WP1

Il setting richiesto, ovvero un turno di ballottaggio durante le elezioni amministrative, prevede due candidati che sono interessati ad ottenere la carica di sindaco, ed elettori che vogliono votare il candidato più vicino ai propri ideali, idee o interessi.
Gli obiettivi contrastanti fra diversi elettori e i due candidati, oltre che tra i candidati stessi, introducono chiaramente rischi di cheating che possono provenire da un singolo agente o da una coalizione di agenti.

La votazione si svolge in nove fasi. In una prima finestra temporale T0-T1, gli elettori hanno il compito di controllare se possiedono tutti i requisiti necessari per partecipare alla fase di votazione elettronica.
Tra T1-T2 si identificano con SPID attraverso un Identity Provider (IdP) autorizzato. In questo modo, se abilitati, si prenotano tramite identificativo univoco per l'ottenimento di credenziali necessarie per accedere alle votazioni. In questo lasso di tempo, è previsto, inoltre, che l'elettore possa cambiare la sua chiave univoca fino al tempo T2. L'ultima inserita è l'unica e sola valida.
Tra T2-T3 avviene una fase di controllo in cui ogni elettore deve verificare se l'abilitazione è andata a buon fine; quindi, se le informazioni sono corrette.
Tra T3-T4 vengono generate le credenziali all'utente per poter accedere e votare.
Tra T4-T5 l'utente deve recuperare le credenziali a lui associate e verificare se sono presenti informazioni corrette.
Tra T5-T6 ogni elettore esprime la sua preferenza. In questo lasso di tempo, è previsto, inoltre, che l'elettore possa cambiare la sua preferenza arbitrariamente fino al tempo T6. L'ultima preferenza espressa è l'unica e sola valida.
Tra T6-T7 avviene lo spoglio dei voti online.
Tra T7-T8 viene pubblicato l'elenco dei voti validi a cui vengono associati le credenziali degli elettori e il numero di voti effettuati da questi ultimi.
In questa fase ogni elettore può esprimere la volontà di votare fisicamente attraverso dei seggi, sia se ha già effettuato il voto online, sia se ha già ottenuto le credenziali per il voto online ma non ha effettuato alcun voto, sia se non ha effettuato nessuna delle operazioni descritte fino a questo lasso di tempo.
Nel caso in cui l'elettore che voglia votare fisicamente abbia ricevuto le credenziali per il voto online, a prescindere se quest'ultimo abbia votato o meno, deve obbligatoriamente sporgere denuncia prima di poter essere abilitato al voto fisico.
Infine, in tempo T8-T9 avviene il conteggio dei voti online e fisici e viene determinato il vincitore che otterrà la carica di sindaco.
Nel caso in cui gli utenti non abbiano fatto richiesta delle credenziali prima del tempo T3, non sono abilitati a partecipare alla sessione di votazione online. Il numero di elettori non deve superare il numero di persone che fanno parte dell'elettorato attivo del Comune interessato.

Nei comuni con più di 15.000 abitanti, il ballottaggio viene vinto dal candidato sindaco che ottiene la maggioranza assoluta dei voti, vale a dire il 50% + 1 dei voti validamente espressi. Invece nei Comuni fino a 15.000 abitanti il ballottaggio viene vinto dal candidato che ottiene la maggioranza relativa dei voti; quindi, non serve che raggiunga il 50% più uno, ma basta che ottenga un numero di voti maggiore di quello dell'avversario.

Noi consideriamo il Ministero per l'innovazione tecnologica e la transizione digitale (MITD), il Ministero dell'Interno (MI), il Ministero della Difesa (MD) e il Ministro per la Pubblica Amministrazione (MPA).
Esiste, inoltre, una blockchain permissioned pubblica, la cui governance è affidata a MITD e MI. Se entrambi sono d'accordo, possono dare la possibilità di operare su di essa o visualizzarne il contenuto. È presente, quindi, una piattaforma web gestita da MI su cui sono presenti le informazioni necessarie alla votazione (finestre temporali, livello di SPID richiesto, candidati coinvolti, comune di riferimento, codici open source e riferimento alla blockchain).
Assumiamo che in casi sporadici si possa coinvolgere anche la giustizia, che ha un'identità ben nota J. Ha senso che J venga utilizzato solo in caso di controversia o supervisione, evitando il più possibile il suo coinvolgimento. Una controversia è intrinsecamente possibile, infatti una volta che la votazione è conclusa, il candidato sconfitto potrebbe presentare un'istanza di ricorso per irregolarità dovute a fattori esterni al sistema.

Completeness

Siano C_1 *e* C_2i potenziali candidati e $E_1 \dots E_n$ gli elettori.

Al tempo T0-T1, ciascun elettore interessato a partecipare alla votazione, accede alla piattaforma web gestita da MI per ottenere le informazioni utili; inoltre, genera il suo identificativo univoco. Contemporaneamente, MITD, MI, MPA e MD ricevono informazioni utili per eseguire lo spoglio alla fine della votazione.
Al tempo T1-T2 l'elettore fornisce al MITD e MPA l'identificativo x_i per i=1, …, n. Ciascun elettore per essere autorizzato a fornire x_i , deve autenticarsi presso un IdP autorizzato. Per ogni elettore viene salvato sulla blockchain x_i e Codice Fiscale.
Al tempo T2-T3 ogni elettore deve controllare sulla blockchain se le informazioni presenti sono corrette.
Al tempo T3-T4, per ogni CF-x_i presente, MI e MD generano le credenziali per accedere alla fase di votazione in modo che l'elettore riconosca solo le credenziali che gli interessano.
Al tempo T4-T5 l'utente deve recuperare le credenziali a lui associate e verificare se sono presenti informazioni corrette.
Al tempo T5-T6 avviene la fase di votazione in cui ogni elettore si identifica, tramite le sue credenziali, comunicando con la blockchain permissioned. Genera un input v_i che rappresenta la preferenza espressa. v_i è un intero positivo di dominio $\{0, \dots, 3\}$. I valori di v_i rappresentano: 0, per la volontà di consegnare una scheda bianca; 1, per la votazione a favore di C_1; 2, per la votazione a favore di C_2; 3 per la volontà di astenersi dalla votazione. Fornisce v_i alla blockchain, che si occuperà di creare una transazione contente le credenziali dell'elettore e il voto associato. Essendo pubblica, l'elettore E_i può controllare se v_i è stato ricevuto correttamente. L'elettore può cambiare v_i fino al tempo T6.
Al tempo T6-T7, avviene lo spoglio di tutti gli input v_n da parte di almeno tre attori coinvolti al tempo T0-T1.
Al tempo T7-T8, l'elettore E_i potrà effettuare un ultimo controllo per verificare se v_i è quello desiderato. Inoltre, l'elettore E_i potrà verificare il numero di volte in cui attraverso le sue credenziali è stato cambiato v_i attraverso il parametro Nv_i. L'elettore potrà, quindi, decidere se partecipare alla sessione di voto fisico, annullando la scelta effettuata durante la sessione di voto online.
Alla sessione di voto fisico possono accedere quattro tipologie di utente:

1. colui che non ha richiesto le credenziali in tempo utile per la votazione online, ma voglia comunque esprimere la sua preferenza;
2. colui che ha richiesto le credenziali in tempo utile, ma non abbia partecipato alla sessione di voto online;
3. colui che ha richiesto le credenziali in tempo utile, ha partecipato alla sessione di voto online ma voglia cambiare la sua preferenza;
4. colui che ha richiesto le credenziali in tempo utile, ha partecipato alla sessione di voto online ma ha il sospetto che durante la fase di voto ci sia stato un furto di identità o problemi indipendenti dal sistema di voto.

Per il secondo, terzo e quarto caso è necessario che l'elettore E_i sporga denuncia e fornisca le credenziali a J in modo da permettere che in fase di spoglio i propri voti online non vengano conteggiati e dargli la possibilità di votare fisicamente.
Infatti, gli scrutinatori controllano i CF presenti sulla blockchain e quelli degli utenti che hanno sporto denuncia e abilitano, quindi, il voto fisico soltanto a chi non è presente su nessuno dei due e a chi è presente su entrambi.
Infine, in tempo T8-T9 avviene il conteggio dei voti online e fisico e viene decretato il candidato vincente.

La giustizia J può essere invocata durante una controversia.

Threat model

Fake elector. Un cittadino non autorizzato a partecipare al ballottaggio (cittadino di un altro comune o minorenne) esprime il suo voto rispetto ad un candidato. Questo avversario realisticamente non ha né un grande potere computazionale né capacità di controllo del canale di comunicazione. Potrebbe assistere all'inserimento delle credenziali e alla sessione di votazione di un elettore autorizzato.

Mafia candidate. Un candidato costringe elettori, minacciandoli, a votare per lui controllando la preferenza espressa. Questo avversario potrebbe ragionevolmente avere un grande potere computazionale data la sua posizione di rappresentante di più persone. Non ha controllo del canale di comunicazione. Potrebbe assistere all'inserimento delle credenziali e alla sessione di votazione di un elettore autorizzato. Potrebbe collaborare con tutti i cittadini disonesti che vogliono che salga al potere. Inoltre, potrebbe collaborare anche con i MITD, MI, MD, MPA.

Extremist voter. Un elettore, che vuole necessariamente che un certo candidato salga al potere, invia molteplici volte il suo voto. Questo avversario realisticamente non ha grandi risorse a disposizione.

Eavesdropper: Un elettore osserva, per scopi di lucro, l'andamento del ballottaggio. Questo avversario realisticamente non ha né un grande potere computazionale né capacità di controllo del canale di comunicazione. Potrebbe assistere all'inserimento delle credenziali e alla sessione di votazione di un elettore autorizzato. Potrebbe collaborare con un Mafia Candidate.

Invalidator. Un cittadino invalida il voto di un elettore per non permettergli di esprimere il suo diritto al voto. Questo avversario realisticamente non ha né un grande potere computazionale né capacità di controllo del canale di comunicazione. Potrebbe assistere all'inserimento delle credenziali e alla sessione di votazione di un elettore autorizzato.

Doppelganger. Un cittadino vota con le credenziali di un altro elettore per non permettergli di esprimere la sua preferenza. Questo avversario realisticamente non ha né un grande potere computazionale né capacità di controllo del canale di comunicazione. Potrebbe assistere all'inserimento delle credenziali e alla sessione di votazione di un elettore autorizzato.

Corrupted Identity provider - block. I Gestori di Identità Digitale, per favorire un candidato, bloccano l'accesso a elettori dichiaratamente del partito avversario. In questo modo, non possono usufruire dei servizi utili alla partecipazione alla votazione. Questo avversario ha un enorme potere computazionale e può visionare alcune comunicazioni di autenticazione che avvengono sul canale. Questo avversario potrebbe collaborare con un Corrupted MITD.

Corrupted Identity provider – exchange identity. I Gestori di Identità Digitale, per favorire un candidato, permettono ad un elettore di votare utilizzando credenziali di cittadini dichiaratamente del partito avversario. In questo modo, il cittadino non può usufruire dei servizi utili alla partecipazione alla votazione. Questo avversario ha un enorme potere computazionale e può visionare alcune comunicazioni di autenticazione che avvengono sul canale. Questo avversario potrebbe collaborare con un MITD.

Corrupted MI – allow scope. Il ministero dell'interno, per favorire un candidato, blocca l'accesso alla votazione a elettori dichiaratamente del partito avversario. Questo avversario realisticamente ha un enorme potere computazionale. Possiede la governance della blockchain permissioned e può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con un altro ministero è improbabile.

Corrupted MI – vote scope. Il ministero dell'interno è interessato a modificare, aggiungere e cancellare i voti del ballottaggio per favorire un candidato. Questo avversario realisticamente ha un enorme potere computazionale. Possiede la governance della blockchain permissioned e può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con un altro ministero è improbabile.

Corrupted MI – final result scope. Il ministero dell'interno è interessato a modificare il risultato finale ottenuto dal conteggio dei voti validi per favorire un candidato. Questo avversario realisticamente ha un enorme potere computazionale. Possiede la governance della blockchain permissioned e può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con un altro ministero è improbabile.

Corrupted MI – view scope. Il ministero dell'interno è interessato a ottenere una corrispondenza voto - elettore per minare la privacy dei votanti. Questo avversario realisticamente ha un enorme potere computazionale. Possiede la governance della blockchain permissioned e può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con un altro ministero è improbabile.

Corrupted MI – service denial scope. Il ministero dell'interno è interessato a invalidare l'intera sessione di voto. Possiede la governance della blockchain permissioned e può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con l'altro ministero è improbabile.

Corrupted MITD – allow scope. Il ministero per l'innovazione tecnologica e la transizione digitale, per favorire un candidato, blocca l'accesso alla votazione a elettori dichiaratamente del partito avversario. Possiede la governance della blockchain permissioned e può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate e/o un Corrupted Identity Provider. Una collaborazione con un altro ministero è improbabile.

Corrupted MITD – vote scope. Il ministero per l'innovazione tecnologica e la transizione digitale è interessato a modificare, aggiungere e cancellare i voti del ballottaggio per favorire un candidato. Possiede la governance della blockchain permissioned e può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate e/o un Corrupted Identity Provider. Una collaborazione con un altro ministero è improbabile.

Corrupted MITD – final result scope. Il ministero per l'innovazione tecnologica e la transizione digitale è interessato a modificare il risultato finale ottenuto dal conteggio dei voti validi per favorire un candidato. Possiede la governance della blockchain permissioned e può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate e/o un Corrupted Identity Provider. Una collaborazione con un altro ministero è improbabile.

Corrupted MITD – view scope. Il ministero per l'innovazione tecnologica e la transizione digitale è interessato a ottenere una corrispondenza voto - elettore per minare la privacy dei votanti. Possiede la governance della blockchain permissioned e può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate e/o un Corrupted Identity Provider. Una collaborazione con un altro ministero è improbabile.

Corrupted MITD – service denial scope. Il ministero per l'innovazione tecnologica e la transizione digitale è interessato a invalidare l'intera sessione di voto. Possiede la governance della blockchain permissioned e può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate e/o un Corrupted Identity Provider. Una collaborazione con l'altro ministero è improbabile.

Corrupted MPA – allow scope. Il ministro per la Pubblica Amministrazione, per favorire un candidato, blocca l'accesso alla votazione a elettori dichiaratamente del partito avversario. Può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con un altro ministero è improbabile.

Corrupted MPA – vote scope. Il ministro per la Pubblica Amministrazione è interessato a modificare, aggiungere e cancellare i voti del ballottaggio per favorire un candidato. Può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con un altro ministero è improbabile.

Corrupted MPA – final result scope. Il ministro per la Pubblica Amministrazione è interessato a modificare il risultato finale ottenuto dal conteggio dei voti validi per favorire un candidato. Può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con un altro ministero è improbabile.

Corrupted MPA – view scope. Il ministro per la Pubblica Amministrazione è interessato a ottenere una corrispondenza voto - elettore per minare la privacy dei votanti. Può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate e/o un Corrupted Identity Provider. Una collaborazione con un altro ministero è improbabile.

Corrupted MPA – service denial scope. Il ministero per la Pubblica Amministrazione è interessato a invalidare l'intera sessione di voto. Può intercettare le connessioni tra qualsiasi coppia di parti che comunicano attraverso il canale. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con un altro ministero è improbabile.

Corrupted MD – allow scope. Il ministero della difesa, per favorire un candidato, blocca l'accesso alla votazione a elettori dichiaratamente del partito avversario. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con un altro ministero è improbabile.

Corrupted MD – vote scope. Il ministero della difesa è interessato a modificare, aggiungere e cancellare i voti del ballottaggio per favorire un candidato. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con un altro ministero è improbabile.

Corrupted MD – final result scope. Il ministero della difesa è interessato a modificare il risultato finale ottenuto dal conteggio dei voti validi per favorire un candidato. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con un altro ministero è improbabile.

Corrupted MD – view scope. Il ministero della difesa è interessato a ottenere una corrispondenza voto - elettore per minare la privacy dei votanti. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con un altro ministero è improbabile.

Corrupted MD – service denial scope. Il ministero della difesa è interessato a invalidare l'intera sessione di voto. Potrebbe collaborare con un Mafia Candidate. Una collaborazione con un altro ministero è improbabile.

Hacktivism. Un gruppo di cittadini interni o esterni alle elezioni che vuole impedire l'avvenuta delle stesse per motivi etici. Questo avversario realisticamente ha un potere computazionale tale da poter invalidare il funzionamento di un server o un database centralizzato per un lasso di tempo limitato.

Scrutinatori. Un gruppo di cittadini interni o esterni alle elezioni, chiamati per svolgere il ruolo di scrutinatore, che vuole permettere ad un elettore di votare più volte. Questo avversario realisticamente non ha un grande potere computazionale.

Colluding cheaters. Unione di più avversari durante un attacco.

Integrità

In presenza di avversari, il sistema dovrebbe comunque garantire che:

- **I.1 Authenticity.** Le possibilità di vincere per un candidato devono essere dipendenti dal solo voto degli elettori autorizzati (i.e. solo elettori con i requisiti necessari per la specifica votazione possono partecipare al voto);
- **I.2 Freedom.** Ogni elettore autorizzato deve aver la possibilità di votare secondo la sua volontà senza obblighi esterni;
- **I.3 Uniformity.** Ogni elettore può esprimere molteplici voti ma solo uno può essere riconosciuto come valido;
- **I.4 Availability.** I metodi di voto devono essere resi disponibili ai cittadini autorizzati;
- **I.5 Fairness.** Ogni azione scorretta sarà scoraggiata o punita ai sensi della legge;
- **I.6 Correctness.** Ogni voto espresso dagli elettori deve essere correttamente valutato ed i risultati pubblicamente divulgati dopo lo spoglio devono essere corretti.

Privacy

In presenza di avversari, il sistema dovrebbe comunque garantire che:

- **P.1 Secretly.** Nessuno all'infuori del votante deve essere a conoscenza della preferenza (o della mancata votazione) espressa;
- **P.2 Confidentiality.** I risultati delle elezioni devono restare segreti fino allo spoglio.

WP2

Per semplicità, dividiamo l'analisi delle scelte progettuali effettuate in base agli istanti di tempo in cui esse avvengono.

Per tutti i successivi istanti di tempo assumiamo che:

- sia presente un'unica blockchain permissioned pubblica, la cui governance è affidata a 7 membri del MITD e 7 membri del MI; quindi, per accettare l'inserimento di un blocco è necessario che 8 su 14 siano di comune accordo;
- ogni utente abbia tutti i dispositivi personali privi di malware, in good shape e con OS corretto;
- ogni utente si trovi in un contesto digitalizzato e abbia le capacità per effettuare tutte le operazioni descritte;
- durante le fasi di controllo, tutti i cittadini abilitati al voto controllino che non ci siano irregolarità associate alla loro identità.

T0-T1

Consideriamo la presenza di:

- macchina in good shape, con OS corretto, non connesso alla rete e che contiene in memoria unicamente il codice da eseguire;
- quattro ministeri in un'unica stanza che agiscono sul pc precedentemente descritto;
- piattaforma dell'MI per le informazioni utili alla votazione;
- utente che genera il suo identificativo.

MPA: Utilizza l'algoritmo di generazione della chiave dello schema di cifratura di ElGamal, per ottenere la coppia (PK_{MPA}, SK_{MPA}).

MD: Utilizza l'algoritmo di generazione della chiave dello schema di cifratura di ElGamal, per ottenere la coppia (PK_{MD}, SK_{MD}).

MITD: Utilizza l'algoritmo di generazione della chiave dello schema di cifratura di ElGamal, per ottenere la coppia (PK_{MITD}, SK_{MITD}) e pubblica sulla piattaforma web di voto gestita dal MI gli 8 codici open source insieme al risultato dello SHA256 di questi codici. Questi codici servono rispettivamente per: effettuare l'algoritmo di generazione della ***chiave segreta*** per permettere lo spoglio dei voti; generare gli ***schemi di cifratura di ElGamal*** da parte dell'utente e dei ministeri; codice lato server per permettere la comunicazione con IdP e blockchain per l'***autorizzazione dell'inserimento*** dei dati dell'utente; codice lato utente per la comunicazione con IdP e server per l'***inserimento del*** PK_{USER}; generare le ***credenziali*** per permettere l'accesso alla votazione a partire dai PK_{USER} inseriti; codice lato utente per la comunicazione con blockchain per l'***inserimento del voto***; effettuare lo ***spoglio*** dei voti online; effettuare il ***conteggio*** dei voti totali.

MITD, MI, MPA, MD: Avviano un codice sulla macchina, precedentemente controllato e visionato da tutti e quattro gli attori coinvolti. I ministeri lo controllano e ne controllano il software, verificando che il risultato dello SHA256 coincida con quello pubblicato sulla piattaforma web del MI. Il codice utilizza l'algoritmo di generazione della chiave dello schema di cifratura di ElGamal per ottenere la coppia (PK_{SS}, SK_{SS}). Il segreto SK_{SS} è diviso seguendo lo schema Shamir Secret Sharing (3,4) tra MITD, MI, MPA, MD. Le share criptate tramite l'algoritmo di cifratura di ElGamal vengono visualizzate come segue:

Enc(p(x_i), PK_{MITD}), Enc(σ_i: $Sig(SK_{SS}, \mathrm{p}(x_i))$, PK_{MITD})

Enc(p(x_i), PK_{MI}), Enc(σ_i: $Sig(SK_{SS}, \mathrm{p}(x_i))$, PK_{MI})

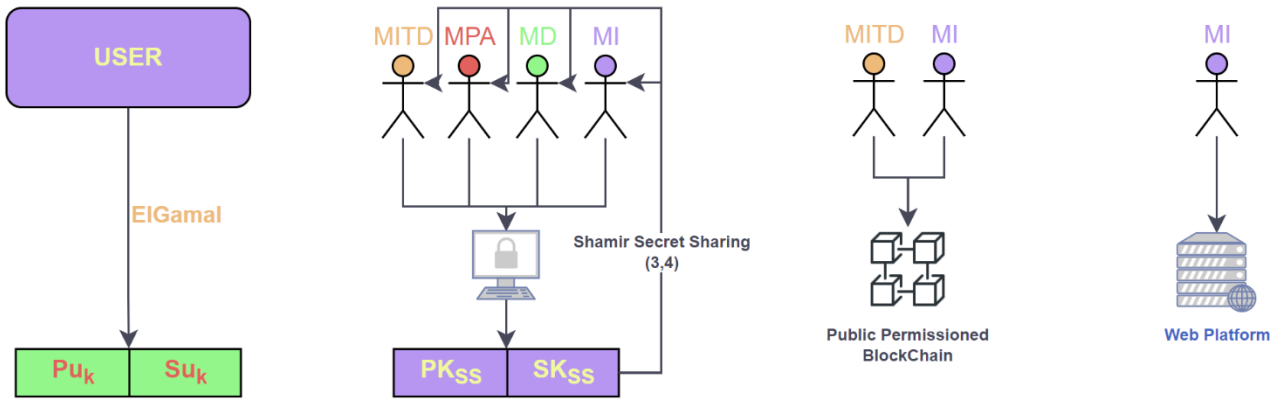
Enc(p(x_i), PK_{MPA}), Enc(σ_i: $Sig(SK_{SS}, \mathrm{p}(x_i))$, PK_{MPA})

Enc(p(x_i), PK_{MD}), Enc(σ_i: $Sig(SK_{SS}, p(x_i))$, PK_{MD})

firmate seguendo lo schema di Schnoor per la firma digitale, con 1<=i<=4 e p(x) polinomio di grado 3 generato dallo schema di Shamir Secret Sharing avendo come termine noto SK_{SS}. In questo modo, ogni attore coinvolto, in privato, potrà decriptare la sua parte di share. Inoltre, verrà visualizzata PK_{SS}.
Ad operazioni finite, la macchina verrà resettata completamente eliminandone tutti i dati.

MI: Utilizza l'algoritmo di generazione della chiave dello schema di cifratura di ElGamal, per ottenere la coppia (PK_{MI}, SK_{MI}).
Pubblica sulla piattaforma web da lui gestita:
Lista degli IdP abilitati aventi almeno SPID di livello 2
Nomi dei 2 candidati
Comune di riferimento votazioni
Data e ora dell'inserimento sulla piattaforma (T0)
Data e ora inizio prenotazione (T1)
Data e ora inizio prima fase di controllo (T2)
Data e ora rilascio credenziali(T3)
Data e ora inizio seconda fase di controllo (T4)
Data e ora inizio votazioni online (T5)
Data e ora inizio votazioni fisiche (T6)
Data e ora spoglio (T7)
Indirizzo tramite il quale connettersi al server
Riferimento alla blockchain permissioned pubblica
PK_{MITD}
PK_{MI}
PK_{MPA}
PK_{MD}
PK_{SS}

Utente: Utilizza l'algoritmo di generazione della chiave dello schema di cifratura di ElGamal, per ottenere la coppia (PK_{USER}, SK_{USER}). Controlla sulla piattaforma web di MI se possiede tutti i requisiti per partecipare alla votazione e se il codice proposto dai ministeri è corretto.



T1-T2

Consideriamo la presenza di:

- file su cui sono salvati tutti i Codici Fiscali dei cittadini abilitati alla votazione; ciascun nodo della blockchain ha accesso al file;
- macchina in good shape, con OS corretto, che abbia tutte le protezioni necessarie per la connessione alla rete e che contiene in memoria unicamente il codice da eseguire;
- server hostato da MPA sul pc precedentemente descritto, ma supervisionato da MITD;
- IdP;
- utente che comunica con il server.

Vengono effettuate, inoltre, operazioni di inserimento sulla blockchain permissioned pubblica tramite il server.

Codice open-source T1-T2 lato utente: Il client eseguirà il seguente protocollo per connettersi al server:

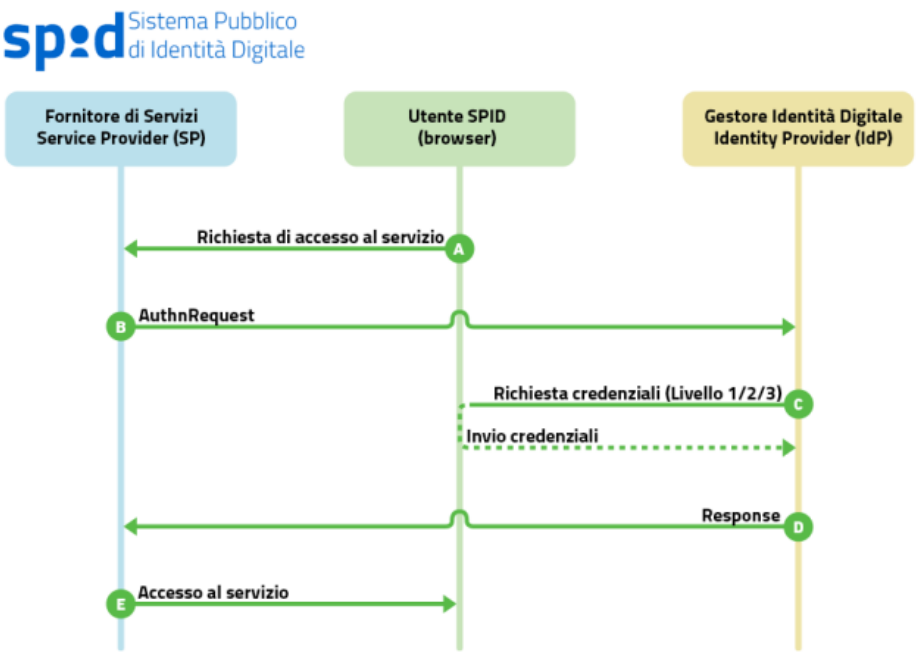
1. L'utente richiede l'accesso al servizio.
2. L'utente sceglie un IdP presso cui autenticarsi scegliendo tra quelli proposti dal server.
3. L'utente riceve la richiesta di identificazione tramite credenziali SPID.
4. L'utente invia le credenziali SPID.
5. L'utente conferma di voler permettere l'accesso al suo codice fiscale, data di nascita e comune di residenza.
6. All'utente viene chiesto di confermare i dati forniti dall'Idp.
7. L'utente inserisce la propria PK_{USER}.
8. L'utente riceve dal server la firma $\sigma_{u_{MPA}}: Sig(SK_{MPA}, PK_{USER})$.

Appuriamo che queste 8 comunicazioni descritte sono protette da TLS 1.3.

Codice open-source T1-T2 lato server: All'avvio il codice richiede l'inserimento di SK_{MPA}. Successivamente aprirà una porta e creerà una socket che utilizza TLS 1.3 per agire da server. Rimarrà in attesa di ricevere comunicazioni e quando ne rileva una, esegue il seguente protocollo:

1. Riceve la richiesta di accesso al servizio.
2. Invia all'utente la lista di IdP tra cui scegliere.
3. Manda un AuthnRequest all' IdP scelto dall'utente.
4. Riceve la Response in maniera autentica con i dati di interesse (CF, data di nascita e comune di residenza).
5. Gestisce un meccanismo esplicito di chiusura della sessione di autenticazione con SPID.
6. Richiede all'utente di confermare i dati forniti dall'Idp, inviandoglieli.
7. Riceve la conferma dei dati. Se è negativa, termina il protocollo e notifica l'incongruenza salvando su un file l'indirizzo IP e i dati relativi alla comunicazione.
8. Controlla la data di nascita e il comune di residenza dell'utente per verificare se è abilitato alla sessione di votazione. Se non lo è, chiude la connessione; altrimenti, viene eseguito il punto successivo.
9. Riceve il PK_{USER} dell'utente.
10. Invia alla Blockchain la coppia CF – PK_{USER} per l'esecuzione dello smart contract.
11. Genera la firma $\sigma_{u_{MPA}}: Sig(SK_{MPA}, PK_{USER})$ seguendo lo schema di Schnoor per la firma digitale.
12. Invia all'utente la firma $\sigma_{u_{MPA}}$ e avviene la chiusura della connessione.

Appuriamo che queste 12 comunicazioni descritte sono protette da TLS 1.3.



Ai sensi dell'art 28 del regolamento *Modalità attuative per la realizzazione dello SPID*, per i livelli 2 e 3 SPID, allo scopo di garantire la massima sicurezza e stabilità del sistema, non si prevede la possibilità di mantenimento di sessioni condivise di autenticazione. Pertanto:
1) il gestore dell'identità digitale non deve mantenere alcuna sessione di autenticazione con l'utente;
2) ogni fornitore di servizi deve gestire per proprio conto l'eventuale sessione con l'utente. Per la chiusura dovranno essere forniti meccanismi espliciti per il logout.
Inoltre, il comma 2 dell'articolo 13 del DPCM obbliga i fornitori di servizi (Service Provider) alla conservazione per ventiquattro mesi delle informazioni necessarie a imputare alle singole identità digitali le operazioni effettuate sui propri sistemi. A tal fine il Service provider dovrà mantenere un Registro delle transazioni contenente i tracciati delle richieste di autenticazione servite negli ultimi 24 mesi. L'unità di memorizzazione di tale registro dovrà rendere persistente per ogni transazione la coppia dalla <AuthnRequest> e della relativa <Response>.

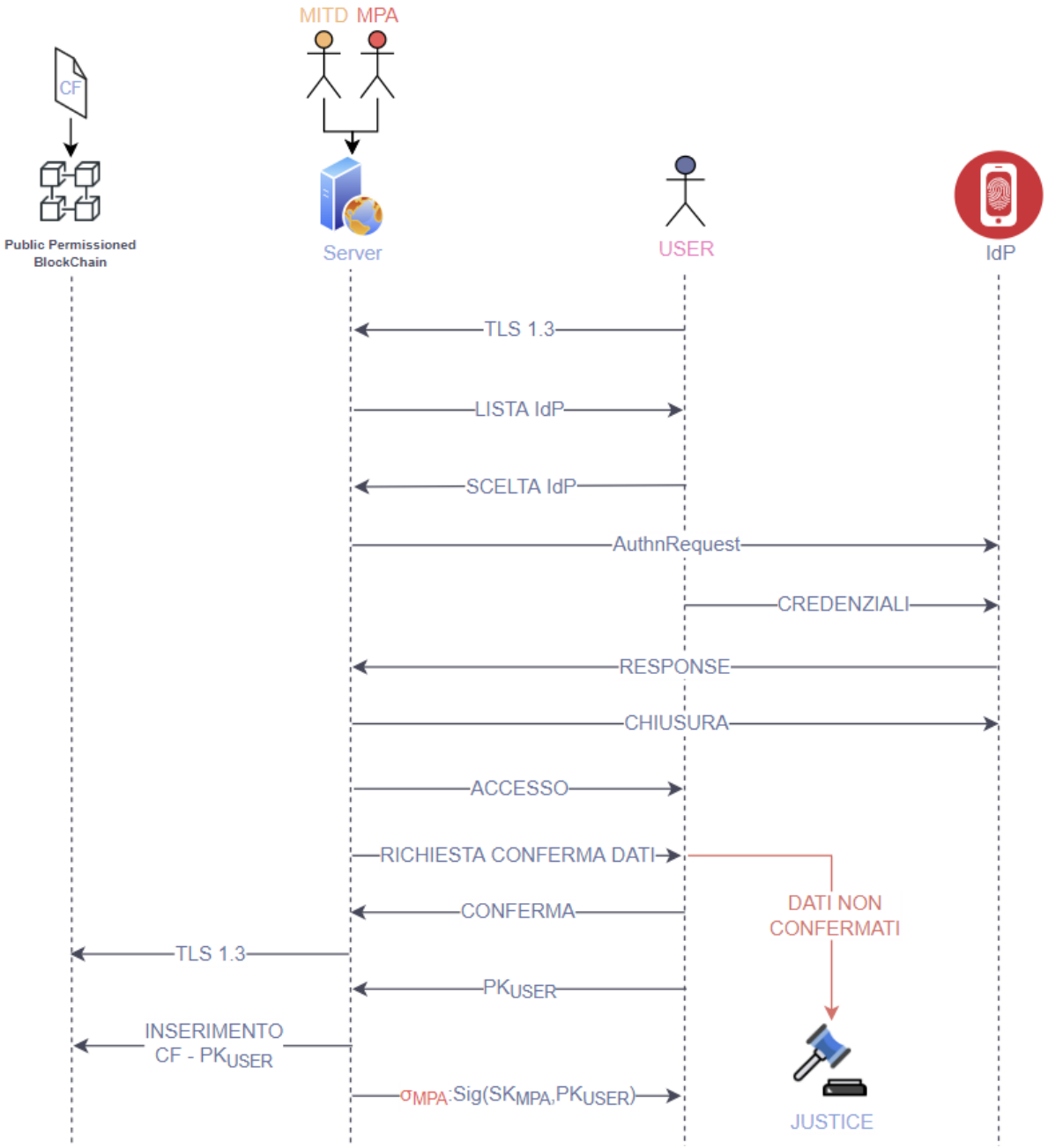
Blockchain permissioned: Sulla blockchain è implementato uno smart contract che si assicura che, ricevuti CF e PK_{USER}, il CF sia presente nella lista di persone abilitate alla votazione (quindi viene controllato il file) e che non esista già un PK_{USER} con lo stesso valore di quello inserito.

MITD e MPA: Avviano il codice open-source lato server sulla macchina. MITD e MPA lo controllano e ne controllano il software, verificando che il risultato dello SHA256 coincida con quello pubblicato sulla piattaforma web del MI. Generano l'indirizzo a cui far connettere gli utenti e lo pubblicano sulla piattaforma web del MI. Viene lanciato il codice e inserisce quando richiesto SK_{MPA}. Da questo momento in poi questo computer viene lasciato senza poter essere manomesso da alcuna delle parti dato che l'altra funge da supervisore. Alla fine del tempo T2 consegnano a Justice il file contenente l'indirizzo IP e i dati relativi alle presunte comunicazioni errate.

Utente: Sul suo dispositivo personale viene lanciato il codice open-source lato client. L'utente seguirà i passaggi richiesti da tale codice facendo dunque l'accesso con credenziali SPID e confermando la propria identità. Inviato PK_{USER} e ottenuta la firma associata, per verificarne la correttezza utilizza l'algoritmo $Ver(PK_{MPA}, PK_{USER}, \sigma_{u_{MPA}})$. Può ripetere questa operazione fino al tempo T2. L'ultimo PK_{USER} inserito è l'unico e solo valido.

IdP: Ingaggia una connessione TLS 1.3 con l'utente a seguito della richiesta da parte del server. Quest'ultima specifica quali dati personali dell'utente vengono richiesti. L'utente, dunque, inserisce le credenziali, decide se acconsentire all'invio dei dati personali precedentemente richiesti; quindi, l'IdP

procede a mandare questi dati al server e chiude subito la connessione dato che si tratta di uno SPID di livello 2.



T2-T3

Utente: Gli viene garantita la visione dell'intera blockchain (ormai non più modificabile) per verificare la presenza di incongruenze. Qualora ve ne siano, l'utente ha la possibilità di avviare una controversia ricorrendo a Justice. Nel caso in cui non è possibile risolvere la controversia entro il tempo T3, l'utente interessato ricorrerà al voto fisico, cancellando successivamente la propria associazione CF-PK_{USER}.

Successivamente a questa fase di controllo, le denunce relative ad un errato inserimento del CF-PK_{USER} non verranno prese in considerazione.

T3-T4

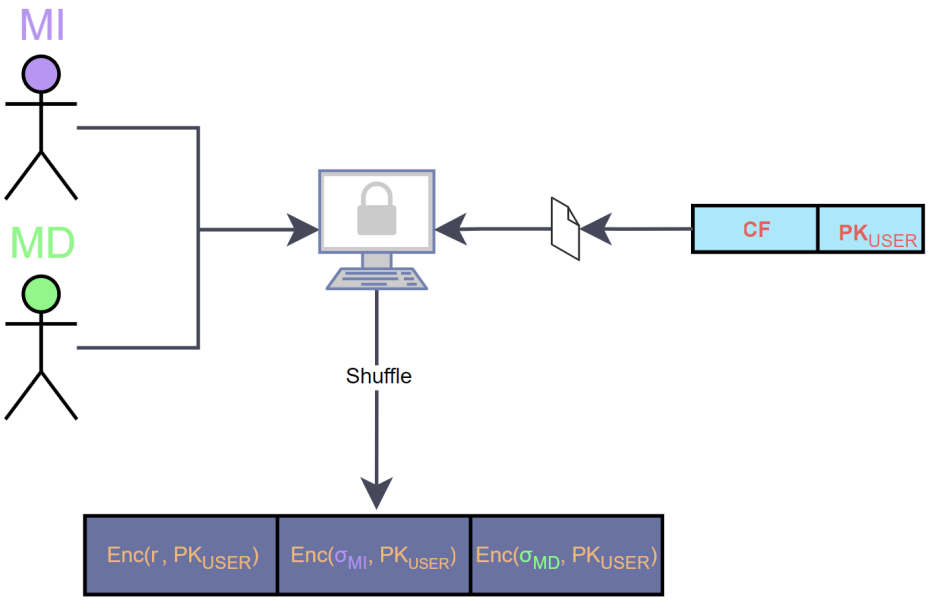
In questo lasso di tempo vengono generate le credenziali per ogni utente che ha effettuato le operazioni fino ad ora descritte. Queste credenziali saranno necessarie per poter accedere alla fase di votazione.

Consideriamo la presenza di:

- computer non collegato in rete, non attaccabile, in good shape, con OS corretto, al cui interno è presente il codice da eseguire;
- MI e MD che utilizzano e supervisionano il pc precedentemente descritto;
- file contenente tutte le transazioni della blockchain.

Codice open-source T3-T4: Richiede di inserire le chiavi segrete di MI e MD (SK_{MI}, SK_{MD}). Accede al file contenenti i CF e PK_{USER}. Per ogni CF, riconosce come valido l'ultimo PK_{USER} inserito.
Per ogni CF, genera una r di 256 bit random CS (cryptographically strong) e controlla che non esistano r duplicate. Genera le firme $\sigma_{MI}: Sig(SK_{MI}, r), \sigma_{MD}: Sig(SK_{MD}, r)$ seguendo lo schema di Schnoor per la firma digitale.
L'output consiste in una lista di:
Enc(r, PK_{USER}), Enc(σ_{MI}, PK_{USER}), Enc(σ_{MD}, PK_{USER}).
criptate tramite l'algoritmo di cifratura di ElGamal, su cui è stato effettuato uno shuffle.

MI e MD: Avviano il codice open-source sulla macchina. MI e MD lo controllano e ne controllano il software, verificando che il risultato dello SHA256 coincida con quello pubblicato sulla piattaforma web del MI. Inseriscono le proprie chiavi segrete. Da questo momento in poi questo computer viene lasciato senza poter essere manomesso da alcuna delle parti dato che l'altra funge da supervisore.
Generato l'output, lo pubblicano sulla piattaforma web di MI. Ad operazioni finite, la macchina verrà resettata completamente eliminandone tutti i dati.



T4-T5

Utente: si collega alla piattaforma web del MI. Accede al file contenenti
Enc(r, PK_{USER}), Enc(σ_{MI}, PK_{USER}), Enc(σ_{MD}, PK_{USER}) e con la sua SK_{USER} decripta tutto il file, pulendolo dai rand. Verifica le firme utilizzando gli algoritmi: Ver(PK_{MI}, r, σ_{MI}) e Ver(PK_{MD}, r, σ_{MD}). Riconosciuta la corrispondenza per cui entrambe le Ver vanno a buon fine, prende la tripla (r, σ_{MI}, σ_{MD}) come propria. Qualora vi siano incongruenze, l'utente ha la possibilità di avviare una controversia ricorrendo a Justice. Nel caso in cui non è possibile risolvere la controversia entro il tempo T5, l'utente interessato ricorrerà al voto fisico.

T5-T6

Ogni elettore esprime la sua preferenza. In questo lasso di tempo, è previsto, inoltre, che l'elettore possa cambiare la sua preferenza arbitrariamente fino al tempo T6. Alla fine del tempo T6, l'ultima preferenza espressa è l'unica e sola valida. Consideriamo la presenza della blockchain su cui viene inserito il voto espresso.

Utente: L'utente calcola un voto *v* utilizzando 8 bit per esprimere la propria preferenza. In particolare, gli 8 bit esprimono:

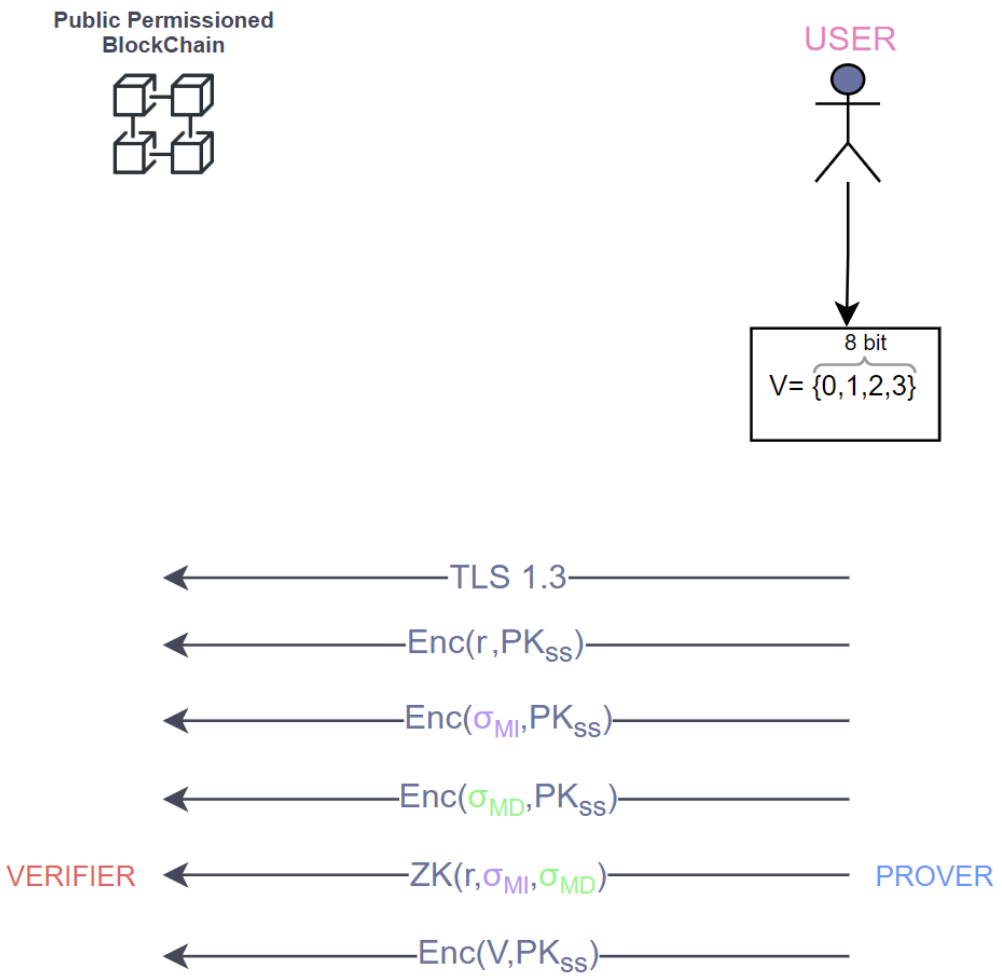
- il valore 0 il voto è annullato;
- il valore 1 il voto è per il primo candidato;
- il valore 2 il voto è per il secondo candidato;
- il valore 3 o valori differenti da quelli presentati fin ora, il voto è invalidato.

Stabilisce una comunicazione con la blockchain tramite canale protetto da TLS 1.3.
Invia alla blockchain la seguente quadrupla per l'esecuzione dello smart contract:
Enc(r, PK_{SS}), Enc(σ_{MI}, PK_{SS}), Enc(σ_{MD} , PK_{SS}), Enc(v, PK_{SS})
criptate tramite l'algoritmo di cifratura di ElGamal.

Rispetto alla blockchain, l'utente agirà da prover per la zero-knowledge proof sulla tripla Enc(r, PK_{SS}), Enc(σ_{MI}, PK_{SS}), Enc(σ_{MD} , PK_{SS}), provando che sia well-formed poiché costruita basandosi su una tripla (r, σ_{MI}, σ_{MD}) autentica.

Blockchain permissioned: Sulla blockchain è implementato uno smart contract che agisce come segue:

- Ricevuta la transazione, contenente la seguente quadrupla Enc(r, PK_{SS}), Enc(σ_{MI}, PK_{SS}), Enc(σ_{MD} , PK_{SS}), Enc(v, PK_{SS}), agirà da verifier per la zero-knowledge proof sulla tripla Enc(r , PK_{SS}), Enc(σ_{MI}, PK_{SS}), Enc(σ_{MD} , PK_{SS}), verificando che sia well-formed poiché costruita basandosi su una tripla (r, σ_{MI}, σ_{MD}) autentica.
- Se la ZKP non va a buon fine, la transazione viene rifiutata. In caso contrario, la transazione viene inserita all'interno della blockchain se il voto è well-formed, quindi la lunghezza è di 256 bit.



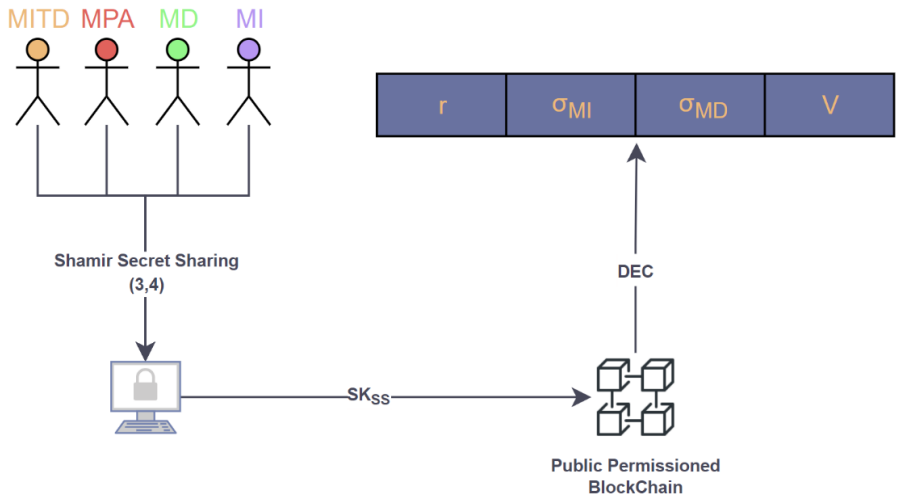
T6-T7

In questo istante di tempo, è prevista la fase di spoglio dei voti online.

Codice open-source T6-T7: Esegue l'accesso in lettura alla blockchain e legge tutto il suo contenuto. Richiede che siano inseriti almeno tre p(x_i), e le tre firme ad esse associate. Con queste firme il codice funge da verifier per la ZKP sulla coppia ($u^{p(x_i)}$, σ_i: $Sig(SK_{SS}, p(x_i))$ fornita da tutti e tre gli attori coinvolti che fungono da prover. La ZKP si occupa di verificare che la coppia sia well-formed poiché costruita basandosi su una coppia (p(x_i), σ_i) autentica, per cui Ver(PK_{SS}, p(x_i), σ_i) = 1. Nel caso in cui una ZKP fallisca, viene notificato a Justice per richiederne l'intervento nei confronti del cheater. In caso contrario,
il codice si occupa di verificare se ci sono Enc(r, PK_{SS}), Enc(σ_{MI}, PK_{SS}), Enc(σ_{MD} , PK_{SS}) ripetute in modo da considerare valida solo la prima entry associata ad esse eliminando le rimanenti.
Successivamente viene eseguito l'algoritmo di decifratura utilizzando la chiave ottenuta dalla ricostruzione del segreto (SK_{SS}) tramite la ricostruzione descritta da Shamir Secret Sharing:
Dec(Enc(r, PK_{SS}), SK_{SS}), Dec(Enc(σ_{MI} , PK_{SS}), SK_{SS}), Dec(Enc(σ_{MD} , PK_{SS}), SK_{SS}), Dec(Enc(v, PK_{SS}),SK_{SS}) decriptate tramite l'algoritmo di cifratura di ElGamal.
Per ogni *r* vengono conteggiati tutte le istanze di voto effettuate selezionando l'ultimo *v* come voto valido.
Le triple (*r*, *v*, *conteggio*) selezionate vengono salvate su un file in ordine casuale.
SK_{SS} non potrà mai essere visualizzato da nessuno degli attori coinvolti.

MITD, MI, MD, MPA: Avviano un codice su una macchina in good shape, con OS corretto, non connesso alla rete e che contiene in memoria unicamente il codice e la blockchain. Almeno tre degli attori coinvolti, controllano il software verificando che il risultato dello SHA256 coincida con quello pubblicato sulla piattaforma web del MI. All'avvio, ognuno di essi fungono da prover per la ZKP sulla coppia ($u^{p(x_i)}$,

σ_i: $Sig(SK_{SS}, p(x_i))$ fornita, verificando che sia well-formed poiché costruita basandosi su una coppia ($p(x_i)$, σ_i) autentica, per cui Ver(PK_{SS}, $p(x_i), \sigma_i$) = 1. Se le ZKP sono verificate, avviene la ricostruzione del segreto che permette di decifrare le informazioni presenti sulla blockchain.
Generato l'output, lo pubblicano sulla piattaforma web di MI. Ad operazioni finite, la macchina verrà resettata completamente eliminandone tutti i dati.



T7-T8

In questo istante di tempo è prevista la possibilità di votare fisicamente attraverso dei seggi.

Utente: Gli viene garantita la visione del file per verificare la presenza di incongruenze. Qualora ve ne siano, l'utente ha la possibilità di avviare una controversia ricorrendo a Justice e recarsi ai seggi per votare fisicamente.
In questo caso, se ha già richiesto le credenziali per il voto online, fornisce a Justice la tripla (r, σ_{MI}, σ_{MD}) e CF in modo da permettere che il suo voto online non venga conteggiato. Deve dimostrarne l'autenticità attraverso l'utilizzo del proprio PK_{USER} e SK_{USER} grazie alle quali è riuscito ad ottenere le credenziali di accesso.
Inoltre, un utente, che non ha credenziali SPID o non ha richiesto le credenziali al tempo T1-T2, può andare a votare fisicamente, fornendo il suo CF e tessera elettorale.

Justice: Genera due file: uno in cui sono presenti i CF di chi ha presentato la denuncia, utile per gli scrutinatori; un altro in cui è presente la r di chi ha presentato la denuncia, utile ai ministeri per effettuare l'annullamento dei voti contestati.

Scrutinatori: Si occupano di controllare se il CF dell'elettore, che ha intenzione di votare fisicamente, sia presente sul file fornito da Justice o non sia presente sulla blockchain in cui sono salvate le associazioni CF-PK_{USER}. Nel caso in cui fosse presente sulla blockchain ma non nel file fornito da Justice, è necessario presentare la denuncia prima di poter andare a votare fisicamente.

T8-T9

In questo tempo è previsto il conteggio dei voti online e fisico decretando il candidato vincente.

Codice open-source T8-T9: Esegue l'accesso in lettura al file pubblicato nel tempo T7-T8 e legge tutto il suo contenuto. Richiede l'inserimento di un file in cui sono presenti le r di tutte le persone che hanno sporto denuncia.
Delle triple (*r*, *v*, *conteggio*) le cui r non sono presenti nel file inserito, valuta il voto stabilendo se esso è associato al candidato 1, al candidato 2, ad una scheda bianca o ad una scheda nulla.
Di questi voti ne effettua il conteggio e genera in output un file contenente il risultato finale di quest'ultimo.

MI, MITD, Justice: Avviano un codice su una macchina in good shape, con OS corretto, non connesso alla rete e che contiene in memoria unicamente il codice. Gli attori coinvolti controllano il software verificando che il risultato dello SHA256 coincida con quello pubblicato sulla piattaforma web del MI.
Justice si occupa di inserire il file contente le r di tutte le persone che hanno sporto denuncia.
Generato l'output, lo pubblicano sulla piattaforma web di MI. Ad operazioni finite, la macchina verrà resettata completamente eliminandone tutti i dati.

Voto fisico: Gli scrutinatori, supervisionati da Justice, effettuano il conteggio delle schede elettorali, assicurandosi che il numero di schede sia uguale al numero di elettori registrati fisicamente.
Justice si occupa di conservare il risultato delle elezioni fisiche e salvarlo su file.

I risultati dell'elaborazione fisica e online, già salvati su file, sono messi sotto la supervisione di tutti sul sito del MI utilizzando un canale sicuro protetto dal TLS 1.3. Controversie per gli share o controversie per il risultato sia da parte dei candidati che da parte degli elettori sono successivamente gestiti dall'intervento sotto richiesta di Justice.

WP3

Completeness

Ispezionando i protocolli descritti nel work package precedente è possibile determinare con certezza che se tutti i player seguono il protocollo descritto non ci sono dispute, terminando il protocollo con la vittoria del candidato che ha ricevuto più consensi.

Justice può essere successivamente richiamata per dispute riguardanti lesioni al diritto al voto o per violazioni della privacy che sono mitigate dalla presenza più che soddisfacente di prove nel caso un player voglia agire in maniera disonesta.

Integrity

Discuteremo ora tutti gli avversari che minano tale proprietà e successivamente le loro aggregazioni.

Fake elector: Questo avversario si trova in gravi difficoltà dato che, per ottenere le credenziali di accesso alla sessione di voto, ha bisogno di ottenere la tripla (r, σ_{MI}, σ_{MD}). Quest'ultima è ottenibile solo dopo la verifica delle informazioni personali tramite l'accesso con SPID. L'MITD e MPA, infatti, controllano che il CF, comune di residenza e data di nascita, siano validi per permettere l'accesso alla votazione. Dato che l'avversario non ha capacità di controllo sul canale di comunicazione, protetto da TLS 1.3, non può manomettere i dati personali forniti all'MITD e all'MPA tramite SPID.
Un altro modo che avrebbe sarebbe quello di falsificare questa tripla ma questo significherebbe, per riduzione, riuscire a risolvere il logaritmo discreto dato che dovrebbe eseguire due firme digitali partendo unicamente dalle chiavi pubbliche rilasciate.
Questo avversario potrebbe pensare di mandare la quadrupla Enc(r , PK_{SS}), Enc(σ_{MI} ,PK_{SS}), Enc(σ_{MD} ,PK_{SS}), Enc(v, PK_{SS}) copiando quella presente sulla blockchain. In questo caso, viene considerato come voto valido solo il primo inserito con quell'Enc.
Potrebbe pensare di manomettere il server, inviando alla blockchain CF e PK_{USER} ma la blockchain non accetterebbe la transazione in quanto il CF non è presente nella lista dei votanti abilitati.
Il sistema garantisce la proprietà I.1 e I.4.

Mafia candidate: Questo avversario, per quanto può abusare della coercizione per far votare un elettore per lui, non ha modo di impedire all'elettore di cambiare la propria public key o il suo voto quando non sarà più presente per costringerlo.
Uno dei modi per mandare a segno questo attacco è costringere l'elettore a votare alla scadenza del periodo T6. L'attacco, per essere efficace, implicherebbe che l'avversario sia presente accanto al votante fino alla scadenza delle elezioni, limitando di molto la portata dell'attacco.
Nel caso in cui costringa l'elettore ad inserire una PK_{USER}, il Mafia Candidate potrebbe riconoscere che è stato effettuato un cambio di credenziali ma non avrebbe modo di impedirlo se non essere presente accanto al votante fino alla scadenza del tempo T2, riducendo la portata dell'attacco.
Un altro modo potrebbe essere quello di costringere l'elettore a fornirgli le credenziali per l'accesso alla votazione. Il mafia candidate, inoltre, potrebbe costringere l'utente a fornire la tripla (r, σ_{MI}, σ_{MD}) ma, non essendo in grado di decriptare la blockchain, non riuscirebbe ad avere nessuna informazione in merito al voto. Inoltre, gli utenti avrebbero la possibilità di rivolgersi a Justice e andare a votare fisicamente. Infatti, durante lo spoglio, anche se è visibile l'associazione r – voto, non è possibile sapere se il voto è stato

conteggiato in quanto il votante potrebbe anche essere andato a votare fisicamente. ll mafia candidate non ha accesso alle denunce presentate.
Il mafia candidate potrebbe bloccare all'elettore l'accesso al seggio per il voto fisico ma anche qui, dovrebbe essere presente accanto all'elettore fino alla fine del tempo T8, riducendo così la portata dell'attacco.
Nel caso in cui il mafia candidate costringesse l'utente ad agire per suo conto senza avere informazioni su credenziali quali PK_{USER}, SK_{USER} o tripla, il sistema di voto prevede che guardando la blockchain non è possibile distinguere i differenti rivoti.
Il sistema garantisce la I.2 e I.5 in quanto viene data la possibilità di denunciare, andare a votare fisicamente e la garanzia di dimostrare di essere il legittimo possessore delle credenziali.

Extremist voter: Questo avversario potrebbe provare a richiedere molteplici credenziali di voto. Per provare ad eseguire questo attacco, dovrebbe fare in modo che l'IdP fornisse CF sempre diversi. Dato che non ha potere sul canale di comunicazione tra IdP e MPA (che funge da Service Provider) questo attacco non è eseguibile.
Potrebbe pensare di accedere a SPID molteplici volte per farsi generare le credenziali di accesso. Non va a buon fine in quanto alla fine del tempo T2 non è più possibile inserire coppia CF – PK_{USER} e nella fase di rilascio credenziali viene considerato per ogni CF l'ultimo PK_{USER} inserito.
Inoltre, potrebbe pensare di provare ad esprimere molteplici volte il suo voto nell'arco di tempo che va da T5 a T6. La blockchain permetterebbe che, date le giuste credenziali, tutte le transazioni vengano accettate, ma il codice open source per il tempo T8-T9 mantiene unicamente un voto valido per ogni credenziale valida.
Il sistema garantisce la I.3.

Invalidator: Questo avversario non porta a termine l'attacco in quanto per invalidare il voto di un elettore dovrebbe fare un accesso tale da poter modificare la blockchain permissioned gestita da MITD e MI. Non è possibile eseguire questo attacco in quanto non possiede le risorse necessarie. Dato che le connessioni sono basate su TLS 1.3, gli risulterebbe complesso operare un attacco attivo.
L'unica strategia d'attacco valida potrebbe essere basata sulla possibilità di ottenere delle credenziali di un utente e votare in modo da annullare il suo voto. Tale attacco è prevenuto parzialmente dalla possibilità dell'utente di rivotare e controllare eventuali irregolarità sugli esiti della votazione online.
Il sistema garantisce la I.5 e I.2 permettendo all'utente di verificare sugli esiti della votazione online quante votazioni sono state eseguite utilizzando le sue credenziali.

Doppleganger: Questo avversario potenzialmente si può appropriare di diverse credenziali:

- SPID: potrebbe appropriarsi delle credenziali SPID (nome utente, password) dell'elettore. Per eseguire l'attacco, dovrebbe possedere anche il dispositivo o un modo per accedervi, in quanto SPID livello 2 o superiore prevede che su di esso ci sia un livello di sicurezza da superare (es: OTP, QR code, messaggio etc.). La riuscita dell'attacco comporterebbe la possibilità, non trascurabile, che al legittimo proprietario arrivi una notifica di avvenuto accesso, potendo chiedere l'intervento di Justice. Se anche questo deterrente venisse superato, il nostro sistema comunque tiene conto dell'associazione diretta tra CF e PK_{USER} sulla blockchain pubblica. Quindi, in caso di irregolarità, è possibile modificare il PK_{USER}.
- PK_{USER}, SK_{USER}: ottenendo queste credenziali, in particolare il SK_{USER}, avrebbe la possibilità di decriptare la tripla (r, σ_{MI}, σ_{MD}) dell'utente legittimo e votare al suo posto. L'utente potrebbe accorgersi di questo attacco nel momento in cui, accedendo agli esiti della votazione online, verifica che ci più transazioni di quelle aspettate. Ottenere le credenziali è complesso dato che sono dati sensibili che vengono utilizzati pochissime volte, per breve tempo e solo per questo ballottaggio. Una strategia di attacco potrebbe essere quella di manomettere il server, inserendo una PK_{USER} diversa rispetto a quella emessa dall'utente. Questo attacco non va a buon fine in quanto non solo la blockchain è pubblica, ma la firma risultante potrebbe non superare l'algoritmo di Verify. Nel caso

in cui succedesse, è possibile ricorrere a Justice per fare denuncia; di conseguenza, viene richiesta l'invalidazione dei voti presenti sulla blockchain associati alla tripla, dimostrando di esserne il legittimo proprietario tramite associazione PK_{USER} e CF.

- La tripla (r, σ_{MI}, σ_{MD}): grazie alla tripla, l'avversario potrebbe votare al posto dell'utente proprietario della tripla. L'utente potrebbe accorgersi di questo attacco nel momento in cui, accedendo agli esiti della votazione online, verifica che ci più transazioni di quelle aspettate. Entrare in possesso di queste credenziali, presuppone che l'utente le abbia salvate in chiaro, senza criptarle. É irragionevole pensarlo in quanto sono credenziali sensibili che vengono utilizzate pochissime volte, per breve tempo e solo per questo ballottaggio. Nel caso in cui succedesse, non avrebbe comunque la capacità di riconoscere quanti voti sono stati effettuati con quelle credenziali. Per andare a buon fine dovrebbe, inoltre, essere certo che il voto da lui inserito sia l'ultimo.
  Nel caso in cui succedesse, è possibile ricorrere a Justice per fare denuncia; di conseguenza, viene richiesta l'invalidazione dei voti presenti sulla blockchain associati alla tripla, dimostrando di esserne il legittimo proprietario tramite associazione PK_{USER} e CF.
  Infatti, durante lo spoglio, anche se è visibile l'associazione r – voto, non è possibile sapere se il voto è stato conteggiato in quanto il votante potrebbe anche essere andato a votare fisicamente. Infatti, il doppleganger non ha possibilità di presentare una denuncia in quanto la tripla è collegata indirettamente al CF del reale votante.
- Enc(r, PK_{SS}), Enc(σ_{MI},PK_{SS}), Enc(σ_{MD},PK_{SS}): grazie alla tripla, l'avversario potrebbe pensare di prenderle per votare al posto dell'utente che le ha generate. L'utente potrebbe accorgersi di questo attacco nel momento in cui, accedendo in lettura alla blockchain, verifica che ci sono transazioni aggiuntive con quella tripla. Anche se ottenesse queste credenziali, supererebbe la ZK e quindi avrebbe la possibilità di inserire il voto, ma in fase di spoglio il voto verrebbe scartato in quanto non è il primo voto associato a quella tripla.

Per tutte queste considerazioni, per quanto il sistema riesca a garantire la I.5 e I.2, in caso di furto di credenziali di larga portata, alla pari di scambio di persona, il sistema non è in grado di effettuare una distinzione se non viene effettuata la denuncia.

Corrupted Identity provider – block: Ha un solo modo di operare, ovvero nel momento in cui l'utente inserisce le credenziali SPID, riconosce che esse sono associate ad un utente dichiaratamente del partito avversario e gli nega l'accesso, notificandolo al Service Provider. In questo caso, l'utente o sporge un reclamo oppure si reca a votare fisicamente, annullando del tutto l'attacco. Essendo un attacco facilmente riconoscibile, il sistema garantisce la I.2 e I.5.

Corrupted Identity provider – exchange identity: Ha un solo modo di operare, ovvero fornire al Service Provider un CF non associato al reale utente che ha fatto accesso. In questo modo, possono verificarsi due situazioni:

1. l'utente è onesto e nel momento in cui viene richiesta la conferma dei dati, esso nega che quelle credenziali siano veritiere, annullando così l'attacco, con la conseguente registrazione sul log di failure che verrà consegnato a Justice. Questo implica un grande rischio per l'IdP;
2. l'utente è un attaccante passivo, accetta il CF non veritiero e inserisce un PK_{USER} proprio. Questo attacco ha grande possibilità di fallire in quanto il proprietario del CF ha l'obbligo di controllare se i dati sulla blockchain sono veritieri.

Il sistema garantisce la I.5.

Corrupted MI – allow scope: Ha tre possibilità:

- Impedire l'inserimento della coppia CF-PK_{USER} nella blockchain; questo attacco è irrealizzabile in quanto non solo l'utente ha la possibilità di visionare la blockchain e ricorrere a Justice, ma in più la governance della blockchain è anche in mano all'MITD.
- Impedire che la coppia CF-PK_{USER} generi la tripla utile in fase di votazione. Quest'attacco è irrealizzabile dato che la costruzione della tripla è presente su un PC non connesso alla rete, con codice open source e supervisionato anche dal MD. Inoltre, in fase di pubblicazione della blockchain l'utente può ricorrere a Justice se non riesce a identificare le sue credenziali.
- Potrebbe ricevere per vie esterne al sistema le triple associate alle persone dichiaratamente del partito avversario e non accettare l'inserimento del voto nella blockchain. Questo attacco è irrealizzabile in quanto non solo l'utente ha la possibilità di visionare la blockchain e ricorrere a Justice, in più la governance della blockchain è anche in mano all'MITD.

Corrupted MI – vote scope: Potrebbe modificare le transazioni in fase di votazione per modificare l'istanza di voto nella blockchain. Questo attacco è irrealizzabile in quanto non solo l'utente ha la possibilità di visionare la blockchain e ricorrere a Justice, in più la governance della blockchain è anche in mano all'MITD.

Corrupted MI – final result scope: Potrebbe voler manomettere l'esito delle votazioni. Potrebbe effettuare un attacco in due fasi:

- Durante la fase di spoglio, ma questo attacco è irrealizzabile in quanto sono presenti almeno altri due ministeri e l'operazione avviene su un PC non connesso alla rete con codice open source.
- Durante la fase di conteggio, ma questo attacco è irrealizzabile in quanto sono presenti Justice e un altro ministero e l'operazione avviene su un PC non connesso alla rete con codice open source.

Corrupted MI – service denial scope: Potrebbe voler invalidare tutta la sessione di voto. Può farlo in quattro metodologie:

- Non accettare o modificare le transazioni sulla blockchain in qualsiasi momento, quindi non permettere sia l'inserimento di CF-PK_{USER} sia l'inserimento dei voti. Questo comporterebbe una cattiva gestione della governance della blockchain permissioned, che verrebbe notata e notificata da MITD a Justice. Questo attacco non è realizzabile.
- Potrebbe inoltre in fase di rilascio credenziali, inserire una SIK errata in modo da generare firme non valide. Questo attacco è irrealizzabile in quanto è possibile effettuare l'algoritmo di Verify della firma tramite PK_{MI} pubblicato sulla piattaforma MI e quindi sarebbe facilmente riconoscibile dagli utenti e dal MD che lo supervisiona.
- Potrebbe in fase di spoglio fornire una share dello shamir secret sharing errato, ma questo attacco risulta inefficace in quanto è prevista la ZK. Inoltre, la shamir secret sharing è (3,4), quindi considera valida l'assenza di un ministero.
- Potrebbe effettuare modifiche sulla piattaforma web o renderla inaccessibile. In entrambi i casi, il tentativo di cheating viene identificato dagli altri ministeri e dagli utenti rendendo l'attacco non fattibile in quanto: in tutte le fasi in cui sussiste il caricamento di file o informazioni sulla piattaforma web, è presente almeno un altro ministero; inoltre, un caricamento errato viene identificato anche dagli utenti.

Corrupted MITD – allow scope: Ha quattro possibilità:

- Impedire l'inserimento della coppia CF-PK_{USER} nella blockchain; questo attacco è irrealizzabile in quanto non solo l'utente ha la possibilità di visionare la blockchain e ricorrere a Justice, in più la governance della blockchain è anche in mano all'MI.
- Potrebbe manomettere il server per non accettare cittadini con CF abilitati. Questo attacco è irrealizzabile in quanto il codice è controllato da MPA e non è possibile farne modifiche. Inoltre, l'utente riconoscerebbe che non gli viene garantita la possibilità di votare e può decidere di ricorrere a Justice ed effettuare il voto fisico.
- Potrebbe ricevere per vie esterne al sistema le triple associate alle persone dichiaratamente del partito avversario e non accettare l'inserimento del voto nella blockchain. Questo attacco è irrealizzabile in quanto non solo l'utente ha la possibilità di visionare la blockchain, in più la governance della blockchain è anche in mano all'MI.
- Modificare il CF dell'utente con uno di una persona che non vuoi che voti durante l'inserimento del PK_{USER} nel server. Questo attacco dovrebbe poter essere effettuato facendo visualizzare all'utente il suo CF per richiedere la conferma, inserendo però un CF non appartenente all'utente in questione. Questo attacco non va a buon fine in quanto i proprietari di entrambi i CF hanno l'obbligo di controllare se è i dati sulla blockchain sono veritieri. Inoltre, l'operazione è supervisionata anche da MPA, quindi, non può effettuare questa modifica localmente.

Corrupted MITD – vote scope: Potrebbe modificare le transazioni in fase di votazione per modificare l'istanza di voto nella blockchain. Questo attacco è irrealizzabile in quanto non solo l'utente ha la possibilità di visionare la blockchain e ricorrere a Justice, in più la governance della blockchain è anche in mano all'MI. Inotre, potrebbe modificare:

- PK_{USER}: potrebbe inserire una PK_{USER} non veritiera e generare sia una signature sbagliata, sia la signature corretta. Nel primo caso, l'utente se ne accorge tramite Verify e Blockchain; nel secondo caso, tramite Blockchain pubblicata.
- Aggiungere CF-PK_{USER}: potrebbe inserire una coppia non reclamata ma il proprietario del CF ha il dovere di controllare se i dati sono veritieri.
- Modificare CF-PK_{USER}

In tutti questi casi, è tutto pubblico, facilmente riconoscibile e l'utente può inserire più volte il suo $PK_{USER.}$

Corrupted MITD – final result scope: Potrebbe voler manomettere l'esito delle votazioni. Potrebbe effettuare un attacco in due fasi:

- Durante la fase di spoglio, ma questo attacco è irrealizzabile in quanto sono presenti almeno altri due ministeri e l'operazione avviene su un PC non connesso alla rete con codice open source.
- Durante la fase di conteggio, ma questo attacco è irrealizzabile in quanto sono presenti Justice e un altro ministero e l'operazione avviene su un PC non connesso alla rete con codice open source.

Corrupted MITD – service denial scope: Potrebbe voler invalidare tutta la sessione di voto. Può farlo in tre metodologie:

- Non accettare o modificare le transazioni sulla blockchain in qualsiasi momento, quindi non permettere sia l'inserimento di CF-PK_{USER} sia l'inserimento dei voti. Questo comporterebbe una cattiva gestione della governance della blockchain permissioned, che verrebbe notata e notificata da MI a Justice. Questo attacco non è realizzabile.
- Potrebbe voler manomettere il server rendendolo inutilizzabile. Non è possibile in questo caso stimare in che modalità viene effettuato e per quanto tempo viene reso inutilizzabile. Essendo un server che funge da filtro per l'identificazione dell'utente sulla blockchain, non prevede hardware dedicato ed è quindi semplice da sostituire.

- Potrebbe in fase di spoglio fornire una share dello shamir secret sharing errato, ma questo attacco risulta inefficace in quanto è prevista la ZK. Inoltre, la shamir secret sharing è (3,4), quindi considera valida l'assenza di un ministero.

Corrupted MPA – allow scope: Ha due possibilità:

- Potrebbe manomettere il server per non accettare cittadini con CF abilitati. Questo attacco è irrealizzabile in quanto il codice è controllato da MITD e non è possibile farne modifiche. Inoltre, l'utente riconoscerebbe che non gli viene garantita la possibilità di votare e può decidere di ricorrere a Justice ed effettuare il voto fisico.
- Modificare il CF dell'utente con uno di una persona che non vuole che voti durante l'inserimento del PK_{USER} nel server. Questo attacco dovrebbe poter essere effettuato facendo visualizzare all'utente il suo CF per richiedere la conferma, inserendo però un CF non appartenente all'utente in questione. Questo attacco non va a buon fine in quanto i proprietari di entrambi i CF hanno l'obbligo di controllare se è i dati sulla blockchain sono veritieri. Inoltre, l'operazione è supervisionata anche da MITD, quindi, non può effettuare questa modifica localmente.

Corrupted MPA – vote scope: Potrebbe pensare di modificare le transazioni in fase di votazione per modificare l'istanza di voto nella blockchain. Questo attacco è irrealizzabile in quanto non possiede la governance della blockchain.
Inoltre, potrebbe modificare:

- PK_{USER}: potrebbe inserire una PK_{USER} non veritiera e generare sia una signature sbagliata, sia la signature corretta. Nel primo caso, l'utente se ne accorge tramite Verify e Blockchain; nel secondo caso, tramite Blockchain pubblicata.
- Aggiungere CF-PK_{USER}: potrebbe inserire una coppia non reclamata ma il proprietario del CF ha il dovere di controllare se i dati sono veritieri.
- Modificare CF-PK_{USER}

In tutti questi casi, è tutto pubblico, facilmente riconoscibile e l'utente può inserire più volte il suo PK_{USER}.

Corrupted MPA – final result scope: Potrebbe effettuare un attacco durante la fase di spoglio, ma questo attacco è irrealizzabile in quanto sono presenti almeno altri due ministeri e l'operazione avviene su un PC non connesso alla rete con codice open source.

Corrupted MPA – service denial scope: Potrebbe voler invalidare tutta la sessione di voto. Può farlo in due metodologie:

- Potrebbe voler manomettere il server rendendolo inutilizzabile. Non è possibile in questo caso stimare in che modalità viene effettuato e per quanto tempo viene reso inutilizzabile. Essendo un server che funge da filtro per l'identificazione dell'utente sulla blockchain, non prevede hardware dedicato ed è quindi semplice da sostituire.
- Potrebbe in fase di spoglio fornire una share dello shamir secret sharing errato, ma questo attacco risulta inefficace in quanto è prevista la ZK. Inoltre, la shamir secret sharing è (3,4), quindi considera valida l'assenza di un ministero.

Corrupted MD – allow scope: Ha la possibilità di impedire che la coppia CF-PK_{USER} generi la tripla utile in fase di votazione. Quest'attacco è irrealizzabile dato che la costruzione della tripla è presente su un PC non connesso alla rete, con codice open source e supervisionato anche dal MI. Inoltre, in fase di pubblicazione della blockchain l'utente può ricorrere a Justice se non riesce a identificare le sue credenziali.

Corrupted MD – vote scope: Potrebbe pensare di creare credenziali aggiuntive per permettere l'inserimento o la modifica di un voto. Questo non è realizzabile in quanto l'operazione di rilascio credenziali avviene su un PC non connesso alla rete con codice open source supervisionato da MI.
Non può effettuare la cancellazione di un voto in quanto non possiede la governance della blockchain.

Corrupted MD – final result scope: Potrebbe effettuare un attacco durante la fase di spoglio, ma questo attacco è irrealizzabile in quanto sono presenti almeno altri due ministeri e l'operazione avviene su un PC non connesso alla rete con codice open source.

Corrupted MD – service denial scope: Potrebbe voler invalidare tutta la sessione di voto. Può farlo in due metodologie:

- Potrebbe in fase di rilascio credenziali, inserire una SIK errata in modo da generare firme non valide. Questo attacco è irrealizzabile in quanto è possibile effettuare l'algoritmo di Verify della firma tramite PK_{MI} pubblicato sulla piattaforma MI e quindi sarebbe facilmente riconoscibile dagli utenti e da MI che lo supervisiona.
- Potrebbe in fase di spoglio fornire una share dello shamir secret sharing errato, ma questo attacco risulta inefficace in quanto è prevista la ZK. Inoltre, la shamir secret sharing è (3,4), quindi considera valida l'assenza di un ministero.

Per tutti i threat model di ogni ministero il sistema rispetta le proprietà di integrità, a meno della I.4 in maniera parziale, a causa di possibili disservizi dovuti a manomissione del server.

Hacktivism: questo avversario ha due modi di agire:

- Attaccare il server di prenotazione delle credenziali, dato che hanno le risorse computazionali per poter effettuare un attacco DDoS, ma non possono bloccare in nessun caso la blockchain date le sue proprietà. Inoltre, attacchi sul server possono essere prevenuti o identificati tramite meccanismi di intrusion detection o firewall.
- Fornire falsa testimonianza a Justice per invalidare le elezioni affermando in massa di aver subito un furto di identità. In questo caso, se vengono effettuate le denunce prima del termine della fase di controllo, si prendono provvedimenti abilitando il voto fisico; altrimenti non vengono presi provvedimenti dato che è dovere degli utenti controllare prima della fine del periodo di controllo.

Scrutinatori: questo avversario potrebbe pensare di modificare il file consegnatogli da Justice per dare la possibilità ad alcuni CF, che hanno votato online, di votare fisicamente senza aver sporto denuncia. In questo modo non si terrebbe traccia della transazione online da cancellare e quindi i CF aggiunti avrebbero espresso due voti validi. Questo attacco è mitigato dalla presenza di Justice nei seggi fisici e sarebbe, inoltre, facilmente identificabile effettuando un controllo incrociato tra la blockchain con l'associazione CF-PK_{USER} e il file ufficiale di Justice.

Colluding cheaters:

Extremist voter/ Eavesdropper / Invalidator / Doppelganger con Fake elector: il fake elector come descritto precedentemente non ha alcun modo di poter portare a termine il suo attacco; quindi, non porterebbe nessun aiuto aggiuntivo ai precedenti cheater.

Eavesdropper e Extremist voter: entrambi potrebbero collaborare in quanto l'Extremist voter chiede all'Eavesdropper di verificare l'andamento delle elezioni. In questo modo, il suo interesse è quello di comprendere quanti voti deve inserire per far si che un candidato salga al potere. Nessuno dei due riescono ad assicurarsi che l'andamento delle votazioni non venga cambiato da una denuncia o richiesta di votazione fisica.

Invalidator / Doppelganger e Extremist voter: l'Extremist voter non ha interesse a collaborare con gli altri cheater, in quanto agiscono contro il suo obiettivo.

Invalidator / Doppelganger e Eavesdropper: potrebbero collaborare in quanto l'Eavesdropper per comprendere l'andamento delle elezioni si confronta con i cheater per comprendere quali sono i voti invalidati e quelli validi ma generati con furto di identità. L'Eavesdropper non riesce ad assicurarsi che l'andamento delle votazioni non venga cambiato da una denuncia o richiesta di votazione fisica. Per questo motivo la loro collaborazione potrebbe essere risultata futile.

Invalidator e Doppelganger: questo colluding potrebbe portare problematiche in quanto grazie alle credenziali fornite dal Doppelganger, non solo può essere espressa una preferenza diversa ma anche una creazione di voto invalido. Questo problema è mitigato dal fatto che il legittimo proprietario delle credenziali ha intervalli di tempo e mezzi per poter dimostrare un furto di identità e conseguente modifica del voto.

Hacktivism e MITD: Potrebbe voler compromettere la blockchain ma MI detiene l'altro 50% della governance quindi non può farlo. Potrebbe voler mettere fuori uso il server ma agisce soltanto sotto supervisione. Potrebbe non consegnare la share del segreto ma esso verrebbe ricostruito comunque, in quanto bastano 3 share.

Hacktivism e MI: Potrebbe voler compromettere la blockchain ma MITD detiene l'altro 50% della governance quindi non può farlo. Potrebbe non consegnare la share del segreto ma esso verrebbe ricostruito comunque, in quanto bastano 3 share. Potrebbe voler mettere fuori uso la piattaforma web ma sarebbe facilmente riconoscibile. Potrebbe non consegnare la sua vera firma ma sarebbe riconosciuto.

Hacktivism e MPA: Potrebbe non consegnare la share del segreto ma esso verrebbe ricostruito comunque, in quanto bastano 3 share. Potrebbe non consegnare la sua vera firma ma sarebbe riconosciuto. Potrebbe voler mettere fuori uso il server ma agisce soltanto sotto supervisione.

Hacktivism e MD: Potrebbe non consegnare la share del segreto ma esso verrebbe ricostruito comunque, in quanto bastano 3 share. Potrebbe non consegnare la sua vera firma ma sarebbe riconosciuto.

Hacktivism e IdP: Potrebbe falsificare i propri servizi ma sarebbe facilmente contestabile da MITD che potrebbe addirittura eliminarlo dall'elenco degli IdP autorizzati.

MITD e IdP: potrebbero collaborare rendendo sia server hostato da MPA che servizi di Idp inaccessibili. Questo potrebbe portare ad un disservizio temporaneo del nostro sistema. Questo lede la proprietà I.4. Inoltre, potrebbero collaborare per manomettere i dati che arrivano al server e alla blockchain inserendo dati errati o facendo visualizzare all'utente (in fase di conferma dei dati) dati errati. Questo attacco non va a buon fine data la trasparenza e il file di log di Justice.

Mafia candidate e Hacktivism e MITD e IdP: il mafia candidate che pensa di non vincere le elezioni, collabora con gli attivisti e con gli altri cheater per far in modo che non avvengano. Potrebbe portare a termine l'attacco ma come detto precedentemente solo per un tempo limitato, non tale da garantire il completo annullamento.

Mafia candidate e MITD/MI/MPA/MD: il mafia candidate collabora con uno dei quattro ministeri per provare a controllare l'andamento delle elezioni e costringerli ad inserire voti a suo favore. L'attacco non va a buon fine in quanto un ministero con ha le capacità di ledere il sistema lavorando in solitaria.

Considerare un colluding cheater tra due o più ministeri è improbabile.

Seppure tutte le proprietà di integrità sono rispettate dal nostro sistema, data la presenza dell'Idp e del server online non abbiamo potuto garantire un'integrità del 100%. Infatti, abbiamo considerato:

- l'eventualità di un disservizio da parte di un Idp che, nonostante non sia gestito da noi, andrebbe a negare il diritto al voto delle persone che lo utilizzano;
- l'eventualità di un disservizio da parte del server online, facilmente mitigabile dato che il server in questione è facilmente sostituibile.

La proprietà che viene parzialmente rispettata è la I.4.

Confidentiality

Eaves dropper: Questo avversario ha una sola opzione di attacco, ovvero leggere la blockchain criptata e decodificare il voto di più di circa metà delle transizioni presenti. Questo avversario si trova in difficolta per due motivi:

- Decifrare un singolo voto richiede la risoluzione dello schema di cifratura di ElGamal. Non ha la capacità necessaria per ottenere anche un singolo risultato entro la fine delle elezioni.
- Qualora riuscisse anche a decifrare i voti avrebbe problemi ad ottenere un risultato affidabile dato che gli utenti hanno la possibilità di rivotare e l'attaccante non potrebbe riconoscere se più voti sono stati effettuati dallo stesso votante. Inoltre, alcuni voti potrebbero essere falsi dato che potrebbero non essere conteggiati dopo la denuncia effettuata e la richiesta di voto fisico.

Corrupted MI – view scope: Potrebbe voler visualizzare la blockchain e capire la corrispondenza voto-persona. Questo attacco potrebbe essere attuato in cinque fasi:

- Nella fase di creazione della tripla, in cui le operazioni avvengono su un PC non connesso alla rete con codice open source supervisionato da MD. Questo attacco risulterebbe rischioso e facilmente identificabile. Inoltre, durante la creazione della tripla l'output viene shufflato in modo da non avere la corrispondenza diretta CF-PK_{USER}-tripla.
- Potrebbe ricevere per vie esterne al sistema le triple e deve richiede la risoluzione dello schema di cifratura di ElGamal per trovare la tripla e ottenere una corrispondenza CF-PK_{USER}-voto. Questo attacco è irrealizzabile in quanto non ha le capacità necessarie a risolvere questa problematica in tempo utile.
- Potrebbe pensare di visualizzare la transazione presente sulla blockchain nell'istante in cui una persona effettua il voto, ma in questo caso la transazione visualizzata sarebbe inconsistente in quanto al votante viene data la possibilità di rivotare e il voto successivo non è correlato a quello precedente. Questo attacco non ha una grande portata ed è molto rischioso in quanto dovrebbe essere logisticamente possibile costringere molte persone ad effettuare il voto nello stesso momento.
- Nel caso in cui il ministero ottenga da fonti esterne al sistema CF-PK_{USER}-SK_{USER}, potrebbe pensare di utilizzare le credenziali per effettuare un voto. Questo è ovviato dal fatto che l'utente può controllare quante transazioni sono state fatte utilizzando le sue credenziali e, eventualmente, andare a votare fisicamente.
- I nodi che si occupano della governance della blockchain potrebbero salvare indirizzi IP relativi a delle transazioni per poter associare un insieme di voti alla relativa zona di provenienza una volta effettuata la fase di spoglio dei voti. Questa non risulta essere una criticità in quanto SK_{SS} non viene visualizzato da nessuno degli attori coinvolti e, quindi, non è possibile decifrare nessuna transazione salvata.

Corrupted MITD – view scope: Potrebbe voler visualizzare la blockchain e capire la corrispondenza voto-persona. Questo attacco potrebbe essere attuato in quattro fasi:

- Potrebbe ricevere per vie esterne al sistema le triple e deve richiede la risoluzione dello schema di cifratura di ElGamal per trovare la tripla e ottenere una corrispondenza CF-PK_{USER}-voto. Questo attacco è irrealizzabile in quanto non ha le capacità necessarie a risolvere questa problematica in tempo utile.
- Potrebbe pensare di visualizzare la transazione presente sulla blockchain nell'istante in cui una persona effettua il voto, ma in questo caso la transazione visualizzata sarebbe inconsistente in quanto al votante viene data la possibilità di rivotare e il voto successivo non è correlato a quello precedente. Questo attacco non ha una grande portata ed è molto rischioso in quanto dovrebbe essere logisticamente possibile costringere molte persone ad effettuare il voto nello stesso momento.
- Nel caso in cui il ministero ottenga da fonti esterne al sistema CF-PK_{USER}-SK_{USER}, potrebbe pensare di utilizzare le credenziali per effettuare un voto. Questo è ovviato dal fatto che l'utente può controllare quante transazioni sono state fatte utilizzando le sue credenziali e, eventualmente, andare a votare fisicamente.
- I nodi che si occupano della governance della blockchain potrebbero salvare indirizzi IP relativi a delle transazioni per poter associare un insieme di voti alla relativa zona di provenienza una volta effettuata la fase di spoglio dei voti. Questa non risulta essere una criticità in quanto SK_{SS} non viene visualizzato da nessuno degli attori coinvolti e, quindi, non è possibile decifrare nessuna transazione salvata.

Corrupted MPA – view scope: Potrebbe voler visualizzare la blockchain e capire la corrispondenza voto-persona. Questo attacco potrebbe essere attuato in tre fasi:

- Potrebbe ricevere per vie esterne al sistema le triple e deve richiede la risoluzione dello schema di cifratura di ElGamal per trovare la tripla e ottenere una corrispondenza CF-PK_{USER}-voto. Questo attacco è irrealizzabile in quanto non ha le capacità necessarie a risolvere questa problematica in tempo utile.
- Potrebbe pensare di visualizzare la transazione presente sulla blockchain nell'istante in cui una persona effettua il voto, ma in questo caso la transazione visualizzata sarebbe inconsistente in quanto al votante viene data la possibilità di rivotare e il voto successivo non è immediatamente correlato a quello precedente. Questo attacco non ha una grande portata ed è molto rischioso in quanto dovrebbe essere logisticamente possibile costringere molte persone ad effettuare il voto nello stesso momento.
- Nel caso in cui il ministero ottenga da fonti esterne al sistema CF-PK_{USER}-SK_{USER}, potrebbe pensare di utilizzare le credenziali per effettuare un voto. Questo è ovviato dal fatto che l'utente può controllare quante transazioni sono state fatte utilizzando le sue credenziali e, eventualmente, andare a votare fisicamente.

Corrupted MD– view scope: Potrebbe voler visualizzare la blockchain e capire la corrispondenza voto-persona. Questo attacco potrebbe essere attuato in quattro fasi:

- Nella fase di creazione della tripla, in cui le operazioni avvengono su un PC non connesso alla rete con codice open source supervisionato da MI. Questo attacco risulterebbe rischioso e facilmente identificabile. Inoltre, durante la creazione della tripla l'output viene shufflato in modo da non avere la corrispondenza diretta CF-PK_{USER}-tripla.
- Potrebbe ricevere per vie esterne al sistema le triple e deve richiede la risoluzione dello schema di cifratura di ElGamal per trovare la tripla e ottenere una corrispondenza CF-PK_{USER}-voto. Questo attacco è irrealizzabile in quanto non ha le capacità necessarie a risolvere questa problematica in tempo utile.

- Potrebbe pensare di visualizzare la transazione presente sulla blockchain nell'istante in cui una persona effettua il voto, ma in questo caso la transazione visualizzata sarebbe inconsistente in quanto al votante viene data la possibilità di rivotare e il voto successivo non è immediatamente correlato a quello precedente. Questo attacco non ha una grande portata ed è molto rischioso in quanto dovrebbe essere logisticamente possibile costringere molte persone ad effettuare il voto nello stesso momento.
- Nel caso in cui il ministero ottenga da fonti esterne al sistema CF-PK_{USER}-SK_{USER}, potrebbe pensare di utilizzare le credenziali per effettuare un voto. Questo è ovviato dal fatto che l'utente può controllare quante transazioni sono state fatte utilizzando le sue credenziali e, eventualmente, andare a votare fisicamente.

In generale il nostro sistema garantisce un ottimo livello di confidenzialità.
Per garantire la massima trasparenza, però, è prevista la pubblicazione in chiaro del Codice Fiscale delle persone che hanno richiesto le credenziali di voto.
Ciò va a penalizzare la confidenzialità ma un qualsiasi utente, nonostante conosca chi ha richiesto le credenziali, non ha modo di avere ulteriore conoscenza relativa al voto e alla mancanza di votazione.
Un altro modo per ledere alla confidenzialità è l'appropriazione di SK_{USER} o di un elemento della tripla (r, σ_{MI}, σ_{MD}). Questa particolare situazione permette al sistema di garantire la P.1 solo fino alla fine della fase di votazione. In fase di spoglio e pubblicazione degli esiti della votazione online, la P.1 non è rispettata. La P.2, in qualunque caso, viene rispettata.

Efficiency

I Ministeri interessati non vengono coinvolti tutti contemporaneamente, se non nelle fasi cruciali di distribuzione del segreto e, conseguentemente, nella fase di ricostruzione del segreto che permette di decriptare i voti. Le operazioni che vengono effettuate che interessano l'efficienza sono:

1. Generazione delle credenziali a partire dalla coppia CF, PK_{USER}
2. Acquisizione delle credenziali da parte dell'utente
    - Questa fase richiede molta computazione da parte dell'utente, in quanto è previsto che l'utente provi a decriptare tutte le credenziali con la propria chiave privata e prenda soltanto quelle che verificano le firme.

In particolare, nel primo punto vengono generati per ogni entry CF-PK_{USER} 256 bit rand CS. Quindi, il periodo di generazione aumenta all'aumentare della popolosità dei comuni. Si può effettuare preventivamente una stima in base al numero di CF abilitati presenti nel file.

Nel secondo punto, l'operazione di Decodifica e Verify per ogni singola entry viene effettuata in tempo costante. Non essendo il numero di entry predeterminato, ma dipendente dal numero di votanti, per comuni più popolosi questa fase risulta più onerosa.

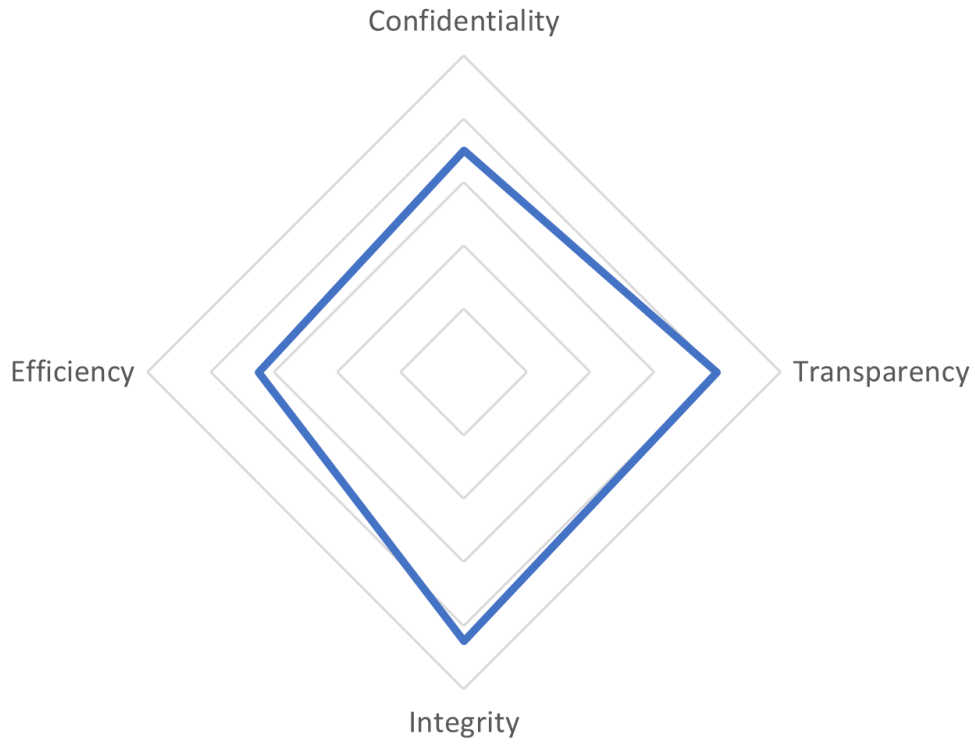
In compenso, lavorando con Blockchain permissioned le transazioni non richiedono proof of work e le operazioni di ZKP, CF abilitato e lunghezza del voto sono operazioni eseguite in tempo costante.

Transparency

Il nostro sistema prevede la pubblicazione di tutti i codici sorgenti dei programmi utilizzati e l'intera blockchain criptata.
In questo modo, chiunque può controllare tutte le operazioni che vengono effettuate durante le varie fasi.

Così facendo, abbiamo garantito un'alta trasparenza, caratteristica essenziale per un sistema di voto pubblico.

Tool usati in WP2	**Seguiti da**	**Usato per la proprietà**
Correlazione delle transazioni	Assunzioni su Blockchain	P.1, P.2
CPA-security dell'Enc	Assunzioni su ElGamal encryption	P.1, P.2, I.5
CCA- security dell'Enc	Assunzioni su ElGamal e TLS 1.3	P.1, P.2, I.5
Non-falsificazione della Sig	Assunzioni su Schnorr Sig	I.4, I.5
Minimal della Secret Sharing	Assunzioni su Shamir Secret Sharing	I.5
ZK della ZKP	Assunzioni su Zero-knowledge Proof	P.1, P.2, I.1, I.4, I.5
Completezza della ZKP	Assunzioni su Zero-knowledge Proof	P.1, P.2, I.1, I.4, I.5
Solidità della ZKP	Assunzioni su Zero-knowledge Proof	P.1, P.2, I.1, I.4, I.5
Uniformità CRHF	Assunzioni su SHA256	I.5



Analisi aggiuntive

Durante l'esposizione del funzionamento del nostro sistema abbiamo assunto che i dispositivi degli utenti siano liberi da malware di ogni tipo.

Questa assunzione è molto forte, dunque, di seguito analizzeremo cosa potrebbe accadere al nostro sistema se esposto a malware analizzando quelli più comuni:

- *Backdoors*: Questi malwares permettono ad un attaccante di bypassare l'autenticazione per accedere a un sistema andando a comprometterlo.
  Nel nostro caso un malware di questo tipo, se presente sulla macchina di un utente, potrebbe causare il furto delle credenziali dell'utente (se esso le mantiene salvate sulla macchina corrotta).
  Tale attacco è mitigato dalla possibilità di votare fisicamente.
- *Trojan horses*: Questi malwares si identificano come un programma legittimo, fornendo anche le funzionalità legittime, ma permette all'attaccante di ottenere elevati privilegi sul sistema.
  Un malware di questo tipo potrebbe infettare il server ma, dato che viene controllato il codice che gira sulla macchina, il rischio che possa venir infettato è minimo.
  Potrebbe infettare un dispositivo utente, rendendo semplice all'attaccante rubare le credenziali di voto ma tale attacco è mitigato dalla possibilità di votare fisicamente.
- *Worms*: Questo tipo di malware tipicamente infetta una rete di computer replicandosi al loro interno, inoltre, non richiede l'esecuzione di programmi per attivarsi.
  Potrebbe essere un avversario per la nostra blockchain, dato che, attaccando la rete di un ministero, potrebbe riuscire ad ottenere il 50% della governance.
  Questo attacco è mitigato dalla possibilità di poter sostituire le macchine infette nella blockchain. Inoltre, con solo il 50% della governance, l'attaccante non riuscirebbe a inserire voti o credenziali false.
- *Ransomware*: Questo tipo di malware blocca un computer o un dispositivo, tipicamente ne cripta i dati per permettere all'attaccante di chiedere un riscatto. Questo tipo di attacco non ha effetto sulla blockchain data la sua architettura ma potrebbe essere efficace sui dispositivi degli utenti. Questi potrebbero avere i propri dispositivi bloccati, non potendo procedere alla votazione online.
  Tale attacco è mitigato dalla possibilità di votare fisicamente o da un altro dispositivo.
- *Rootkits*: Questi malwares conferiscono all'attaccante privilegi di altissimo livello essendo installati nel BIOS della macchina.
  Nel nostro caso, si potrebbe attaccare con un malware di questo tipo le macchine degli utenti.
  Tale attacco è mitigato dalla possibilità di votare fisicamente o da un altro dispositivo.
- *Spyware*: Questo tipo di malware ha come scopo raccogliere informazione all'interno del dispositivo.
  Nel nostro caso, potrebbe essere presente all'interno del dispositivo di un utente permettendo all'attaccante di conoscere l'esito del voto dell'utente e le sue credenziali. L'attaccante però non è incentivato a modificare il voto dato che l'utente, rendendosene conto, potrebbe richiedere l'intervento di Justice e votare fisicamente.

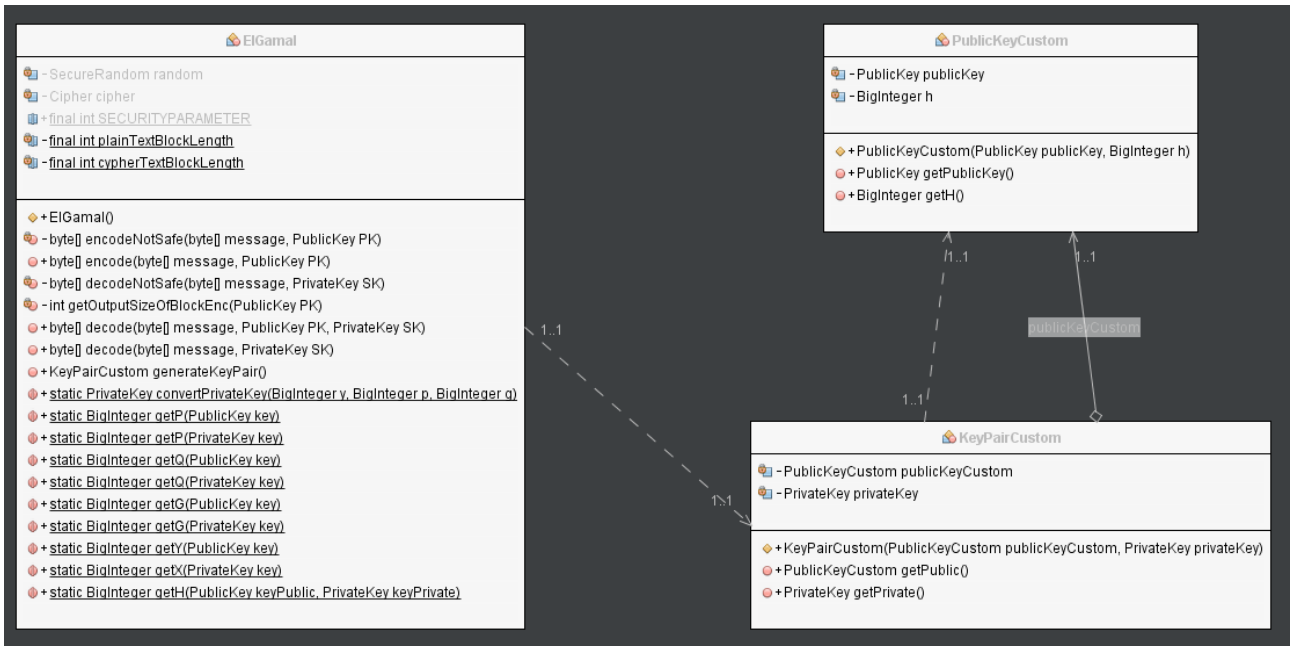
WP4

Durante lo sviluppo del sistema abbiamo adoperato due macro-semplificazioni:

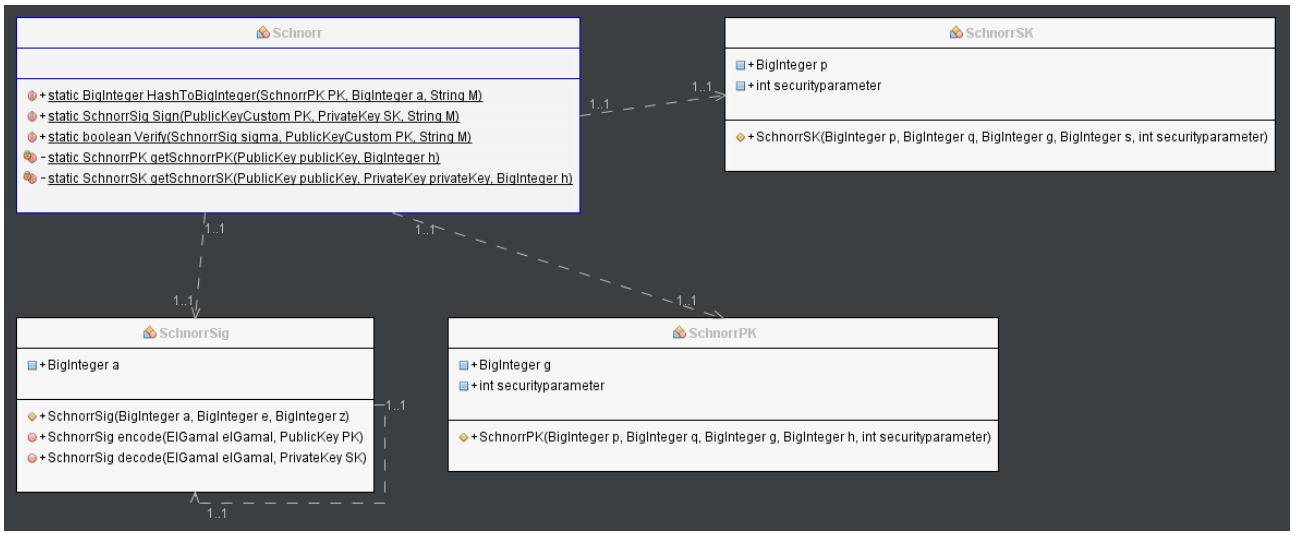
- Blockchain: Dato che non era possibile sviluppare una blockchain permissioned, è stata simulata nel codice utilizzando un semplice file contenente i dati inviati come descritti nel WP2.
- Identity Provider: Dato che non era possibile simulare fedelmente un Identity Provider abbiamo eliminato il suo coinvolgimento nell'implementazione.

Per avere ulteriore chiarezza, consultare il file README.md.

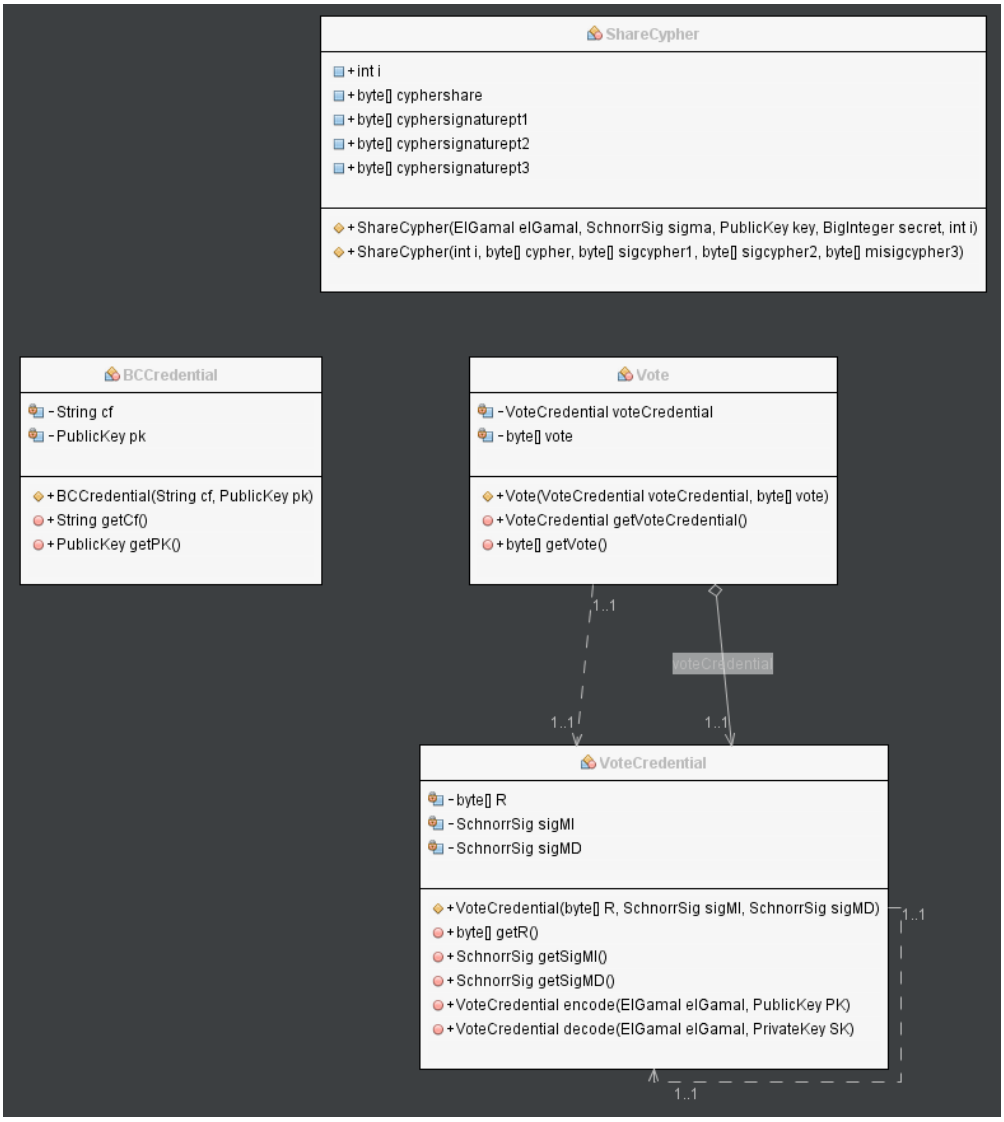
Riportiamo qui gli UML dei package messi a disposizione:



Package: ElGamal



Package: Schnorr



Package: Primitives