *et al.* in [BFP⁺01] hanno proposto uno schema che, partendo da quello di Hirt e Sako, mira a risolvere questi problemi. Come prima cosa gerarchizzano le autorità in una struttura ad albero in modo da ridurre il numero di elettori per ciascuna e convergere le somme parziali verso poche autorità alla radice. Inoltre, fanno uso del sistema crittografico di Paillier che, diversamente da ElGamal, sfrutta una trapdoor per calcolare efficientemente la somma dei voti.

Juels *et al.* in [JCJ05] osservano che l'assunzione che l'elettore possa conoscere quali sono le autorità oneste è troppo forte.

Infine, l'assunzione di utilizzare canali di comunicazione di tipo *untappable* è sconveniente perché preclude automaticamente l'utilizzo del protocollo su Internet.

5 Schema di Acquisti

A differenza dello schema di Hirt e Sako, il protocollo di Acquisti non si basa su assunzioni fisiche ad-hoc (come i canali *untappable*). È uno schema teorico ma si presta maggiormente ad essere schierato con diverse configurazioni: votazioni elettroniche con schede di carta, cabine elettorali, smartcards, la rete Internet, ecc.

Un'altra differenza fondamentale è che con questo schema non si decifra la somma dei voti, ma ogni singolo voto.

5.1 Panoramica

L'idea che sta dietro a questo protocollo è molto semplice. Il processo di elezione è controllato da una "Autorità delle elezioni", un soggetto composto da un certo numero di server indipendenti (detti autorità). Le schede elettorali sono divise in "quote", ognuna delle quali viene fornita da una precisa autorità. Ogni legittimo elettore ha diritto ad una *credenziale*¹: anche questa è divisa in quote (dette credenziali) e vengono fornite sempre dai server dell'Autorità.

Le autorità pubblicano su un bulletin board delle "copie" equivalenti delle quote delle schede e delle credenziali che sono state fornite agli utenti.

Questi ultimi, affinché il loro voto venga conteggiato nello scrutinio finale, devono fornire tutte le credenziali e tutte le quote della scheda che hanno ricevuto. Il loro voto viene pubblicato sul bulletin board.

Prima dello scrutinio i voti degli elettori vengono mescolati in una *mix-net* per far perdere ogni relazione. Durante lo scrutinio vengono conteggiati solo i voti degli utenti che hanno fornito tutte le credenziali e tutte le quote della scheda.

5.2 Fondamenta

Lo schema di Acquisti è costruito a partire dai seguenti elementi:

- Versione con soglia del sistema crittografico di Paillier nel quale la chiave segreta viene distribuita tra più entità (descritto nel capitolo 3.4). Di questo sistema vengono sfruttate in particolare le proprietà di *omomorfismo* (capitolo 3.3) e *self-blinding* (proprietà 2 del capitolo 4.1). Per i dettagli sulla generazione e la distribuzione delle chiavi di Paillier con soglia si veda l'appendice dell'articolo [Acq04].
- Una rete di server del tipo *mix-net*.
- Le prove di tipo *zero-knowledge*, usate in particolare per mostrare l'equivalenza di due crittogrammi derivati dallo stesso testo in chiaro ma

¹La si consideri un documento che certifica che l'elettore è ammesso al voto.

generati con chiavi pubbliche diverse (queste prove sono simili a quelle descritte per la proprietà 4 nel capitolo 4.1). Il metodo di creazione viene descritto nell'appendice dell'articolo di Acquisti [Acq04].

- Un *bulletin board* per memorizzare le operazioni di interesse pubblico (capitolo 3.8).

5.3 Presupposti

L'autore stabilisce alcune assunzioni indispensabili:

- Esiste una infrastruttura a livello nazionale per la gestione delle chiavi pubbliche e dei certificati dei cittadini.
- Le autorità sono a conoscenza delle chiavi pubbliche degli aventi diritto al voto.
- Un attaccante non può controllare tutte le possibili comunicazioni tra un elettore e una autorità (questo è più ragionevole da ottenere rispetto ai canali untappable, ad esempio tramite broadcast anonimo o mix-net). Questa assunzione non mira tanto a garantire la privacy (che è mantenuta comunque), quanto ad evitare che si incorra in attacchi del tipo forced-abstention (capitolo 2.1).
- Il numero di autorità che possono essere corrotte è inferiore al numero minimo di autorità che servono per decifrare i crittogrammi. Indicheremo questa soglia con y .

5.4 Fasi preliminari

Sia \mathcal{A} l'Autorità delle elezioni composta da s server A_1, \dots, A_s . Indichiamo con \mathcal{BB} il bulletin board. Inoltre, sia l il numero di elettori legittimi, indicati con v_j , per $j = 1, \dots, l$. I loro nomi sono pubblicati nel \mathcal{BB} prima dell'inizio delle elezioni.

Le autorità assumono più ruoli durante il processo:

- creano e distribuiscono le quote delle schede e le credenziali agli elettori;
- agiscono come proxy servers all'interno della mix-net;
- eseguono lo scrutinio dei voti pubblicati in \mathcal{BB} .

Supponiamo che il numero di scelte ammesse dalla scheda sia il valore fissato T . Indichiamo con b_i^t , dove $i = 1, \dots, s$ e $t = 1, \dots, T$, la i -esima quota della scelta t -esima della scheda elettorale. Ad esempio: b_1^3 rappresenta la quota della terza scelta fornita dalla autorità A_1 . Definiamo una possibile scelta come la somma di tutte le quote fornite dalle autorità per quella scelta: $B^t = \sum_{i=1, \dots, s} b_i^t$.

5.4.1 Creazione delle chiavi

Prima delle votazioni \mathcal{A} genera e distribuisce alle autorità due set di chiavi pubbliche/private di Paillier:

- il set C per le credenziali: PK^C e SK_i^C .
- il set V per le quote delle schede: PK^V e SK_i^V .

Indicheremo con $E^C()$ e $E^V()$ le rispettive operazioni di cifratura. Inoltre, \mathcal{A} crea un terzo set di chiavi distribuite grazie ad un sistema a soglia non omomorfo (tipo RSA): PK^S e SK_i^S . Scriveremo $E^S()$ per indicare la relativa operazione di codifica. Tutte le chiavi pubbliche sono rese disponibili tramite il \mathcal{BB} .

5.4.2 Creazione delle quote delle scelte

Viene creata la lista delle scelte ammesse B^t . Ogni autorità A_i crea le proprie quote per ciascuna scelta ammessa b_i^t . Le quote non sono altro che dei numeri generati casualmente. Ogni A_i cifra b_i^t una prima volta con PK^C e una seconda volta con PK^V usando numeri casuali differenti e indipendenti (si veda il capitolo 3.2 per i dettagli); si ottengono rispettivamente $E^C(b_i^t)$ e $E^V(b_i^t)$. I due crittogrammi vengono firmati dall'autorità la quale crea anche una prova zero-knowledge per dimostrare che essi codificano lo stesso messaggio. I crittogrammi e la prova vengono pubblicati sul bulletin board in una sezione riservata alle schede in modo tale che siano chiari:

- la distinzione tra il crittogramma ottenuto con PK^C e quello ottenuto con PK^V ;
- la scelta della scheda a cui si riferiscono;
- da quale autorità sono stati prodotti.

5.5 Il protocollo

Il protocollo si articola in tre fasi: preparazione, votazione e scrutinio.

Preparazione

Ciascuna autorità A_i genera l numeri casuali che rappresentano le credenziali per gli l elettori v_j . Identifichiamo ciascuna credenziale con $c_{i,j}$ dove $j = 1, \dots, l$ e $i = 1, \dots, s$. A_i cifra ogni $c_{i,j}$ da lui generata usando PK^C e firma il crittogramma ottenuto con SK_i^C . Il crittogramma $E^C(c_{i,j})$ firmato viene pubblicato sul \mathcal{BB} in uno spazio riservato alle credenziali dell'elettore v_j .

Successivamente A_i cifra $c_{i,j}$ usando la chiave PK^V e un opportuno valore casuale. Invece di firmare il crittogramma, gli allega una *designated*

verifier proof P_{v_j} , che prova l'equivalenza dei testi in chiaro dei crittogrammi $E^C(c_{i,j})$ e $E^V(c_{i,j})$. La prova è costruita in modo da essere verificabile solo dall'elettore v_j .

A questo punto le autorità cifrano il secondo crittogramma e la prova e spediscono il risultato all'elettore: $E^{v_j}(E^V(c_{i,j}), P_{v_j})$. E^{v_j} rappresenta l'operazione di cifratura usando la chiave pubblica dell'elettore (che può essere di tipo RSA). Le autorità non memorizzano i crittogrammi $E^V(c_{i,j})$.

Votazione

L'elettore ha ricevuto tutti i crittogrammi delle credenziali. Per ognuno verifica la correttezza della prova: ciò può essere fatto perché $E^C(c_{i,j})$ è pubblicato nel \mathcal{BB} . Se la prova ha successo, allora moltiplica tra loro le credenziali:

$$\bigotimes_{i=1,\dots,s} E^V(c_{i,j}) = E^V\left(\bigoplus_{i=1,\dots,s} c_{i,j}\right) \equiv E^V(C_j)$$

Gli operatori \otimes e \oplus indicano rispettivamente la produttoria dei crittogrammi e la sommatoria dei testi in chiaro secondo le regole dell'omomorfismo del sistema di Paillier (si veda il capitolo 3.3).

In maniera analoga, recupera le quote $E^V(b_1^t), \dots, E^V(b_s^t)$ corrispondenti alla propria scelta di voto t dal \mathcal{BB} , le somma tra loro e ottiene $E^V(B_j^t)$.

Quindi, moltiplica i due crittogrammi ottenuti:

$$E^V(C_j) \otimes E^V(B_j^t) = E^V\left(\bigoplus_{i=1,\dots,s} c_{i,j} \oplus \bigoplus_{i=1,\dots,s} b_{i,j}^t\right) \equiv E^V(C_j \oplus B_j^t)$$

Alla fine, cifra il risultato con la chiave pubblica non-omomorfica PK^S e lo spedisce al \mathcal{BB} : $E^S(E^V(C_j \oplus B_j^t))$.

Scrutinio

Allo scadere del tempo, \mathcal{A} moltiplica tra di loro le credenziali che aveva pubblicato nel \mathcal{BB} :

$$\forall j, \bigotimes_{i=1,\dots,s} E^C(c_{i,j}) = E^C\left(\bigoplus_{i=1,\dots,s} c_{i,j}\right) \equiv E^C(C_j)$$

Successivamente, mescola e ri-cifra questi crittogrammi usando la chiave pubblica PK^C attraverso la mix-net di autorità: si ottiene $E^C(C_{\pi(j)})$, dove π è una permutazione casuale.

Separatamente, decifra tutti i crittogrammi $E^S(E^V(C_j \oplus B_j^t))$ ricevuti dagli utenti². \mathcal{A} permuta e ri-cifra i crittogrammi ricavati usando la chiave

²Possono essere in numero superiore a l dal momento che gli utenti possono aver più volte utilizzato delle credenziali false o già usate. Sarà chiaro più avanti che ciò non costituisce un problema.

pubblica PK^V e la mix-net: si ottiene $E^V(C_{\pi(j)} \oplus B_{\pi(j)}^t)$, dove π è una permutazione casuale.

Infine, ricava i crittogrammi delle possibili scelte $E^C(B^t)$ sommando tra di loro le quote $E^C(b_1^t), \dots, E^C(b_s^t)$ pubblicate sul \mathcal{BB} .

Ora \mathcal{A} possiede gli elementi necessari per effettuare lo scrutinio:

- Da un lato ha l'elenco dei voti $E^V(C_{\pi(j)} \oplus B_{\pi(j)}^t)$ spediti dagli elettori, cifrati usando il set di chiavi V e opportunamente permutati usando la mix-net.
- Dall'altro ha la lista delle credenziali di ogni utente $E^C(C_{\pi(j)})$ mescolate e la lista delle possibili scelte $E^C(B^t)$: entrambe contengono crittogrammi ottenuti col set di chiavi C .

Tutte e tre le liste sono rese pubbliche nel \mathcal{BB} . Ricordiamo che, per poter decifrare qualunque crittogramma, devono collaborare almeno y autorità (capitolo 5.3).

Si noti che, dalla proprietà di omomorfismo (capitolo 3.3), segue:

$$E^C(C_{\pi(j)}) \otimes E^C(B^t) \equiv E^C(C_{\pi(j)} \oplus B^t) \quad (1)$$

L'ultima operazione è la ricerca dei voti validi. L'algoritmo segue i seguenti passi:

1. Si sceglie una credenziale $E^C(C_{\pi(j)})$ dalla lista (a quale elettore appartenga ora è ignoto).
2. Si prende una scelta cifrata $E^C(B^t)$ dalla lista delle possibili scelte offerte dalla scheda.
3. Grazie all'equivalenza 1 si controlla se esiste un voto tale che:

$$D(E^C(C_{\pi(j)} \oplus B^t)) \stackrel{?}{\equiv} D(E^V(C_{\pi(j)} \oplus B_{\pi(j)}^t)) \quad (2)$$

4. Se un tale voto esiste, allora è certamente valido e deve essere contato. In questo caso, si aggiunge il voto al conteggio, si rimuove la credenziale $E^C(C_{\pi(j)})$ dalla lista e si riparte dal punto 1. Se nessun voto viene trovato l'algoritmo riparte dal punto 2 pescando una nuova scelta $E^C(B^t)$: se tutte le schede sono state provate invano, l'algoritmo rimuove la credenziale $E^C(C_{\pi(j)})$ e riprende dal punto 1.
5. Quando tutte le credenziali sono state considerate, lo scrutinio è terminato.

5.6 Sicurezza

Di seguito verrà analizzata la sicurezza offerta all'elettore da questo schema.

5.6.1 Idoneità

L'algoritmo non pone vincoli al numero di persone che possono esprimere un voto né al numero di voti esprimibile da ciascuno di essi. Ciò detto, la fase di conteggio ripara a questa "carenza" ammettendo come validi solo i voti degli elettori che hanno presentato le opportune credenziali. L'idoneità è quindi garantita dall'elenco di credenziali che vengono pubblicate sul \mathcal{BB} .

5.6.2 Privacy

La privacy è garantita dall'utilizzo del sistema crittografico di Paillier e dall'uso della mix-net per mescolare e ri-cifrare le credenziali e i voti.

La segretezza del voto è preservata anche se un elettore spedisce il proprio voto in modo non anonimo perché nessuna autorità è in grado di ottenere tutte le quote/credenziali che compongono il voto (essendo queste cifrate diversamente da quelle rese pubbliche nel \mathcal{BB}).

Per violare la privacy servirebbero almeno y autorità che collaborino tra loro per: decifrare tutte le credenziali $E^C(C_j)$ originariamente pubblicate sul \mathcal{BB} ; decifrare i voti degli elettori $E^V(C_j \oplus B_j^t)$; effettuare gli opportuni confronti. Ma per le assunzioni fatte in 5.3 questo non può succedere.

5.6.3 Correttezza

Solo un voto per credenziale può essere contato dall'algoritmo descritto nel capitolo 5.5.

Nessuna delle autorità può usare le credenziali di un elettore per votare al suo posto perché:

- le credenziali spedite all'elettore sono cifrate con un set di chiavi diverso da quello usato per cifrare le credenziali di dominio pubblico;
- le credenziali $E^C(C_{\pi(j)})$ non vengono mai decriptate singolarmente ma solo dopo che sono state moltiplicate con una opportuna scelta $E^C(B^t)$.

I crittogrammi dei voti degli elettori, anche se sono pubblicati, sono protetti dalla cifratura non omomorfa della chiave PK^S : pertanto il loro contenuto non è accessibile da meno di y autorità.

Nel caso in cui venisse spedita due o più volte una scheda duplicata (attacco di tipo *replay*), l'algoritmo conterebbe il voto una sola volta.

Affinché tutto ciò possa essere vero, però, è necessario che la lunghezza delle quote e delle credenziali sia sufficientemente grande da rendere trascurabile la probabilità che due voti diversi collidano. Contemporaneamente, bisogna prestare attenzione a che la somma delle credenziali o delle quote non sia più grande del dominio dei crittogrammi.

5.6.4 Verificabilità

Gli elettori possono fare diversi controlli:

- grazie alla prova P_{v_j} , possono verificare che le credenziali che hanno ricevuto e quelle pubblicate siano equivalenti;
- possono accertare che il loro voto $E^V(C_j \oplus B_j^t)$ compaia nella lista dei voti sul bulletin board (cifrato con la chiave PK^S);

5.6.5 Robustezza

Un'autorità che non pubblichi e firmi le sue credenziali $E^C(C_{i,j})$ sul \mathcal{BB} può essere scoperta e sostituita prima che la fase di voto inizi.

Un'autorità che non fornisca agli elettori le credenziali e l'opportuna prova di equivalenza può essere scoperta dagli elettori stessi nella fase di preparazione. Anche in questo caso si può provvedere alla sua sostituzione. Nel caso in cui il problema persista, dopo che un certo numero di autorità sono state sostituite, l'Autorità \mathcal{A} può concludere che l'elettore sta mentendo.

Non è necessario che tutti gli elettori registrati votino: l'operazione di scrutinio inizia semplicemente allo scadere del tempo.

Infine, come ripetuto più volte, tutte le operazioni di decodifica dei crittogrammi possono avvenire solo se almeno y server collaborano tra di loro.

5.6.6 Receipt-freeness

Consideriamo il caso di un estorsore o di un compratore di voti: entrambi hanno bisogno di ricevere dall'elettore delle prove che dimostrino il modo con cui ha votato. Vediamo per quali ragioni non potranno avere la certezza che una prova fornita loro non è falsa:

- Né l'elettore, né le autorità conoscono le vere credenziali: quando vengono pubblicate sono in forma cifrata e durante lo scrutinio non vengono mai decifrate singolarmente ma solo dopo esser state sommate ai voti possibili (che a loro volta sono generati in maniera distribuita).
- Per certificare la validità delle credenziali, le autorità spediscono all'elettore delle prove di tipo *designated verifier*: nessuno, eccetto l'elettore, può essere convinto che le credenziali non sono state falsificate.
- Lo scrutinio ha inizio dopo che i voti sono stati mescolati, per cui nessuno è in grado di risalire al proprio voto, anche se questo viene decifrato e reso pubblico.
- La chiave privata è il mezzo con cui l'elettore si identifica all'autorità e attraverso il quale instaura un canale di comunicazione privato. Un attaccante potrebbe chiedere all'elettore di rivelare la chiave

privata. Tuttavia, a questa richiesta l'elettore potrebbe rispondere con un chiave privata falsa, costruita appositamente per ingannarlo. Supponiamo, infatti, di estendere il protocollo introducendo una fase iniziale di “handshaking” durante la quale l'elettore genera un nuovo set temporaneo di chiavi pubblica/privata che verrà usato nelle le fasi successive per criptare il traffico e per generare la *designated verifier proof*. A questo punto, è chiaro che l'attaccante non può più essere convinto che la chiave che ha ricevuto è quella temporanea che è stata impiegata veramente.

- Supponiamo allora di aver introdotto la fase di handshaking. Ora, in luogo della ricevuta, l'estorsore potrebbe richiedere alla vittima di mostrargli come creare il voto $E^S(E^V(C_j \oplus B_j^t))$ che sia compatibile con $E^{v_j}(E^V(c_{i,j}), P_{v_j})$. Però, dato che la *designated verifier proof* P_{v_j} non è firmata dall'autorità, anche l'elettore è in grado di costruirne una simile a partire da credenziali false e una seconda chiave temporanea, finta, creata ad-hoc. Così facendo, l'attaccante non può più stabilire se la prova zero-knowledge che ha ricevuto è riferita alle vere credenziali e alla chiave privata utilizzata realmente.

Il protocollo di Acquisti è immune ad un attacco di tipo *randomization*: infatti, lo scrutinio non rivela il contenuto del voto, ma si limita a mostrare la decodifica della somma delle credenziali e di una scelta possibile. Così, si riduce l'insieme di informazioni al quale un attaccante può avere accesso. Inoltre, siccome non c'è limite al numero di schede che possono essere spedite alle autorità, un elettore potrebbe inviare una scelta non valida per accontentare l'estorsore e subito dopo una valida. La prima non verrebbe conteggiata, la seconda sì.

Anche un attacco di tipo *forced-abstention* non sarebbe efficace: l'elettore, infatti, potrebbe usare un canale anonimo per comunicare. Per di più, le autorità non possono rivelare chi ha votato e chi no perché, durante la fase di scrutinio e grazie alla mix-net, si perde ogni relazione tra la scheda e l'elettore (si ricordi che le credenziali non vengono mai decifrate singolarmente).

6 Conclusioni

Per superare la diffidenza nei confronti del voto elettronico, i primi protocolli rilasciavano agli utenti una sorta di ricevuta cartacea del proprio voto. Talvolta, però, è bene non essere in grado di dimostrare come si è votato. Un sistema di e-voting che non sia in grado di imporre la segretezza del voto non sarà mai più vantaggioso del metodo tradizionale.

Receipt-freeness non significa strettamente “assenza di ricevuta”: un attaccante potrebbe risalire al modo con cui ha votato un utente anche se a

quest'ultimo non viene rilasciata alcuna ricevuta. Per esempio, un sistema tradizionale di voto, in cui l'elettore è chiamato a scegliere dei candidati e ad elencarli in ordine di preferenza sulla scheda non è receipt-free. Un estorsore che sia presente ai seggi, potrebbe richiedere alla vittima di votare certi candidati in una certa sequenza; assistendo allo spoglio, poi, potrebbe capire se quella sequenza è uscita o meno, cioè se la sua vittima ha agito come richiesto oppure no.

Con questa ricerca sono stati analizzati in dettaglio due diversi protocolli di e-voting il cui scopo è assicurare la proprietà di receipt-freeness. Entrambi sfruttano il fatto che l'estorsore (o il compratore di voti) non può essere convinto della veridicità delle prove mostrate dall'elettore; pertanto sarà scoraggiato dall'estorcere (o acquistare) un voto.

I protocolli garantiscono anche il requisito di verificabilità del processo elettorale, grazie alla proprietà di omomorfismo rispetto alla somma offerta da certi sistemi crittografici. I due schemi, però, sfruttano questa caratteristica in modo diverso: quello di Hirt e Sako ne fa uso per ottenere la somma dei voti senza “aprire” le singole schede; lo schema di Acquisti, invece, se ne serve per “camuffare” i voti degli elettori e rendere inaccessibile il contenuto (sebbene poi le schede vengano decifrate una per una).

Purtroppo, la proprietà di receipt-freeness da sola non basta a garantire che il voto non verrà venduto o estorto. Consideriamo il caso in cui un attaccante abbia costretto un elettore a consegnargli la propria chiave privata prima della votazione: senza ulteriori controlli, potrebbe simulare l'elettore e votare al suo posto (per esempio via internet). Per contrastare questo attacco (detto *simulation* [JCJ05]), l'unica soluzione consiste nel vincolare gli elettori a presentarsi ai seggi e a farsi riconoscere utilizzando un documento d'identità. Entrambi gli schemi considerati possono essere impiegati in combinazione con le cabine elettorali.

Riferimenti bibliografici

- [Acq04] Alessandro Acquisti. Receipt-free homomorphic elections and write-in ballots. Cryptology ePrint Archive, Report 2004/105, 2004. <http://eprint.iacr.org/>.
- [BFP⁺01] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Guillaume Poupard. Practical multi-candidate election system. In *PODC '01: Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, pages 274–283, New York, NY, USA, 2001. ACM Press.
- [BT94] Josh Benaloh and Dwight Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *STOC '94: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 544–553, New York, NY, USA, 1994. ACM Press.
- [CGHGN01] Dario Catalano, Rosario Gennaro, Nick Howgrave-Graham, and Phong Q. Nguyen. Paillier’s cryptosystem revisited. In *CCS '01: Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 206–214, New York, NY, USA, 2001. ACM Press.
- [CGS97] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *Advances in Cryptology – Proceedings of EUROCRYPT'97*, volume 1233, pages 103–118, 1997.
- [Cha81] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, 1981.
- [HS00] Martin Hirt and Kazue Sako. Efficient receipt-free voting based on homomorphic encryption. In *Advances in Cryptology – EUROCRYPT 2000: International Conference on the Theory and Application of Cryptographic Techniques*, pages 539–557. Springer Berlin/Heidelberg, 2000. <http://www.springerlink.com/content/vrwq2k6gvbd3341x>.
- [JCJ05] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-resistant electronic elections. In *WPES '05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 61–70, New York, NY, USA, 2005. ACM Press.
- [JdV06] H.L. Jonker and E. de Vink. Formalising receipt-freeness. In S.K. Katsikas, J. Lopez, M. Backes, S. Gritzalis, and B. Preneel, editors, *Proceedings ISC 2006*, pages 476–488. LNCS 4176, 2006.

- [JSI96] Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. *Lecture Notes in Computer Science*, 1070:143–155, 1996.
- [Mao03] Wenbo Mao. *Modern Cryptography: Theory and Practice*. Prentice Hall PTR, 2003.
- [MvOV01] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2001.
- [Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. *Lecture Notes in Computer Science*, 1592:223–238, 1999.
- [TR06] Thomas Tjøstheim and Geir Røsland. Remote electronic voting using variable chain encryption. In *Fundamenta Informaticae XX*, pages 1–15. IOS Press, 2006.
- [Wik] Wikipedia. Baby-step giant-step algorithm. http://en.wikipedia.org/wiki/Baby-step_giant-step.