Progettazione e implementazione di un sistema di Truth Discovery sfruttando tecniche di cifratura omomorfica

Tesi di Laurea Magistrale in Ingegneria Informatica

Giuseppe Messina

Relatore: Prof. Salvatore Gaglio

Correlatori: Ing. Vincenzo Agate
Ing. Pierluca Ferraro

UNIVERSITÀ DEGLI STUDI DI PALERMO
DIPARTIMENTO DI INGEGNERIA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

PROGETTAZIONE E IMPLEMENTAZIONE DI UN SISTEMA DI TRUTH DISCOVERY SFRUTTANDO TECNICHE DI CIFRATURA OMOMORFICA

Tesi di Laurea di
Giuseppe Messina

Relatore:
Ch.mo Prof. Salvatore Gaglio

Correlatori:
Ingg. Vincenzo Agate e Pierluca Ferraro

Sommario

La grossa diffusione di dispositivi mobili, quasi sempre equipaggiati con una grande quantità di sensori, consente di registrare e acquisire moltissimi dati e informazioni relativamente all'ambiente circostante. Tale diffusione ha dato vita alla tecnica del *mobile crowdsensing*, ossia la possibilità da parte di una comunità di utenti di potere condividere dati di qualsiasi natura, al fine di estrarre informazioni di utilità per la comunità stessa.

L'obiettivo di questo lavoro di tesi sarà progettare ed implementare un framework che utilizzi approcci basati sul paradigma di Truth Discovery per l'analisi aggregata di dati di varia natura, in funzione delle attendibilità dei singoli utenti, e che utilizzi approcci basati sul paradigma della Secure Multi Party Computation per garantire la privacy degli stessi utenti. La prima parte del lavoro sarà effettuare un'analisi degli approcci presenti in letteratura relativamente ai due paradigmi, evidenziando quelli più promettenti. Successivamente, verrà descritto il modello proposto e le fasi di implementazione. Il modello utilizzerà tecniche di Truth Discovery per l'analisi aggregata dei dati e sistemi di crittografia omomorfica per garantire la privacy degli utenti. Infine, il modello verrà valutato sperimentalmente in funzione di diversi contesti di utilizzo.

Indice

Introduzione

Nell'ultimo decennio l'intera popolazione mondiale ha assistito ad una vera e propria esplosione del numero di dispositivi mobili, come smartphone, smartwatch e così via. Quasi tutti questi dispositivi sono disponibili in commercio equipaggiati da una grande quantità di sensori, in grado di potere registrare e acquisire moltissime informazioni dell'ambiente circostante. Proprio a fianco di questa grossa disponibilità di dispositivi e sensori nasce il paradigma del *mobile crowdsensing*, ossia quell'insieme di tecnologie che consentono ad una comunità di persone di potere condividere dati di qualsiasi natura, al fine di poter estrarre informazioni relative a misure, analisi e predizioni circa qualsiasi processo di interesse della comunità.

Un classico esempio è rappresentato dall'utilizzo del sensore GPS e di temperatura spesso presenti all'interno di uno smartphone. In un contesto di crowdsensing, i sensori GPS possono raccogliere e memorizzare una enorme quantità di dati e informazioni circa la localizzazione, e quindi le posizioni geografiche registrate dagli utenti, mentre i sensori di temperatura possono acquisire informazioni strettamente legate all'ambiente circostante. Sulla base di questi dati, è possibile quindi effettuare un'analisi aggregata circa le variazioni climatiche di determinate posizioni geografiche, o più in generale, effettuare analisi di tipo meteorologico.

Il vantaggio risiede nel fatto che tutto ciò è effettuato sull'invio volontario dei dati da parte dei possessori dei vari smartphone, che acquisiranno e invieranno tali informazioni. Risulta immediato pensare come un sistema basato sul mobile crowdsensing possa avere un importante impatto sociale ed economico e come possa integrarsi perfettamente in contesti di monitoraggio del traffico urbano, monitoraggio ambientale, controllo del trasporto pubblico intelligente, gestione della sanità, votazioni e aste elettroniche. Tuttavia, i dati acquisiti (o dichiarati) dalla popolazione attraverso il sistema di mobile crowdsensing spesso possono essere affetti da rumore, o comunque possono rappresentare informazioni non attendibili. Le cause sono molteplici: possibile rumore ambientale, sensibilità del sensore utilizzato dallo smartphone, cattivo utilizzo dell'applicazione da parte dell'utente e così via.

La soluzione più immediata a questa tipologia di problemi potrebbe essere l'aggregazione di dati relativi ad uno stesso oggetto d'esame. Nonostante tutto, l'aggregazione di questi dati, seguita dall'utilizzo di metodologie statistiche "classiche", come la media dei valori o la scelta del valore più frequente, porterebbe a non ottenere un risultato aggregato finale accurato e preciso. Da questa problematica, nasce l'esigenza di potere catturare e rappresentare le differenze di qualità e precisione presenti all'interno delle informazioni dichiarate dai vari utenti. Questa esigenza si traduce quindi nell'attribuire dei gradi di "attendibilità" ai singoli utenti, al fine di poter discriminare le informazioni più precise e veritiere. Sebbene questa idea possa essere promettente, bisogna scontrarsi con la difficoltà di potere conoscere questi gradi di attendibilità, che risultano sconosciuti a priori.

La migliore soluzione a questa sfida è rappresentata da un paradigma di trattamento dei dati, conosciuto in letteratura come *Truth Discovery*. L'obiettivo del paradigma di Truth Discovery è sostanzialmente quello di potere ottenere un risultato corretto e attendibile a partire da un insieme di dati sia corretti che rumorosi. Il principio di funzionamento è quello di attribuire un alto grado di attendibilità agli utenti che hanno dichiarato un valore prossimo al risultato aggregato finale e penalizzare gli utenti che hanno dichiarato dei valori rumorosi o non attendibili. Il grado di attendibilità dei vari utenti può essere rappresentato attraverso un indicatore numerico che prende il nome di *peso*.

Gli approcci basati sul modello di Truth Discovery sono degli ottimi strumenti per l'elaborazione di una informazione aggregata attendibile, ma non garantiscono alcuna riservatezza sui dati. Facendo riferimento proprio agli esempi precedenti sul mobile crowdsensing, quasi sempre i dati che si andranno a trattare con approcci di Truth Discovery conterranno informazioni sensibili e attribuibili ai singoli utenti, e che quindi necessitano di essere tutelati. Ad esempio, l'invio dei dati acquisiti dal sensore GPS potrebbe far trapelare la posizione in cui si trova un determinato utente.

Tale situazione risulta estremamente più grave nel momento in cui si decidesse di utilizzare gli approcci di Truth Discovery, ad esempio, in un contesto sanitario: infatti, i dati forniti dai vari utenti (che in questo contesto sono rappresentati dalle aziende ospedaliere) potrebbero rappresentare una grave minaccia per la privacy dei propri pazienti. Inoltre, anche lo stesso peso, ossia il grado di attendibilità attribuito ai singoli utenti, potrebbe essere un'altra informazione da tutelare, poiché rappresenterebbe un'altra minaccia in termini di privacy.

Al fine di fornire riservatezza sia ai dati che ai pesi attribuiti ai singoli partecipanti, è quindi necessario sfruttare sistemi e approcci derivanti dal mondo della crittografia. Il problema diventa

quindi il seguente: *è possibile sviluppare un approccio di Truth Discovery in grado di potere effettuare le proprie elaborazioni partendo da soli dati cifrati e utilizzando i pesi attribuiti ai singoli partecipanti in maniera cifrata?*

In questo caso, la soluzione adatta a questa problematica è rappresentata dal paradigma di *Secure Multi Party Computation*. Tale paradigma è sostanzialmente un ramo della crittografia che si pone come obiettivo quello di sviluppare delle metodologie che consentano a diversi utenti, denominati *partecipanti*, di potere calcolare in maniera congiunta una qualsiasi espressione matematica, o funzione, a partire dai dati forniti dai singoli partecipanti.

In questo contesto, il vincolo da dover rispettare è che i dati di input di questa funzione devono essere riservati, ossia ogni singolo partecipante è in grado di conoscere solo il proprio valore, ma non quello degli altri partecipanti. Al contrario, il risultato della funzione appena calcolata può essere nota a tutti i partecipanti.

Obiettivo di questo lavoro di tesi sarà quello di progettare e sviluppare un sistema che possa garantire la riservatezza e la privacy dei partecipanti all'interno di un sistema distribuito, attraverso approcci derivanti dal mondo della Secure Multi Party Computation, consentendo inoltre di potere individuare e isolare eventuali falsi feedback derivanti da utenti malevoli, discriminando i risultati ottenuti e considerando quelli più attendibili, il tutto attraverso approcci derivanti dal mondo della Truth Discovery.

La prima parte di questo lavoro di tesi sarà un'analisi dello stato dell'arte, quindi incentrata sull'effettuare uno studio approfondito dei paradigmi di Truth Discovery e Secure Multi Party Computation e dei relativi lavori presenti in letteratura. Tale studio riguarderà i seguenti aspetti per entrambi i paradigmi:

- *definizione*: durante questa fase di studio verrà stilata una definizione formale del paradigma in esame. Saranno quindi descritti gli obiettivi principali e il contesto che ha reso necessaria la nascita del paradigma;
- *contesti di utilizzo*: durante questa fase di studio saranno elencati e descritti i principali contesti di utilizzo del paradigma in esame. Saranno quindi descritti esempi che renderanno più chiari tali contesti;
- *caratteristiche peculiari*: durante questa fase di studio saranno descritti gli elementi presenti in ogni approccio attribuibile al paradigma in esame. Saranno quindi elencate e descritte le metodologie e le tecniche utilizzate che accomunano gli approcci appartenenti al paradigma;

- *approcci in letteratura*: durante questa fase di studio saranno citati gli approcci più significativi e promettenti presenti in letteratura relativamente al paradigma in esame. Gli approcci saranno analizzati singolarmente, saranno mostrati eventuali risultati e ne verranno evidenziati vantaggi e svantaggi, in relazione all'obiettivo finale di questo lavoro di tesi.

Sulla base degli approcci presenti in letteratura, la seconda parte di questo lavoro di tesi sarà invece incentrata sulla scelta degli approcci più adatti da utilizzare, partendo da quelli analizzati durante la prima parte, al fine di poter soddisfare l'obiettivo principale. Successivamente, si effettuerà una formulazione dell'approccio proposto, che sarà basato su entrambi i paradigmi, e la sua successiva implementazione e valutazione sperimentale. Infatti, per potere soddisfare l'obiettivo principale di questo lavoro di tesi, sarà necessario progettare un approccio che possa coniugare in maniera funzionale entrambi gli aspetti derivanti dai due paradigmi.

L'integrazione dei due paradigmi può essere fatto analizzando quale dei due aspetti sia effettivamente in grado di completare l'altro, cioè definire quale dei due paradigmi rappresenta effettivamente la base da cui partire per la progettazione e lo sviluppo dell'approccio finale.

La struttura completa del lavoro di tesi sarà la seguente: il Capitolo 1 introdurrà il lettore nello studio del paradigma di Secure Multi Party Computation, elencherà i principali contesti applicativi, definirà ulteriori paradigmi strettamente legati all'aspetto teorico del problema, elencherà i modelli implementativi a cui fare riferimento, tra cui il concetto di crittografia omomorfica, e infine descriverà gli approcci più promettenti presenti in letteratura; il Capitolo 2 introdurrà il lettore nello studio del paradigma di Truth Discovery, elencherà i principali contesti applicativi, definirà gli approcci generali di base e i modelli implementativi a cui fare riferimento, e infine descriverà gli approcci più promettenti presenti in letteratura, tra cui il framework CRH; il Capitolo 3 esporrà delle importanti considerazioni relativamente allo studio condotto sugli approcci analizzati nei due capitoli precedenti, esporrà il modello proposto e descriverà tutte le fasi di progettazione e implementazione di tale modello; il Capitolo 4 descriverà tutta fase di valutazione sperimentale del modello proposto, definendo il dataset utilizzato per questo scopo e discriminando le tipologie di valutazioni effettuate sul sistema; le Conclusioni esporranno un giudizio complessivo sul modello proposto, sulla base dei risultati sperimentali ottenuti e descritti nel Capitolo 4, evidenziando eventuali aspetti positivi e negativi, ed infine elencherà le possibili migliorie che possono essere incluse, nonché elencherà i possibili scenari di sviluppo futuro.

Capitolo 1

Secure Multi Party Computation

La *Secure Multi Party Computation* è un ramo della crittografia il cui obiettivo fondamentale è quello di creare dei metodi che consentano a diverse *parti* (o partecipanti), di calcolare congiuntamente una funzione sulla base dei loro input, mantenendoli privati.

A differenza della crittografia tradizionale, il cui obiettivo principale è quello di assicurare la sicurezza e l'integrità della comunicazione, mantenendo fuori dalla comunicazione dei due partecipanti l'eventuale *avversario*, nel modello di Secure Multi Party Computation l'obiettivo principale è proteggere la privacy tra tutti i partecipanti all'interno del protocollo.

Le radici per la Secure Multi Party Computation furono gettate verso gli albori degli anni '70, con la necessità di modellare dei sistemi di sicurezza che permettessero di formalizzare problemi di tipo *mental poker* [1], cioè una serie di problemi crittografici il cui obiettivo è quello di permettere ai partecipanti di giocare (o elaborare informazioni) a distanza, senza la necessaria presenza di una terza parte che verifichi o regolarizzi il gioco e l'elaborazione delle informazioni.

Successivamente, tali teorie si concretizzarono nella *Secure Two Party Computation*, che al giorno d'oggi rappresenta un caso particolare della Secure Multi Party Computation.

Il concetto di Secure Multi Party Computation fu introdotto formalmente nel 1982 da Andrew Yao in [2] e [3], dove enunciò il famoso *problema del milionario*, approfondito nella sezione 1.2.5, in cui m partecipanti desiderano calcolare congiuntamente una funzione del tipo $f(x_1, x_2, \ldots, x_m)$, dove x_i rappresenta l'input dell'i-esimo partecipante. Negli anni successivi, Yao introdusse il *Garbled Circuits Protocol*, che ha rappresentato la soluzione formale al problema del milionario e rimane ancora oggi la base da cui partire per implementare dei protocolli

efficienti che sfruttino il modello della Secure Multi Party Computation.

Nei successivi vent'anni, la Secure Multi Party Computation è stato un argomento "attrattivo" solo da un punto di vista teorico. Infatti, è solo durante i primi anni del 2000, grazie al miglioramento dell'efficienza degli algoritmi e al miglioramento delle risorse computazionali, che si iniziò a raggiungere un punto in cui diventò realistico pensare all'implementazione reale di sistemi general-purpose basati sul modello della Secure Multi Party Computation.

Fairplay [4] è considerata la prima implementazione degna di nota di un sistema general-purpose basato sul modello Secure Multi Party Computation. Infatti, l'implementazione Fairplay dimostrò la possibilità da parte di un programma di protezione della privacy, di essere espresso in un linguaggio ad alto livello, compilato ed infine eseguito, sulla base dei dati posseduti dai partecipanti, e utilizzando il modello della Secure Multi Party Computation. Nonostante tutto, la sua scalabilità e le sue performance erano abbastanza limitate, di conseguenza tale progetto è stato considerato, nel contesto del suo utilizzo, come "programma giocattolo".

A prescindere da ciò, la velocità di esecuzione dei protocolli basati sulla Secure Multi Party Computation è aumentata di diversi ordini di grandezza, grazie soprattutto alla combinazione di miglioramenti nella crittografia, nei protocolli, nelle reti di telecomunicazione e nell'hardware utilizzato. Tutto ciò ha consentito alle applicazioni basate sulla Secure Multi Party Computation di raggiungere finalmente un grande spazio di interesse, soprattutto per la sua importanza applicativa.

1.1 Definizione formale

Sia dato un numero di partecipanti $p_1, p_2, \ldots, p_N$, ognuno dei quali possiede dei dati privati, rispettivamente $d_1, d_2, \ldots, d_N$. L'obiettivo dei partecipanti è quello di calcolare una funzione che è nota a tutti, ma che lavori sui dati verificati di ciascun partecipante, cioè una funzione del tipo $F(d_1, d_2, \ldots, d_N)$, mantenendo sempre segreti i propri parametri di input.

A titolo di esempio, si supponga di avere tre partecipanti A, B e C, i cui input sono rispettivamente x, y e z. Si ipotizzi che i tre input rappresentino gli stipendi mensili dei tre partecipanti. L'obiettivo di ognuno dei tre partecipanti è quello di individuare il valore dello stipendio più alto, senza però rivelare agli altri il valore del proprio salario.

Da un punto di vista matematico, ciò si traduce nel voler calcolare la seguente funzione:

$$F(x, y, z) = \max(x, y, z) \tag{1.1}$$

Se all'interno di tale approccio fosse presente un utente esterno verificato T in grado di conoscere i valori degli stipendi di ogni partecipante, allora potrebbe calcolare il massimo dei tre valori e, successivamente, restituire il valore ottenuto ai tre partecipanti. Ma l'obiettivo della Secure Multi Party Computation è proprio quello di progettare un protocollo che permetta di scambiare messaggi solo con gli altri partecipanti, cioè solo tra gli utenti A, B e C, i quali conoscono solo la funzione $F(x, y, z)$, senza rivelare a nessuna altro, quindi nemmeno alla parte fidata T, la propria informazione.

L'aspetto interessante è che i partecipanti possono ottenere nuove informazioni confrontando tra di loro il valore dell'output della funzione e il proprio input. Se, ad esempio, l'output della funzione fosse z, allora uno dei tre partecipanti, si supponga C, scoprirà che il suo valore di stipendio è il massimo tra tutti quelli dei partecipanti; allo stesso modo, l'utente A e B scopriranno che i loro valori sono diversi dal valore massimo (ovviamente sotto l'ipotesi in cui i valori di x, y e z siano distinti).

Tale scenario può ovviamente essere generalizzato al caso in cui i partecipanti abbiano a disposizione diversi input e output e la funzione restituisca differenti output a differenti partecipanti.

In generale, le proprietà di base che un protocollo basato sulla Secure Multi Party Computation deve rispettare sono:

- *privacy degli input*: durante l'esecuzione del protocollo, non è possibile desumere alcuna informazione riguardo i dati privati conservati dai partecipanti. L'unica informazione che può essere dedotta sulla base dei dati privati è l'output della funzione;

- *correttezza*: qualsiasi sottoinsieme di partecipanti avversari disposti a condividere informazioni o a deviare dalle istruzioni durante l'esecuzione del protocollo non dovrebbe essere in grado in alcun modo di forzare i partecipanti onesti a produrre un risultato errato. La correttezza si pone quindi come obiettivo quello di garantire ai partecipanti onesti di ottenere sempre un output corretto (il protocollo deve quindi essere *robusto*), in caso contrario il protocollo deve essere arrestato.

1.2 Contesti applicativi

1.2.1 Vendita all'asta

La vendita all'asta è un contesto in cui il bisogno di privacy è ben comprensibile. Infatti, il poter fare affidamento sulla privacy e sulla non malleabilità delle offerte fatte durante l'asta diventa un aspetto cruciale per tutti i partecipanti, sia per gli offerenti che per i venditori.

La *privacy dell'offerta* richiede che nessun partecipante possa apprendere l'offerta di qualsiasi altro partecipante, mentre la *non malleabilità dell'offerta* indica che l'offerta di un partecipante non deve essere manipolata al fine di generare una nuova offerta affine.

Ad esempio, se un partecipante all'asta fa un'offerta pari a $\$n$, allora nessun altro partecipante dovrebbe essere in grado di utilizzare questa offerta per rilanciarne una nuova pari a $\$n + 1$. Bisogna notare che, in generale, la privacy dell'offerta non necessariamente implica la non malleabilità dell'offerta.

Queste proprietà sono cruciali in molti processi d'asta: ad esempio, un'*asta ad offerte sigillate* è un'asta in cui gli offerenti presentano offerte private (quindi sigillate) per potere acquistare delle proprietà da vendere al miglior offerente. Chiaramente, il valore dell'offerta del primo offerente deve essere tenuto segreto agli altri potenziali offerenti, per evitare che tali offerenti abbiano un vantaggio sleale. Allo stesso modo, la malleabilità dell'offerta può consentire a un offerente malevolo di presentare un'offerta leggermente al di sopra dell'offerta di un altro partecipante, ottenendo di nuovo un vantaggio sleale. Infine, l'asta stessa deve essere condotta correttamente, assegnando l'oggetto al miglior offerente per l'importo della sua offerta.

Un'*asta di Vickrey* è un tipo di asta ad offerta sigillata dove vince l'offerta più alta, ma viene pagato il prezzo indicato nella seconda massima offerta. Questo tipo di asta offre agli offerenti un incentivo a dichiarare il vero valore della propria offerta, ma richiede la privacy e la non malleabilità di ciascuna offerta, ma soprattutto correttezza nel determinare il vincitore e il prezzo.

In questo contesto, il modello della Secure Multi Party Computation può essere utilizzato per raggiungere facilmente tutte le caratteristiche elencate precedentemente e per fare in modo che tutti i partecipanti possano fare affidamento su tale protocollo per ottenere la massima sicurezza che l'asta sia condotta in modo confidenziale ed equo.

1.2.2 Votazioni elettroniche

Il contesto della votazione elettronica sicura può essere semplicemente rappresentato dal calcolo di una funzione additiva in grado di determinate il risultato del voto. Anche in questo caso, la privacy e la non malleabilità del voto (proprietà discusse precedentemente) sono essenziali per ragioni abbastanza simili a quelle dell'asta. Inoltre, poiché il voto è un processo civile di importanza fondamentale, queste proprietà sono spesso sostenute e confermate dalla legislazione.

Il problema principale all'interno del contesto del voto elettronico è la possibilità da parte degli elettori di poter dimostrare a terzi cosa hanno votato. Se una tale prova è consentita, allora è possibile che possa presentarsi una forma di *coercizione degli elettori*. Anche sotto questo aspetto, l'implementazione di protocolli basati sulla Secure Multi Party Computation è di enorme supporto.

1.2.3 Applicazioni mediche

Uno degli ambiti in cui la tutela della privacy è predominante è quello dell'assistenza sanitaria e del trattamento dei dati dei pazienti. Il contesto particolare è quello in cui i dati medici di un paziente sono caricati su una piattaforma di *cloud computing*: in questo caso, solo l'utente a cui fanno riferimento questi dati è il proprietario di quest'ultimi, pertanto questi dati, prima di essere conservati nella piattaforma cloud, devono essere cifrati in maniera tale che solo il paziente stesso possa decifrarli.

A partire da questi dati cifrati, che potrebbero contenere informazioni riguardo la pressione sanguigna, la frequenza cardiaca, il peso o il valore di glicemia nel sangue del paziente, la piattaforma di cloud computing (o comunque chiunque sia in possesso di questi dati) potrà essere in grado di effettuare delle accurate analisi predittive grazie ad approcci di machine learning, al fine, ad esempio, di prevedere la probabilità che si verifichino determinate patologie o, più in generale, solo per tenere traccia della salute dell'utente, il tutto senza che la piattaforma possa accedere direttamente a questi dati.

Il vantaggio maggiore, in termini di accuratezza dell'analisi predittiva, è ottenibile consentendo tale analisi sulla base delle letture di dati ottenibili da varie fonti, quindi analizzando le informazioni dei pazienti appartenenti a diverse strutture ospedaliere, senza dover divulgare questi dati a nessun altro elemento. Lavori degni di nota su questo ambito sono presenti in [5], [6] e [7].

1.2.4 Machine Learning

Il modello della Secure Multi Party Computation può essere utilizzato per ottenere privacy nelle fasi di inferenza statistica ed apprendimento dei sistemi di machine learning.

L'inferenza consente ad un client *C* di inviare una richiesta ad un server *S*, in possesso di un modello pre-addestrato di machine learning, mantenendo tale richiesta sconosciuta per il server *S*, e mantenendo il modello pre-addestrato sconosciuto per il client *C*. Con questa configurazione, gli input del protocollo basato sulla Secure Multi Party Computation sono il modello pre-addestrato di *S* e l'input del test da parte di *C*; infine l'output del protocollo, che verrà restituito solo al client *C*, sarà la previsione del modello pre-addestrato del server *S*.

Un esempio di lavoro recente che si basa su questa configurazione è rappresentato da *MiniONN* [8], che ha fornito un meccanismo per consentire a qualsiasi rete neurale standard di essere convertita in un servizio con un modello pre-addestrato, utilizzando una combinazione di Secure Multi Party Computation e tecniche di *crittografia omomorfica*.

Nella fase di apprendimento, la Secure Multi Party Computation può essere utilizzata per consentire a un gruppo di partecipanti di formare un modello basato sui loro dati combinati, senza esporli agli altri partecipanti del gruppo. Nonostante tutto, nel caso di grandi insiemi di dati, necessari per la maggior parte delle applicazioni di machine learning, l'addestramento basato su insiemi di dati privati e sulla Multi Computation diventa una cosa poco fattibile. Proprio per questo motivo sono stati progettati approcci ibridi che combinano la Secure Multi Party Computation con la crittografia omomorfica [9] [10] o che sviluppano protocolli personalizzati per eseguire operazioni aritmetiche sicure in modo efficiente [11]. Questi approcci possono adattarsi abbastanza bene ad insiemi di dati contenenti molti milioni di elementi, utili per il buon apprendimento.

1.2.5 Secure Two Party Computation

La Secure Two Party Computation è un sotto-problema della più generica Secure Multi Party Computation, il cui obiettivo è quello di creare un generico protocollo che permetta a due partecipanti distinti di calcolare in maniera aggregata una funzione arbitraria degli input forniti dalle due parti, senza che vi sia alcuna condivisione degli input tra le due parti opposte. Il più celebre esempio di Secure Two Party Computation è il *problema del milionario*, formulato da Andrew Yao in [1] nel 1982.

Il problema tratta di due milionari, interessati a conoscere chi dei due possiede la maggiore ricchezza, senza che sia svelata la ricchezza dei singoli.

Il problema è esprimibile in maniera più generale, in cui vi sono due numeri a e b e l'obiettivo è quello di determinare se la disuguaglianza $a \geq b$ è vera oppure falsa, senza quindi rivelare il proprio attuale valore di a o b.

Tale problema riscuote grande importanza all'interno della crittografia, la cui soluzione è spesso utilizzata nell'*e-commerce* e nel *data-mining*. Nel corso degli anni, le numerose soluzioni avanzate a questo problema, tra cui quella creata dallo stesso Yao, soffrono di un pesante costo computazione, sia in termini di tempo che di spazio, ed è di ordine esponenziale.

1.3 Modelli di interesse

In letteratura, esistono varie funzioni considerate particolarmente utili nella costruzione di protocolli basati sulla Secure Multi Party Computation. Possono essere definite come gli elementi essenziali per garantire la privacy tra i partecipanti e possono differenziarsi nel caso in cui ci trovassimo in un contesto di Secure Two Party Computation o Secure Multi Party Computation.

L'approccio basato sulla Secure Two Party Computation è particolarmente interessante non solo dal punto di vista delle applicazioni, ma anche per il fatto che questo modello permette di realizzare speciali tecniche che non possono essere utilizzabili nel caso più generale Multi Party. Basti pensare che il primo esempio di implementazione di Secure Multi Party Computation è stato realizzato proprio in un ambiente Two Party, come ad esempio il *protocollo garbled circuit* (o *circuito confuso*) di Yao descritto in [3].

La maggior parte dei protocolli basati sulla Secure Multi Party Computation, al contrario di quelli basati sulla Two Party Computation, utilizzano una sorta di *condivisione segreta*. Di conseguenza, i singoli partecipanti non svolgono dei ruoli speciali all'interno del protocollo, a differenza del Garbled Circuit di Yao, dove i partecipanti invece assumono i ruoli predefiniti di *sender* e *receiver*. Al contrario, i dati associati a ciascun ingresso del circuito logico sono condivisi tra tutti i partecipanti ed è necessario l'utilizzo di un ulteriore protocollo per valutare ciascuna porta logica.

Quindi, in un contesto di condivisione segreta, la funzione è definita come un circuito dove i valori utilizzabili ricadono all'interno di un campo finito, al contrario dei circuiti logici booleani utilizzati nell'approccio di Yao. In letteratura, questo tipo di circuito prende il nome di *circuito*

aritmetico ed è costituito da porte che effettuano operazioni di addizione e moltiplicazione i cui valori sono definiti su un campo finito. Ad esempio, la condivisione segreta consente di distribuire un dato segreto tra un numero di partecipanti, assegnando diversi "compiti" a ciascun partecipante. Generalmente, sono usati due tipi di schemi di condivisione segreta, cioè la *condivisione segreta di Shamir* e la *condivisione segreta additiva*. In entrambi i casi, i compiti sono numeri casuali appartenenti ad un campo finito che sono sommati in maniera segreta, secondo le proprietà del campo.

1.3.1 Crittografia omomorfica

La *crittografia omomorfica* è una forma di crittografia che consente il calcolo direttamente sui dati cifrati, generando un risultato cifrato che, una volta decifrato, equivale al risultato che si sarebbe ottenuto se si fossero eseguite le stesse operazioni sui valori in chiaro [12]. La crittografia omomorfica può essere utilizzata sia in un contesto di memorizzazione dei dati, sia nei casi in cui è necessario utilizzare risorse di calcolo esterne su dati di cui si vuole mantenere la riservatezza.

Il sistema omomorfico consente ai dati di essere cifrati ed essere così forniti a piattaforme cloud commerciali per l'elaborazione, mantenendo sempre la riservatezza dei dati. In settori altamente regolamentati, come l'assistenza sanitaria, la crittografia omomorfica può essere utilizzata per consentire l'utilizzo di nuovi servizi, rimuovendo le barriere sulla privacy dei dati, che inibiscono la condivisione e l'elaborazione degli stessi. Ad esempio, l'analisi predittiva nell'assistenza sanitaria può essere difficile da applicare proprio a causa di restrizioni dovute alla tutela della privacy dei dati sanitari dei pazienti, ma se il fornitore di servizi di analisi predittiva può operare su dati cifrati, tali problemi di privacy sono ridotti.

Quella omomorfica è quindi una forma di crittografia in grado di garantire una metodo di calcolo sui dati cifrati, senza l'esplicito utilizzo della chiave segreta per cifrare i suddetti dati, di conseguenza il risultato di tale calcolo rimane cifrato. La crittografia omomorfica può essere vista come ulteriore estensione nel mondo della crittografia, affiancandosi alle ben più comuni forme di crittografia a chiave simmetrica e a chiave pubblica.

Il termine "omomorfico" si riferisce all'omomorfismo dell'algebra astratta, cioè un'applicazione tra due strutture algebriche dello stesso tipo (ad esempio due gruppi, due anelli o due spazi vettoriali) in grado di preservare la loro struttura e le operazioni in esse definite [13]. Più formalmente, data un'applicazione del tipo $f : A \rightarrow B$, dove A e B sono due insiemi aventi la

stessa struttura algebrica, allora f è definibile un *omomorfismo* tra i due insiemi A e B se, assumendo che sia un operatore della struttura algebrica dei due insiemi, per ogni coppia di valori $(x, y) \in A$ vale che:

$$f(x \quad y) = f(x) \quad f(y) \tag{1.2}$$

Pertanto, le funzioni di cifratura e decifratura possono essere considerate come omomorfismi tra lo spazio del testo in chiaro e lo spazio del testo cifrato, nel caso si utilizzino sistemi crittografici omomorfici.

La crittografia omomorfica include diversi tipi di schemi di cifratura, in grado di eseguire diverse tipologie di operazioni matematiche sui dati crittografati [14]. Alcuni tipi comuni di sistemi omomorfici sono:

- *sistemi partially homomorphic*: questi sistemi prevedono una sola tipologia di operazioni sui dati cifrati, ad esempio addizione o moltiplicazione;
- *sistemi somewhat homomorphic*: questi sistemi prevedono il calcolo di due tipologie di operazioni, ma soltanto su un sotto-insieme di problemi;
- *sistemi leveled fully homomorphic*: questi sistemi prevedono il calcolo di qualsiasi funzione arbitraria, ma soltanto su un sotto-insieme di problemi;
- *sistemi fully homomorphic* (FHE): questi sistemi prevedono il calcolo di qualsiasi funzione arbitraria su un qualsiasi insieme di problemi, e ovviamente sono i sistemi omomorfici più potenti.

Per la maggior parte degli schemi di crittografia omomorfica, la profondità moltiplicativa dei circuiti è la principale limitazione pratica nell'esecuzione di calcoli su dati cifrati. Nel contesto di questo lavoro di tesi, sono di maggior rilievo i sistemi partially homomorphic, sia per il loro basso costo computazionale, sia perché utilizzano sistemi di cifratura già molto consolidati e ampiamente utilizzati in letteratura e, in generale, nel mondo della crittografia. Alcuni esempi di sistemi partially homomorphic sono:

- algoritmo *RSA* (sistema moltiplicativo) [15]: sia n il valore con cui sono state calcolate la chiave pubblica e privata e sia e l'esponente da utilizzare per la cifratura; considerando

che in RSA la cifratura di un messaggio m è data da $E(m) = m^e \bmod n$, allora l'algoritmo RSA soddisfa la seguente proprietà omomorfica:

$$E(m_1) \cdot E(m_2) = m_1^e \cdot m_2^e \bmod n = \tag{1.3}$$

$$= (m_1 \cdot m_2)^e \bmod n = \tag{1.4}$$

$$= E(m_1 \cdot m_2) \tag{1.5}$$

- algoritmo *ElGamal* (sistema moltiplicativo) [15]: sia (q, a, Y_a) la chiave pubblica, dove $Y_a = a^{X_a} \bmod q$ e sia X_a la chiave privata; considerando che in ElGamal la cifratura di un messaggio m è data da $E(m) = (a^r \bmod q,\ m \cdot Y_a^r \bmod q)$, dove r è un valore casuale con $r \in \{0, \ldots, q-1\}$, allora l'algoritmo ElGamal soddisfa la seguente proprietà omomorfica:

$$E(m_1) \cdot E(m_2) = (a^{r_1} \bmod q, m_1 \cdot Y_a^{r_1} \bmod q) \cdot (a^{r_2} \bmod q, m_2 \cdot Y_a^{r_2} \bmod q) = \tag{1.6}$$

$$= (a^{r_1+r_2} \bmod q, (m_1 \cdot m_2) \cdot Y_a^{r_1+r_2} \bmod q) = \tag{1.7}$$

$$= E(m_1 \cdot m_2) \tag{1.8}$$

- algoritmo *Goldwasser–Micali* (sistema additivo modulo 2) [16] [17]: sia x la chiave pubblica, cioè il residuo non quadratico mod n; considerando che in Goldwasser–Micali la cifratura di un bit b di messaggio è data da $E(b) = x^b \cdot r^2 \bmod n$, dove r è un valore casuale con $r \in \{0, \ldots, n-1\}$, allora l'algoritmo Goldwasser–Micali soddisfa la seguente proprietà omomorfica:

$$E(b_1) \cdot E(b_2) = (x^{b_1} \cdot r_1^2) \cdot (x^{b_2} \cdot r_2^2) \bmod n = \tag{1.9}$$

$$= x^{b_1+b_2} \cdot (r_1 \cdot r_2)^2 \bmod n = \tag{1.10}$$

$$= E(b_1 \quad b_2) \tag{1.11}$$

- algoritmo di *Paillier* (sistema additivo) [18]: sia (n, g) la chiave pubblica; considerando che in Paillier la cifratura di un messaggio m è data da $E(m) = g^m \cdot r^n \bmod n^2$, dove r è un valore casuale con $r \in \{0, \ldots, n - 1\}$, allora l'algoritmo di Paillier soddisfa la seguente proprietà omomorfica:

$$E(m_1) \cdot E(m_2) = (g^{m_1} \cdot \underset{1}{r^n}) \cdot (g^{m_2} \cdot \underset{2}{r^n}) \bmod n^2 = \tag{1.12}$$

$$= g^{m_1+m_2} \cdot (r_1 \cdot r_2)^n \bmod n^2 = \tag{1.13}$$

$$= E(m_1 + m_2) \tag{1.14}$$

L'algoritmo di Paillier verrà approfondito nella prossima sezione, in 1.3.2,

1.3.2 Sistema omomorfico additivo di Paillier

Il *sistema omomorfico additivo di Paillier* è un algoritmo di cifratura a chiave asimmetrica probabilistico ideato da Pascal Paillier nel 1999 [18]. La caratteristica principale è quella di essere un sistema omomorfico additivo, cioè permette di ottenere la somma di due valori cifrati o il prodotto di un valore cifrato per un valore in chiaro, dove il risultato delle operazioni corrisponde alla versione cifrata del risultato che si avrebbe se si effettuassero le stesse operazioni sui valori in chiaro. Il sistema crittografico di Paillier segue il seguente schema semplificato:

$$E(a) \cdot E(b) = E(a + b) \tag{1.15}$$

$$E(a)^b = E(ab) \tag{1.16}$$

Lo schema dell'algoritmo di Paillier è composto da tre fasi principali: generazione della chiave, cifratura e decifratura.

Generazione della chiave

Siano due interi primi p e q molto grandi tali che:

$$\gcd(pq, (p-1)(q-1)) = 1 \tag{1.17}$$

La 1.17 garantisce che entrambi i numeri primi scelti siano dello stesso ordine di grandezza. Successivamente, si calcola il prodotto $n = p \cdot q$ e la *funzione di Carmichael* definita come:

$$l = \text{lcm}(p-1, q-1) \tag{1.18}$$

Nella 1.18 lcm rappresenta il *minimo comune multiplo* tra i valori $p-1$ e $q-1$. Successivamente, si seleziona un intero $g \in Z_{n^2}$ tale che:

$$\gcd(L(g^l \bmod n^2), n) = 1 \tag{1.19}$$

La 1.19 assicura che n e il valore ottenuto da $L(g^l \bmod n^2)$ siano coprimi, dove L è una funzione del tipo:

$$L(u) = \frac{u-1}{n} \tag{1.20}$$

In questo modo è possibile garantire che esista il seguente inverso moltiplicativo in n:

$$\mu = L^{-1}(g^l \bmod n^2) \tag{1.21}$$

Alla fine, la *chiave pubblica* è rappresentata da $p_k = (n, g)$, mentre la *chiave privata* è rappresentata da $s_k = (l, \mu)$.

Cifratura

Si ipotizzi di voler cifrare il valore $m \in Z_{n^2}$, allora il corrispondente valore cifrato c è ottenuto come:

$$c = g^m r^n \bmod n^2 \tag{1.22}$$

dove r è un valore casuale con $0 < r < n$ e $r \in Z_{n^2}$ e $\gcd(r, n) = 1$.

Decifratura

Si ipotizzi di voler decifrare il valore $c \in Z_{n^2}$, allora il corrispondente valore decifrato m è ottenuto come:

$$m = \frac{L(c^l \bmod n^2)}{L(g^l \bmod n^2)} \bmod n \tag{1.23}$$

Proprietà

Dato che questa funzione di cifratura è *omomorfica additiva*, è possibile descrivere le seguenti identità:

- *I proprietà*: il prodotto di 2 valori cifrati sarà decifrato come la somma dei loro corrispondenti valori in chiaro:

$$D(E(m_1, r_1) \cdot E(m_2, r_2) \bmod n^2) = m_1 + m_2 \bmod n \tag{1.24}$$

  Inoltre, il prodotto di un valore cifrato per un valore in chiaro che eleva g sarà decifrato come la somma dei corrispondenti valori in chiaro:

$$D(E(m_1, r_1) \cdot g^{m2} \bmod n^2) = m_1 + m_2 \bmod n \tag{1.25}$$

- *II proprietà*: un valore cifrato elevato ad un valore in chiaro, sarà decifrato come il prodotto dei due valori in chiaro:

$$D(E(m_1, r_1)^{m2} \bmod n^2) = m_1 \cdot m_2 \bmod n \tag{1.26}$$

$$D(E(m_2, r_2)^{m1} \bmod n^2) = m_1 \cdot m_2 \bmod n \tag{1.27}$$

  Più in generale, un valore cifrato elevato ad una costante k sarà decifrato nel prodotto tra il valore in chiaro e la costante:

$$D(E(m_1, r_1)^k \bmod n^2) = k \cdot m_1 \bmod n \tag{1.28}$$

1.3.3 Condivisione segreta di Shamir

L'algoritmo di condivisione segreta di Shamir è un algoritmo di crittografia creato da Adi Shamir e può essere ricondotto ad una forma di *condivisione segreta*, cioè quella tecnica in cui un dato segreto viene diviso in varie parti, assegnando ad ogni partecipante la propria parte. Per ricostruire il dato segreto originale, è richiesto comunque un numero minimo di partecipanti, ed in genere questo numero è minore del numero totale di partecipanti.

L'algoritmo di Shamir viene utilizzato per rendere sicura un'informazione segreta in maniera distribuita, molto spesso con lo scopo di rendere sicure altre chiavi di cifratura. L'informazione da tutelare viene quindi suddivisa in più parti, denominate *share*, utilizzate per la ricostruzione. Per ricostruire l'informazione segreta di partenza, è necessario un numero minimo di share denominato *threshold*.

Ad esempio, nel caso in cui una società debba proteggere la password per accedere ai propri dati personali, può facilmente utilizzare un algoritmo standard come l'AES, ma cosa succederebbe se il titolare della chiave non fosse disponibile o morisse? O più in generale, cosa potrebbe succedere se l'informazione relativa alla chiave non fosse più disponibile, o in mano di avversari della società?

In questo caso, l'algoritmo di Shamir può essere utilizzato per cifrare la password e generare un certo numero di share, che possono essere assegnate a ciascun dirigente all'interno della società. In questo modo, solo se i vari dirigenti uniscono le proprie informazioni, sarà possibile ottenere la decifratura della chiave. La threshold può essere impostata in modo appropriato per il numero di dirigenti, in modo che le sole persone autorizzate possano accedere alle informazioni segrete della società. Di conseguenza, se una o due share cadessero nelle mani sbagliate, non potrebbero decifrare la password, a meno che gli altri dirigenti non cooperino alla decifratura.

Da un punto di vista matematico, l'algoritmo prevede di dividere un segreto S in n share di dati del tipo $S_1, \ldots, S_n$ tali che:

- la conoscenza di k o più share S_i renda semplice l'elaborazione di S. In questo modo, l'informazione completa segreta S può essere ricostruita facilmente dalla combinazione delle k share di dati;
- la conoscenza di $k - 1$ o meno share S_i, lascia l'informazione di S del tutto indeterminata. Di conseguenza, l'informazione segreta S non può essere ricostruita con meno di k share.

L'algoritmo di Shamir possiede anche una "visione" geometrica del problema, infatti sulla base dello schema di threshold adottato, due punti sono sufficienti per ricostruire una retta, tre punti sono sufficienti per ricostruire una parabola, quattro punti sono sufficienti per ricostruire una curva cubica e così via. In altri termini, presi k punti è possibile definire un polinomio di grado $k - 1$.

Si supponga di voler usare uno schema di threshold del tipo (k, n) per condividere l'informazione segreta S, che rappresenta un elemento in un campo finito F di dimensione P dove $0 < k \leq n < P; S < P$ e P è un intero primo.

Si scelgano casualmente $k - 1$ interi positivi del tipo $a_1, \ldots, a_{k-1}$ con $a_i < P$ e sia $a_0 = S$; successivamente, si costruisca il polinomio $f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + ... + a_{k-1}x^{k-1}$. Dal polinomio costruito, si estragga una serie di n punti, del tipo $i = 1, 2, \ldots, n$, in modo da ottenere le coppie $(i, f(i))$. Ad ogni partecipante, viene assegnato quindi un punto (cioè un intero non nullo come input del polinomio e il corrispondente output del polinomio), oltre all'intero primo che definisce il campo finito su cui lavorare. Dato un qualsiasi sottoinsieme di k di queste coppie, è possibile definire i coefficienti del polinomio usando un'interpolazione. L'informazione segreta sarà contenuta all'interno del coefficiente a_0.

In genere, gli schemi basati sulla condivisione segreta possono tollerare un partecipante avversario in grado di controllare fino a t partecipanti su un totale di n partecipanti, in cui t varia in base allo schema di threshold utilizzato, se l'avversario è passivo o attivo oppure se vi sono diverse ipotesi sul "potere" dell'avversario.

Nel caso della condivisione segreta di Shamir, lo schema è sicuro contro un avversario passivo quando $t < \frac{n}{2}$, e un avversario attivo quando $t < \frac{n}{3}$, riuscendo a mantenere la sicurezza nelle informazioni. Tutto ciò significa che, anche se l'avversario avesse un potere computazionale illimitato, non può ugualmente apprendere alcuna informazione segreta all'interno della condivisione.

Le principali proprietà che uno schema di threshold dovrebbe avere, all'interno dell'algoritmo di Shamir, sono:

- *sicurezza*: deve essere garantita la sicurezza teorica dell'informazione da proteggere;
- *minimalismo*: la dimensione di ogni share non deve eccedere la dimensione del dato originale;
- *estensibilità*: mentre k viene mantenuto fisso, le D_i share possono essere dinamicamente aggiunte o rimosse, senza influenzare le altre share;

- *dinamicità*: la sicurezza può essere facilmente raggiunta senza cambiare l'informazione segreta, ma cambiando occasionalmente il polinomio costruito e generando nuove share per i partecipanti;
- *flessibilità*: nelle organizzazioni in cui la gerarchia è importante, è possibile fornire ad ogni partecipante un numero di share diverse, sulla base dell'importanza del partecipante all'interno dell'organizzazione (il presidente di una società può aprire la cassaforte da solo, oppure sono necessari tre o più impiegati insieme).

Il protocollo *BGW*, descritto in [19], definisce le metodologie su come calcolare l'addizione e la moltiplicazione all'interno di condivisioni segrete e viene spesso utilizzato come implementazione della condivisione segreta di Shamir.

1.3.4 Condivisione segreta additiva

Gli schemi basati sulla condivisione segreta additiva possono tollerare avversari in grado di controllare tutti i partecipanti, cioè nel caso in cui $t < n$, riuscendo a mantenere la sicurezza nelle informazioni, anche contro un avversario passivo e attivo con potere computazionale illimitato. Alcuni protocolli richiedono una fase di inizializzazione, che può essere sicura solo contro un avversario limitato dal punto di vista computazionale.

Tra tutte le implementazioni basate sul modello della condivisione segreta, utilizzando l'approccio della Secure Multi Party Computation, la più popolare è *SPDZ*, descritto in [20].

1.3.5 Threshold Paillier

In letteratura, esiste una variante del sistema omomorfico additivo di Paillier, studiata durante diverse implementazioni in [21], [22] e [23], che prende il nome di *Threshold Paillier*, spesso indicata con (p, t)-*Threshold Paillier*. Questa variante, oltre a mantenere le tipiche proprietà di omomorfismo dell'algoritmo originale circa le operazioni nel dominio cifrato, possiede la peculiare caratteristica di permettere la decifratura di un qualsiasi valore cifrato solamente nel caso in cui si posseggano almeno t tra tutte le p chiavi private distribuite ad ogni partecipante del protocollo. Infatti, come nei comuni cifrari asimmetrici, anche nel Threshold Paillier sono generate una chiave pubblica (utilizzata per la cifratura) e una chiave privata (utilizzata per la decifratura), ma con l'importante differenza che la chiave privata generata viene "suddivisa"

in p parti, dove p rappresenta proprio il numero di partecipanti al protocollo, ed ogni parte successivamente distribuita casualmente ad ogni singolo partecipante. Di conseguenza, affinché si possa decifrare un valore cifrato con la chiave pubblica, saranno necessarie almeno t chiavi private con $t \leq p$: sono quindi necessari almeno t partecipanti attivi al protocollo affinché si possa decifrare un particolare valore cifrato. In questo contesto, la decifratura "parziale" ottenuta da un partecipante prende il nome di *share*, come nella condivisione segreta di Shamir. Una volta che ogni partecipante calcola la propria share, queste dovranno essere "combinate" tra loro per ottenere il valore in chiaro finale. L'algoritmo a cui faremo riferimento è quello descritto in [21].

La variante Threshold Paillier, rispetto all'originale, conserva dietro di se una pesante trattazione matematica, che coinvolge anche la condivisione segreta di Shamir. È sufficiente dire che la variante prevede l'utilizzo di quattro numeri primi p, p', q, e q' (e non solo due come nell'algoritmo originale), tali che $p = 2p' + 1$ e $q = 2q' + 1$; sono successivamente calcolati i prodotti $n = pq$ e $m = p'q'$.

Viene quindi costruita una funzione polinomiale del tipo:

$$f(X) = \sum_{i=0}^{k-1} a_i X^i \bmod n * m \tag{1.29}$$

Nella (1.29) i coefficienti a_i sono valori casuali presi all'interno dell'insieme $\{0, \ldots, n * m - 1\}$ con $a_0 = d$, dove d rappresenta un intero casuale tale che $d \equiv 0 \bmod m$ e $d \equiv 1 \bmod n$. In questo caso, la chiave privata del i-esimo partecipante sarà data dall'elemento $s_i = f(i)$ con $1 \leq i \leq l$, dove l rappresenta il numero di partecipanti al protocollo. La chiave pubblica rimane invariata a quella dell'algoritmo di Paillier originale. Anche la cifratura in Threshold Paillier rimane identica a quella dell'algoritmo originale.

La differenza rispetto all'algoritmo originale è rappresentata principalmente dalla parte di decifratura: in questo caso l'i-esimo partecipante potrà calcolare la propria share c_i come segue:

$$c_i = c^{2\Delta s_i} \tag{1.30}$$

Nella (1.30), la Δ equivale al fattoriale del numero di partecipanti al protocollo, cioè $\Delta = l!$. La parte finale prevede infine la combinazione delle singole c_i da parte del server, tramite l'algoritmo descritto in [21].

1.4 Sistema per il voto elettronico

Le soluzioni per il voto elettronico di solito mirano ad aumentare la partecipazione degli elettori e a migliorare i risultati delle elezioni, affrontando i problemi relativi ai sistemi di voto tradizionali. Il sistema di voto elettronico consiste nella registrazione, autenticazione, votazione, conteggio dei voti e verifica del conteggio dopo il voto. Inoltre, la rapida crescita dei sistemi di autenticazione ha cambiato la visione delle persone verso il modo in cui questi sistemi gestiscono i dati, di conseguenza fornire una sicurezza al sistema di voto è una questione di grande importanza.

Il modello proposto da P. Sanyasi Naidu et al. in [24] si pone l'obiettivo di raggiungere l'autenticità e la sicurezza nella non rintracciabilità e anche il rispetto della riservatezza del voto. Tale approccio viene proposto combinando lo schema biometrico con uno schema di *Visual Cryptography* basato sui modelli di Naor e Shamir, dove lo schema biometrico ha il compito di occuparsi di metodologie per la certificazione dell'identità di una persona in base a caratteristiche comportamentali o fisiologiche.

I sistemi di crittografia più comunemente utilizzati all'interno dei sistemi di voto elettronico sono tecniche interattive di condivisione segreta, *protocolli a conoscenza zero*, *blind signature* e utilizzo della Secure Multi Party Computation. La condivisione segreta viene utilizzata nel voto elettronico al fine di rendere robusto il protocollo di voto contro il controllo delle varie autorità. Il sistema proposto fa uso delle caratteristiche della Secure Multi Party Computation per il conteggio finale dei voti. Infine, l'approccio utilizza anche uno schema di condivisione segreta per autenticare la registrazione degli elettori.

1.4.1 Modelli utilizzati

La *Visual Cryptography*, descritta in [25], è un tipico schema crittografico in grado di decodificare le immagini cifrate senza calcoli crittografici, cioè consente di cifrare le informazioni visive in modo tale che la decifratura possa essere eseguita dal sistema visivo umano, senza l'ausilio di un calcolatore. Lo schema di Visual Cryptography elimina i costosi calcoli computazionali nel processo di decifratura delle immagini cifrate. Questa proprietà rende la Visual Cryptography particolarmente utile ove i requisiti di calcolo sono molto stringenti.

L'*e-voting*, descritto in [26], è un particolare sistema di voto basato sul web che consente agli elettori di votare indipendentemente dalla posizione in cui si trovano. L'obiettivo è quel-

lo di fornire sicurezza a qualsiasi dato o informazione, e ciò è possibile offrendo agli utenti un'autenticazione sicura attraverso lo schema di condivisione segreta di Shamir, descritto in 1.3.3.

1.4.2 Descrizione del sistema

L'obiettivo principale del voto è consentire agli elettori di esercitare il loro diritto di esprimere le proprie scelte su questioni specifiche, atti legislativi, iniziative dei cittadini, emendamenti costituzionali o scegliere il proprio governo e i rappresentanti politici. Nel sistema proposto in [24], l'utente deve registrare la propria impronta digitale per ottenere una propria ID card; successivamente, lo scanner per impronte digitali creerà l'immagine per tale impronta digitale.

L'approccio si svolge su quattro fasi:

- registrazione dell'elettore;
- autenticazione dell'elettore;
- votazione e registrazione della votazione;
- conteggio dei voti e pubblicazione dei risultati.

Registrazione dell'elettore

Ogni elettore deve iscriversi alla votazione, al fine di esercitare i propri diritti di voto. All'interno del modulo di iscrizione, verrà utilizzato lo scanner di impronte digitali per scansionare l'impronta di ciascun elettore. Successivamente, verrà utilizzato l'algoritmo di condivisione segreta, in modo da iscrivere e autenticare gli elettori: questa fase comprenderà due parti, cioè la *costruzione condivisa* e la *ricostruzione segreta*.

L'algoritmo di costruzione condivisa viene utilizzato per costruire due *share* (vedi 1.3.3) dell'immagine dell'impronta digitale. Una share verrà archiviata nel database e un'altra share verrà memorizzata nella ID card dell'elettore. In questo modo, sia il database che la ID card dell'elettore non conterranno l'immagine reale della sua impronta digitale: infatti, finché non saranno disponibili entrambe le share, non sarà possibile ricostruire l'immagine originale dell'impronta digitale. Quindi, ogni elettore possiederà una ID card contenente le informazioni private dell'elettore e una parte dell'impronta digitale.

Autenticazione dell'elettore

Durante la fase di autenticazione, l'elettore deve fornire la propria ID card contenente una parte dell'immagine dell'impronta digitale, quindi un lettore leggerà la share della ID card, mentre l'altra share dell'impronta digitale è memorizzata nel database. Queste due share sono fornite all'algoritmo di ricostruzione segreta in modo che sia ricostruita l'immagine dell'impronta digitale originale dell'elettore. Nello stesso momento, uno scanner acquisirà un'immagine aggiornata dell'impronta digitale dell'elettore e verrà confrontata con l'immagine ricostruita. Se le due immagini corrispondono, allora l'elettore è autenticato, in caso contrario non gli sarà permesso di votare.

Al fine del confronto delle due immagini, si ricorre all'*algoritmo di estrazione dei dettagli*, utilizzato e descritto in [27], cioè dati due insiemi di dettagli, uno ottenuto dall'immagine ricostruita e l'altro dalla scansione diretta dell'utente da autenticare, l'algoritmo di estrazione dei dettagli determinerà se i due insiemi di dettagli sono uguali o meno.

Votazione e registrazione della votazione

All'interno dell'applicativo per il voto, sono presenti dei pulsanti per ogni candidato, o più in generale, per ogni opzione che l'utente può selezionare. Ad esempio, nel caso in cui l'elettore prema il pulsante numero 2, verrà generato un segnale binario in cui verrà impostato il secondo bit, cioè 01000; quindi l'i-voto può essere espresso nella forma:

$$V_i = [V_{i1}, V_{i2}, \ldots, V_{iN}] \tag{1.31}$$

Nella (1.31), $V_{ij} = 1$ se viene scelto il j-candidato, altrimenti $V_{ij} = 0$.

Nonostante tutto, all'interno del database questo voto non viene memorizzato esattamente nella stessa forma binaria. Sia d un numero maggiore del numero totale di elettori in una determinata area; sia n il numero del candidato o dell'opzione scelta durante il voto; infine siano $R_1, R_2, \ldots, R_n$ dei n numeri casuali, uno per ogni candidato/opzione. Se, ad esempio, l'elettore selezionasse il secondo candidato/opzione, il suo voto verrà archiviato nel database nella forma:

$$(R_1 \cdot d), (1 + R_1 \cdot d), (R_1 \cdot d), \ldots, (R_1 \cdot d) \tag{1.32}$$

Quindi, il database dei voti espressi dagli utenti sarà simile a quello indicato nella tabella della figura 1.1.

$Candidate_1$	$Candidate_2$	.....	$Candidate_n$
$(0+R_{11}*d)$	$(0+R_{12}*d)$	.....	$(1+R_{1n}*d)$
$(0+R_{21}*d)$	$(1+R_{22}*d)$	.....	$(0+R_{2n}*d)$
$(1+R_{31}*d)$	$(0+R_{32}*d)$	.....	$(0+R_{3n}*d)$
. . .	. . .	....	. . .
$(1+R_{m1}*d)$	$(0+R_{m2}*d)$	.....	$(0+R_{mn}*d)$

Figura 1.1: Pubblicazione dei risultati dell'e-voting.

Conteggio dei voti e pubblicazione dei risultati

Ogni colonna all'interno del database rappresenta i voti ottenuti da ogni candidato/opzione, di conseguenza il totale dei voti ottenuto dall'i candidato/opzione è calcolato come $C_i = Y_i \,\%\, d$, dove Y_i è ottenuto come:

$$Y_i = [(V_{1i} + R_{1i} \cdot d) + (V_{2i} + R_{2i} \cdot d) \ldots + (V_{ni} + R_{ni} \cdot d)] \tag{1.33}$$

Considerazioni

Poiché due elettori non potranno mai avere le stesse caratteristiche biometriche, allora la sicurezza basata su questi sistemi è un ottimo approccio all'interno di votazioni elettroniche. Il problema della falsificazione viene superato, poiché l'autorizzazione di un utente avrà successo solo quando fornirà la sua ID card assieme ad una nuova immagine dell'impronta digitale. La riservatezza delle informazioni biometriche degli elettori viene garantita archiviando le share dell'immagine dell'impronta digitale in differenti zone. Infine, con il supporto della Secure Multi Party Computation, viene superato anche il problema della falsificazione dei risultati.

1.5 Privacy preserving nell'analisi dei segnali

Confrontare due segnali è uno dei compiti più significativi e rilevanti all'interno dell'elaborazione dei segnali, ma comunque anche nel contesto più generale dell'analisi dei dati. Il problema posto dagli autori di [28] è il seguente: dato un insieme di segnali e uno di riferimento scelto come modello, tra tutti i segnali, quali sono quelli più "vicini" al modello appena scelto?

Più formalmente, il precedente quesito può essere espresso in termini matematici: dato un insieme *S* contenente un generico numero di punti appartenenti ad uno spazio metrico *M* e un punto di riferimento $x \in M$, l'obiettivo è trovare quel punto appartenente ad *S* tale che risulti essere il più vicino al punto di riferimento x. Questo problema prende il nome di *Nearest Neighbor* (NN) e può essere generalizzato nel caso in cui si volessero trovare *K* punti, parlando quindi di K-NN.

Nel contesto analizzato dagli autori di [28], i suddetti punti possono rappresentare segnali di ogni tipologia, quindi immagini, video, suoni e qualsiasi altra forma di dato, mentre il concetto di "più vicino" può essere espresso in termini di distanza metrica, come la distanza euclidea o la distanza di Manhattan tra due punti scelti all'interno dello spazio metrico.

Sempre in questo contesto, se trovare il punto vicino tra tutti quelli presenti all'interno del dominio può essere un problema affrontabile facilmente, ciò non si può dire nel caso in cui i punti in questione debbano essere mantenuti privati, cioè non debbano essere rilevati al partecipante del sistema che conduce l'elaborazione per la risoluzione del problema. Sulla base di questo vincolo, gli autori si pongono il problema di capire se sia possibile calcolare la distanza tra due segnali senza conoscere nulla di loro, e di conseguenza se sia possibile determinare la minima di queste distanze.

Lo scopo della trattazione fatta dagli autori di [28] è quello di fornire un approccio di base in grado di potere rispondere in maniera affermativa ai problemi posti precedentemente, riassumendolo in un unico approccio denominato *Privacy-Preserving Nearest Neighbor* (PPNN), capace di sintetizzare in due macro-modelli il problema formulato dagli autori: il metodo *Privacy-Preserving* (PP) ha come obiettivo quello di calcolare tutte le singole distanze dei punti presenti all'interno dell'insieme *S*, mantenendo la privacy e la segretezza di questi punti, mentre il metodo *Nearest Neighbor* (NN) ha come obiettivo quello di trovare la minima tra tutte le distanze calcolate dal metodo PP.

I partecipanti al protocollo possono comportarsi sia seguendo il paradigma del partecipante *semi-onesto*, cioè il partecipante segue le regole del protocollo e sfrutta tutte le informazioni rese

disponibili per cercare di indovinare le informazioni generate dagli altri partecipanti, sia seguendo il paradigma del partecipante *malevolo*, cioè il partecipante può discostarsi arbitrariamente dalle regole del protocollo con lo scopo di rilevare le informazioni generate dagli altri partecipanti e scegliere di colludere con un altro partecipante con l'obiettivo di forzare il protocollo a restituire un risultato errato o di compromettere i dati di proprietà di una partecipante ignaro. Ovviamente, implementare un protocollo che possa contrastare la presenza di partecipanti malevoli risulta molto più oneroso, rispetto al caso più semplice di partecipanti semi-onesti, di conseguenza gli autori di [28] assumono di trattare solo questa particolare condizione, motivazione dettata dal fatto che gran parte delle ricerche presenti in letteratura sull'elaborazione sicura dei segnali ha preso quasi sempre in considerazione la presenza di soli partecipanti semi-onesti, e non malevoli. Successivamente, gran parte di questi approcci possono in seguito essere "rafforzati" per accogliere partecipanti malevoli tramite approcci di condivisione segreta, descritta in 1.3.3, sempre se si possa accettare l'incremento del costo computazione che si avrebbe adottando queste tipologie di modelli.

In sintesi, il modello proposto dagli autori deve inizialmente calcolare in maniera cifrata le distanze tra tutte le coppie di segnali ottenibili a partire da un database di segnali Y presenti in un server, e un segnale di query x proposto da un client. Successivamente, ottenuto l'insieme di tutte le distanze cifrate, il server dovrà essere in grado di trovare la minima tra tutte le distanze calcolate.

1.5.1 Strumenti di calcolo cifrato

Lo strumento di calcolo crittografico più adatto per ottenere le distanze cifrate tra tutti i punti è la crittografia omomorfica a chiave pubblica. Come già descritto in 1.3.2, un sistema crittografico omomorfico possiede l'importante proprietà che il calcolo in maniera cifrata di una funzione di alcune variabili, può essere ottenuto eseguendo operazioni direttamente sulle versioni cifrate delle variabili, quindi per un generico modello omomorfico vale che:

$$E(a) \cdot E(b) = E(a + b) \tag{1.34}$$

$$E(a)^b = E(ab) \tag{1.35}$$

Uno dei risultati degni di nota all'interno del mondo della crittografia degli ultimi anni è

stato quello di provare l'esistenza della crittografia completamente omomorfica, che consente il calcolo cifrato di funzioni polinomiali arbitrarie con diverse variabili [29] [30] [31].

Tra tutti i possibili sistemi di crittografia omomorfica, gli autori prediligono quello additivo, perché tale sistema è stato il primo ad essere utilizzato per altri modelli basati sul PPNN, come quelli presenti in [32] [33], ed è di facile implementazione. Si assume inoltre che sia il client che il server utilizzino una chiave pubblica compatibile con il sistema omomorfico.

Ottenuto l'insieme delle distanze cifrate tra i vari punti, la scelta del punto più vicino a quello scelto dal client implica innanzitutto la ricerca della distanza più piccola tra tutte quelle calcolate, quindi è necessario restituire questo risultato al client, sempre in maniera cifrata. Per ottenere ciò, gli autori decidono di ricorrere al protocollo *Oblivious Transfer*, descritto per la prima volta in [34].

Nel contesto del PPNN, l'*Oblivious Transfer* può essere utilizzato per essere integrato in un protocollo basato sul problema del milionario 1.2.5, cioè dati due input *a* e *b* ottenuti da due partecipanti non attendibili, il protocollo sarà in grado di rilevare quale partecipante ha dichiarato il valore maggiore, senza rivelare ad ogni singolo partecipante i valori dichiarati dagli altri partecipanti. Tale protocollo è onnipresente ogni volta che devono essere effettuati confronti, quindi ricerche di valori massimi o minimi sotto vincoli stringenti di privacy.

OMISSIS

1.5.2 Rilevamento della distanza minima

Una volta ottenute tutte le distanze cifrate del tipo $E(d(\mathrm{x}, \mathrm{y}^i))$ tra un segnale modello scelto e tutti i segnali presenti nel database del server, il prossimo passo è quello di identificare la distanza minore, cioè, sempre in maniera privata, trovare un valore $\mathrm{y}^{(NN)}$ tale che $d(\mathrm{x}, \mathrm{y}^{(NN)}) \leq d(\mathrm{x}, \mathrm{y}^i)$ per qualsiasi valore di i.

Poiché i calcoli per ottenere le distanze tra i punti sono stati eseguiti rispettano la privacy prima del rilevamento della distanza minima, è fondamentale che il protocollo di rilevamento della distanza minima non riveli l'indice $i^* = \mathrm{argmin}_i\, d(\mathrm{x}, \mathrm{y}^i)$ del punto più vicino sia al client che al server. In alcuni contesti, è anche necessario che il valore di $d(\mathrm{x}, \mathrm{y}^{(NN)})$ sia celato allo stesso server.

Un approccio di base che permette di raggiungere i suddetti obiettivi è innanzitutto quello di applicare più volte un'operazione di *permutazione/confusione*, con lo scopo di nascondere l'ordine delle distanze appena calcolate ad entrambi i partecipanti [36]. Questa fase è seguita da una serie di confronti in maniera sicura, utilizzando l'approccio seguito nel problema dei

milionari, quindi utilizzando vari circuiti confusi. L'idea è quella di implementare un protocollo simile a quello proposto per risolvere il problema del milionario, che funzioni in maniera sicura ed efficiente usando vari circuiti confusi, ma cercando in questo caso di comporre più circuiti confusi in maniera aggregata, al fine di trovare la distanza minima tra tutte le distanze calcolate.

Il modello di circuito confuso finale, progettato dagli autori di [37] e [38] consiste principalmente di semplici porte XOR, e il vantaggio che lo caratterizza è quello di avere un costo computazionale dimezzato rispetto ai classici metodi che utilizzano circuiti confusi. Spesso, il costo computazionale di tale approccio è inferiore a quello che si avrebbe se si utilizzasse un sistema crittografico omomorfico.

Ovviamente, lo scenario è stato volutamente descritto in maniera semplice, ma ciò non pregiudica il fatto che questi approcci non possano essere scalati in un contesto con più partecipanti. Ad esempio, sarebbe troppo restrittivo richiedere che il server conservi il valore $y^{(i)}$ in chiaro: basti pensare ad un contesto in cui il server non sia considerato un partecipante attendibile e che sia utilizzato sia come risorsa di calcolo in cloud, sia per la conservazione di dati privati appartenenti ad altri partecipanti.

Per effettuare il calcolo delle distanze all'interno di suddetto contesto, esistono due possibili strategie: la prima prevede il calcolo di valori incrociati, come ad esempio $E(x_i)^{-2y_i}$, nel caso i partecipanti utilizzino la distanza di Hamming o la distanza euclidea. Tale approccio purtroppo prevede un aumento del costo computazionale da parte di ogni partecipante e un maggior utilizzo della rete, al fine di trasmettere i valori cifrati tra il server e tutti i partecipanti.

La seconda strategia è quella di utilizzare sistemi crittografici omomorfici in grado di potere calcolare direttamente espressioni quadratiche o interi polinomi, come quelli descritti in [39] [29]. Questa strategia consente ai proprietari dei dati di inviare le loro versioni cifrate all'eventuale server non attendibile e di rimuovere completamente il costo computazionale per il calcolo della distanza, previsto nella prima strategia.

Capitolo 2

Truth Discovery

Prima di affrontare il paradigma di Truth Discovery, è necessario preliminarmente dare un'introduzione del concetto di Data Fusion. Generalmente per *Data Fusion* si intende quel processo teorico che ha come obiettivo quello di produrre dati molto più consistenti, accurati e che forniscano molte più informazioni all'utente finale, a partire da diverse sorgenti di dati che analizzate da sole non forniscono alcun tipo di informazione. Lo scopo è quindi quello di integrare i dati per migliorare il processo decisionale su questi, tenendo conto delle imperfezioni e delle diversità di cui sono costituiti i dati prodotti da queste sorgenti. Nonostante tutto, è fondamentale analizzare il concetto che sta alla base di questo processo di elaborazione, cioè l'estrapolazione di informazione. Il termine *informazione* include al suo interno sia il concetto di *dati*, che quello di *conoscenza*. Secondo gli autori di [40], questi tre concetti sono indipendenti tra loro, ma strutturati secondo un ordine gerarchico preciso. Più precisamente, possiamo prendere in esame le definizioni proposte da Russel Ackoff in [42], in merito ai livelli che compongono la *piramide della conoscenza*:

- *dato*: è un'entità espressa tramite simboli o segni in grado di rappresentare stimoli e segnali ottenuti da osservazioni. I dati presi singolarmente non hanno utilità perché non espressi in forma utilizzabile;
- *informazione*: rappresenta un insieme di dati che, raccolti insieme, risultano "utili". Le informazioni possono essere ricavabili dai dati sulla base di quesiti e domande (Chi? Come? Dove? Quando? Perché?);

- *conoscenza*: non esiste una definizione formale di conoscenza e spesso è un riferimento ad un insieme di informazioni. La conoscenza può rappresentare un insieme di informazioni che sono state elaborate, organizzate e strutturate secondo un certo criterio, oppure applicate;

- *saggezza*: è l'abilità di sapere utilizzare la conoscenza per ottenere maggiore successo con gli obiettivi.

A partire da questi elementi, il termine "Data Fusion" è stato coniato dai *Joint Working Group Directors of Laboratories* (JDL) del Dipartimento di Difesa degli Stati Uniti d'America nel 1986, definendolo come "*un processo che si occupa dell'associazione, della correlazione e della combinazione di dati e informazioni da sorgenti singole e multiple per ottenere stime di posizione e identità raffinate e valutazioni complete e tempestive di situazioni e minacce e del loro significato. Il processo è caratterizzato da continui raffinamenti delle sue stime, dalla necessità di valutazione di ulteriori fonti e dalla modifica del processo stesso, per ottenere risultati sempre migliori*" [43]. Generalmente, un'architettura di Data Fusion prevede una serie di input prodotti da sorgenti multiple, mentre l'output dell'architettura rappresenterà l'input di un sistema di interazione uomo-macchina. Il nucleo dell'architettura è composto dai seguenti elementi:

- *raffinamento dell'oggetto*: in questa fase avviene la combinazione dei dati sensoriali;

- *raffinamento della situazione*: questa fase consente di creare un'espressione dinamica delle relazioni tra entità ed eventi, quindi sintetizzare un'immagine dell'ambiente;

- *raffinamento dell'obiettivo*: questa fase mette in relazione la situazione attuale con quella futura e ne disegna le conclusioni;

- *raffinamento del processo:* questa fase serve per valutare e migliorare le prestazioni del sistema in tempo reale.

La nascita della Truth Discovery è motivata dalla forte necessità di rimuovere i conflitti presenti all'interno di informazioni "rumorose" provenienti da più sorgenti di dati, come nel caso dei database, dei siti Web e dei dati raccolti attraverso il *crowdsensing*. Contrariamente agli approcci che si pongono come obiettivo la votazione di più partecipanti, in cui le fonti di informazioni sono considerate equivalenti indiscriminatamente, al contrario, la Truth Discovery

ha l'obiettivo di attribuire dei gradi di attendibilità ai vari partecipanti, al fine di ottenere dati e informazioni più attendibili, ed è quindi considerata una particolare applicazione del concetto di Data Fusion.

I dati sono intrecciati in ogni aspetto della nostra vita e sono generati continuamente attraverso una varietà di canali, come social network, blog, forum di discussione e piattaforme di crowdsensing. Successivamente, i dati sono analizzati sia a livello individuale che di popolazione per scoprire nuove conoscenze, a partire da tante informazioni. Nonostante tutto, i dati, pur descrivendo spesso la stessa entità o evento, a causa della diversità delle fonti da cui provengono, potrebbero essere in conflitto tra loro a causa di errori, record mancanti, rumore o informazioni non aggiornate. Preliminarmente, il primo approccio che si potrebbe pensare per ridurre al minimo la discontinuità tra i vari dati è quello di scegliere l'informazione più frequente o effettuare una media dei dati, ma tale approccio presuppone che tutte le fonti siano ugualmente attendibili, che è proprio l'aspetto che dovrebbe essere evitato.

In letteratura, generalmente, la stima dell'attendibilità delle fonti e le fasi di ricerca della verità sono basate sul principio secondo cui alle fonti che forniscono informazioni vere più frequentemente, siano assegnati livelli di attendibilità più elevati, in modo che le informazioni generate da queste fonti siano considerate come dati di verità (*ground truth*) per il confronto dei dati generati successivamente. La stragrande maggioranza degli approcci che implementano la Truth Discovery basano la propria struttura su questo principio, poiché consente di catturare e modellare una grande varietà di scenari, di contesti e fare differenti assunzioni anche sulla base dei dati di input, tra le relazioni presenti tra le varie sorgenti e così via. La Truth Discovery gioca un ruolo importante nell'era dell'informazione, infatti, se da un lato nasce la necessità di avere sempre più informazioni accurate, dall'altro è inevitabile che si vada incontro alla presenza di dati incoerenti, proprio a causa della varietà dei *big data*. Lo sviluppo della Truth Discovery può quindi aiutare lo sviluppo di diversi campi in cui è fondamentale prendere decisioni critiche basate su informazioni attendibili estratte da fonti diverse, come ad esempio il campo dell'assistenza sanitaria, del crowdsensing e così via.

2.1 Definizione formale

Al fine di fornire una definizione formale del concetto di Truth Discovery, è necessario preliminarmente definire alcuni concetti più elementari, descritti tutti in [44]:

- un *oggetto o* è un'entità di nostro interesse a cui attribuire delle informazioni;
- una *sorgente s* descrive il luogo da cui è possibile raccogliere le informazioni sugli oggetti;
- un *valore* v_o^s rappresenta l'informazione fornita dalla sorgente *s* riguardo l'oggetto *o* di nostro interesse;
- un'*osservazione*, conosciuta anche come *record*, è una tupla di tre elementi costituita rispettivamente da un oggetto, una sorgente e il suo valore fornito;
- la *verità identificata* per un oggetto v_o^* rappresenta l'informazione selezionata come la più attendibile tra tutte i possibili valori candidati su questo oggetto;
- il *peso* della sorgente w_s riflette la probabilità che la sorgente *s* trasmetta informazioni attendibili. Un valore elevato indica che le informazioni provenienti da questa fonte hanno maggiori probabilità di essere accurate e di essere scelte.

Sulla base delle seguenti nozioni, è possibile adesso enunciare una definizione formale del concetto di Truth Discovery. Assumendo di avere un insieme di oggetti O di interesse le cui informazioni relative possono essere raccolte a partire da un insieme di sorgenti S, allora l'obiettivo della Truth Disocvery è quello di trovare la verità v_o^* per ogni oggetto $o \in O$ risolvendo i conflitti tra le informazioni provenienti dalle diverse sorgenti, quindi tra i valori $\{v_o^s\}_{s \in S}$. In questo contesto, gli approcci di Truth Discovery hanno come obiettivo di stimare i pesi delle sorgenti $\{w_s\}_{s \in S}$ che saranno utilizzati per ottenere il valore vero.

2.2 Contesti applicativi

Al fine di rendere più comprensibile l'importanza e il contesto applicativo della Truth Discovery, si presenterà un esempio tratto da [44]. Si ipotizzi di avere tre sorgenti distinte che forniscano, tramite votazione, il luogo di nascita di sei personaggi famosi, che in questo caso rappresentano gli oggetti di interesse. L'obiettivo di questo esempio è quello di determinare il corretto luogo di nascita per ogni personaggio in esame, sulla base delle informazioni conflittuali provenienti dalle tre sorgenti. In particolar modo, le ultime due righe della tabella 2.1 descrivono rispettivamente (1) la decisione che si otterrebbe se si scegliesse come metodo di inferenza il valore più frequente dalle sorgenti e (2) la decisione che si ricaverebbe da tecniche generali di Truth Discovery.

	George Washington	Abraham Lincoln	Mahatma Gandhi	John Kennedy	Barack Obama	Franklin Roosevelt
Source 1	Virginia	Illinois	Delhi	Texas	Kenya	Georgia
Source 2	Virginia	Kentucky	Porbandar	Massachusetts	Hawaii	New York
Source 3	Maryland	Kentucky	Mumbai	Massachusetts	Kenya	New York
Majority Voting	Virginia	Kentucky	Delhi	Massachusetts	Kenya	New York
Truth Discovery	Virginia	Kentucky	Porbandar	Massachusetts	Hawaii	New York

Figura 2.1: Elenco dei valori generati dalle 3 sorgenti.

Confrontando i risultati ottenuti dai due metodi, è possibile notare che questi non si differenziano, eccetto in due particolari casi:

- oggetto *Mahatma Gandhi*: questo rappresenta un *caso vincolato*, cioè il metodo basato sul valore più frequente non può far altro che tirare a sorte nel determinare il valore di verità tra i vari voti, proprio perché non esiste un valore che primeggia sugli altri (scegliendo Delhi). Al contrario, le tecniche di Truth Discovery sono in grado di distinguere le varie sorgenti, in funzione del loro grado di attendibilità, consentendo quindi di rompere il vincolo del pareggio dei voti grazie alla distribuzione dei pesi associati alle sorgenti (scegliendo Porbandar);

- oggetto *Barack Obama*: questo rappresenta il caso in cui le tecniche di Truth Discovery scelgono come valore quello meno votato tra tutti gli altri valori, cioè Hawaii, a differenza del più votato Kenya. Questo contesto descrive il caso in cui sono presenti molte sorgenti con un basso grado di attendibilità, contro poche sorgenti con attendibilità molto più alta. Ovviamente, il metodo basato sulla maggioranza di voto non può far altro che scegliere l'informazione più votata, che risulta sbagliata. Nonostante tutto, le tecniche di Truth Discovery sono in grado di distinguere tra sorgenti attendibili e inattendibili, determinando l'informazione corretta sempre sulla base del grado di attendibilità della sorgente che ha emesso la votazione.

2.3 Approcci generali

A prescindere di quale dei seguenti metodi sia scelto, la procedura generale per qualsiasi approccio che si basi sul paradigma della Truth Discovery deve seguire i seguenti passaggi: (1) l'algoritmo inizia associando dei valori di partenza ai pesi delle sorgenti; (2) sono calcolati in

maniera iterativa il valore di verità e la stima dei pesi da associare alle sorgenti; (3) viene stabilita una condizione che permetta di controllare il numero di iterazioni, ad esempio stabilendo che l'algoritmo termini quando la differenza tra i valori ottenuti tra due iterazioni è minore di una certa soglia. Le nozioni sono sempre tratte da [44].

OMISSIS

2.3.1 Aspetti implementativi

L'*inizializzazione* dei valori di verità, cioè i primi valori di verità da cui partire per avviare il processo di iterazione, può essere ottenuta utilizzando degli approcci già esistenti per la risoluzione dei conflitti. Nel caso di CRH, gli autori hanno dedotto che gli approcci basati sul *Voting*/*Averaging* in genere sono una buona scelta, difatti l'inizializzazione non influirà sui risultati finali, perché il problema di ottimizzazione è di per se convesso, di conseguenza il punto di inizio da cui minimizzare l'errore è ininfluente.

La *convessità* del problema dipende dalle funzioni di perdita e dalla funzione di regolarizzazione che si adottano nell'implementazione. Un esempio di famiglia di funzioni di perdita che permettono di modellare il problema in maniera convessa è la *divergenza di Bregman* [49], che include una varietà di funzioni di perdita come la funzione di perdita quadratica, la funzione di perdita logistica, la distanza Itakura-Saito, la distanza euclidea, la distanza di Mahalanobis e la divergenza di Kullback-Leibler.

Un altro problema importante da dovere affrontare è la *normalizzazione* delle deviazioni dei valori dichiarati dalle sorgenti su ogni proprietà dell'oggetto in esame. Come osservabile nella (2.8), le deviazioni dei valori di verità sono considerate insieme, nonostante riguardino proprietà diverse dello stesso oggetto. Se varie funzioni di perdita applicate su proprietà diverse di uno stesso oggetto hanno scale significativamente diverse, il valore del peso restituito dall'equazione sarà distorto e tenderà verso la proprietà che ha un intervallo maggiore nella deviazione. Per risolvere questo problema, è sufficiente normalizzare il valore restituito da ciascuna funzione di perdita su ciascuna proprietà, in modo che la deviazione calcolata su tutte le proprietà rientri nello stesso intervallo.

Nel caso di *outlier*, per motivi di semplicità, all'interno del modello si suppone che tutte le sorgenti osservino tutti gli oggetti su tutte le proprietà per garantire il corretto funzionamento del modello, ma quest'ultimo può essere facilmente modificato per gestire anche valori mancanti, nei casi in cui sorgenti diverse osservino sotto-insiemi di oggetti diversi su diversi sotto-gruppi di proprietà. Quando il numero di osservazioni fatte dalle sorgenti è abbastanza diversificato, è possibile normalizzare la distanza complessiva di ciascuna sorgente con il numero di osservazioni fatte dalla sorgente.

Capitolo 3

Sistema proposto e implementazione

Il problema che ci si trova di fronte, e che rappresenta l'obiettivo di questo lavoro di tesi, è quello di voler aggregare in un unico modello quelli che sono due aspetti apparentemente inconciliabili tra loro: la *Secure Multi Party Computation* 1 e la *Truth Discovery* 2. Lo scopo è quello di modellare un framework che garantisca la privacy dei partecipanti all'interno di un sistema distribuito, consentendo allo stesso tempo al sistema di individuare e di isolare falsi feedback forniti da eventuali utenti malevoli, quindi potere assegnare un grado di attendibilità per ciascun partecipante, consentendoci di discriminare i dati dichiarati dai vari partecipanti, estrapolando solo le informazioni più attendibili. In questo modello l'aspetto di tutela della privacy dovrà essere garantito da sistemi derivanti dal mondo della Secure Multi party Computation, mentre la stima del grado di attendibilità dei partecipanti al protocollo dovrà essere garantito da sistemi basati sulla Truth Discovery.

3.1 Sintesi dei sistemi di Secure Multi Party Computation

Dallo studio effettuato nel Capitolo 1, è emerso che in letteratura, ma generalmente nel mondo della crittografia, esiste un sistema in grado di potere effettuare calcoli direttamente su dati cifrati, senza la necessità di dovere effettuare la decifratura dei dati stessi per potere ottenere il risultato desiderato, che prende il nome di *crittografia omomorfica*. La crittografia omomorfica ha visto diverse implementazioni in letteratura e una di queste è quella attribuibile a Pascal Paillier, che prende il nome di *sistema crittografico additivo di Paillier* [7] [18]. In

breve, tale sistema permette di ottenere la somma di due valori cifrati o il prodotto di un valore cifrato per un valore in chiaro e il risultato delle operazioni corrisponde alla versione cifrata del risultato che si avrebbe se si effettuassero le stesse operazioni sui valori in chiaro. Il sistema crittografico di Paillier segue il seguente schema semplificato:

$$E(a) \cdot E(b) = E(a + b) \tag{3.1}$$

$$E(a)^b = E(ab) \tag{3.2}$$

Ovviamente, tale strumento è di grandissima utilità se il dominio di applicazione prevedesse il calcolo di semplici somme o prodotti, o comunque calcoli che possano essere ricondotti facilmente a queste due operazioni aritmetiche. Nello studio affrontato in [28], il problema posto è quello di modellare degli approcci che fossero in grado di confrontare più segnali cifrati con uno scelto come modello. In questo scenario, i segnali possono essere rappresentati da punti appartenenti ad uno spazio metrico *M*, di conseguenza il problema assume un aspetto geometrico: dati un insieme di punti e uno di riferimento scelto come modello, è possibile trovare il punto con la distanza minore dal punto scelto come modello? In questo caso, i punti rappresentano segnali cifrati, di conseguenza, è fondamentale trovare una metodologia che consenta di calcolare le distanze tra i punti rimanendo nel dominio cifrato.

Gli autori di [28] affrontano il problema sfruttando proprio il sistema crittografico di Paillier, consentendoci quindi di potere calcolare le distanze tra i vari punti, senza la necessità di dovere esplicitamente decifrare i vari segnali. Nello studio si pone attenzione al fatto che il calcolo della distanza deve essere fatto in funzione delle tipologie di dati che si stanno trattando, andando quindi a discriminare varie funzioni di calcolo della distanza, ognuna per ogni tipologia di dati, come ad esempio la *distanza di Hamming* per segnali binari, la *distanza euclidea* per segnali continui e la *edit-distance* per sequenze temporali.

Il risultato più significativo è che ognuna di queste distanze può essere calcolata utilizzando proprio il sistema crittografico di Paillier, a patto che si conosca la tipologia di dati da cui ricavare le distanze.

Infine, gli autori propongono anche un modello per il rilevamento della minima tra tutte le distanze calcolate, utilizzando sistemi basati su *Garbled Circuits*, simili a quelli presenti nel *problema del milionario*, ma questo aspetto non è indicativo per il problema affrontato in questo lavoro di tesi, poiché l'obiettivo è poter calcolare distanze tra punti, mantenendoci nel dominio

cifrato.

3.2 Sintesi dei sistemi di Truth Discovery

L'ultima parte del Capitolo 2 ha incluso un'analisi sulle tecniche di Truth Discovery presenti allo stato dell'arte, con un grande occhio di riguardo anche alla possibilità di potere integrare tali tecniche con l'obiettivo prefissato, cioè che possano essere utilizzate in un contesto di *privacy preserving*.

Il sistema di Truth Discovery di base presente in letteratura, e a cui molti altri si sono via via ispirati, prende il nome di TruthFinder [50]. Partendo da più sorgenti contrastanti di informazioni relativi a vari oggetti d'esame, l'obiettivo di TruthFinder è quello di identificare l'informazione reale (o più attendibile) tra tutte quelle dichiarate dalle sorgenti partecipanti. Sulla base di alcune euristiche descritte dettagliatamente in [50], il principio base di TruthFinder è che, se un determinato valore è fornito da molte sorgenti considerate attendibili, è probabile che quel valore rispecchi la verità; al contrario, se un valore è in conflitto con i valori forniti da molte sorgenti attendibili, è molto improbabile che quel valore rispecchi la verità. Di conseguenza, il grado di attendibilità di una sorgente e il valore di verità finale sono determinati l'uno dall'altro, utilizzando un approccio iterativo.

Più in particolare, l'algoritmo si basa su due grandezze, la *Confidence* e la *Trustworthiness*, che rappresentano rispettivamente il grado di verità attribuibile ad un valore dichiarato da una sorgente e il grado di attendibilità di tale sorgente.

L'algoritmo inizia fissando dei valori t casuali (o scelti sulla base di una particolare euristica) rappresentanti i gradi di attendibilità delle varie sorgenti; successivamente, in funzione dei valori di t, è possibile calcolare i valori di s , rappresentanti i valori di verità delle informazioni dichiarate dalle sorgenti, avente ognuna un grado di attendibilità sempre pari a t, durante la stessa iterazione; infine, in funzione del valore sia di t che di s , è possibile aggiornare il grado di attendibilità della sorgente, ottenendo quindi un nuovo valore t', che verrà utilizzato nella prossima iterazione. L'algoritmo termina quando i valori raggiungono uno stato di convergenza, ergo quando i valori di t subiscono una variazione minima.

Un altro sistema molto diffuso in letteratura e che si presta ad essere il miglior candidato per integrarsi nel problema è il *Conflict Resolution on Heterogenous Data* (CRH) [46].

Come il TruthFinder, l'obiettivo di CRH è quello di risolvere i conflitti che si possono presentare tra sorgenti multiple di dati eterogenei. L'obiettivo quindi è quello di ottenere l'in-

formazione più veritiera tra tutte quelle dichiarate dalle varie sorgenti, dove ad ogni sorgente è attribuito un grado di attendibilità tramite un peso. Il vantaggio principale di questo sistema è quello di potere utilizzare differenti funzioni di perdita all'interno del framework, in modo da riconoscere più facilmente le caratteristiche dei vari tipi di dati. Difatti, gran parte degli approcci di Truth Discovery presenti in letteratura prevedono il loro utilizzo su una singola tipologia di dati, di conseguenza non sono in grado di riconoscere differenze tra diverse tipologie di dati generati dalle sorgenti.

L'idea alla base di CRH è anch'essa simile a quella di TruthFinder, cioè che sorgenti attendibili debbano fornire osservazioni attendibili, quindi i valori di verità degli oggetti in esame dovrebbero essere molto simili alle osservazioni emesse dalle sorgenti attendibili. Il tutto si traduce nel minimizzare la deviazione pesata tra i valori di verità relativi agli oggetti in esame e i valori emessi dalle sorgenti, dove il peso riflette il grado di attendibilità delle sorgenti.

Il problema è affrontato, anche in questo caso, in maniera iterativa, seguendo due fasi:

1. *aggiornamento dei pesi delle sorgenti*: in questa fase sono aggiornati i gradi attendibilità delle sorgenti partecipanti al protocollo;

2. *aggiornamento dei valori di verità*: in funzione dei valori dichiarati dalle sorgenti e dai gradi di attendibilità di quest'ultimi, è possibile determinare il valore di verità cercato.

3.3 Un sistema di Truth Discovery sfruttando tecniche di cifratura omomorfica

OMISSIS

Conclusioni

L'obiettivo di questo lavoro di tesi è stato quello di progettare e implementare un framework che garantisca la riservatezza e la privacy degli utenti all'interno di un sistema distribuito sottostante, consentendo inoltre di potere individuare e isolare eventuali falsi feedback forniti da utenti malevoli, e quindi di potere assegnare delle "etichette" di attendibilità a ciascun partecipante, al fine di discriminare i risultati ottenuti ed estrapolare solo quelli più attendibili. Durante una ricerca preliminare dei sistemi e degli approcci presenti in letteratura, è emerso che i due paradigmi fondamentali per l'implementazione del suddetto framework sono quelli di *Secure Multi Party Computation* e di *Truth Discovery*. Il paradigma di Secure Multi Party Computation ha fornito gli strumenti necessari per la progettazione dell'aspetto di tutela della privacy degli utenti e della riservatezza dei dati all'interno del framework, mentre il paradigma di Truth Discovery ha riguardato l'aspetto della stima di attendibilità dei partecipanti. Sono stati quindi studiati e illustrati gli aspetti fondamentali di entrambi i paradigmi, partendo da una definizione formale e illustrandone gli obiettivi, per poi elencare successivamente per ognuno le componenti peculiari e le tecniche più diffuse e di maggiore interesse per l'obiettivo di questa tesi.

Relativamente allo studio del paradigma di Secure Multi Party Computation effettuato nel Capitolo 1, è emerso che il sistema più promettente ai fini dell'implementazione del framework è quello di *crittografia omomorfica additiva di Paillier*. Tale sistema permette di effettuare calcoli aritmetici utilizzando direttamente i valori cifrati, ottenendo un risultato equivalente, se si effettuasse la stessa operazione sui valori in chiaro.

Relativamente allo studio del paradigma di Truth Discovery effettuato nel Capitolo 2, è emerso che il sistema più promettente ai fini dell'implementazione del framework è quello di *Conflict Resolution on Heterogenous data* (CRH). Tale sistema permette di restituire l'informazione più "veritiera" tra tutte quelle dichiarate dalle varie sorgenti, attribuendo a ciascuna sorgente un grado di attendibilità attraverso un opportuno peso. Il sistema gode dell'importante vantaggio di poter utilizzare differenti funzioni di perdita, in funzione della natura dei dati da trattare.

L'approccio basato sul calcolo delle distanze utilizzato all'interno del sistema CRH ha consentito una perfetta integrazione dell'utilizzo del sistema crittografico omomorfico additivo di Paillier. Il nuovo modello ha quindi integrato e utilizzato le tecniche di Truth Discovery utilizzate in CRH, cioè le elaborazioni relative all'aggiornamento dei pesi e del valore finale di verità, ma rispettando la riservatezza dei partecipanti, integrando quindi il sistema crittografico di Paillier alle due suddette elaborazioni. Inoltre, il calcolo basato sulle distanze, ha consentito una progettazione che prevedesse la gestione e la corretta elaborazione sia di dati continui che categorici.

Relativamente alla parte di riservatezza del framework, si è deciso di utilizzare la variante *Threshold Paillier* del già citato sistema crittografico omomorfico. Tale variante, oltre a garantire le classiche proprietà di omomorfismo relative al dominio cifrato, gode della caratteristica di permettere la decifratura di un qualsiasi valore cifrato sono nel caso in cui si posseggano almeno t tra tutte le p chiavi private distribuite ai singoli partecipanti. Infatti, in questo sistema, la chiave privata generata assieme a quella pubblica viene "suddivisa" in p chiavi private distinte, dove p rappresenta il numero totale di partecipanti. Solo successivamente ad una *combinazione matematica* delle decifrature parziali dei singoli partecipanti, denominate *share*, è possibile ottenere il valore in chiaro finale.

L'obiettivo è stato quindi garantire la riservatezza a più componenti possibili durante l'elaborazione dei dati forniti dagli utenti, grazie all'utilizzo del sistema crittografico di Paillier. Di conseguenza, durante l'esecuzione del nuovo framework, sono celate tutte le informazioni relative ai valori dichiarati dai singoli partecipanti, il grado di attendibilità attribuito ad ogni singolo partecipante e le identità dei partecipanti. L'unico dato in chiaro restituito dal framework è il valore di verità finale, calcolato considerando i gradi di attendibilità dei singoli partecipanti.

L'implementazione finale del framework prevede una struttura sequenziale suddivisa in 4 fasi: una *Fase 0* caratterizzata dalla generazione e dalla modellazione dei partecipanti e del server; una *Fase 1* caratterizzata dall'inizializzazione del framework, ossia la generazione delle chiavi del cifrario e la distribuzione delle chiavi e dei parametri del cifrario a tutti i partecipanti; una *Fase 2* caratterizzata dalla dichiarazione dei valori da parte dei partecipanti; una *Fase 3* caratterizzata dall'inizializzazione dei parametri statistici (nel solo caso di dati continui); una *Fase 4* caratterizzata dall'esecuzione vera e propria del processo iterativo.

La valutazione sperimentale è stata effettuata utilizzando un dataset contenente informazioni meteorologiche su diverse città americane durante diverse giornate. Il dataset contiene i valori dichiarati da 16 partecipanti, sia le relative ground truth. Tutti i valori dichiarati sono rappresen-

tati sia in maniera continua che categorica, consentendo quindi una valutazione del framework durate l'utilizzo di entrambe le tipologie di dati.

Gli aspetti maggiormente presi in considerazione durante la valutazione sono stati il tempo impiegato dal framework per ottenere il valore di verità finale e la precisione dello stesso valore, in relazione alle ground truth presenti all'interno del dataset. Tali valutazioni sono state effettuate facendo variare il numero di partecipanti, la dimensioni in bit delle chiavi utilizzate per il cifrario e il *parametro di arrotondamento* (quest'ultimo assieme al relativo *limite di convergenza*). La valutazione sperimentale è stata eseguita separatamente sia durante l'utilizzo di dati continui, sia durante l'utilizzo di dati categorici. Nel solo caso della valutazione al variare del numero dei partecipanti, i risultati sono stati confrontati con quelli ottenuti da CRH, ossia un modello di riferimento che non garantisce alcun aspetto sulla privacy degli utenti.

Dai risultati è emerso che il tempo di esecuzione richiesto dal framework per restituire il valore di verità finale è fortemente influenzato dal parametro di arrotondamento e dalla dimensione in bit delle chiavi utilizzate per il sistema crittografico di Paillier. Il parametro di arrotondamento preserva la parte decimale dei valori continui durante la loro cifratura con il sistema crittografico di Paillier, semplicemente moltiplicando il valore decimale per un fattore moltiplicativo. Di conseguenza, all'aumentare del numero di cifre decimali da preservare, è necessario incrementare anche la precisione del limite di convergenza, portando quindi il framework ad arrestare la propria esecuzione solo nel momento in cui l'ultima cifra decimale imposta non sia variata considerevolmente durante due iterazioni consecutive.

Ad influenzare maggiormente il tempo di esecuzione del framework però è la dimensione in bit delle chiavi utilizzate con il sistema crittografico di Paillier. Nel caso dei cifrari asimmetrici infatti (e quindi anche del sistema omomorfico additivo di Paillier), le operazioni di cifratura e decifratura risultano abbastanza onerose da un punto di vista computazionale, a causa dell'utilizzo di numeri di enorme dimensioni e il calcolo di esponenziazioni con tali numeri.

Relativamente sia al parametro di convergenza che alla dimensione in bit delle chiavi, la scelta di entrambi è comunque in funzione del contesto di utilizzo del framework, e di conseguenza dei dati che si tratteranno. Prendendo in considerazione i risultati sperimentali ottenuti durante le valutazioni del framework, è emerso che una dimensione di 1024 bit è un ottimo compromesso tra sicurezza e prestazioni. Allo steso modo, a meno che non fosse richiesta una precisione elevata, il parametro di arrotondamento può essere configurato per preservare una o due cifre decimali.

Dai risultati è anche emerso come la discrepanza tra il risultato finale e la ground truth

dipenda molto di più dal numero di partecipanti, piuttosto che da altri fattori. I risultati mostrano come un numero di partecipanti maggiore tenda a ridurre, o comunque a far convergere, l'errore commesso dal framework rispetto alla ground truth. Tale risultato ha riguardato sia il caso di dati continui, che di quelli categorici. In definitiva, il framework è in grado di fornire le migliori prestazioni al crescere del numero partecipanti, a patto che la dimensione in bit delle chiavi utilizzate non sia abbastanza elevata da incrementare in maniera esponenziale il tempo di esecuzione.

Concludendo, è possibile affermare che, allo stato attuale, l'implementazione ha soddisfatto pienamente l'obiettivo di questo lavoro di tesi, aprendo la porta ad una serie di possibili scenari e migliorie che possono essere incluse e implementate al fine di incrementare l'efficienza dell'approccio, o crearne di nuovi completamente differenti. Da un punto di vista più critico, è possibile affermare che l'efficienza non è sicuramente il punto forte dell'intera implementazione, sebbene quest'aspetto può sicuramente essere migliorato studiando e sviluppando i seguenti punti:

- *riduzione dell'overhead*: con l'utilizzo di sistemi crittografici omomorfici adatti, si potrebbe pensare di affidare tutte le fasi di elaborazione del valore di verità finale al solo server, eliminando quindi tutte le fasi in cui i partecipanti sono coinvolti per l'elaborazione delle singole share o per il calcolo del valore pesato in forma cifrata. Di conseguenza, il partecipante si limiterà solo a dichiarare il proprio valore. Tale tipo di approcci predilige la presenza di più server, al fine di distribuire in maniera più efficiente la raccolta delle informazioni dei partecipanti e il calcolo del valore di verità finale. Possibili studi che hanno affrontato questo tipo di problema sono [54] e [57]. In particolare, l'approccio sviluppato in [54] possiede anche caratteristiche di *fault tolerant*, nei casi in cui alcuni partecipanti non siano presenti durante le fasi di elaborazione, nonostante abbiano effettuato una votazione;

- *scelta delle informazioni a cui garantire la riservatezza*: nell'implementazione si è deciso di rendere riservata qualsiasi informazione, eccetto il valore di verità finale. Ovviamente, un maggior numero di informazioni da proteggere porta ad un inevitabile aumento del costo computazionale dovuto alle fasi di cifratura aggiuntive. In funzione del contesto di utilizzo, è possibile rilassare le condizioni di riservatezza imposte, e quindi ridurre le fasi di cifratura all'interno del framework. In tale contesto, gli autori di [58] hanno sviluppato un approccio basato sempre su tecniche di crittografia omomorfica e di Truth Discovery che si limita a garantire la riservatezza ai soli dati forniti dai partecipanti;

- *utilizzo di sistemi crittografici omomorfici moltiplicativi*: il sistema omomorfico utilizzato in questa implementazione è stato unicamente quello additivo di Paillier, ossia un sistema in grado di effettuare esclusivamente somme tra valori cifrati e prodotti tra un valore in chiaro e un valore cifrato. In questo contesto, l'utilizzo di un sistema omomorfico moltiplicativo ridurrebbe drasticamente il numero di fasi all'interno di una stessa iterazione del framework. Ad esempio, nel caso si utilizzasse un sistema omomorfico moltiplicativo, è possibile rimuovere la fase in cui il server è costretto a mandare il peso cifrato ai singoli partecipanti, affinché quest'ultimi possano calcolare il corrispettivo valore pesato in forma cifrata, da mandare successivamente al server. Esempi di possibili sistemi omomorfici moltiplicativi sono gli algoritmi di cifratura RSA e ElGamal, descritti in 1.3.1. Nonostante tutto, l'integrazione di due sistemi crittografici omomorfici è tutt'altro che banale;

- *utilizzo di sistemi crittografici omomorfici completi*: come già descritto in 1.3.1, un sistema omomorfico completo (o *fully homomorphic*) prevede, da un punto di vista teorico, il calcolo di qualsiasi funzione matematica, a partire da un insieme di dati, rimanendo all'interno del dominio cifrato. Sicuramente, rappresenterebbe lo strumento migliore, più potente, e quindi l'alternativa più promettente. Nonostante tutto, gran parte degli esempi proposti in letteratura non risultano il massimo in termini di efficienza e di complessità di implementazione. Quasi sempre, i sistemi fully homomorphic sono soggetti a rumore e la loro efficienza è di gran lunga inferiore rispetto ai comuni sistemi partially homomorphic, come ad esempio il sistema crittografico di Paillier. Pertanto, l'adozione di tali sistemi avrebbe reso estremamente inefficiente, e quindi inutilizzabile, il modello implementato. Una lista in continuo sviluppo di possibili sistemi omomorfici completi è presente in [59].

Bibliografia

[1] A. Shamir, R. Rivest e L. Adleman. «Mental Poker». In: *Technical Report LCS/TR-125, Massachusetts Institute of Technology* (1979).

[2] A. C. Yao. «Protocols for secure computations». In: (1982).

[3] A. C. Yao. «How to Generate and Exchange Secrets». In: *27th Annual Symposium on Foundations of Computer Science* (1986), pp. 162–167.

[4] D. Malkhi, N. Nisan, B. Pinkas e Y. Sella. «Fairplay-Secure Two-Party Computation System». In: 2004.

[5] K. Lauter. «Practical Applications of Homomorphic Encryption». In: *Proceedings of the 2012 ACM Workshop on Cloud Computing Security Workshop*. 2012, pp. 57–58.

[6] M. Naehrig, K. E. Lauter e V. Vaikuntanathan. «Can homomorphic encryption be practical?» In: *Proceedings of the 3rd ACM Cloud Computing Secu- rity Workshop, CCSW*. 2011, pp. 113–124.

[7] M. Marwan, A. Kartit e H. Ouahmane. «Applying Secure Multi-Party Computation to Improve Collaboration in Healthcare Cloud». In: *Third International Conference on Systems of Collaboration (SysCo)*. 2016.

[8] J. Liu, M. Juuti, Y. Lu e N. Asokan. «Oblivious Neural Network Predictions via MiniONN Transformations». In: *ACM CCS 17: 24th Conference on Computer and Communications Security*. 2017.

[9] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft e D. Boneh. «Privacy-preserving matrix factorization». In: *ACM CCS 13: 20th Conference on Computer and Communications Security*. 2013.

[10] A. Gascón, P. Schoppmann, B. Balle, M. Raykova, J. Doerner, S. Zahur e D. Evans. «Privacy-Preserving Distributed Linear Regression on High- Dimensional Data». In: *Proceedings on Privacy Enhancing Technologies.* (2017), pp. 248–267.

[11] P. Mohassel e Y. Zhang. «SecureML: A System for Scalable Privacy- Preserving Machine Learning». In: *2017 IEEE Symposium on Security and Privacy.* (2017), pp. 19–38.

[12] URL: https://en.wikipedia.org/wiki/Homomorphic_encryption.

[13] URL: https://en.wikipedia.org/wiki/Homomorphism.

[14] F. Armknecht, C. Boyd, C. Carr, K. Gjøsteen, A. Jäschke, C. A. Reuter e M. Strand. «A Guide to Fully Homomorphic Encryption». In: *Cryptology ePrint Archive, Report 2015/1192* (2015).

[15] W. Stallings. *Cryptography And Network Security - Principles And Practice*. Seventh Edition. Pearson, 2017.

[16] S. Goldwasser e S. Micali. «Probabilistic encryption and how to play mental poker keeping secret all partial information». In: *Proc. 14th Symposium on Theory of Computing*. 1982, pp. 365–377.

[17] S. Goldwasser e S. Micali. «Probabilistic encryption». In: *Journal of Computer and System Sciences* 28.2 (1984), pp. 270–299.

[18] P. Paillier. «Public-Key Cryptosystems Based on Composite Degree Residuosity Classes». In: *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques*. A cura di Springer. 1999, pp. 223–238.

[19] M. Ben-Or, S. Goldwasser e A. Wigderson. «Completeness theorems for non-cryptographic fault-tolerant distributed computation». In: (1988), pp. 1–10.

[20] I. Damgård, V. Pastro, N. Smart e S. Zakarias. «Multiparty computation from somewhat homomorphic encryption». In: *Crypto 2012* Springer LNCS 7417 (2012), pp. 643–662.

[21] I. Damgård e M. Jurik. «A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System». In: *Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography (PKC '01)*. 2001.

[22] P.-A. Fouque, G. Poupard e J. Stern. «Sharing Decryption in the Context of Voting or Lotteries». In: *Financial Cryptography, 4th International Conference*. 2000.

[23] K. Kaya, B. Dündar, S. Kalkan e A. Selçuk. «Threshold Paillier and Naccache-Stern Cryptosystems Based on Asmuth-Bloom Secret Sharing». In: (2006).

[24] P. S. Naidu, R. Kharat, R. Tekade, P. Mendhe e V. Magade. «E-Voting System Using Visual Cryptography and Secure Multi-party Computation». In: *Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India* (2016).

[25] M. Naor e A. Shamir. «Visual eryptography». In: *Proceedings of the advances in cryptology Eurocrypt* (1995).

[26] A. Walake e P. Chavan. «Efficient Voting system with (2,2) Secret Sharing Based Authentication». In: *IJCSIT,* 6 (2015).

[27] R. Bansal, P. Sehgal e P. Bedi. «Minutiae Extraction from Fingerprint Images - a Review». In: *IJCSI* 8 (2011).

[28] S. Rane e P. T. Boufounos. «Privacy-preserving nearest neighbor methods: comparing signals without revealing them». In: *IEEE Signal Processing Magazine* 30.2 (2013), pp. 18–28.

[29] C. Gentry. «Computing arbitrary functions of encrypted data». In: *Communications of the ACM* 53.3 (2010), pp. 97–105.

[30] M. V. Dijk, C. Gentry, S. Halevi e V. Vaikuntanathan. «Fully homomorphic encryption over the integers». In: *Advances in Cryptology–EUROCRYPT* (2010), pp. 24–43.

[31] C. Aguilar-Melchor, S. Fau, C. Fontaine, G. Gogniat e R. Sirdey. «Recent Advances in Homomorphic Encryption: A Possible Future for Signal Processing in the Encrypted Domain». In: *IEEE Signal Processing Magazine* 30.2 (2013), pp. 108–117.

[32] M. Shaneck, Y. Kim e V. Kumar. «Privacy preserving nearest neighbor search». In: *Proc. of the Sixth IEEE Intl. Conf. Data Mining*. 2006, pp. 541–545.

[33] Y. Qi e M. J. Atallah. «Efficient privacy-preserving k-nearest neighbor search». In: *Intl. Conference on Distributed Computing Systems*. Vol. 0. 2008, pp. 311–319.

[34] M. O. Rabin. «How to exchange secrets with oblivious transfer». In: *Technical Report TR-81, Aiken Computation Lab, Harvard University* (1981).

[35] C. C. Aggarwal. *Data Mining: The Textbook*. Springer, 2015.

[36] M. Atallah, F. Kerschbaum e W. Du. «Secure and private sequence comparisons». In: *Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society*. 2003, pp. 39–44.

[37] V. Kolesnikov e T. Schneider. «Improved garbled circuit: Free XOR gates and applications». In: *International Colloquium on Automata, Languages, and Programming*. 2008, pp. 486–498.

[38] V. Kolesnikov, A. Sadeghi e T. Schneider. «Improved garbled circuit building blocks and applications to auctions and computing minima». In: *Cryptology and Network Security (CANS)*. 2009, pp. 1–20.

[39] D. Boneh, E. J. Goh e K. Nissim. «Evaluating 2-DNF formulas on ciphertexts». In: *Theory of Cryptography Conference*. 2005, pp. 325–341.

[40] E. Bosse, J. Anne-Laure e M. Patrick. «Knowledge, uncertainty and belief in information fusion and situation analysis». In: *Data Fusion for Situation Monitoring, Incident Detection, Alert and Response Management* 198 (2005), pp. 61–80.

[41] V. Agate, M. Curaba, P. Ferraro, G. Lo Re, and M. Morana. «Secure e-Voting in Smart Communities». In: Proceeding of the Fourth Italian Conference on Cyber Security (ITASEC 2020).

[42] R. L. Ackoff. «La pyramide des connaissances (DIKW) Knowledge Management». In: (2013), pp. 4–5.

[43] F. E. White. «Data Fusion Subpanel of the Joint Directos of Laboratories Technical Panel for C3». In: (1991).

[44] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan e J. Han. «A Survey on Truth Discovery». In: *ACM SIGKDD Explorations Newsletter* 17.2 (2016).

[45] D. P. Bertsekas. «Non-linear Programming». In: *Athena Scientific 2nd edition* (1999).

[46] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan e J. Han. «Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation». In: *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 2014.

[47] F. Concone, G. Lo Re, M. Morana. «SMCP: a Secure Mobile Crowdsensing Protocol for fog-based applications». In: Journal of Human-centric Computing and Information Sciences (HCIS 2020).

[48] F. Restuccia, P. Ferraro, S. Silvestri, S. K. Das, and G. Lo Re. «IncentMe: Effective mechanism design to stimulate crowdsensing participants with uncertain mobility». In: IEEE Transactions on Mobile Computing, 18(7), 1571-1584.

[49] A. Banerjee, S. Merugu, I. S. Dhillon e J. Ghosh. «Clustering with Bregman Divergences». In: *Journal of Machine Learning Research* (2005).

[50] X. Yin, J. Han e P. S. Yu. «Truth Discovery with Multiple Conflicting Information Providers on the Web». In: *IEEE Transactions on Knowledge and Data Engineering* 20.6 (2008), pp. 796–808.

[51] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao e K. Ren. «Cloud-Enabled Privacy-Preserving Truth Discovery in Crowd Sensing Systems». In: *13th ACM Conference*. 2015, pp. 183–196.

[52] URL: https://coderzcolumn.com/tutorials/python/threshold-paillier.

[53] URL: https://bit.ly/2XfT1se.

[54] C. Zhang, L. Zhu, C. Xu, K. Sharif, X. Du e M. Guizani. «LPTD: Achieving light-weight and privacy-preserving truth discovery in CIoT». In: *Future Generation Computer Systems* 90 (2019), pp. 175–184.

[55] F. Concone, P. Ferraro, G. Lo Re. «Towards a Smart Campus through Participatory Sensing». In: Proceedings of the 4th IEEE International Conference on Smart Computing (SMARTCOMP 2018).

[56] F. Restuccia, P. Ferraro, T. S. Sanders, S. Silvestri, S. K. Das, and G. Lo Re. «FIRST: A framework for optimizing information quality in mobile crowdsensing systems». In: ACM Transactions on Sensor Networks (TOSN), 15(1), 1-35.

[57] Y. Zheng, H. Duan, X. Yuan e C. Wang. «Privacy-Aware and Efficient Mobile Crowdsensing with Truth Discovery». In: *IEEE Transactions on Dependable and Secure Computing* 17.1 (2020), pp. 121–133.

[58] C. Miao, L. Su, W. Jiang, Y. Li e M. Tian. «A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems». In: *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, Atlanta, GA*. 2017, pp. 1–9.

[59] URL: http://homomorphicencryption.org/introduction/.