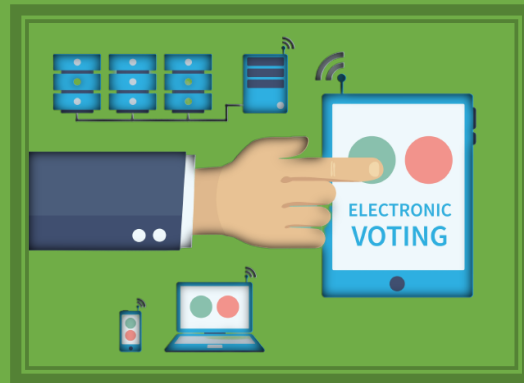


Università degli Studi di Salerno
DIEM, a.a. 2021-22
Corso: ALGORITMI E PROTOCOLLI PER LA SICUREZZA
Professori: Ivan Visconti – Vincenzo Iovino



e-Ballot by NAKAMOTEAM

*Enrico Maria Di Mauro
Allegra Cuzzocrea
Andrea De Gruttola
Salvatore Grimaldi*

0612701706
0612701707
0612701880
0612701742

e.dimauro5@studenti.unisa.it
a.cuzzocrea2@studenti.unisa.it
a.degruttola@studenti.unisa.it
s.grimaldi29@studenti.unisa.it



WP1

Descrizione

L'obiettivo è realizzare un sistema di **e-voting** per lo svolgimento di un **ballottaggio**: un tipo di votazione a confronto diretto tra due candidati, che può rappresentare anche l'ultima eventuale fase di un'elezione. In particolare, secondo la normativa vigente, in Italia il ballottaggio può avere luogo in occasione di elezioni regionali e amministrative.

Nell'ambito di questa trattazione verranno utilizzati i termini "e-voting" e "voto elettronico" per indicare un sistema di voto elettronico/remoto. Tale sistema permette di utilizzare tecnologie elettroniche ed informatiche per lo svolgimento delle funzioni principali dell'attività di voto e di adoperare Internet per la trasmissione del voto dal device dell'elettore all'urna elettronica.

Le funzioni principali dell'attività di voto sono:

- Autorizzazione al voto
- Voto
- Registrazione dei voti
- Conteggio dei voti
- Tabulazione del risultato

Un sistema di voto elettronico è soggetto a molteplici aspetti da considerare ed importanti criticità nella sua realizzazione. È possibile risolvere e/o mitigare queste ultime implementando correttamente tale sistema, garantendo un risultato molto più sicuro ed affidabile di una tipica votazione al seggio.

L'e-voting, infatti, presenta numerosi vantaggi rispetto al classico voto cartaceo:

- Aumento dell'affluenza alle urne per 2 motivi:
 1. Possibilità di voto per gli elettori che non si trovano nei pressi del luogo di residenza
 2. Possibilità di voto per i disabili senza assistenza
- Costi economici nettamente ridotti in quanto si suppone che i cittadini abbiano già a disposizione l'hardware necessario ad usufruire del voto elettronico. Il voto in presenza al seggio, invece, è costoso in quanto richiede una miriade di seggi distribuiti capillarmente sul territorio, ciascuno con personale adibito alle operazioni di voto e forze dell'ordine per questioni di sicurezza

- Costi di tempo ridotti in quanto non viene richiesto ai cittadini di recarsi presso il seggio elettorale e di aspettare il proprio turno, ma bastano pochi click in un qualsiasi momento della giornata. Inoltre, il conteggio finale dei voti avviene in automatico e non richiede l'apertura e la lettura di ogni scheda elettorale
- Riduzione di alcune anomalie, come ad esempio schede mancanti, falsificazione dei conteggi, ecc.
- Maggiore privacy in quanto il voto elettronico permette di votare senza influenze ed osservatori presenti nel singolo seggio
- Riduzione del consumo di carta dovuto al non utilizzo di schede elettorali, registri e tessere elettorali

Le elezioni rappresentano un momento estremamente significativo e delicato della vita politica di una comunità all'interno della quale vi sono sicuramente interessi contrastanti che potrebbero sfociare in rischi di frodi e brogli. Sebbene l'utilizzo dell'e-voting possa semplificare e rendere accessibile a molte più persone l'attività di voto, vi è il problema degli oppositori dell'innovazione che, anziché studiare le nuove tecnologie, non fanno altro che indicarne genericamente i rischi con il solo scopo di lasciare tutto così com'è, riferendosi a chi le studia col termine "tecnocrati". Sono in genere allarmisti che puntano sul fatto che tutti i dispositivi possano essere compromessi, la sicurezza assoluta non esiste, la democrazia è troppo importante e chi propone il voto elettronico ha forse l'obiettivo di abusarne le deficienze. Per evitare facili strumentalizzazioni da parte di tali complottisti è quindi necessario che un sistema di voto remoto/elettronico sia trasparente, nel senso che non debba affidarsi eccessivamente ad una presunta parte fidata, ma abbia invece una progettazione ed analisi che permetta a tutti di verificarne la bontà, limitando il danno che può essere causato da un qualunque avversario.

Attori

Le categorie di persone che prendono parte all'e-voting vengono chiamate "attori" e comprendono sia i protagonisti della fase di voto sia coloro che si occupano dell'organizzazione. I principali attori sono:

- **ELETTORE:** il cittadino che, in accordo alla legge, gode del diritto di voto. Ciascun elettore può esprimere un singolo voto
- **CANDIDATO:** il cittadino che, secondo la legge, gode del diritto a candidarsi
- **OSSERVATORE:** il cittadino che, previa accettazione della richiesta per tale ruolo, si occupa del monitoraggio dell'andamento delle votazioni. Egli ha il diritto di osservare tutte le attività elettorali e di segnalare, eventualmente, una violazione della legge
- **MINISTERO DEGLI INTERNI:** il ministero il cui compito è quello di curare gli adempimenti preparatori ed organizzativi per lo svolgimento di tutte le consultazioni elettorali

- **GIUSTIZIA:** entità che, in casi eccezionali, può essere coinvolta per risolvere controversie legate allo svolgimento del voto ed al suo risultato. Una contestazione è intrinsecamente possibile poiché vi possono essere richieste di revoca o riconteggi del risultato della votazione

Completeness

Il ballottaggio elettronico, a cui per brevità d'ora in poi si farà riferimento con il termine **e-ballot**, si svolge nel modo seguente, assumendo che tutti gli attori si comportino correttamente:

1. Il cittadino richiede al Ministero degli Interni **MI** le credenziali necessarie ad effettuare il voto presentando un certificato digitale ottenuto in precedenza da una Certification Authority CA. Una volta in possesso di tali credenziali il cittadino assume il ruolo di elettore **E_i** con $i=1, \dots, N$
2. All'interno della finestra temporale **[T1-T2]** ciascun elettore **E_i** utilizza le proprie credenziali precedentemente ricevute per accedere alla piattaforma di e-ballot. In questa fase, l'osservatore **O_i** con $i=1, \dots, k$ (con k proporzionale ad N) si occupa del monitoraggio dei log della piattaforma
3. L'elettore **E_i** esprime il proprio voto **V_i**, dove $V_i \in \{\mathbf{A}, \mathbf{B}, \mathbf{W}\}$, con **A**, **B** e **W** rappresentanti rispettivamente il primo candidato, il secondo candidato e la scheda bianca
4. Al termine della finestra temporale **[T1-T2]** l'elettore **E_i** non può più accedere alla piattaforma di e-ballot per esprimere la propria preferenza. Si osservi che nel corso di **[T1-T2]** **E_i** ha la facoltà di cambiare/annullare il proprio voto
5. Gli m voti espressi, con $m \leq n$, vengono conteggiati. Siano **m_A**, **m_B** e **m_W** il totale dei voti associati rispettivamente ad **A**, **B** e **W**. Si ottiene che:
 - Se $m_A > m_B$ il vincitore è **A**
 - Se $m_B > m_A$ il vincitore è **B**
 - In tutti gli altri casi, il voto termina in parità
6. Viene comunicato il vincitore
7. In seguito all'annuncio del risultato elettorale tutti i server coinvolti nel processo di voto cancellano i dati raccolti per poi essere distrutti

La giustizia **J** può essere invocata durante o dopo la finestra temporale **[T1-T2]** e, in caso di illeciti, può interrompere il processo di voto annullando i risultati.

Threat model

La valutazione dei rischi è un passaggio cruciale nel processo di protezione di un'organizzazione e delle sue funzioni. Determinando la probabilità che un certo attore identifichi e sfrutti una particolare vulnerabilità nel metodo di voto è possibile decidere quali controlli sono necessari per mitigare i rischi a un livello accettabile.

Come per il voto classico, anche per quello elettronico è possibile individuare "avversari" che possono compromettere la riuscita della votazione. In particolare, è possibile distinguere 3 tipi di "attacco":

- Attacco di Manipolazione
- Attacco alla Segretezza
- Attacco alla Fiducia pubblica

Attraverso tali compromissioni è possibile che un avversario riesca ad aggiungere o sottrarre voti, individuare le preferenze di determinati elettori o addirittura convincere le persone che il metodo di voto utilizzato sia insicuro, e quindi comprometterne la riuscita.

I principali avversari sono:

- **ELETTORE MANIPOLATO:** si tratta del singolo elettore che viene corrotto o minacciato affinché esprima una determinata preferenza. In seguito alla votazione deve fornire una prova per poter ricevere il denaro promesso (in caso di corruzione) o essere risparmiato (in caso di minaccia). Questo avversario necessita di una potenza computazionale bassa in quanto non deve far altro che accedere alla piattaforma di voto con le proprie credenziali, selezionare ed inviare il voto richiesto e fornire una conferma di ciò che ha votato
- **CANDIDATO MANIPOLATO:** si tratta del singolo candidato che viene corrotto o minacciato affinché si ritiri dalle elezioni per far vincere automaticamente l'altro candidato. In seguito al suo abbandono riceve il denaro promesso (in caso di corruzione) o viene risparmiato (in caso di minaccia). Questo avversario necessita di una potenza computazionale bassa in quanto non deve far altro che comunicare il proprio ritiro

- **OSSERVATORE MANIPOLATO:** si tratta del singolo osservatore che viene corrotto o minacciato affinché non svolga in modo onesto il proprio compito. Esso, infatti, nota la presenza di anomalie all'interno dei log che sta monitorando, ma non le segnala poiché in accordo con chi sta commettendo la frode. In seguito alla sua accondiscendenza per tutta la durata delle elezioni riceve il denaro promesso (in caso di corruzione) o viene risparmiato (in caso di minaccia). Questo avversario necessita di una potenza computazionale bassa in quanto non deve far altro che assecondare i malintenzionati.

I log sono dei file che tengono traccia di tutti i movimenti eseguiti dai server. I log non mostrano i dati ma solo l'andamento del sistema.

- **IMBROGLIONE:** si tratta del singolo individuo che si occupa di corrompere o minacciare elettori, candidati e/o osservatori. Può essere una persona interna o esterna al sistema ed agisce al fine di perseguire un determinato obiettivo. Se l'imbroglione è un candidato, il suo scopo è vincere le elezioni. Se, invece, l'imbroglione è un elettore, un osservatore o una persona esterna, il suo scopo è favorire un determinato candidato o compromettere il ballottaggio al fine di farlo annullare. Questo avversario necessita di una potenza computazionale bassa in quanto non deve far altro che interloquire con i soggetti manipolati
- **HACKER:** si tratta del singolo individuo che si occupa di penetrare nel sistema al fine di compromettere l'integrità dei voti (modificandoli, aggiungendoli e/o eliminandoli). Può essere una persona interna o esterna al sistema ed agisce al fine di perseguire un determinato obiettivo. Se l'hacker è un candidato, il suo scopo è vincere le elezioni. Se, invece, l'hacker è un elettore, un osservatore o una persona esterna, il suo scopo è favorire un determinato candidato o compromettere il ballottaggio al fine di farlo annullare. Questo avversario necessita di una potenza computazionale alta sia se ha già accesso al sistema sia se ha bisogno di superare i meccanismi di sicurezza, in quanto in entrambi i casi ha bisogno di manipolare i voti
- **FICCANASO:** si tratta del singolo individuo che si occupa di intercettare ed eventualmente modificare informazioni al fine di compromettere la confidenzialità dei voti (scoprire il voto di un determinato elettore). Può essere una persona interna o esterna al sistema e può utilizzare le informazioni ottenute a scopo di lucro o per attuare strategie in futuro. Questo avversario necessita di una potenza computazionale bassa se è già in possesso di informazioni o ha già accesso al canale di comunicazione point-to-point tra elettore e urna elettronica e/o al sistema; in caso contrario, necessita di una potenza computazionale alta
- **NO-EVOX:** si tratta del singolo cittadino(sauro) o di movimenti che mirano a diminuire la fiducia pubblica nei confronti del sistema di voto, indicando unicamente i possibili rischi derivanti dal suo utilizzo. Gli attacchi alla fiducia pubblica spesso mescolano solide argomentazioni con la demagogia ed il loro obiettivo potrebbe andare oltre la demonizzazione dell'e-voting,

sfociando ad esempio nella revoca di un risultato elettorale o nell'annullamento di un metodo di voto. Attacchi di questo tipo sono semplici da realizzare ed apparentemente privi di rischi: è sufficiente sollevare problematiche, ipotetiche criticità, possibili contraddizioni con la legge o differenze con le tornate elettorali precedenti e presentarli al pubblico come se fossero veri e propri attacchi alla democrazia. Questo avversario necessita di una potenza computazionale bassa in quanto può perseguire il suo obiettivo agendo solamente dal punto di vista sociale e psicologico

- **OSTRUZIONISTA:** si tratta di un avversario il cui scopo è quello di mandare offline le urne elettroniche o i server che contribuiscono al funzionamento del sistema di voto, impossibilitando l'invio di un certo numero di voti o l'intero svolgimento del ballottaggio. Tale avversario potrebbe attuare strategie differenti a seconda dei mezzi a sua disposizione: avendo accesso alle strutture fisiche del sistema di voto (ad esempio i server) non avrebbe bisogno di una potenza computazionale elevata; in caso contrario, egli dovrebbe effettuare attacchi da remoto (ad esempio del tipo DDoS) per cui è richiesta un'alta potenza computazionale. A prescindere dalla riuscita, totale o parziale, dei suoi obiettivi, l'attaccante potrebbe mettere in luce delle vulnerabilità del sistema e la fiducia pubblica potrebbe risentirne.
- **GRUPPO MALEVOLO:** si tratta di più tipologie di avversari che colludono al fine di incrementare le probabilità di successo dei loro obiettivi. Esempi di gruppi sono:
 - IMBROGLIONE + FICCANASO: l'imbrogliatore minaccia una persona e fornisce la prova al ficcanaso che può rivenderla facilmente grazie alle sue conoscenze
 - HACKER + NO-EVOX: il no-evox influenza più facilmente gli elettori utilizzando come prova la compromissione dell'integrità dei voti eseguita dall'hacker
 - OSTRUZIONISTA + NO-EVOX: essi possono allearsi al fine di dimostrare l'inefficienza e l'insicurezza dell'e-ballot. Un eventuale malfunzionamento del sistema di voto, infatti, rappresenterebbe un'ulteriore prova a sostegno delle tesi no-evox

Proprietà

Per un sistema di e-voting è fondamentale garantire il mantenimento di determinate proprietà anche in presenza di attacchi. Inoltre, è necessario tener conto che qualsiasi tipo di attacco può essere visto come forma di ingiustizia e dovrebbe essere scoraggiato attraverso sanzioni. Vi sono 4 pilastri fondamentali da considerare:

- **CONFIDENZIALITA':** i dati sensibili dovrebbero restare confidenziali anche in presenza di attacchi

- C.1. Nessuno al di fuori del singolo elettore (compresi coloro che gestiscono il sistema) dovrebbe poter conoscere il voto da lui inviato sia durante sia dopo le votazioni
- C.2. Nessuno dovrebbe poter sapere se un determinato elettore abbia espresso o meno il voto sia durante sia dopo le votazioni
- C.3. Il risultato elettorale dovrebbe essere pubblicato solo al termine delle votazioni
- C.4. Non dovrebbe essere possibile ottenere risultati intermedi durante il periodo delle votazioni
- **INTEGRITA':** il sistema dovrebbe realizzare le funzionalità previste anche in presenza di attacchi
 - I.1. Al termine della finestra temporale $[T1-T2]$ non dovrebbe essere possibile aggiungere, modificare o annullare il voto
 - I.2. Non dovrebbe essere possibile aggiungere, modificare o annullare il voto a nome di un'altra persona
 - I.3. Ad ogni elettore dovrebbe essere associato al massimo 1 voto
 - I.4. Il risultato elettorale dovrebbe essere calcolato tramite il corretto conteggio di tutti e soli i voti legittimi
 - I.5. Un voto dovrebbe essere considerato legittimo solo se effettuato in seguito all'apposito riconoscimento delle proprie credenziali
 - I.6. Il conteggio dei risultati non dovrebbe poter essere modificato
- **TRASPARENZA:** il sistema non dovrebbe essere basato su algoritmi segreti
 - T.1. Dovrebbe essere evitata la coercizione, cioè dovrebbe essere possibile votare liberamente malgrado ci sia qualcuno interessato a far votare secondo le sue indicazioni
 - T.2. Chi vota dovrebbe poter controllare che il proprio voto sia stato contato correttamente
- **EFFICIENZA:** il sistema dovrebbe essere utilizzabile senza eccessivi costi e ritardi

WP2

Di seguito si espone una soluzione correlata a quanto descritto precedentemente. Tale soluzione è caratterizzata da una combinazione di elementi centralizzati e decentralizzati

Assunzioni

Si riportano di seguito alcune assunzioni che riguardano il sistema.

Si assume che:

- tutti i server presenti nel sistema siano onesti
- i voti vengano distribuiti in maniera equa nei server senza causare il sovraccarico di uno di essi
- l'hardware ed il software che caratterizzano i dispositivi utilizzati dagli attori del sistema non siano corrotti

Inoltre, si considera la versione più recente di TLS, ovvero TLS 1.3

Ottenimento del certificato digitale

Ogni cittadino, per poter partecipare all'e-ballot, necessita di richiedere ad un Server Certification Authority **S_{CA}**, noto ed affidabile, un certificato digitale **cd**. Tale certificato attesta l'identità del cittadino e garantisce che quest'ultimo sia in possesso di una determinata coppia di chiavi (**PK_c**, **SK_c**).

cd è costituito da:

1. Versione
2. Numero Seriale del certificato
3. Algoritmo di Firma
4. Emittente
5. Periodo di Validità del certificato
6. Intestatario del certificato: Nome, Cognome, Codice Fiscale e Nazionalità
7. PK dell'intestatario del certificato
8. Firma dell'emittente
9. Estensioni

Un certificato digitale può essere revocato in 3 casi:

- problemi di identificazione a causa dello smarrimento o del furto di SK_c
- termine del periodo di validità
- volontà del cittadino in possesso del certificato di cessarne la validità

L'unico modo per ottenere la revoca di un certificato è fare richiesta a S_{CA} che si occuperà di inserire il cd all'interno di una revocation list. I cittadini il cui certificato è presente all'interno di tale lista non avranno più il diritto di usufruire dei servizi ad esso collegati.

Inoltre, si assume che tutti i server presenti nella soluzione abbiano un proprio certificato digitale che permette la loro autenticazione.

Schema di cifratura a chiave pubblica (El Gamal)

Lo schema di cifratura a chiave pubblica di El Gamal è costituito dalla terna di algoritmi (Gen, Enc, Dec):

- **Gen(1^n)** → prende in input una stringa di n bit 1 (n è chiamato "parametro di sicurezza") e restituisce $p, q, g, g^x=y$ in modo da ottenere la coppia $PK=(p, q, g, y)$ e $SK=(x)$. In particolare, $p=2q+1$ e q sono numeri primi, g genera un sottogruppo ciclico G_q del gruppo Z_{p^*} e x , che rappresenta il logaritmo discreto (DLog), appartiene al gruppo Z_q
- **Enc(PK, m)** → prende in input la chiave pubblica PK ed il messaggio da cifrare m e restituisce il cipher text c . In particolare, prende r in Z_q e calcola $u=g^r$ e $v=m*y^r$. Il cipher text in output è la coppia (u, v)
- **Dec(SK, c)** → prende in input la chiave segreta SK ed il cipher text c e restituisce il messaggio m . In particolare, si effettua tale calcolo:
$$v*u^{-x}=m*y^r*(g^r)^{-x}=m*y^r*g^{-xr}=m*y^r*y^{-r}=m*y^0=m$$

Threshold El Gamal Decryption

Una singola entità in possesso della chiave segreta SK è in grado di decifrare tutto ciò che viene cifrato attraverso PK mediante lo schema di cifratura a chiave pubblica di El Gamal. Per evitare ciò e rendere più complicato il furto o la diffusione della chiave si utilizza la Threshold El Gamal Decryption. In questo modo la decifratura si svolge in maniera decentralizzata.

La chiave SK di El Gamal viene divisa in n "porzioni", dette "shares", e distribuita ad n players differenti che, messi insieme, sono in grado di eseguire la decifratura.

Il funzionamento prevede che un dealer, che possiede la SK , utilizzi lo strumento di crittografia Shamir Secret Sharing (t, n) in cui il segreto è x , ossia SK . Attraverso questo strumento, il dealer definisce un polinomio $p(x)$ ed il player i -esimo ottiene $(i, p(i))$. In fase di decifratura è necessario che almeno t degli n players cooperino tra loro calcolando $u^{p(i)}$ e condividendo il risultato ottenuto. Uno dei t players raccoglie i contributi di tutti gli altri ed attraverso un calcolo matematico ottiene u^x . In questo modo il player è in grado di decifrare il ciphertext di El Gamal senza mai entrare in possesso dell'effettiva SK .

Generazione e distribuzione di PK_v e SK_v

Antecedentemente alla finestra temporale $[T1-T2]$ un server S_{GEN} genera una coppia di chiavi El Gamal (PK_v, SK_v) necessarie a cifrare e decifrare i voti di tutti gli elettori. Si assume che tale server sia una parte fidata del Ministero degli Interni MI e che il suo software sia open source e non corrotto.

Per utilizzare la Threshold El Gamal Decryption, S_{GEN} calcola le shares di SK_v e le distribuisce agli n S_{BAL} , ossia i server di ballottaggio, che al termine della finestra temporale $[T1-T2]$, si occuperanno di effettuare il conteggio locale dei ciphertext.

La distribuzione delle shares di SKv da S_{GEN} agli S_{BAL} viene effettuata connettendosi via TLS con autenticazione bidirezionale. Con questo tipo di autenticazione si garantisce che:

- S_{GEN} invii le shares agli S_{BAL} corretti
- ogni S_{BAL} riceva una share dal S_{GEN} corretto

Inoltre, S_{GEN} firma le share cifrate prima di consegnarle agli S_{BAL} per permettere successive verifiche in fase di decifratura.

S_{GEN} , una volta distribuite le shares di SKv, deve essere disconnesso da internet ed il suo disco deve essere totalmente ripulito. In questo modo si evita il riuso del suo contenuto e che qualcuno possa entrare in possesso della chiave segreta. Per questo motivo, prima della disconnessione del server, è necessario inviare PKv al server S_{PLAT} che, durante la finestra temporale $[T1-T2]$, si occuperà di far votare gli elettori.

L'invio di PKv da S_{GEN} a S_{PLAT} viene effettuato connettendosi via TLS con autenticazione bidirezionale. Con questo tipo di autenticazione si garantisce che:

- S_{GEN} invii la PKv al S_{PLAT} corretto
- S_{PLAT} riceva la PKv dal S_{GEN} corretto

Autorizzazione al voto

A seconda del Paese in cui ci si trova esistono delle caratteristiche che un cittadino deve rispettare per poter esprimere il proprio voto.

All'interno della finestra temporale $[T1-T2]$, il singolo cittadino richiede la registrazione sulla piattaforma di voto (caratterizzata dal server S_{PLAT}) attraverso il proprio certificato digitale connettendosi via TLS con autenticazione bidirezionale. Con questo tipo di autenticazione si garantisce che:

- Il cittadino comunichi con l' S_{PLAT} corretto
- S_{PLAT} possa verificare l'identità del cittadino ed effettuare i dovuti controlli per abilitarlo al voto

In particolare, si verifica la presenza del codice fiscale all'interno di un database contenente i codici fiscali di tutti i cittadini aventi diritto di voto. Tale database è fornito da MI ed è presente in S_{PLAT} . Se tale verifica avviene con successo il cittadino:

- assume il ruolo di elettore E_i con $i=1,...,N$
- riceve un codice randomico, univoco ed indipendente dall'identità di E_i . Esso non è sostituibile in caso di furto o smarrimento
- deve impostare una password che verrà associata al codice

Tutti i codici assegnati agli elettori e le rispettive password vengono memorizzati all'interno di un ulteriore database. Ogni password impostata dagli elettori non viene memorizzata in chiaro, ma viene cifrata in accordo al meccanismo di Password Hashing che prevede l'applicazione di SHA256 alla concatenazione di un salt, ossia un insieme randomico di bit, e la password. Grazie a questo meccanismo

vengono scongiurati attacchi di tipo Offline Dictionary. Tale database è costituito da 5 campi:

- ID univoco
- $\text{SHA256}(\text{salt} || \text{password})$
- Salt
- Cyphertext
- Firma del cyphertext

L'elettore, una volta impostata la password, ha la possibilità di accedere alla piattaforma per votare, modificare o annullare il proprio voto. L'accesso richiede di connettersi via TLS (con autenticazione bidirezionale), attraverso il proprio certificato digitale, e di inserire il codice ricevuto al primo accesso e la password ad esso associata.

All'interno del database contenente tutti i codici fiscali degli aventi diritto di voto è presente un campo di check. S_{PLAT} inserisce un bit 1 all'interno di tale campo in corrispondenza del codice fiscale del singolo cittadino che è stato registrato.

Quando un cittadino tenta di accedere alla piattaforma attraverso il proprio cd, S_{PLAT} legge il valore del campo check associato al suo codice fiscale.

- Se il campo check è vuoto, il cittadino viene verificato ed eventualmente registrato
- Se il campo check contiene il bit 1, l'accesso viene negato e viene richiesto di immettere il codice e la password posseduti

È possibile che durante il periodo di votazione un cittadino revochi il proprio certificato digitale. Sono previsti 2 casi:

- La revoca avviene prima della richiesta di registrazione sulla piattaforma: il cittadino non ha la possibilità di registrarsi e, quindi, di esprimere il proprio voto fino all'ottenimento di un nuovo certificato
- La revoca avviene dopo la registrazione sulla piattaforma: il cittadino perde la possibilità di esprimere, modificare o annullare il proprio voto fino all'ottenimento di un nuovo certificato

Variazione dello schema di cifratura a chiave pubblica (El Gamal)

La variazione dello schema di cifratura a chiave pubblica di El Gamal prevede delle differenze durante la fase di cifratura e decifratura del messaggio. Invece di calcolare una cifratura di m come $(g^r, m * y^r)$ la si può calcolare come $(g^r, g^{m * y^r})$. Ottenuto g^m , la decifratura prevede un ulteriore passaggio, quello di trovare i possibili valori che hanno senso come plaintext finché non viene identificato quello che corrisponde a g^m . Questa variazione conferisce ad El Gamal la proprietà di omomorfismo additivo. L'operazione omomorfa che combina due ciphertext ne produce un terzo il cui messaggio è la somma dei plaintext di partenza. In particolare, dati due ciphertext $c1 = (g^{r1}, g^{m1 * y^{r1}})$ e $c2 = (g^{r2}, g^{m2 * y^{r2}})$, il loro prodotto genera un ciphertext $c3 = (g^{r1 + r2}, g^{m1 + m2 * y^{r1 + r2}})$.

Voto

Una volta effettuato il login alla piattaforma, l'elettore ha la possibilità di esprimere la propria preferenza selezionando ed inviando una delle voci disponibili $\{A, B, W\}$ con A, B e W rappresentanti rispettivamente il primo candidato, il secondo candidato e la scheda bianca.

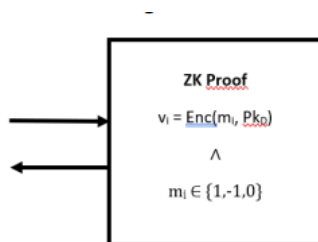
Ogni voto inviato costituisce un plaintext m con:

- $m="1" \rightarrow A$
- $m="-1" \rightarrow B$
- $m="0" \rightarrow W$

Per la cifratura del voto viene adoperata la variazione dello schema di cifratura di El Gamal spiegata nel paragrafo precedente. Questa cifratura permette di sfruttare la proprietà di omomorfismo additivo, che servirà successivamente per il conteggio dei voti. Il singolo elettore cifra il voto da inviare attraverso la chiave pubblica PK_v (mantenuta da S_{PLAT}) e firma il ciphertext prodotto utilizzando la propria chiave segreta SK_c .

Per dimostrare a S_{PLAT} la correttezza del voto inviato senza mostrarlo, E_i deve utilizzare una Zero Knowledge Proof.

Il claim è il seguente:



Ovviamente, l'elettore E_i è l'unico a conoscenza di m_i .

Inoltre, l'elettore invia a S_{PLAT} :

- ciphertext
- firma del ciphertext
- chiave pubblica PK_c

S_{PLAT} , quindi, esegue la Verify. Nel caso in cui quest'ultima restituisca 1, la firma del voto cifrato è valida e S_{PLAT} , dopo essersi convinto della legittimità del voto attraverso la Zk Proof ricevuta, inserisce all'interno del database il ciphertext e la firma associata in corrispondenza delle credenziali di E_i che sta esprimendo il voto; altrimenti la firma del voto cifrato non è valida e l'elettore viene invitato a riprovare. È da tenere presente che S_{PLAT} non salva all'interno del proprio database la PK_c poiché un eventuale avversario che riesce ad appropriarsi di tale database potrebbe risalire all'elettore associato. Non otterrebbe i voti, ma gli E_i che li hanno espressi.

Una volta inseriti i dati nel database, S_{PLAT} invia il ciphertext, la firma e PKc all'apposito S_{BAL} connettendosi via TLS con autenticazione bidirezionale. Con questo tipo di autenticazione si garantisce che:

- S_{PLAT} invii le informazioni al S_{BAL} corretto
- S_{BAL} riceva le informazioni dal S_{PLAT} corretto

Nel caso in cui un E_i modifichi o annulli il proprio voto, il procedimento sopra citato subisce un'alterazione: S_{PLAT} , una volta individuati voto e firma precedenti corrispondenti alle credenziali di E_i , prima di sovrascriverli deve salvarli per poi inviarli all'apposito S_{BAL} insieme a ciphertext e firma nuovi e PKc.

Registrazione dei voti

La registrazione dei voti consiste nel determinare se ogni voto giunto ai vari S_{BAL} sia effettivamente valido per il conteggio. Tale controllo avviene ogni volta che un S_{BAL} riceve la tripla (ciphertext, firma, PKc) da S_{PLAT} . Si noti che PKc non viene memorizzata all'interno del database di ogni S_{BAL} .

Il controllo prevede l'utilizzo dei dati ottenuti da S_{PLAT} per eseguire la Verify e verificare che la firma associata al ciphertext sia valida. I ciphertext corrispondenti a risultati negativi della Verify vengono scartati; mentre quelli corrispondenti a risultati positivi vengono memorizzati con le rispettive firme.

In caso di modifica/annullamento di un voto da parte di un E_i S_{BAL} individua voto e firma precedenti ricevuti da S_{PLAT} nel database e li sovrascrive con le nuove informazioni ricevute.

Conteggio dei voti

Una volta terminata la finestra temporale $[T1-T2]$, non vengono più accettate connessioni tra E_i , S_{PLAT} e S_{BAL} e si procede con il conteggio di tutti i voti presenti all'interno dei S_{BAL} . Da questo momento in poi assumiamo che tutte le connessioni tra i S_{BAL} avvengano via TLS con autenticazione bidirezionale.

Ogni S_{BAL} moltiplica localmente i ciphertext presenti nel proprio database ed ottiene un nuovo ciphertext di El Gamal. In questo modo viene sfruttato l'omomorfismo additivo che caratterizza la variazione della cifratura di El Gamal.

Una volta terminato il proprio conteggio, tutti gli n S_{BAL} inviano il risultato agli altri $n-1$ S_{BAL} . Ogni S_{BAL} moltiplica tutti i ciphertext ottenuti dagli altri S_{BAL} producendo a sua volta un nuovo ciphertext di El Gamal $c_{fin}=(u_{fin}, v_{fin})$ che rappresenta il ciphertext finale da decifrare.

Secondo quanto previsto dalla Threshold El Gamal Decryption, ognuno dei t S_{BAL} con $t \leq n$ calcola $u_{fin}^{p(i)}$, dove $p(i)$ costituisce la share del segreto che ha ricevuto da S_{GEN} . Una volta effettuato il calcolo, ognuno invia il proprio $u_{fin}^{p(i)}$ agli altri $t-1$ S_{BAL} fornendo una ZK Proof per garantire la correttezza della computazione.

Uno dei t S_{BAL} , una volta sicuro che tutti i contributi ottenuti siano corretti, procede con la decifratura ottenendo $g^{m1 + m2 + m3 + \dots + mw}$ con w pari al numero di elettori

che hanno votato ($w \leq N$). Dopo una serie di tentativi, infine, ottiene il plaintext m_{fin} dato che ogni m appartiene ad un insieme finito di valori.

Tabulazione dei risultati

Il risultato ottenuto, ossia m_{fin} , può essere di 3 tipi:

- maggiore di 0 \rightarrow il vincitore è A
- minore di 0 \rightarrow il vincitore è B
- uguale a 0 \rightarrow parità

La correttezza del risultato finale dell'e-ballot è garantita dalla possibilità di effettuare nuovamente la decifratura e fare svolgere i medesimi calcoli ad un altro S_{BAL} . Questo può essere ripetuto per tutti gli S_{BAL} . In questo modo, questi ultimi possono fungere da testimoni per eventuali disonestà da parte del S_{BAL} che ha prodotto il risultato finale.

Una volta annunciato il risultato dell'e-ballot, tutti i S_{BAL} inviano ad un server S_{TAB} le coppie (ciphertext, firma) da loro possedute connettendosi via TLS con autenticazione bidirezionale.

S_{TAB} , quindi, contiene un database pubblico e non modificabile, ossia una bacheca che permette a tutti gli elettori di verificare che il loro voto sia stato considerato nel conteggio e che sia quello da loro inserito.

Al termine dell'e-ballot, ogni server e dispositivo cancella tutti i dati posseduti. La bacheca, invece, rimane consultabile.

WP3

Di seguito si analizzano le proprietà definite nel WP1 per determinare se siano soddisfatte dal sistema considerando i possibili attaccanti:

- **CONFIDENZIALITA'**: i dati sensibili dovrebbero restare confidenziali anche in presenza di attacchi

C.1. Nessuno al di fuori del singolo elettore (compresi coloro che gestiscono il sistema) dovrebbe poter conoscere il voto da lui inviato sia durante sia dopo le votazioni

I possibili attaccanti sono:

- **IMBROGLIONE**: potrebbe corrompere o minacciare un elettore inducendolo ad esprimere una determinata preferenza. Potrebbe essere fisicamente presente durante la scelta e l'invio del voto dell'elettore oppure potrebbe richiedere una prova del rispetto dell'accordo tramite, ad esempio, un video o una videochiamata. Questo avversario potrebbe anche appropriarsi delle credenziali dell'elettore coinvolto e votare al suo posto.

Inoltre, potrebbe forzare via software un elettore. Questo compromette la segretezza del voto.

Come già spiegato, è prevista la possibilità di modificare o annullare un voto precedentemente inviato. In questo modo, questo avversario non può essere completamente sicuro che l'elettore coinvolto mantenga il voto promesso fino alla fine dell'e-ballot. Per questo motivo gli attacchi dovuti a questo tipo di avversario possono essere compromessi.

Inoltre, per mitigare gli attacchi a distanza si assume che l'hardware ed il software utilizzati dagli attori onesti non siano e non possano essere corrotti e quindi controllati da eventuali malware. In caso contrario, sarebbe opportuno aggiungere ulteriori misure di sicurezza al sistema come, ad esempio, l'utilizzo di software che bloccano il dispositivo durante la votazione.

Infine, si assume che questo avversario coinvolga una piccola quantità di elettori (meno della metà) in quanto in caso contrario, sarebbe difficile per lui non essere scoperto o denunciato. Se gli elettori minacciati fossero tanti, infatti, si rivolgerebbero alla Giustizia.

Per i motivi appena citati, questo avversario non compromette in modo rilevante questa proprietà.

Il vantaggio della possibilità di modificare o annullare il proprio voto per ripristinare la segretezza elettorale potrebbe venire meno nel caso in cui colui che corrompe o minaccia sia un ficcanaso o un hacker. Infatti:

- il ficcanaso potrebbe intercettare il messaggio contenente la modifica o l'annullamento del voto da parte dell'elettore coinvolto. Per i motivi sopra citati, questo non è possibile grazie all'utilizzo di TLS e grazie al fatto che i singoli ciphertext non possono essere decifrati
- l'hacker, se a conoscenza delle credenziali dell'elettore, potrebbe penetrare nei database e verificare la presenza del voto concordato.

- **FICCANASO:** potrebbe cercare di intercettare informazioni legate ai voti.

Il sistema non rende possibile la lettura di messaggi che viaggiano sui canali di comunicazione in quanto vengono utilizzate connessioni TLS con autenticazione bidirezionale. In questo modo, i messaggi scambiati possono essere letti solo dagli attori interni alla connessione e non da quelli esterni. Grazie a TLS, i canali di comunicazione sono sicuri.

Inoltre, l'autenticazione bidirezionale permette ad ogni end-point di verificare l'identità dell'altro e, quindi, di impedire attacchi caratterizzati dal tentativo di un avversario di fingersi un determinato server e comunicare con un altro per ottenere le informazioni a cui è interessato.

In aggiunta, in presenza di eventuali modifiche di messaggi importanti inviati non vengono provocati danni al sistema di e-voting in quanto il ricevente effettua una verifica sulla firma inviata. Se quest'ultima non corrisponde alla firma del messaggio ricevuto, quest'ultimo non viene accettato in quanto considerato compromesso.

Si noti che, grazie all'utilizzo di **Threshold ElGamal Decryption**, un avversario che riesce ad intercettare dati significativi non ha modo di decifrare i ciphertext in quanto la decifratura decentralizzata richiede almeno t SBAL in possesso delle share per decifrare i voti. Inoltre, attraverso questo tipo di decifratura è possibile decifrare solo il ciphertext finale e non quello dei singoli elettori.

Quindi, questo avversario non compromette la proprietà in modo significativo.

- **HACKER:** potrebbe tentare di accedere ai database dei server del sistema in modo da ottenere informazioni sull'identità degli elettori associati ai ciphertext. Questo attacco viene mitigato del tutto grazie al modo in cui i database sono stati progettati. Infatti, in essi non è presente alcun tipo di correlazione tra l'identità di un elettore ed il voto a lui associato. Infatti, nei database presenti nei server SPLAT e SBAL le credenziali di un elettore ed il voto da lui espresso non sono associati al codice fiscale. Inoltre, la PKe inviata dall'elettore insieme al voto ed alla firma non viene salvata.

Per questo motivo, questo avversario non rappresenta un pericolo rilevante per questa proprietà.

- **ELETTORE MANIPOLATO:** potrebbe esprimere una determinata preferenza in quanto egli viene corrotto o minacciato. In questo modo, viene meno la segretezza del voto.

Il sistema prevede, oltre all'aggiunta del proprio voto, la possibilità di modificarlo o annullarlo. In questo modo, l'elettore manipolato, può esprimere la preferenza che gli viene imposta e, dopo aver fornito la prova dell'invio della stessa, ha la possibilità di:

- annullarla, se non è intenzionato a votare
- modificarla, se è intenzionato a votare ma la sua preferenza è diversa da quella inviata

in modo da ripristinare la segretezza.

Inoltre, si ipotizza che almeno più della metà degli aventi diritto di voto non sia disposta ad essere corrotta in quanto gli elettori possiedono un senso del dovere nei confronti del sistema politico che li porta a non rinunciare al proprio diritto di libertà di voto.

In aggiunta, si ritiene che sia difficile minacciare un gran numero di elettori senza essere scoperti e, quindi, verrebbe coinvolta una piccola parte anche in caso di minaccia.

Per questi motivi, gli attacchi dovuti a questo avversario non rappresentano una particolare minaccia per il soddisfacimento di questa proprietà.

In conclusione, **questa proprietà viene quasi sempre garantita** grazie ai mezzi utilizzati ed alla progettazione dei database che evitano alcun tipo di correlazione tra un voto e l'identità dell'elettore ad esso associato.

C.2. Nessuno dovrebbe poter sapere se un determinato elettore abbia espresso o meno il voto sia durante sia dopo le votazioni

I possibili attaccanti sono:

- **FICCANASO + HACKER:** potrebbe intercettare l'identità di un elettore e le sue credenziali su un canale di comunicazione e li fornisce le credenziali all'hacker che penetra nel database presente nel server SPLAT e verifica la presenza o meno del voto ad esso associato. Come già esplicitato in precedenza, gli attacchi dovuti al ficcanaso sono mitigati attraverso l'utilizzo di TLS e per questo motivo questo gruppo malevolo non compromette questa proprietà.
- **IMBROGLIONE:** potrebbe costringere un elettore a votare anche se quest'ultimo non è intenzionato a farlo. Questo compromette questa proprietà ma, per i motivi citati all'interno dell'analisi del medesimo

avversario nell'ambito della proprietà precedente, esso non rappresenta un pericolo rilevante per il sistema. Inoltre, grazie alla possibilità di annullare il proprio voto, questo avversario non può essere sicuro che l'elettore da lui coinvolto non annulli il voto inviato. Quindi, questo avversario non compromette significativamente la proprietà.

- **IMBROGLIONE + HACKER:** potrebbe appropriarsi delle credenziali di un elettore corrompendolo o minacciandolo e fornirle all'hacker che, penetrando nel database, potrebbe scoprire se esista un voto associato a quelle credenziali. Questo attacco non può essere del tutto mitigato ma per i motivi già citati non rappresenta un pericolo rilevante per questa proprietà.

La proprietà, quindi, è quasi sempre soddisfatta.

- C.3. Il risultato elettorale dovrebbe essere pubblicato solo al termine delle votazioni
- C.4. Non dovrebbe essere possibile ottenere risultati intermedi durante il periodo delle votazioni

I possibili avversari per queste due proprietà sono:

- **HACKER:** potrebbe riuscire ad accedere ai database dei server contenenti i voti e trarre informazioni sul risultato elettorale. Come già spiegato nell'ambito dell'analisi delle proprietà precedenti, esso non rappresenta un avversario pericoloso grazie all'utilizzo di un meccanismo di decifratura decentralizzato (Threshold ElGamal Decryption). Non è, infatti, possibile ottenere risultati parziali o il risultato elettorale né da parte di avversari esterni né da parte delle urne (server SBAL) prima del termine della finestra temporale $[T1-T2]$. Questo attacco, quindi, non rappresenta un ostacolo significativo per il soddisfacimento delle proprietà.
- **FICCANASO:** potrebbe intercettare i ciphertext inviati sui canali di comunicazione per ottenere risultati parziali o il risultato elettorale. Come già spiegato in precedenza, ciò non è possibile grazie all'utilizzo di TLS e di Threshold ElGamal Decryption.

Queste proprietà, quindi, sono sempre garantite.

- **INTEGRITA':** il sistema dovrebbe realizzare le funzionalità previste anche in presenza di attacchi

- I.1. Al termine della finestra temporale $[T1-T2]$ non dovrebbe essere possibile aggiungere, modificare o annullare il voto

Questa proprietà è soddisfatta in quanto il sistema è progettato in modo che al termine della finestra temporale $[T1-T2]$ vengano interrotte tutte le connessioni tra elettori, piattaforma ed urne.

I possibili avversari sono:

- **HACKER:** potrebbe penetrare nel sistema ed in particolare nei server SPLAT e SBAL che contengono i ciphertext per aggiungere, modificare o annullare dei voti al termine della finestra temporale $[T1-T2]$. Egli potrebbe favorire uno dei due candidati oppure compromettere il ballottaggio al fine di farlo annullare. Si assume che i server siano caratterizzati da sistemi di sicurezza di alta qualità che impediscono ogni tipo di accesso da parte di esterni. Inoltre, ci si affida all'onestà degli osservatori che smascherano comportamenti anomali in quanto si assume che la maggior parte degli osservatori non sia manipolata. Senza queste assunzioni, l'attacco da parte di un hacker potrebbe rivelarsi pericoloso.
- **IMBROGLIONE + HACKER:** l'imbroglione, in collusione con l'hacker, potrebbe corrompere o minacciare l'osservatore per non comunicare anomalie causate dall'hacker. Le anomalie possono riguardare attacchi al termine della finestra temporale $[T1-T2]$. Si tratta, quindi, di un gruppo malevolo. Per mitigare questo attacco si fa riferimento alle assunzioni già citate in merito agli attacchi hacker (sicurezza dei server, fiducia riposta negli osservatori, maggioranza degli osservatori non manipolata).

Questa proprietà, infine, è quasi sempre soddisfatta.

I.2. Non dovrebbe essere possibile aggiungere, modificare o annullare il voto a nome di un'altra persona

I possibili avversari sono:

- **IMBROGLIONE:** potrebbe impossessarsi delle credenziali di un elettore minacciandolo o corrompendolo e votare a suo nome. Per mitigare questo attacco valgono le considerazioni riportate nell'ambito di proprietà precedenti
- **FICCANASO:** potrebbe intercettare sul canale le credenziali di un determinato elettore ed usarle per votare al suo posto. Potrebbe, inoltre, intercettare un voto e modificarlo prima che arrivi ad un server. Come già esplicitato in precedenza, gli attacchi dovuti a questo avversario sono mitigati dall'utilizzo di TLS. Infine, anche nel caso in cui questo avversario riesca ad intercettare un messaggio e modificarlo, il sistema prevede il controllo sulla firma e quindi il server ricevente si accorgerebbe della manipolazione

In conclusione, **questa proprietà è quasi sempre soddisfatta.** Se non fosse possibile in alcun modo entrare in possesso delle credenziali di un elettore sarebbe completamente soddisfatta. Le connessioni TLS

con autenticazione bidirezionale garantiscono che ogni elettore sia l'unico possessore delle proprie credenziali. È necessario, però, che egli non le divulghi.

I.3. Ad ogni elettore dovrebbe essere associato al massimo 1 voto

I possibili avversari sono:

- **HACKER:** potrebbe accedere al sistema ed aggiungere più voti ad uno stesso elettore. Questo non è possibile in quanto i database dei server sono stati progettati al fine di non ammettere più voti per uno stesso elettore. Inoltre, la possibilità di modificare un voto implica che nel caso in cui un elettore, dopo aver votato, invii un ulteriore voto, il voto precedente venga sovrascritto da quello nuovo.

Per i motivi appena citati, **questa proprietà viene sempre garantita.**

I.4. Il risultato elettorale dovrebbe essere calcolato tramite il corretto conteggio di tutti e soli i voti legittimi

Possibili avversari:

- **HACKER:** potrebbe compromettere il conteggio di tutti e soli i voti legittimi in due modi:
 - durante la finestra temporale $[T1-T2]$ potrebbe penetrare nei server SPLAT e SBAL rendendo i voti non legittimi
 - al termine della finestra temporale $[T1-T2]$ potrebbe penetrare nei server SBAL durante il processo di decifratura e quindi di conteggio

Questi attacchi vengono mitigati considerando le assunzioni riportate in merito alle proprietà precedenti.

- **IMBROGLIONE + HACKER:** questo gruppo malevolo viene mitigato per gli stessi motivi già riportati precedentemente (proprietà I.1)
- **FICCANASO:** potrebbe intercettare e modificare i voti sul canale e renderli non legittimi. Come già spiegato in merito alla proprietà I.2, questo attacco viene mitigato.

Inoltre, ogni voto non autentico viene individuato grazie al meccanismo delle firme. Infatti:

- il server SPLAT che riceve i voti dagli elettori effettua una verifica dell'autenticità del voto
- i server SBAL che ricevono i voti da SPLAT effettuano una verifica dell'autenticità del voto

Infine, la garanzia della legittimità dei voti è data attraverso l'utilizzo delle ZK Proof:

- in fase di Cifratura ogni elettore fornisce una ZK Proof per dimostrare di avere inviato un voto legittimo (1, 0, -1)

- in fase di Decifratura, tutti gli SBAL forniscono una ZK Proof per garantire la correttezza della computazione. In questo modo, il risultato elettorale risulta corretto

In conclusione, **questa proprietà viene quasi sempre garantita.**

I.5. Un voto dovrebbe essere considerato legittimo solo se effettuato in seguito all'apposito riconoscimento delle proprie credenziali

Questa proprietà è soddisfatta in parte in quanto è possibile accedere alla piattaforma attraverso un certificato digitale e delle credenziali univoche. Chiunque sia a conoscenza delle credenziali di un elettore e possiede un certificato digitale valido può accedere alla piattaforma e votare al suo posto. Questo accade in quanto non sono presenti correlazioni tra identità e credenziali e quindi non possono essere effettuati controlli in merito.

I possibili avversari sono:

- **ELETTORE MANIPOLATO:** potrebbe divulgare le proprie credenziali perché corrotto o minacciato. Gli attacchi dovuti a questo avversario sono mitigati dai motivi già citati nell'ambito delle proprietà precedenti
- **FICCANASO:** potrebbe intercettare delle credenziali inviate sul canale di comunicazione. Gli attacchi dovuti a questo avversario sono mitigati dai motivi già citati nell'ambito delle proprietà precedenti
- **IMBROGLIONE:** potrebbe appropriarsi delle credenziali di un elettore corrompendolo o minacciandolo e votare al suo posto. Gli attacchi dovuti a questo avversario sono mitigati dai motivi già citati nell'ambito delle proprietà precedenti

Questa proprietà, quindi, è quasi sempre garantita.

I.6. Il conteggio dei risultati non dovrebbe poter essere modificato

- **HACKER:** potrebbe modificare il conteggio finale prima che il server SBAL comunichi il risultato elettorale. Questo attacco è mitigato considerando le assunzioni riportate in merito alle proprietà precedenti.

Questa proprietà, quindi, è quasi sempre garantita.

• TRASPARENZA

T.1. Dovrebbe essere evitata la coercizione, cioè dovrebbe essere possibile votare liberamente malgrado ci sia qualcuno interessato a far votare secondo le sue indicazioni

Questa proprietà è mitigata in parte dalla possibilità di modificare il proprio voto infinite volte durante la finestra temporale [T1-T2].

I possibili avversari sono:

- **IMBROGLIONE:** potrebbe corrompere o minacciare un elettore privandolo della libertà di voto e compromettendo, quindi, questa proprietà. Gli attacchi dovuti a questo avversario possono essere parzialmente mitigati attraverso le considerazioni fatte nell'ambito delle proprietà precedenti
- **ELETTORE MANIPOLATO:** potrebbe farsi corrompere preferendo una ricompensa allettante in cambio della perdita della libertà di voto. In questo modo compromettere la proprietà. I motivi che mitigano parzialmente gli attacchi dovuti a questo avversario sono spiegati nell'ambito delle proprietà precedenti

T.2. Chi vota dovrebbe poter controllare che il proprio voto sia stato contato correttamente

Questa proprietà è in parte garantita in quanto al termine dell'e-ballot viene pubblicata una bacheca contenente le firme ed i voti cifrati di tutti gli elettori che sono stati considerati nel conteggio. La possibilità di verificare la presenza del proprio voto, inoltre, aumenta la fiducia pubblica compromettendo in parte gli attacchi dovuti ai NO-EVOX.

I possibili avversari sono:

- **OSTRUZIONISTA:** potrebbe mandare offline il server STAB che pubblica la bacheca, impossibilitando gli elettori a verificare la presenza del proprio voto cifrato e della propria firma. Al termine dell'e-ballot, inoltre, gli altri server eliminano tutti i dati in loro possesso, quindi, se questo avversario agisse al termine del ballottaggio, non sarebbe possibile recuperare i voti cifrati e questa proprietà non sarebbe soddisfatta. Questo avversario potrebbe essere in collusione con i NO-EVOX.

La trasparenza, infine, è quasi sempre soddisfatta.

- **EFFICIENZA:** il sistema dovrebbe essere utilizzabile senza eccessivi costi e ritardi

I possibili avversari sono:

- **OSTRUZIONISTA:** potrebbe mandare offline le urne elettroniche o i server che contribuiscono al funzionamento del sistema di voto, impossibilitando l'invio di un certo numero di voti o l'intero svolgimento del ballottaggio. Inoltre, potrebbe mettere in luce la vulnerabilità del sistema e la fiducia pubblica potrebbe risentirne. In particolare, questo avversario potrebbe rendere fuori uso il server

SPLAT e i server SBAL bloccando completamente le elezioni in quanto gli elettori non avrebbero la possibilità di votare.

Un modo per mitigare questo attacco è dato dall'utilizzo della Threshold ElGamal Decryption che prevede il funzionamento di una parte dei SBAL e non necessariamente di tutti. Inoltre, i voti vengono divisi in maniera equa tra i server SBAL in modo da non affidare un elevato numero di dati ad un unico server.

Un miglioramento che potrebbe essere effettuato per irrobustire ancora di più il sistema in termini di efficienza è quello di effettuare dei backup periodici in modo da evitare la perdita di una grande quantità di informazioni in caso di attacco

- **NO-EVOX:** potrebbe agire da solo o all'interno di un movimento. Mira a diminuire la fiducia pubblica nei confronti del sistema di voto, indicando unicamente i possibili rischi derivanti dal suo utilizzo. Gli attacchi alla fiducia pubblica spesso mescolano solide argomentazioni con la demagogia ed il loro obiettivo potrebbe andare oltre la demonizzazione dell'e-voting, sfociando ad esempio nella revoca di un risultato elettorale o nell'annullamento di un metodo di voto. Attacchi di questo tipo sono semplici da realizzare ed apparentemente privi di rischi: è sufficiente sollevare problematiche, ipotetiche criticità, possibili contraddizioni con la legge o differenze con le tornate elettorali precedenti e presentarli al pubblico come se fossero veri e propri attacchi alla democrazia.

Un gruppo di NO-EVOX potrebbe mandare continuamente richieste di registrazione e di voto al server SPLAT in modo da sovraccaricarlo e provocare ritardi significativi del sistema di e-voting. Gli elettori intenzionati a votare in modo serio, quindi, non riuscirebbero a votare nel breve tempo e la fiducia pubblica diminuirebbe a causa dei lunghi tempi di attesa che farebbero apparire la piattaforma malfunzionante.

Questo tipo di attacco non è facile da mitigare. Un miglioramento che si potrebbe fare è quello di impostare un limite massimo di tentativi di comunicazione con la piattaforma. In particolare, se un elettore termina il numero di richieste al server SPLAT a sua disposizione, deve attendere un certo numero di ore per poter effettuare la prossima richiesta.

In conclusione, l'efficienza è abbastanza robusta, ma potrebbero essere introdotti dei miglioramenti che la renderebbero ancora più robusta.

WP4

Di seguito si fornisce un'implementazione in Java per l'e-ballot progettato all'interno del WP2

Implementazione

Considerazioni

- Le ZK Proof sono state considerate ma non implementate
- Si assume che ogni elettore possiede già il proprio certificato digitale self-signed basato su RSA e quindi non è prevista una richiesta ad una CA
- La Threshold ElGamal Decryption è stata implementata attraverso una soluzione basata su Shamir(t, n) con $t = n$. In particolare, è stato utilizzato Shamir(3,3) per la presenza di 3 S_{BAL} .
- Non stati effettuati controlli sul numero di voti e di firme al termine della finestra temporale $[T1-T2]$.
- Non sono stati creati dei veri e propri database, ma sono state utilizzate delle strutture dati adeguate mantenute in RAM
- Dato che sono stati sfruttati dei Certificati self-signed basati su RSA, per firmare con Schnorr sono state sfruttate delle coppie di chiavi generate successivamente e non correlate ai Certificati
- I parametri di sicurezza ideali sono 2048 bit per El Gamal e 512 bit per Schnorr. Nell'implementazione sono stati usati 256 bit per El Gamal e 256 per Schnorr

Costituzione del sistema

- 1 S_{GEN}
- 1 S_{PLAT}
- 3 S_{BAL}
- 1 S_{TAB}
- 1 Timer
- 5 Voters

Tool utilizzati

- Certificati digitali Self-Signed basati su RSA e generati con openssl/keytool con l'aggiunta di appositi campi personalizzati
- Variazione dello schema di cifratura a chiave pubblica (El Gamal)
- Threshold El Gamal Decryption
- Firma digitale di Schnorr
- Password Hashing con SHA256
- TLS con autenticazione bidirezionale
- Libreria Passay per la generazione di ID casuali sfruttando randomness sicura
- Netbeans 8.2, Java 1.8, Bouncycastle 1.71

Strutture dati utilizzate

Per simulare la presenza di database senza dover ricorrere ad essi sono state utilizzate le seguenti strutture dati:

- S_{PLAT} sfrutta:
 - o una HashMap per contenere i codici fiscali di tutti coloro che sono ammessi al voto ed il check per determinare se hanno già effettuato la registrazione in piattaforma
 - o una HashMap per contenere ID, Password Hashata, Salt, Ciphertext del voto, Firma del voto
- S_{BAL} sfrutta:
 - o una HashMap per contenere Ciphertext del voto, Firma del voto
- S_{TAB} sfrutta:
 - o una HashMap per contenere Ciphertext del voto, Firma del voto

Esempio di funzionamento

- Voter 1: richiesta ID ✓
- Voter 1: accesso in piattaforma con proprio ID ✓
- Voter 1: voto ✓
- Voter 1: modifica voto ✓
- Voter 2: richiesta ID ✓
- Voter 2: accesso in piattaforma con ID di Voter 1 ✗
- Voter 2: accesso in piattaforma con proprio ID ✓
- Voter 2: voto ✓
- Voter 3: richiesta ID ✓
- Voter 3: accesso in piattaforma con proprio ID ✓
- Voter 3: voto ✓
- Voter 3: annullamento voto ✓
- Voter 3: voto ✓
- Voter 4: richiesta ID ✓
- Voter 4: accesso in piattaforma con proprio ID ✓
- Voter 4: annullamento voto ✗
- Voter 5: richiesta ID ✗
- Voter 1: richiesta nuovo ID ✗

```
I want an ID
Arriving ID SUCCESS
I want to vote
vote SUCCESS
I want to vote
vote SUCCESS
I want an ID
Arriving ID SUCCESS
Credential check ERROR
I want an ID
Arriving ID SUCCESS
I want to vote
vote SUCCESS
I want an ID
Arriving ID SUCCESS
I want to vote
vote SUCCESS
I want to vote
vote SUCCESS
I want to vote
vote SUCCESS
I want an ID
Arriving ID SUCCESS
I want to vote
No previous vote ERROR
vote ERROR
I want an ID
CD check ERROR
I want an ID
CD check ERROR
```

Voter 1

Voter 2

Voter 3

Voter 4

Voter 5

Voter 1

I voti espressi sono:

Voter 1: -1 → 1

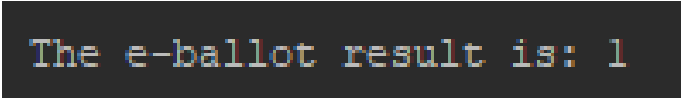
Voter 2: 1

Voter 3: 0 → null → -1

Voter 4: null

Voter 5: non può votare

In questo caso il vincitore sarebbe stato A



```
The e-ballot result is: 1
```

Come avviare su Netbeans

1. Avviare Timer (run file Timer.java)
2. Avviare S_{TAB} (run file Stab.java)
3. Avviare S_{BAL} (run file Sbal.java)
4. Avviare S_{PLAT} (run file Splat.java)
5. Avviare S_{GEN} ed attendere che termini (run file Sgen.java)
6. Avviare Voter (run file Voter.java)

Aggiornamenti effettuati:

WP1:

- eliminazione dell'avversario MINISTERO INFAME in quanto considerato poco significativo per il sistema

WP2:

- Aggiunta di qualche ulteriore cenno teorico
- Aggiunta della sezione "Assunzioni"

WP3:

- Completamento dell'analisi delle proprietà. Per ogni proprietà si riportano riflessioni sugli attacchi dei possibili avversari che potrebbero comprometterla

WP4:

- Si riportano gli aspetti maggiormente significativi dell'implementazione del sistema e-ballot