Università degli Studi di Salerno

.DIEM

Corso di Laurea Magistrale in Ingegneria Informatica

APS 2022/2023

Project Work – I guardiani del bit

Iannaccone Martina (0622702102)

Minichiello Giulia (0622702127)

Ricciardi Andrea Vincenzo (0622702009)

WP1. Modello

Workpackage	Task	Responsabile
WP1	Modello	Cognome1 Nome1

Durante la pandemia, il proprietario di una Sala Bingo, Mister Joker, ha deciso di creare delle stanze virtuali online in cui le persone possono partecipare a vari giochi di fortuna, ossia giochi dove sulla base di valori casuali c'è un vincitore. Per realizzare questa idea, contatta un gruppo di studenti del corso di laurea magistrale in ingegneria informatica dell'università di Salerno per creare una funzionalità che generi in modo casuale **stringhe continue**. L'obiettivo è creare un sistema trasparente che eviti imbrogli.

Il governo ha imposto il divieto di partecipare ad eventi sociali online a chi non possiede il Green Pass. Tuttavia, il garante per la protezione dei dati personali ha vietato l'invio del Green Pass attraverso canali telematici. Il governo ha quindi pubblicato una call aperta a tutti per proposte di formato del Green Pass 2.0 che preveda l'unico uso delle informazioni strettamente necessarie per accedere a un servizio. Mister Joker ha chiesto agli studenti di creare una funzionalità per l'accesso alle sale virtuali della sua Sala Bingo, anche se non sa ancora quali informazioni saranno necessarie nel **Green Pass 2.0**. Tuttavia, questo non è un problema, perché gli studenti intendo progettare un sistema dinamico che permette al proprietario del Green Pass 2.0 di usarlo in sicurezza al variare delle politiche nel tempo circa quali dati bisogna possedere per accedere al servizio. Il Green Pass 2.0 progettato dagli studenti viene emesso dal Ministero della Salute e permette agli utenti di identificarsi con la sala Bingo per accedere al proprio profilo e, successivamente, alla funzionalità di generazione continua di stringhe casuali.

Ci sono oppositori dell'innovazione chiamati *tecnocrati* che puntano sui rischi delle nuove tecnologie e sono considerati allarmisti e complottisti dai sostenitori dell'innovazione. Per evitare la strumentalizzazione, è necessario che il sistema sia trasparente e aperto alla verifica da parte di tutti. Mister Joker ha sottolineato l'importanza di questa trasparenza, citando una canzone di Franco Califano che diceva: "Non mi fido di nessuno".

1.1 Completeness

La procedura di gioco nella sala bingo virtuale è divisa nelle seguenti fasi:

1. La fase $[\boldsymbol{T_0}, \boldsymbol{T_1}]$ rappresenta la fase di pre-gioco, nella quale è prevista l'identificazione del giocatore presso l'Identity Provider della Sala Bingo di Mr. Joker.
2. La fase $[\boldsymbol{T_1}, \boldsymbol{T_2}]$ rappresenta la fase di gioco, nella quale l'utente effettua una giocata e viene verificata la corrispondenza a video.
3. La fase $[\boldsymbol{T_2}, \boldsymbol{T_3}]$ rappresenta la fase di post-gioco, in cui viene mostrato a video le giocate degli eventuali altri giocatori e successivamente il risultato elaborato dal sistema.

Formalizzando, siano $\boldsymbol{C} = \{\boldsymbol{C_1}, \dots \boldsymbol{C_n}\}$ l'insieme dei cittadini in possesso di un Green Pass 2.0 e sia $\boldsymbol{P} = \{\boldsymbol{P_1}, \dots, \boldsymbol{P_m}\} \subseteq \boldsymbol{C}$ l'insieme dei giocatori in possesso dei requisiti necessari per essere ammessi alla sala bingo virtuale. In altre parole, si assume che ogni giocatore P_i sia in possesso di un Green Pass 2.0 valido rilasciato, dopo opportune verifiche, da un'entità governativa di fiducia, quale il **Ministero della Sanità**. Ogni giocatore P_i avrà a disposizione una finestra temporale $[\boldsymbol{T_0}, \boldsymbol{T_1}]$ entro cui identificarsi con il Green Pass 2.0.

Una volta autenticato sul portale, il giocatore $P_i \in P$ può effettuare, durante la finestra temporale $[\boldsymbol{T_1}, \boldsymbol{T_2}]$, al più una giocata $\boldsymbol{B_{P_i}}$. Nel caso in cui uno o più giocatori non effettuino una giocata entro una finestra temporale specificata da Mr. Joker nella fase di definizione dei requisiti, verranno esclusi dall'attuale turno di gioco. Si assume che la finestra temporale abbia una durata di $30s$.

Infine, durante l'intervallo di tempo $[\boldsymbol{T_2}, \boldsymbol{T_3}]$, dopo che il sistema ha elaborato il risultato, il risultato stesso viene mostrato a ciascun giocatore sullo schermo insieme all'esito della giocata ($Vincente$ o $Perdente$) elaborata direttamente dal software di gioco. Inoltre, per favorire la trasparenza del sistema, sono rese pubbliche a ogni giocatore le giocate effettuate dagli altri giocatori della sala, in modo da poter verificare successivamente la correttezza del risultato [1].

[1] Come da specifiche, tutti gli aspetti legati alla componente economica (e.g., importo scommesso, eventuali vincite, etc.) sono stati esclusi dalla trattazione.

In caso di provi concrete di brogli nel sistema di gioco, si può ricorrere alla giustizia J. Anche se il sistema è stato implementato in modo da essere il più trasparente possibile, esiste sempre la possibilità, anche minima, che il sistema possa essere violato. Tuttavia, gli aspetti relativi alle possibili conseguenze penali sono state escluse dalla seguente trattazione.

Gli attori coinvolti nel sistema di gioco sono i seguenti [2]:

1. ***m*-Giocatori**, indicati con la notazione $\boldsymbol{P = \{P_1, \dots, P_m\}}$. Essi sono ritenuti poco affidabili, in quanto potrebbero tentare di manipolare il sistema in modo da ottenere vantaggi illegittimi. Per accedere alla generazione di stringhe casuali (i.e., partecipare ad un gioco della sala bingo virtuale), devono essere in possesso di un Green Pass 2.0 valido. Anche se essi potrebbero trarre vantaggio dal compromettere il sistema, se scoperti, affronterebbero gravi conseguenze penali da parte della giustizia J.

2. **Mr. Joker**, che per semplicità indicheremo con la notazione $\boldsymbol{Mr\ Joker}$. Questo attore è considerato essere affidabile al 70%, ma non gode di una fiducia totale da parte dei giocatori P. Compromettere il sistema sarebbe estremamente rischioso per la sua reputazione e la sua carriera professionale. Oltre alle potenziali perdite di fiducia, affronterebbe gravissime conseguenze penali da parte del sistema giudiziario J.

3. L'**Agenzia delle Dogane e dei Monopoli**, indicata con la notazione $\boldsymbol{ADM}$. Essa è l'entità governativa che supervisiona e governa la blockchain su cui potrebbe essere implementato il sistema di gioco. Essendo un'entità governativa, è considerata come affidabile al 99%. La sua presenza garantisce l'integrità e la trasparenza del sistema di gioco.

[2] Come attori del sistema, è possibile anche considerare gli studenti vincitori della competizione del Green Pass 2.0 (i.e., i Guardiani del Bit), i quali sono responsabili della risoluzione di eventuali problemi tecnici che potrebbero sorgere nel sistema da loro implementato. La loro integrità e competenza sono fondamentali per garantire il corretto funzionamento del sistema.

1.2 Threat Model

In una sala bingo virtuale possiamo identificare vari avversari, che potrebbero essere intenzionati ad attaccare il sistema. Di seguito, vengono riportati gli avversari individuati, distinguendoli in **avversari attivi** (D, *Dishonest*) e in **avversari passivi** (SH, *Semi-Honest*).

1. **The Trickster (D)**: è un soggetto in grado di alterare la propria giocata in modo da risultare vincitore, sebbene la giocata effettiva abbia avuto esito negativo. Si presume che abbia una capacità di elaborazione superiore rispetto a un utente comune, ma è comunque limitato dalle risorse economiche di un singolo individuo e dalle limitazioni delle architetture classiche.

2. **The Insider Trader (SH)**: è un soggetto interno all'organizzazione della sala bingo virtuale che sfrutta il proprio accesso privilegiato per poter ottenere informazioni sui partecipanti, sconosciute alle entità esterne al sistema. Il suo obiettivo è favorire un giocatore specifico fornendo informazioni altamente sensibili. È importante notare che l'Insider Trader opera tramite dispositivi aziendali, il che significa che potrebbe non avere una potenza di calcolo elevata rispetto ad avversari esterni che possono utilizzare risorse hardware più avanzate.

3. **The Meddler (SH)**: è un avversario interessato ad ottenere informazioni sui partecipanti e sulle loro attività di gioco. Si presume che abbia una capacità di elaborazione superiore rispetto a un utente comune, ma è comunque limitato dalle risorse economiche di un singolo individuo e dalle limitazioni delle architetture classiche.

4. **The No-Vax (D):** un avversario che vorrebbe essere ammesso alla Sala Bingo pur non essendo legittimato ad esserlo, in quanto sprovvisto di Green Pass 2.0. Il suo scopo è di convincere, e quindi eludere, il sistema di essere in possesso di un Green Pass 2.0 valido. Si presume che abbia una capacità di elaborazione superiore rispetto a un utente comune, ma

è comunque limitato dalle risorse economiche di un singolo individuo e dalle limitazioni delle architetture classiche.

5. **The Thief (D):** un avversario interessato a rubare le credenziali di accesso (Green Pass 2.0) di un partecipante. Si presume che abbia una capacità di elaborazione superiore rispetto a un utente comune, ma è comunque limitato dalle risorse economiche di un singolo individuo e dalle limitazioni delle architetture classiche.

6. **The Malicious Programmer (D)**: è una persona con una vasta conoscenza ed esperienza nei linguaggi di programmazione che gli permettono di compromettere i sistemi informatici, in modo da creare vantaggi impropri per sé o per altri. Si assume che tale avversario disponga di un'enorme potenza computazionale.

7. **The Overloaders (D)**: gruppo di avversari intenzionati a sovraccaricare il sistema durante la fase di sottomissione della giocata effettuando numerose giocate in successione, in modo da impedire agli altri partecipanti di giocare, rendendo il servizio inutilizzabile. Si presume che la loro capacità di elaborazione sia superiore rispetto a quella di un utente comune, in quanto potrebbero essere in grado di combinare più architetture di calcolo, ottenendo così una potenza di calcolo estremamente elevata. Inoltre, hanno a disposizione considerevoli risorse economiche per sostenere tali attività.

8. **The Bingo Bandits (D)**: gruppi di avversari di cui sopra, intenzionati a collaborare per arrivare ad un obiettivo comune, ossia trarre profitti. Si presume che la loro capacità di elaborazione sia superiore rispetto a quella di un utente comune, in quanto potrebbero essere in grado di combinare più architetture di calcolo, ottenendo così una potenza di calcolo estremamente elevata. Inoltre, hanno a disposizione considerevoli risorse economiche per sostenere tali attività.

1.3 Proprietà

Per i quattro pilastri fondamentali considerati (**confidenzialità**, **integrità**, **trasparenza** ed **efficienza**), vengono elencate le funzionalità che devono essere preservate in presenza di attacchi.

1.3.1 Confidenzialità

In presenza di avversari, il sistema di casinò dovrebbe essere *confidenziale*:

- C1. **Informazioni sanitarie dell'utente**: Il sistema deve garantire la protezione delle informazioni sanitarie dell'utente. Un avversario non dovrebbe essere in grado di determinare le ragioni alla base del rilascio del GP 2.0 a un determinato giocatore P_i.
- C2. **Credenziali d'accesso:** Le credenziali di accesso di un giocatore P_i devono essere completamente segrete.
- C3. **Identità Nascosta:** L'identità di ogni giocatori P_i presenti in una stanza di gioco deve rimanere nascosta a tutti gli altri giocatori P.

1.3.2 Integrità

In presenza di avversari, il sistema di casinò dovrebbe rimanere *integro*:

- I1. **Stabilità del sistema**: Il sistema deve essere ben progettato in modo che un attaccante non possa manipolare la correttezza del risultato generato dal banco B. Ciò è essenziale per garantire che il gioco sia equo e trasparente per tutti i partecipanti.
- I2. **Continuità del servizio**: Il sistema deve garantire la continuità del servizio, evitando interruzioni o downtime che potrebbero causare danni ai partecipanti o al gioco stesso.
- I3. **Idoneità al gioco**: Il processo di generazione di stringhe casuali deve essere garantito solo a coloro in possesso di un GP 2.0 valido.

I4. **Unicità della giocata**: Il sistema deve garantire ai giocatori P che il proprio valore immesso sia integro per tutta la durata della giocata. Ciò significa che per nessun soggetto coinvolto nel sistema deve essere possibile, a partire da un valore lecitamente immesso nel sistema, forgiare un nuovo valore che appare come valido.

1.3.3 Trasparenza

Il sistema di casinò dovrebbe essere *trasparente*, ovvero:

T1. **Algoritmi segreti**: Non dovrebbe essere basato su algoritmi segreti.

T2. **Fiducia Cieca**: Per evitare facili strumentalizzazioni da parte dei *no-fox*, il sistema non dovrebbe affidarsi eccessivamente ad una presunta parte fidata.

T3. **Correttezza delle giocate**: Qualunque soggetto, compreso un giocatore P_i passivo, può verificare, al termine della fase di post-gioco, il risultato prodotto dal sistema e le giocate B_{P_i} dei vari giocatori della stanza.

1.3.4 Efficienza

Il sistema di casinò dovrebbe essere *efficiente*, ovvero:

E1. **Elevato numero di partecipanti**: Il sistema deve assicurare un funzionamento efficiente in caso di un elevato numero di partecipanti.

WP2. Soluzione

Workpackage	Task	Responsabile
WP2	Soluzione	Ricciardi Andrea Vincenzo

Per garantire il corretto funzionamento del sistema, si considerano valide le seguenti assunzioni:

- I dispositivi non sono corrotti, ovvero funzionano correttamente senza essere controllati da eventuali malware e virus.
- Ogni stanza di gioco può ospitare un numero limitato di giocatori G_i, pari all'ordine delle decine.
- Per semplicità ogni volta che è coinvolto un certificato si fa effettivamente riferimento all'elenco concatenato dei certificati fino alla radice (cioè l'autorità di certificazione). In particolare, viene considerata come Root Certification Authority la **Repubblica Italiana**, seguita da due Intermediate CA: la Certification Authority del **Ministero della Salute** per il rilascio del $GP\,2.0$ e la Certification Authority del **Ministero dell'Economia e delle Finanze** per il rilascio del certificato del banco B e dei nodi ADM.
- La privacy e la sicurezza dei dati per le comunicazioni su Internet tra un client e un server, ogni volta che il client si connette al sito utilizzando una connessione HTTPS, viene fornita dal protocollo **TLS**, che crea un canale sicuro tra i due endpoints e permette di identificarne le parti.

Il capitolo in questione è strutturato in quattro paragrafi, ognuno dei quali presenta una descrizione dettagliata dell'argomento trattato, seguita da una fase di formalizzazione. I paragrafi in questione sono:

- **Green Pass 2.0**: include la Fase di Pre-Gioco.
- **Generazione Stringhe Casuali**: include le Fasi di Gioco e di Post-Gioco.
- **JokerChain**: include un'implementazione di *blockchain permissioned* per la gestione delle Fasi di Gioco e di Post-Gioco.

2.1 Green Pass 2.0

Il Green Pass, per semplicità abbreviato come GP, è un certificato digitale rilasciato da una figura considerata affidabile da tutti i cittadini, quale il **Ministero della Salute**, MS. Per rilasciare un GP, il MS genera una coppia di chiavi (p_{k_M}, s_{k_M}), dove la chiave pubblica p_{k_M} è utilizzata per poter effettuare la verifica del GP del cittadino da chiunque, mentre la chiave privata s_{k_M} è utilizzata per firmare la richiesta di certificato caricata dall'utente.

Un possibile scenario per il rilascio del GP può essere descritto come segue:

1. **Procedura di Validazione Sanitaria**. Per ottenere il GP, il cittadino CT_i deve recarsi presso un laboratorio medico accreditato per ricevere il vaccino o per effettuare un tampone. Durante la visita, il cittadino dovrà identificarsi esibendo al personale un documento di validazione (e.g., patente, etc.). Raccolti i dati del cittadino CT_i, il personale addetto li inserisce nella Piattaforma del MS tramite una connessione HTTPS, insieme all'esito del tampone o al certificato di avvenuta vaccinazione.
2. **Elaborazione del GP**. Inseriti i dati, il MS invia al cittadino CT_i un AUTHCODE come prova dell'autorizzazione a ottenere il rilascio del GP. Il cittadino CT_i accede alla pagina web del MS tramite una connessione HTTPS e dopo aver effettuato l'accesso, inserendo nel form l'AUTHCODE e le ultime 8 cifre della tessera sanitaria, genera una coppia di chiavi $(p_{k_{CT}}, s_{k_{CT}})$, prima della creazione di una richiesta di certificato (CSR). Essa contiene la chiave pubblica pk_{CT} e le informazioni che identificano il cittadino CT_i e dovrà essere opportunamente firmata usando la chiave privata $s_{k_{CT}}$.
3. **Rilascio del GP**. Ricevuta la richiesta di certificato, il MS provvederà a firmare il certificato con la sua chiave privata s_{k_M} e quindi al rilascio del GP valido, che sarà scaricabile dal cittadino CT_i nella sua area privata del sito del MS. Allo stesso tempo, il MS renderà pubblica la sua chiave pubblica in modo che chiunque possa verificare la validità del GP di CT_i.

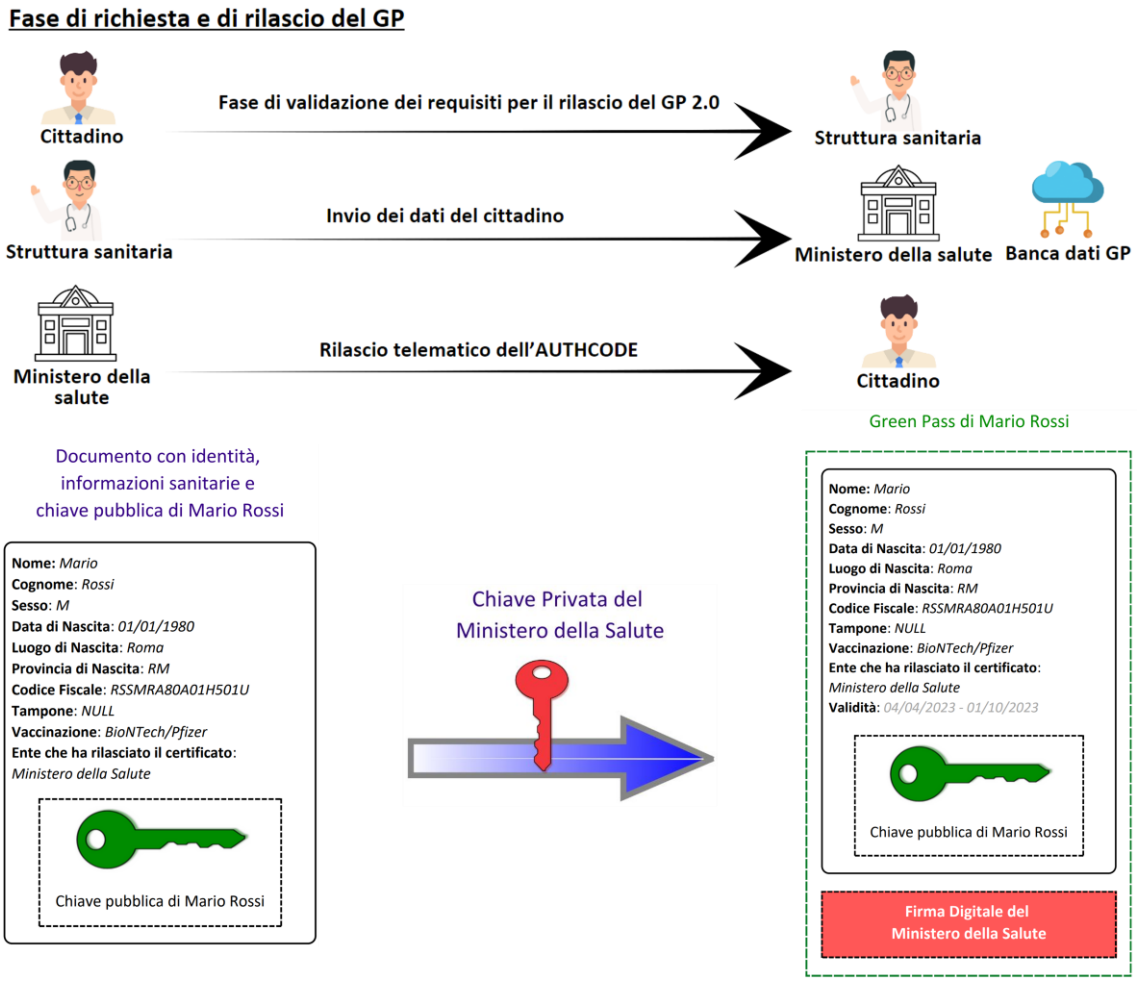

Fig. 2.1 – Possibile scenario per il rilascio del GP. Per semplicità è stato assunto che, indipendentemente dal modo in cui il GP venga ottenuto, tramite vaccinazione o tampone, esista un unico GP contenente le seguente informazioni:

- **Informazioni Personali**: indicano le informazioni personali del cittadino possessore del GP, quali nome, cognome, sesso, data, luogo e provincia di nascita, e codice fiscale.
- **Tampone**: indica il tipo di tampone effettuato: Molecolare o Rapido. Nel caso in cui il GP venga emesso in seguito a una vaccinazione, il campo rimarrà vuoto (NULL).
- **Vaccino**: indica il tipo di vaccino somministrato. Se il GP viene rilasciato a seguito di un tampone, il campo Vaccino rimarrà vuoto (NULL). Invece, nel caso in cui il cittadino, precedentemente positivo al virus e già vaccinato, risulti negativo, verrà generato un nuovo GP valido a partire dalla data dell'ultima somministrazione del vaccino.
- **Validità**: indica il periodo di validazione del GP. Quest'ultimo varia a seconda di come il GP viene ottenuto.
    - Nel caso di un Tampone Rapido, il GP avrà una durata di 48 ore dal rilascio.
    - Nel caso di un Tampone Molecolare, il GP avrà una durata di 72 ore dal rilascio.
    - Nel caso di Vaccino, il GP avrà una durata di 180 giorni dal rilascio.
- **Ente che ha rilasciato la certificazione**: indica l'ente affidabile che ha rilasciato il GP.
- **Identificativo Univoco del Certificato**: indica l'identificativo numerico della chiave pubblica del cittadino.

Formalizzazione

Di seguito, viene formalizzato il processo di generazione della richiesta del GP_{CT_i} da parte del cittadino CT_i e del rilascio dello stesso da parte del MS, dopo aver effettuato l'accesso al sito del MS tramite una connessione HTTPS.

1. Il cittadino CT_i genera una coppia di chiavi pubblica e privata

$$\left(\boldsymbol{p_{k_{CT_i}}} = \langle G, p, q, g, y = g^x \bmod p\rangle, \boldsymbol{s_{k_{CT_i}}} = x\right) \leftarrow \boldsymbol{Gen}(\mathbf{1}^{\boldsymbol{n}}) \qquad x \in \mathcal{Z}^q$$

   e crea, poi, una richiesta di certificato CSR_{CT} contenente le informazioni che identificano il cittadino CT_i e la chiave pubblica $p_{k_{CT}}$. Tale certificato viene firmato usando la chiave privata del cittadino CT_i:

$$\boldsymbol{\sigma} \leftarrow \boldsymbol{Sign}_{s_{k_{CT_i}}}\left(\boldsymbol{CSR}_{\boldsymbol{CT_i}}\right)$$

2. Il MS verifica tramite la firma del cittadino CT_i la veridicità della chiave pubblica $p_{k_{CT}}$, ossia $\boldsymbol{Vrfy}_{p_{k_{CT_i}}}(\boldsymbol{CSR}_{\boldsymbol{CT}}, \boldsymbol{\sigma}) \stackrel{?}{=} \mathbf{1}$.

3. Una volta verificata, il MS genera una coppia di chiavi pubblica e privata

$$\left(\boldsymbol{p_{k_{M,CT_i}}} = \langle G, p, q, g, y' = g^{x'} \bmod p\rangle, \boldsymbol{s_{k_{M,CT_i}}} = x'\right) \leftarrow \boldsymbol{Gen}(\mathbf{1}^{\boldsymbol{n}}) \qquad x' \in \mathcal{Z}^q$$

   e provvede, poi, a firmare il CSR_{CT}. Nel caso in cui il cittadino CT_i risulti positivo ad un tampone, il certificato emesso per la sua chiave pubblica deve diventare invalido. L'invalidazione è realizzato mediante la revoca esplicita del certificato da parte del MS. Per gestire la revoca si prevede che il MS includa un <u>numero seriale</u> in ogni certificato emesso. Pertanto, ogni certificato avrà ora la seguente forma:

$$\boldsymbol{\sigma'} \leftarrow \boldsymbol{Sign}_{s_{k_{M,CT_i}}}\left(\boldsymbol{CSR}_{\boldsymbol{CT_{i_{new}}}}, \#\#\#\right)$$

   dove "**###**" rappresenta il numero seriale del certificato associato a CT_i.

4. A questo punto, chiunque può verificare l'integrità del certificato, ossia il $GP\ 2.0_{CT_i}$, tramite $\boldsymbol{Vrfy}_{p_{k_{M,CT_i}}}(\boldsymbol{GP}_{\boldsymbol{CT_i}}, \boldsymbol{\sigma'}) \stackrel{?}{=} \mathbf{1}$, essendo nota $p_{k_{M,CT_i}}$.

2.1.1 Fase di Pre-Gioco

Il **Garante** $\boldsymbol{GA}$ ha imposto il divieto di invio su canali telematici del GP per garantire la protezione dei dati personali, in quanto mostra dati personali in eccesso rispetto a quanto strettamente necessario per l'accesso ai servizi in questione. Il governo ha deciso di rilasciare un nuovo formato del GP, noto come $\boldsymbol{GP\ 2.0}$, il quale continuerà a prevedere la solita sequenza di informazioni del soggetto (cioè i dati presenti nel GP), associate ad un'unica firma digitale rilasciata dal MS, ma consentirà al cittadino CT_i di esibire telematicamente solo le informazioni strettamente necessarie sulla base del contesto.

Dall'analisi delle informazioni fornite dal $GP\ 2.0$, è individuato un sottoinsieme $Sub_{GP\ 2.0_i}$ di informazioni sufficientemente rappresentative del cittadino CT_i, affinché egli possa accedere al servizio di Sala Bingo Virtuale. Il sottoinsieme di informazioni individuato è il seguente:

- **Data di Nascita**: la data di nascita di un cittadino CT_i gioca un ruolo importante. Questo perché ai cittadini minorenni è impedito l'accesso alla sala bingo virtuale anche se in possesso di un GP 2.0 valido
- **Codice Fiscale**: è un identificativo univoco assegnato ad ogni cittadino, utilizzato per verificare l'identità di una persona.
- **Validità del** $\boldsymbol{GP}$: è importante verificare che il GP sia ancora valido per l'accesso ai servizi della sala bingo virtuale. Questa informazione è ottenibile controllando la data di scadenza specificata nel Green Pass.

È importante sottolineare che all'interno di una sala bingo virtuale, le informazioni riguardanti il tampone, il tipo di vaccino e l'identificativo univoco del certificato non sono necessari per l'identificazione di una persona. Ciò significa che un cittadino CT_i che desidera mantenere riservatezza sulla modalità in cui ha ottenuto il Green Pass, ad esempio un cittadino che ha scelto di non vaccinarsi, può farlo senza dover divulgare tali informazioni all'interno della sala bingo virtuale. In questo modo, viene garantita la protezione della privacy e la tutela del cittadino che desidera mantenere riservate le proprie scelte personali relative alla propria salute.

Formalizzazione

La soluzione per il design del $GP\ 2.0$ è ispirata al funzionamento dello **SPID** [3]. In particolare, per accedere ai servizi forniti dalla sala bingo di $Mr\ Joker$, il cittadino P_i in possesso del $GP\ 2.0_{P_i}$, rilasciato come specificato in precedenza, si connette, dopo aver verificato la correttezza del certificato digitale della sala bingo, tramite una connessione HTTPS al sito di $Mr\ Joker$. Il giocatore P_i viene, quindi, reindirizzato all'IdP del MS ed esibisce telematicamente il $GP\ 2.0_{P_i}$ insieme ad un $Timestamp$ generato al momento della connessione. Il $Timestamp$ viene emesso da una TSA (Time Stamping Authority) in modo da prevenire possibili replay attack. Assunto che il $Timestamp$ non sia scaduto, se

1. $\left[\boldsymbol{Vrfy}_{\boldsymbol{pk}_{M,G_i}}(\boldsymbol{GP\ 2.0}_{\boldsymbol{P_i}}, \boldsymbol{\sigma'}) \neq \boldsymbol{1}\right]$, il requisito di correttezza del $GP\ 2.0_{P_i}$ non è soddisfatto. Ciò potrebbe essere dovuto al fatto che il $GP\ 2.0_{P_i}$ è scaduto oppure che il numero seriale associato a $GP\ 2.0_{P_i}$ appartiene alla lista dei certificati revocati (CRL). Pertanto, è impedito l'accesso al sito a P_i.
2. $\left[\boldsymbol{Vrfy}_{\boldsymbol{pk}_{M,G_i}}(\boldsymbol{GP\ 2.0}_{\boldsymbol{P_i}}, \boldsymbol{\sigma'}) == \boldsymbol{1}\right]$, il requisito di correttezza del $GP\ 2.0_{P_i}$ è soddisfatto. In tal caso, l'IdP del MS genera una coppia di chiavi pubblica $\boldsymbol{p}_{\boldsymbol{k}_{\boldsymbol{IdP}_{\boldsymbol{MS}}}}$ e privata $\boldsymbol{s}_{\boldsymbol{k}_{\boldsymbol{IdP}_{\boldsymbol{MS}}}}$ che utilizzerà per firmare il sottoinsieme di informazioni $\boldsymbol{Sub}_{\boldsymbol{GP\ 2.0}_i}$ ritenuto strettamente necessario dalla sala bingo di $Mr\ Joker$ per accedere ai suoi servizi.

$$\boldsymbol{\sigma}_{\boldsymbol{Sub}_{\boldsymbol{GP\ 2.0}_i}} \leftarrow \boldsymbol{Sign}_{\boldsymbol{s}_{\boldsymbol{k}_{\boldsymbol{IdP}_{\boldsymbol{MS}}}}}(\boldsymbol{Sub}_{\boldsymbol{GP\ 2.0}_i})$$

[3] L'architettura e il funzionamento dello **SPID** (Sistema Pubblico di Identità Digitale) possono essere compresi analizzando gli attori principali e il loro ruolo:

- Soggetto Interessato: Il cittadino che desidera accedere ai servizi attraverso lo SPID.
- Identity Provider: Ente accreditato che fornisce l'identità digitale (Poste Italiane).
- Fornitore di Servizi: Organizzazione che offre servizi online ai cittadini, che accetta l'identità digitale come metodo di aunteticazione.

Durante il processo di autenticazione attraverso SPID, solo alcuni dati minimi e necessari del cittadino vengono inviati al Fornitore di Servizi dall'Identity Provider, dopo che questo ne ha verificato la correttezza, e ciò avviene solo dopo che il cittadino ha dato il proprio consenso. Ad esempio, se consideriamo l'**Università degli Studi di Salerno** come Fornitore di Servizi, saranno trasmessi a quest'ultimo solo i seguenti dati: codice identificativo, nome, cognome e codice fiscale. Questi dati rappresentano solo una parte degli attributi dello SPID, elencati nel dettaglio al link: https://docs.italia.it/italia/spid/spid-regole-tecniche/it/stabile/attributi.html.

Dopo che l'IdP del MS ha inviato il sottoinsieme di informazioni $\boldsymbol{Sub_{GP\ 2.0_i}}$ insieme ad un $Timestamp'$ al sito di $Mr\ Joker$ tramite una connessione HTTPS, quest'ultimo verifica il requisito di correttezza dei dati ricevuti $\boldsymbol{Vrfy_{p_{kIdP_{MS}}}\left(Sub_{GP\ 2.0_i}, \sigma_{Sub_{GP\ 2.0_i}}\right) \stackrel{?}{=} 1}$. Assunto che il $Timestamp$ sia valido, sulla base della risposta e dei dati forniti, il sito di $Mr\ Joker$ decide se concedere o negare l'accesso a P_i, terminando così la fase di pre-gioco. Ad esempio, se il requisito di correttezza è valido ma la data di nascita di P_i indica che è minorenne, gli verrà impedito l'accesso al sito [4].

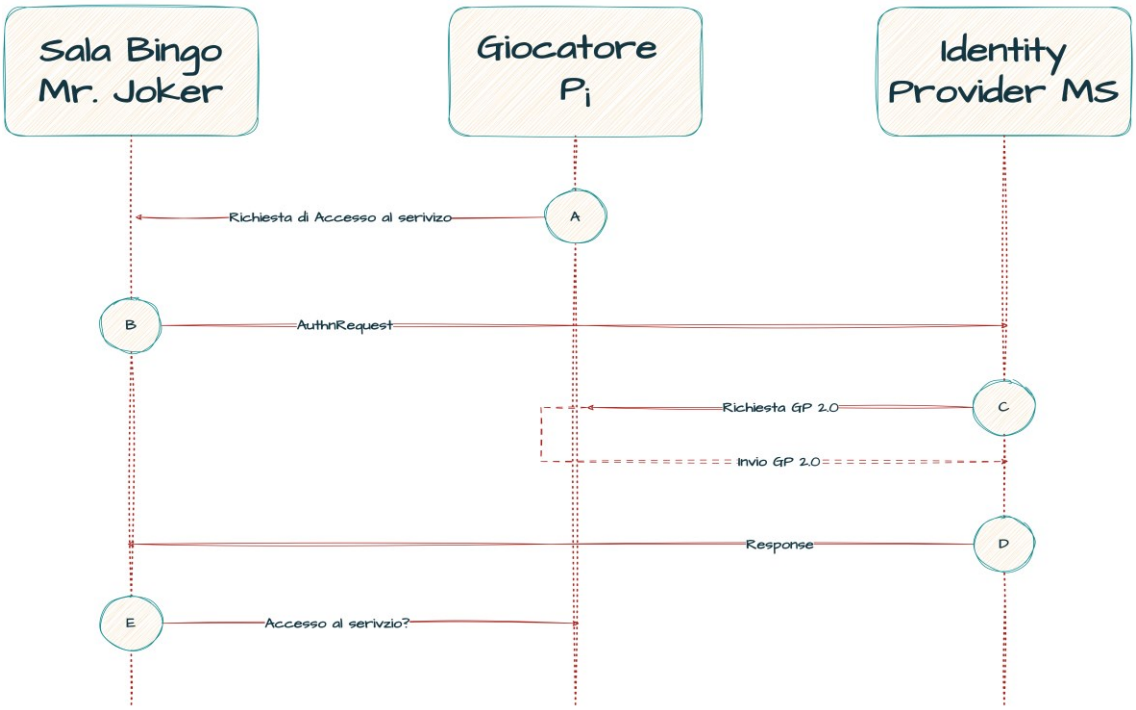

Fig. 2.2 – Meccanismo di autenticazione per la Sala Bingo di Mr. Joker ispirato a quello SPID. Questo prepara una AuthRequest, ossia una richiesta di autenticazione, che il giocatore P_i inoltra all'IdP. Eseguita l'autenticazione, l'utente torna presso il sito di Mr. Joker con un'asserzione firmata dal MS contenente gli attributi richiesti he Mr. Joker può usare per autorizzare l'utente in base alle proprie policy ed erogare il servizio richiesto.

[4] Il sistema del GP 2.0, così concepito, presenta una natura **dinamica**. Questo perché se le condizione di accesso dovessero mutare, non sarebbe necessario dover richiedere al MS di rilasciare un nuovo GP 2.0. Infatti, il sito di Mr. Joker può semplicemente richiedere all'IdP del Ministero della Salute informazioni differenti, a seconda delle politiche in vigore. Ad esempio, nel caso in cui, a causa dell'aggravarsi della pandemia, si vogliano "incoraggiare" i cittadini ad effettuare la vaccinazione, si potrebbe imporre il divieto di partecipare agli eventi, a chi non ha fatto la vaccinazione. Quindi, il sito di Mr. Joker dovrà semplicemente richiedere all'IdP del Ministero della Salute che gli venga fornito anche i dati del campo "Vaccinazione".

2.2 Generazione Stringhe Casuali

Una volta autenticato, al giocatore P_i viene chiesto di scegliere un nickname per la sessione di gioco. A questo punto, il giocatore P_i può effettuare, durante la finestra temporale $[\boldsymbol{T_1}, \boldsymbol{T_2}]$, al più una giocata $\boldsymbol{B_{P_i}}$. Nel caso in cui un giocatore P_i non effettua una giocata, verrà escluso dall'attuale turno di gioco e quindi verrà escluso dalla generazione casuale di stringhe.

Si suppone che, per semplicità, l'unico gioco disponibile all'interno della Sala Bingo virtuale sia la roulette francese. In questo contesto, il giocatore P_i ha la possibilità di selezionare almeno una delle seguenti opzioni di gioco:

- **Numero**: il giocatore P_i può scegliere al più un numero da 0 a 36.
- **Rosso vs. Nero**: il giocatore P_i può scegliere il colore del numero estratto: rosso $(bit = 1)$ o nero $(bit = 0)$.
- **Pari vs. Dispari**: Il giocatore P_i ha la possibilità di scegliere se il numero estratto sarà pari $(bit = 1)$ o dispari $(bit = 0)$.

Una possibile rappresentazione della giocata in bit sarebbe la seguente:

Scelta numero (1 bit)	Numero (6 bit)	Scelta colore (1 bit)	Colore (1 bit)	Scelta parità (1 bit)	Parità (1 bit)
X	XXXXXX	X	X	X	X

dove il bit "$Scelta\ ***$" se settato a 1 indica che il giocatore P_i ha scommesso sulla giocata "$***$", viceversa, se settato a 0. Ad esempio, la sequenza di bit $B_{P_i} = 10010110011$ rappresenta una giocata in cui il giocatore P_i ha scommesso che il numero estratto sarà il numero 11 o che sarà un numero pari.

Infine, al termine del turno di gioco, dopo che ogni giocatore P_i ha impegnato la giocata B_{P_i} effettuata, il banco D e ogni giocatore P_i generano una **stringa casuale** $\boldsymbol{RS_x}$ tramite una PRG, che verrà utilizzata nella fase di post-gioco per determinare il numero estratto per il turno di gioco corrente.

Formalizzazione

Una volta effettuato l'accesso a una stanza della Sala Bingo, un giocatore P_i abilitato al gioco può effettuare al più una giocata B_{P_i}. La fase di gioco, che avviene durante la finestra temporale $[\boldsymbol{T_1}, \boldsymbol{T_2}]$, prevede le seguenti fasi:

1. All'inizio di ogni turno di gioco, il banco D genera una chiave pubblica k_D e la distribuisce tramite una connessione HTTPS a tutti i giocatori P della stanza. Questa verrà utilizzata dagli stessi giocatori per impegnare le loro giocate B_{P_i}:

$$\boldsymbol{k_D = (p, q, g, h)} \qquad con\ h \in \langle g \rangle$$

2. Entro la finestra temporale prefissata, ogni giocatore P_i impegna la propria giocata B_{P_i} e prende un $r_{P_i} \in \mathcal{Z}_q$ uniforme. Poi, calcola il commitment, che invierà al banco D, il quale lo accetta e lo conserva [5].

$$\boldsymbol{com = g^{B_{P_i}} h^{r_{P_i}} \bmod p \in \mathcal{Z}_p^*}$$

3. All'istante temporale $T2$ e, quindi subito dopo che ogni giocatore P_i ha impegnato la giocata B_{P_i}, il banco D genera una stringa casuale $\boldsymbol{RS_D}$ tramite una PRG:

$$\boldsymbol{RS_D \leftarrow PRG(s_D, 1^L)} \qquad con\ s_D \in \{0,1\}^n, L > n$$

   Allo stesso modo, ogni giocatore P_i che ha effettuato una giocata B_{P_i} genera una stringa casuale $\boldsymbol{RS_{P_i}}$ tramite una PRG e la invia attraverso una connessione HTTPS al banco D:

$$\boldsymbol{RS_{P_i} \leftarrow PRG(s_{P_i}, 1^L)} \qquad con\ s_{P_i} \in \{0,1\}^n, L > n$$

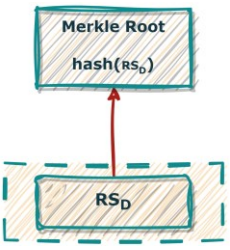

Fig. 2.3 – Merkle Tree di altezza 1 prodotto per l'attuale turno di gioco, a cui non partecipa nessun giocatore presente nella sala. Tutto ciò garantisce una continuità nel gioco, assicurando un'esperienza utente fluida e coerente. Come per lo schema di commitment, tale scelta tende a emulare l'esperienza di un casinò reale dove il croupier gira sempre la roulette, indipendentemente dal fatto che ci siano scommesse o meno.

[5] La scelta di utilizzare uno schema di commitment è da rimandarsi alla volontà di emulare l'atmosfera di un casinò reale, dove le giocate degli altri giocatori sono visibili per la maggior parte dei giochi di fortuna. Ad esempio, quando un giocatore scommette su un numero alla roulette, gli altri giocatori possono vedere dove sono state piazzate le fiches. Tale meccanismo assicura che, anche in un ambiente digitale, le giocate siano trasparenti e che i giocatori non possano alterare le loro decisioni una volta fatte, proprio come avviene in un casinò reale.

Terminata la fase di gioco, inizia la fase di post-gioco identificata dall'intervallo $[\boldsymbol{T_2}, \boldsymbol{T_3}]$. Durante tale fase, ogni stringa casuale prodotta è inserita all'interno di un Merkle Tree costruito dal banco D. Un esempio di Merkle Tree è visibile nelle Fig. 2.3 e Fig. 2.4. Al completamento del Merkle Tree, si estrae la **Merkle Root** $\boldsymbol{MR}$, che sintetizza tutte le stringhe casuali, utilizzata per determinare il numero vincente attraverso un'operazione modulo 37. Il risultato ottenuto viene comunicato a tutti i giocatori P, i quali invieranno a D la coppia $(\boldsymbol{B_{P_i}}, \boldsymbol{r_{P_i}})$ così che il banco possa verificare l'integrità del commitment e, quindi, determinare gli eventuali vincitori dell'attuale turno di gioco.

Infine, al termine della fase di post-gioco ogni giocatore P_i può effettivamente verificare che non è stata commessa alcuna manomissione, andando a confrontare la Merkle Root calcolata MR' da P_i con quella fornita dal banco D. Se si identica una discrepanza, allora i giocatori P possono far intervenire un **meccanismo di giustizia** $\boldsymbol{J}$ per indagare su potenziali irregolarità. Ciò assicura un certo grado di sicurezza e di fiducia nel sistema per tutti i giocatori [6].

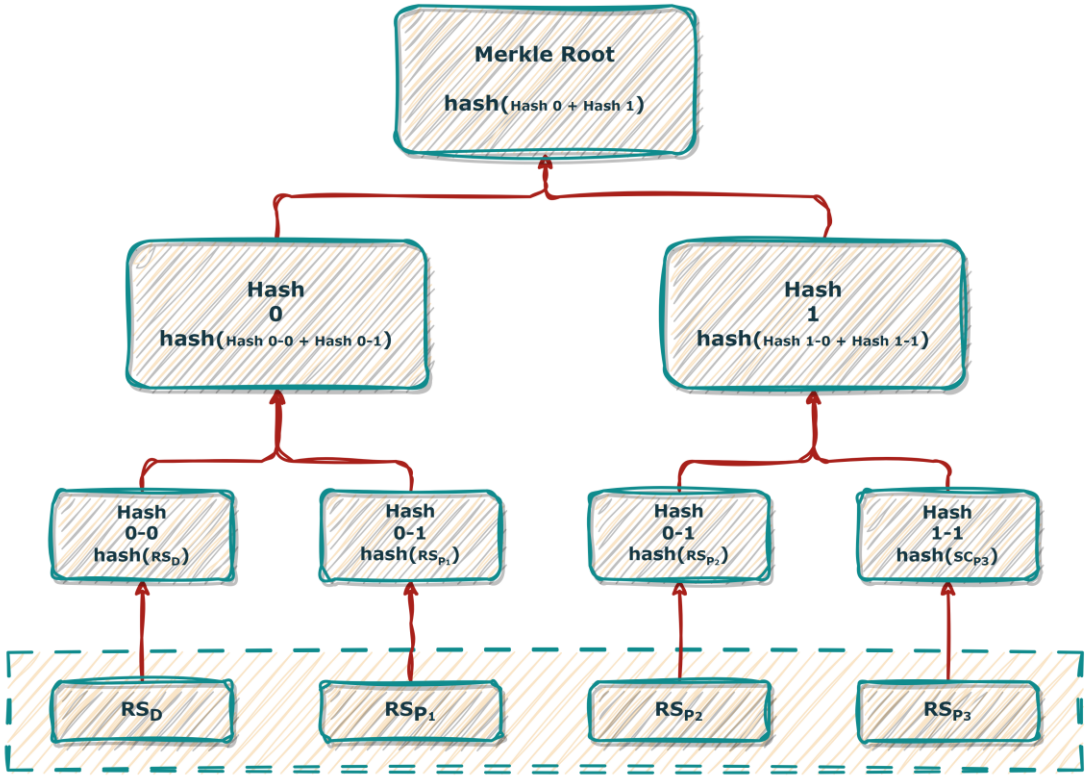

Fig. 2.4 – Merkle Tree di altezza 2 prodotto per l'attuale turno di gioco, a cui partecipano 3 giocatori presenti nella sala.

[6] Come già specificato nel WP1, gli aspetti relativi alle possibili conseguenze penali su Mr. Joker, i giocatori P e sul sistema di casinò sono escluse dalla seguente trattazione.

Formalizzazione

La fase di post-gioco, che avviene durante la finestra temporale $[\boldsymbol{T_2}, \boldsymbol{T_3}]$, prevede le seguenti fasi:

1. Il banco D ha raccolto tutte le stringhe casuali dei giocatori RS_{P_i} inizia a costruire il Merkle Tree. Le stringhe casuali, compresa quella generata dal banco RS_D, fungono da foglie del Merkle Tree. Il banco calcola gli hash di ogni coppia di foglie e continua a costruire l'albero fino a quando non rimane un solo hash, che è la **Merkle Root**.
2. Dopo che sono state aggiunte tutte le stringhe al Merkle Tree, viene calcolato il Merkle Root dell'albero utilizzando come funzione hash crittografica $SHA256$:

   $$\boldsymbol{MR = SHA256\left(\ldots SHA256\left(SHA256(RS_D), SHA256\left(RS_{P_1}\right)\right), \ldots\right)}$$

   Il numero estratto dal banco D viene calcolato attraverso un'operazione di modulo 37 a partire dal valore del Merkle Root: $\boldsymbol{n = MR \bmod 37}$.
3. Il banco D invia il numero estratto a tutti i giocatori P attraverso una connessione HTTPS. A questo punto, ogni giocatore P_i rivela a D il valore di B_{P_i} e di r inviando $\left(\boldsymbol{B_{P_i}}, \boldsymbol{r}\right)$ a quest'ultimo.
4. Il banco D rende pubbliche a ogni giocatore P_i le giocate D_{P_i} effettuate dagli altri giocatori della sala, in modo che tutti possano verificare la correttezza del risultato. Inoltre, fornisce anche la Merkle Root e la Merkle Proof, che rappresenta il percorso di hash dalla stringa casuale del giocatore P_i alla Merkle Root, in modo che ogni P_i possa verificare che la sua stringa sia stata inclusa nel Merkle Tree confrontando la Merkle Root calcolata con quella fornita dal server.

2.3 JokerChain

In termini di efficienza l'approccio proposto è oneroso, il che può portare a ritardi del sistema durante l'esecuzione del programma. Questo perché sono state implementate varie soluzioni crittografiche che implicano un aumento dei costi del sistema, soprattutto al crescere del numero di giocatori. Per mitigare al problema dell'efficienza e garantire un maggior grado di trasparenza, la soluzione è quella di impiegare una blockchain permissioned, nota come **JokerChain**. Non essendo $Mr.Joker$ un attore affidabile al 100%, la governance della rete viene affidata ad un'entità riconosciuta affidabile e sicura da tutti i giocatori P_i della sala, come l'**Agenzia delle Dogane e dei Monopoli**, abbreviata con $\boldsymbol{ADM}$, essendo un ente governativo [7]. La blockchain è quindi costituita da diverse tipologie di blocchi e transazioni:

1. **Blocco Genesi**: è il primo blocco della $JokerChain$, il quale contiene due tipologie di transazioni:
   a. La transazione delle chiavi pubbliche dei certificati dei nodi ADM, utilizzate in primis dal banco e poi dai giocatori per instaurare una connessione HTTPS con gli stessi nodi ADM, così da poter procedere al processo di gioco.

$$\boldsymbol{Tr_1 = p_{k_{C_{ADM_i}}}}$$

   b. La transazione della chiave pubblica del certificato del banco D, che permette al nodo ADM, a cui il banco D vuole connettersi, di poter verificare la correttezza del certificato digitale del banco D, in modo che questo possa instaurare una connessione HTTPS con il nodo ADM, solo se ha verificato la correttezza del certificato digitale del nodo ADM.

$$\boldsymbol{Tr_2 = p_{k_{C_D}}}$$

[7] Possiamo prevedere che i nodi ADM vengano distribuiti in varie aree geografiche della penisola, così da poter ridurre la latenza e migliorare l'esperienza dell'utente.

2. **Blocco Chiavi**: è il blocco contenente le transazioni delle chiavi di gioco. Include la chiave pubblica del Banco D utilizzata dallo stesso per firmare la propria stringa casuale e le chiavi pubbliche dei giocatori P_i ammessi alla sala, generate per firmare le rispettive giocate e stringhe casuali.

$$\boldsymbol{Tr_3 = p_{k_D}} \quad \text{e} \quad \boldsymbol{Tr_3 = p_{k_{P_i}}}$$

3. **Blocco Giocate:** è il blocco contenente le transazioni di gioco. Quando un giocatore P_i effettua una giocata B_{P_i}, opportunamente firmata dallo stesso usando la sua chiave privata $s_{k_{P_i}}$, è registrata in una transazione:

$$\boldsymbol{Tr_4 = B_{P_i}}$$

4. **Blocco Stringhe:** è il blocco contenente le transazioni di stringhe casuali. Sia il banco D che i giocatori P_i, che hanno effettuato una giocata B_{P_i}, generano e inviano le stringhe casuali prodotte RS_x tramite una PRG, firmate dagli stessi usando le rispettive chiavi private:

$$\boldsymbol{Tr_5 = RS_D} \quad \text{e} \quad \boldsymbol{Tr_5 = RS_{P_i}}$$

5. **Blocco Risultato**: è il blocco contenente la transazione risultato. In pratica, il banco legge la Merkle Root MR memorizzata nell'header del Blocco Stringhe e la sottopone a un'operazione modulo 37, per determinare il risultato dell'attutale turno di gioco, che verrà firmato dal banco D con la sua chiave privata s_{k_D}.

$$\boldsymbol{Tr_6 = n} \; (= MR \bmod 37)$$

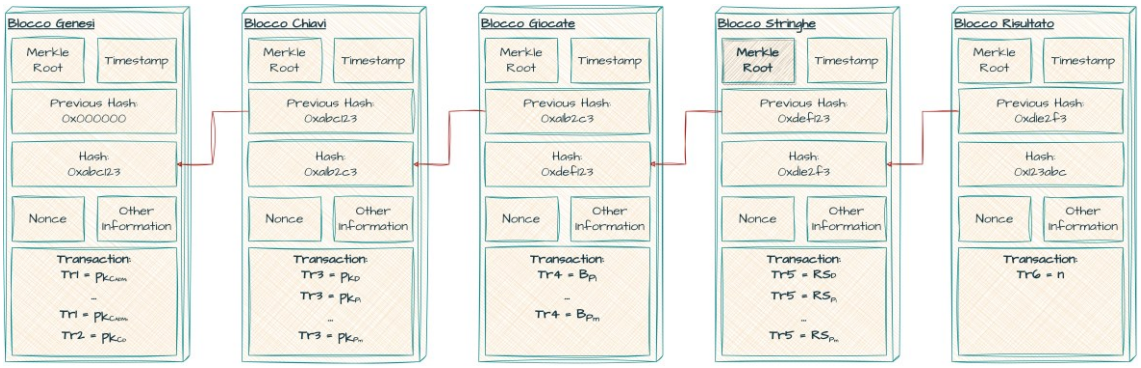

Fig. 2.5 – Architettura della JokerChain. Dopo il Blocco Risultato, inizierà un nuovo turno di gioco, il quale richiede la creazione di un nuovo Blocco Chiavi, seguito da un nuovo Blocco Giocate, da un nuovo Blocco Stringhe e un nuovo Blocco Risultato, e così via. Una possibile rappresentazione del Merkle Root del Blocco Stringhe è presente in Fig. 2.4.

Formalizzazione

Un giocatore P_i, dopo aver esibito il $GP\ 2.0_{P_i}$ sul sito di $Mr\ Joker$ e dopo che questo ne ha verificato la correttezza, verifica l'autenticità del certificato del nodo ADM prima di connettersi allo stesso. Una volta verificato, il giocatore può stabilire una connessione sicura HTTPS con il nodo ADM.

Prima di procedere con qualsiasi azione, il giocatore verifica l'integrità e la cronologia della blockchain assicurandosi che l'ultimo blocco della $JokerChain$ sia immediatamente precedente al Blocco Autenticazione. In tal caso, il giocatore P_i genera una coppia di chiavi, pubblica $\boldsymbol{p_{k_{P_i}}}$ e privata $\boldsymbol{s_{k_{P_i}}}$. Dopo che tutti i giocatori P, compreso il banco D, hanno inviato le loro chiavi pubbliche ai rispettivi nodi ADM, questi nodi collaborano per raggiungere un consenso e pubblicare il Blocco Autenticazione, contenente tutte le chiavi pubbliche generate sotto forma di transazioni [8].

A questo punto, nella finestra temporale $[T_1, T_2]$, il giocatore P_i può effettuare al più una giocata B_{P_i} che verrà, dopo essere stata firmata da P_i con la sua chiave privata $\boldsymbol{\sigma_{B_{P_i}} = Sign_{s_{k_{P_i}}}(B_{p_i})}$, inviata al nodo ADM al quale è collegato P_i.

Il nodo ADM verifica la correttezza della giocata $\boldsymbol{Vrfy_{p_{k_{P_i}}}(B_{P_i}, \sigma_{B_{P_i}}) \stackrel{?}{=} 1}$ e, se dimostrata tale, provvederà ad inserirla nel Blocco Giocate, che verrà aggiunto alla $JokerChain$ solo dopo che tutti i nodi ADM hanno raggiunto il consenso.

All'istante temporale T_2, il giocatore P_i, nel caso in cui ha effettuato una giocata B_{P_i}, genererà una stringa casuale tramite PRG, $\boldsymbol{RS_{P_i} \leftarrow PRG(s, 1^L)}$, che viene firmata da P_i con la sua chiave privata $\boldsymbol{\sigma_{RS_{P_i}} = Sign_{s_{k_{P_i}}}(RS_{P_i})}$.

---

[8] Essendo JokerChain una blockchain permissioned, il **problema del consenso** è più semplice rispetto ad una blockchain permissionless. Tale architettura prevede che ogni nodo ADM riceva transazioni dai giocatore P o dal banco D a cui è connesso, di validare alcune di queste transazioni, andando a controllare le firme, e di trasmetterle agli altri nodi ADM nella rete. Un **nodo leader ADM** propone un nuovo Blocco X una volta che ha raccolto tutte le transazioni. Gli altri nodi ADM ricevono il blocco proposto e lo verificano. Se un nodo ADM concorda con il contenuto dei X, invia un voto positivo al nodo leader ADM, altrimenti un voto negativo. Una volta che il nodo leader ADM riceve i voti da **almeno 2/3 dei nodi ADM** e la maggioranza di questi è positiva, il blocco X viene accettato. Per evitare che un singolo nodo diventi un collo di bottiglia o un punto di attacco, il ruolo del leader viene ruotato tra i nodi. Essendo ADM affidabile al 90%, la probabilità che un nodo ADM agisca in modo malevolo è estremamente bassa. Pertanto, il numero possibile di nodi difettosi è ben al di sotto del limite tollerabile.

Dopo aver generato e firmato le loro stringhe casuali, tutti i giocatori, compreso il banco D, le inviano ai rispettivi nodi ADM, che, dopo averne verificato la correttezza, collaborano per raggiungere un consenso e pubblicare il Blocco Stringhe, contenente tutte le stringhe casuali generate sotto forma di transazioni.

Nell'intervallo temporale $[T_2, T_3]$, il banco D legge il Merkle Root del Blocco Stringhe ed effettua un'operazione modulo 37 per stabilire il risultato per il turno di gioco corrente $\boldsymbol{n = MR \bmod 37}$. Tale numero, dopo essere stato firmato dal banco D con la sua chiave privata $\boldsymbol{\sigma_n = Sign_{s_{k_D}}(n)}$, viene inviato al nodo ADM al quale è collegato D, il quale verifica la correttezza della giocata e, se dimostrata tale, provvederà ad inserirla nel Blocco Risultato, aggiunto alla $JokerChain$ solo dopo che tutti i nodi ADM hanno raggiunto il consenso.

WP3. Analisi

Workpackage	Task	Responsabile
WP3	Analisi	Cognome2 Nome2

Il WP3 ha lo scopo di effettuare un'analisi della soluzione presentata nel WP2 rispetto al modello presentato nel WP1. In particolare, richiede di discutere il modello proposto in termini dei quattro pilastri fondamentali: confidenzialità, integrità, trasparenza ed efficienza. La struttura analizzata è quella che prevede l'utilizzo della $JokerChain$.

3.1 Confidenzialità

Di seguito, vengono discussi uno per uno gli attaccanti che potrebbero compromettere le proprietà di confidenzialità del sistema definite nel WP1:

C1. **Informazioni sanitarie dell'utente**: Il sistema deve garantire la protezione delle informazioni sanitarie dell'utente. Un avversario non dovrebbe essere in grado di determinare le ragioni alla base del rilascio del $GP\ 2.0$ a un determinato giocatore P_i.

I possibili avversari interessati a compromettere la proprietà C1 sono l'**Insider Trader**, il **Thief** e il **Malicious Programmer**. Per garantirla, il sistema è stampo implementato in modo che il giocatore P_i che vuole accedere al sito di $Mr\ Joker$ presenti il suo $GP\ 2.0_{P_i}$, contenete i dati personali e sanitari, all'Identity Provider del MS. In questo modo, il $GP\ 2.0$ non viene mai completamente passato al sito di $Mr\ Joker$. Sarà, poi, l'IdP del MS a fornire al sito di $Mr\ Joker$ il sottoinsieme delle informazioni $Sub_{GP\ 2.0_{P_i}}$ necessarie al sito di $Mr\ Joker$ per convalidare o meno l'accesso al sito.

C2. **Credenziali d'accesso:** Le credenziali di accesso di un giocatore P_i devono essere completamente segrete.

I possibili avversari interessati a compromettere la proprietà **C2** sono l'**Insider Trader**, il **Thief** e il **Malicious Programmer**. Tale proprietà viene garantita in quanto quando il giocatore P_i esibisce il $GP\ 2.0_{P_i}$ telematicamente, lo fa su una connessione sicura HTTPS che garantisce che il certificato inviato all'IdP del MS venga cifrato e non venga intercettato da avversari.

C3. **Identità Nascosta:** L'identità di ogni giocatori P_i presenti in una stanza di gioco deve rimanere nascosta a tutti gli altri giocatori P.

I possibili avversari interessati a compromettere la proprietà **C3** sono l'**Insider Trader**, il **Meddler** e il **Malicious Programmer**. La proprietà è assicurata per il Meddler in quanto è impossibile risalire all'identità associata a P_i, essendo richiesto a P_i di utilizzare un nickname per la sessione di gioco corrente. Tuttavia, il Malicious Programmer e l'Insider Trader potrebbero essere in grado di compromettere tale proprietà, soprattutto se il primo dovesse attaccare il dispositivo di P_i.

3.2 Integrità

Di seguito, vengono discussi uno per uno gli attaccanti che potrebbero compromettere le proprietà d'integrità del sistema descritte nel WP1:

I1. **Stabilità del sistema**: Il sistema deve essere ben progettato in modo che un attaccante non possa manipolare la correttezza del risultato generato dal banco B. Ciò è essenziale per garantire che il gioco sia equo e trasparente per tutti i partecipanti.

Un possibile avversario interessato a compromettere la proprietà **I1** è il **Malicious Programmer**. Essa viene assicurata in quanto, essendo la blockchain trasparente, chiunque può controllare il valore del Merkle Root del Blocco Stringhe. Pertanto, anche nel caso in cui un avversario riuscisse a compromettere il risultato, i giocatori possono verificano la correttezza del risultato, essendo questo ricavato applicando al valore del Merkle Root del Blocco Stringhe un'operazione modulo 37. Nel caso in cui fosse evidenziata una discrepanza, allora i giocatori P possono richiedere l'intervento della giustizia J.

I2. **Continuità del servizio**: Il sistema deve garantire la continuità del servizio, evitando interruzioni o downtime che potrebbero causare danni ai partecipanti o al gioco stesso.

Un possibile avversario interessato a compromettere tale proprietà sono gli **Overloaders**. La continuità del servizio viene garantita dalla specifica per cui il numero massimo di partecipanti per ogni stanza è dell'ordine delle decine. In questo caso, il numero di giocatori in una sala influisce in maniera marginale sull'esperienza di gioco. Pertanto, attacchi di sovraccarico del sistema sono trascurabili.

I3. **Idoneità al gioco**: Il processo di generazione di stringhe casuali deve essere garantito solo a coloro in possesso di un GP 2.0 valido.

Un possibile avversario interessato a compromettere la proprietà **I3** è il **No-Vax**. Tale proprietà viene rispettata in quanto affinché avvenga l'accesso all'interno della Sala Bingo è necessario che il giocatore P_i esibisca un $GP\ 2.0$ valido. Tuttavia, il No-Vax potrebbe sferrare un replay attack verso un giocatore P_i in possesso di un $GP\ 2.0$ valido. Pertanto, il sistema è stato implementato in modo che la possibilità di replay attack,

in cui un avversario registra e rinvia nuovamente l'interazione tra il giocatore P_i e l'IdP del MS sia resa impossibile. In particolare, è stato previsto che il mittente aggiunga un $Timestamp$, il viene generato e firmato da un'autorità di timestamping (TSA) separata e affidabile. In questo modo, se il $Timestamp$ non è entro un certo intervallo di tempo accettabile, la richiesta viene considerata non valida. Questo garantisce che anche se un avversario cattura i dati, non può utilizzarli dopo un certo periodo di tempo.

Tuttavia, nel caso in cui fosse entrato in possesso del GP 2.0 di un altro cittadino, il No-Vax riuscirebbe ad accedere alla Sala Bingo, eludendo così il sistema. Ciononostante, se fosse scoperto, incorrerebbe in gravi conseguenze da parte della giustizia J.

I4. **Unicità della giocata**: Il sistema deve garantire ai giocatori P che il proprio valore immesso sia integro per tutta la durata della giocata. Ciò significa che per nessun soggetto coinvolto nel sistema deve essere possibile, a partire da un valore lecitamente immesso nel sistema, forgiare un nuovo valore che appare come valido.

I possibili avversari interessati a compromettere la proprietà **I4** sono il **Trickster**, l'**Insider Trader**, e il **Malicious Programmer**. Tale proprietà viene rispettata in quanto, una volta che il giocatore P_i ha sottomesso una giocata B_{P_i}, questa viene registrata come transazione all'interno del Blocco Giocate della $Jokerchain$. In particolare, il nodo ADM, a cui è associato il giocatore P_i, verifica l'autenticità della giocata B_{P_i}. Questo viene fatto controllando la firma digitale di B_{P_i} realizzata con la chiave privata $s_{k_{P_i}}$ del giocatore. Poiché solo P_i conosce la sua chiave privata, il nodo ADM può confermare con certezza che la giocata è stata effettuata da P_i. Quindi, una volta che la giocata è stata verificata e aggiunta alla $Jokerchain$, essa diventa immutabile.

Le proprietà di confidenzialità e di integrità valgono nel caso di presenza dei **Bingo Bandits**, ossia una collusione dei possibili avversari citati.

3.3 Trasparenza

Il sistema implementato gode di un'elevata **trasparenza**, in quanto il sistema di gioco viene implementato attraverso la blockchain $Jokerchain$, garantendo quindi che chiunque all'interno della $Jokerchain$ possa esaminare tutte le transazioni a partire dal Blocco Genesi. In questo modo, viene garantita la proprietà **T3** [9].

Inoltre, Il sistema è stato progettato per garantire un alto livello di affidabilità nel funzionamento equo del gioco. Ciò viene ottenuto nuovamente grazie all'ausilio della blockchain $Jokerchain$. Inoltre, la generazione del risultato è un'operazione matematica molto semplice, che chiunque è in grado determinare. In questo modo viene garantita la proprietà **T1**, facendo fronte alle criticità poste dai *no-fox* (tecnocrati del progresso tecnologico).

Per quanto riguarda la proprietà **T2**, l'ADM viene considerata un attore che gode di un'alta credibilità, essendo un ente governativo. Come già espresso in precedenza, la governance di una blockchain permissioned, come $Jokerchain$, deve essere affidata solo ad organizzazioni ben note e autorizzate. $Mr\ Joker$ avrebbe dei validi motivi per compromettere il sistema e, pertanto, non potrà godere mai di una completa fiducia come può essere per un ente governativo.

[9] Come già espresso in precedenza, la decisione di non cifrare, all'interno della Jokerchain, le giocate e le stringhe casuali prodotte per la generazione del risultato è da rimandarsi alla volontà di emulare l'atmosfera di un casinò reale, dove le giocate degli altri giocatori sono visibili per la maggior parte dei giochi di fortuna.

3.4 Efficienza

La soluzione proposta senza blockchain prevedeva l'uso di diverse tecniche crittografiche, come lo schema di commitment, e di numerose comunicazioni tra i client (giocatori P) e il server (banco D) che imponevano un'elevata operosità al sistema. Per garantire una migliore efficienza e al tempo stesso offrire un maggior grado di confidenzialità, integrità e trasparenza, abbiamo introdotto la blockchain permissioned *Jokerchain*. Questa soluzione, pur offrendo numerosi vantaggi, presenta alcune sfide in termini di efficienza. In particolare, la verifica e l'approvazione delle transazioni, l'aggiunta di queste transazioni alla blockchain e la creazione di nuovi blocchi possono introdurre ritardi. Inoltre, l'uso di connessioni HTTPS, sebbene essenziale per la sicurezza, può aggiungere ulteriori ritardi. L'utilizzo dell'algoritmo SHA-256, che è relativamente veloce e ampiamente utilizzato, implica che le operazioni relative al calcolo del risultato siano ottimizzate.

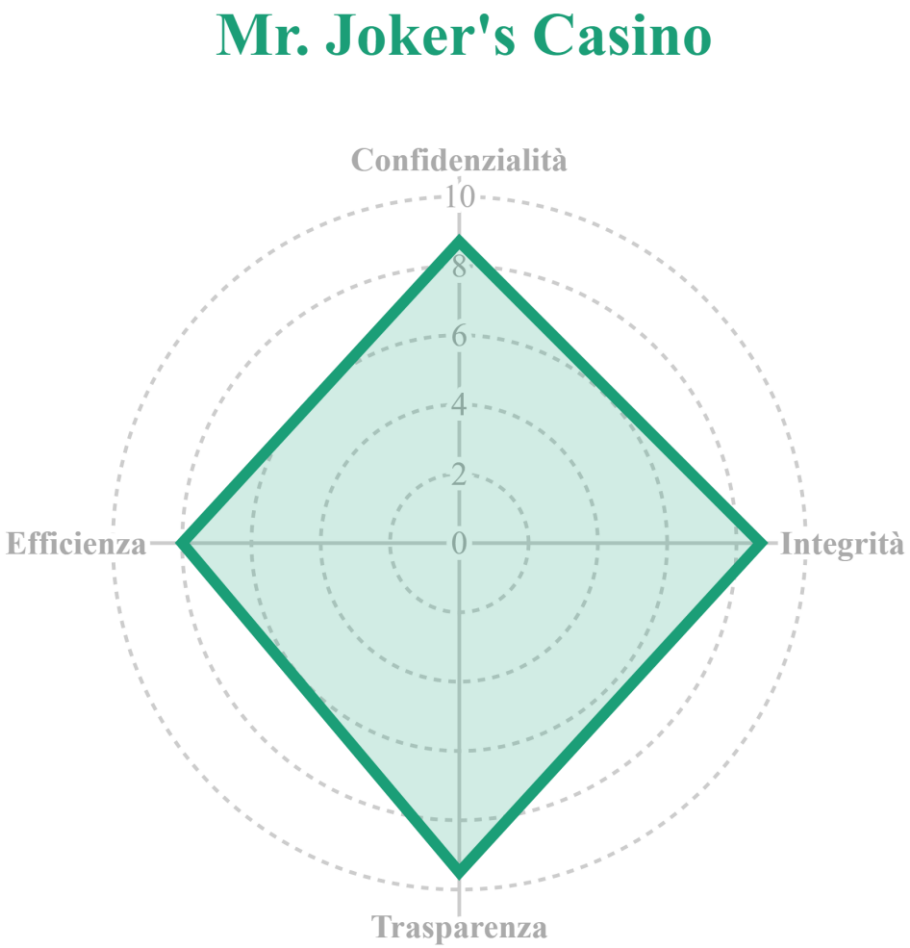

Fig. 3.1 – Grafico radar dei quattro pilastri fondamentali considerati per l'analisi del sistema.

Approfondimento - Malware, Virus e Trojans

Nel caso in cui il dispositivo di un giocatore P_i fosse compromesso da malware, virus o trojan, tutte le misure di sicurezza implementate a livello di rete o di piattaforma potrebbero risultare inutili. Ciò perché un avversario A potrebbe, attraverso il malware, intercettare, alterare o rubare dati direttamente dal dispositivo del giocatore prima che questi vengano trasmessi alla rete.

Per proteggere il dispositivo da tali minacce, un giocatore P_i può utilizzare un sistema di rilevamento delle intrusioni (IDS, *Intrusion Detection System*). Nello specifico, può impiegare un **HIDS** (Host-based IDS), il quale monitora l'attività all'interno di un dispositivo per rilevare qualsiasi comportamento sospetto. Può rilevare tentativi di accesso non autorizzati, modifiche ai file di sistema o altre attività che potrebbero indicare la presenza di malware. Se un software dannoso tenta di bypassare o disabilitare l'HIDS, il sistema è progettato per rilevarlo e bloccarlo. Per garantire che il database dell'HIDS non venga compromesso, può essere conservato su un supporto fisico separato o su un supporto che non può essere facilmente alterato. Oltre all'HIDS, un **NIDS** (Network-based Intrusion Detection System) può monitorare il traffico di rete per rilevare attività sospette. Infatti, è proprio dalla rete che arrivano i maggiori pericoli. Nello specifico, un giocatore P_i può impiegare un'Anomaly-based IDS, in cui le anomalie possono essere rilevate attraverso un'analisi comportamentale del traffico osservato in precedenza (euristica), al contrario di quanto accade nei sistemi Signature-based dove la ricerca è fatta tramite regole, tramite la ricerca di pattern o di firme caratteristiche delle violazioni di sicurezza, il quale può essere affetto da falsi negativi (non rileva attacchi classificati) e falsi positivi (allarmi erronei).

Il rilevamento precoce può permettere agli amministratori di sistema di poter intervenire rapidamente, fermare un'attività malevola in corso e prendere misure per rimuovere la minaccia.

WP4. Implementazione

Workpackage	Task	Responsabile
WP4	Implementazione	Nessuno