UNIVERSITÀ DEGLI STUDI DI CAMERINO

SCUOLA DI SCIENZE E TECNOLOGIE

CORSO DI LAUREA IN INFORMATICA

EASE: una web chat per garantire la privacy delle comunicazioni

ELABORATO FINALE

Studente

Luca Pettinari

MATRICOLA 085748

Relatore

Prof. Marcantoni Fausto

Co-Relatore

Dr. Riccardo Cognini

ANNO ACCADEMICO 2015/2016

1. Introduzione

Attualmente uno dei dibattiti su cui si discute maggiormente è quello che cerca di coniugare il concetto di privacy con quello della sicurezza. Se da un lato gli scontri culturali in atto, già alla fine dello scorso secolo, hanno messo a repentaglio la sicurezza globale, dall'altro la privacy, di cui ciascun individuo ha diritto e che costituisce la base delle libertà individuali, viene spesso penalizzata dalla necessità di assicurare, al livello mondiale, la sicurezza di tutti i cittadini. Questo lavoro tuttavia, non si prefigge di arricchire il dibattito con posizioni di carattere ideologico/filosofico, perché non risponderebbe alla mia esigenza di offrire servizi di comunicazione sicuri a tutti gli utenti. Pertanto mi concentrerò su tutte quelle situazioni di controllo, più o meno nascosto, che si pongono diversi intenti di tipo economico commerciale oppure per l'attuazione di atti delinquenziali, come ad esempio il ricatto o l'eventuale scoperta di dati sensibili come password e codici bancari. La consapevolezza che gran parte dei software odierni, più noti e maggiormente utilizzati, non garantiscano la totale privacy dell'utente, sia dovuta alla negligenza delle grandi aziende e al loro interesse nel raccogliere dati, porta gli utenti più esperti ad utilizzare software di nicchia. Tuttavia questi ultimi comportano alcuni problemi, il primo dei quali derivante dal fatto che, essendo poco utilizzati, non sempre garantiscono la corrispondenza tra i vari partner con cui si comunica. Il problema non è di facile soluzione perché nel caso di sconfinamento della privacy, sia in termini di notizie acquisite, sia di veri e propri profili individuali, spesso le aziende produttrici hanno un riscontro economico non indifferente nell'utilizzo di questi dati. Da questa situazione, ho deciso di analizzare alcuni dei software più noti presenti sul mercato, estrapolarne le caratteristiche migliori e unirle in un solo prodotto che garantisca la massima protezione da attacchi in rete. In seguito a questo studio si è sviluppata una web chat, che andremo a chiamare *EASE*, che si concentra sull'idea di tenere segreto ciò che viene scambiato tra gli utenti, e per farlo sono state adottate diverse tecnologie e algoritmi di crypting recenti. L'obiettivo finale della web chat, sviluppata come tesi, è quindi quello di garantire all'utente la totale riservatezza e la completa cancellazione della sua conversazione, proprio per evitare quei rischi, già evidenziati in precedenza, che si vogliono salvaguardare da terzi. Un esempio pratico può essere uno scambio di idee riguardo una password affidabile da usare per un conto bancario, o il passaggio di dati riguardo un conto postale, o semplicemente confessioni private. Informazioni di questo tipo possono interessare soprattutto i malintenzionati, o hacker e servizi segreti, i quali utilizzerebbero tali dati a proprio vantaggio, per prosciugare un conto, per ricattare o per raccogliere dati per

indagini segrete. Tale chat non va però interpretata come una chat per chi ha dei segreti da nascondere, ma soltanto un servizio web che, al contrario di molte app esistenti, consente di non tenere traccia del proprio passaggio e poter comunicare senza aver timore di essere spiati.

1.1. Review

Come già ampiamente espresso nell'introduzione, il problema principale sul quale mi sono soffermato è il pericolo, da parte di persone estranee, di intercettare la conversazione tra gli utenti di una stanza, o addirittura di manomettere i messaggi. Una delle più grandi minacce sono le aziende stesse, perché come descrive Cristiano Ghidotti sul sito webnews.it (http://www.webnews.it/2015/05/12/google-hangouts-crittografia-end-to-end/), l'applicazione Hangout dichiara, sulle pagine del supporto ufficiale, che la crittografia avviene soltanto "in transit", ovvero i messaggi viaggiano dal client al server, e una volta arrivati possono essere decifrati dall'azienda stessa, soprattutto in caso di formale richiesta relativa a un'indagine. Un altro esempio di totale vulnerabilità da parte dei prodotti più noti, come lo è Whatsapp, è descritto nell'articolo di Andrea Artes, sul sito enjoyphoneblog.it (http://enjoyphoneblog.it/33836/guide/come...), dove viene spiegata la procedura per attaccare un account Whatsapp e prendere il totale possesso dei dati della persona, semplicemente conoscendo il numero e il MAC address del telefono della vittima. Queste falle destano preoccupazioni negli utenti medi a causa del contenuto delle loro conversazioni, che siano di carattere privato, o che contengano dati sensibili.

1.2. Funzionamento di background

Per proteggere i dati, che gli utenti all'interno della chat si scambiano, la soluzione ottimale adottata è stata innanzitutto quella di utilizzare il protocollo HTTPS per lo scambio di informazioni tra client e server, e rendere più difficile un attacco di tipo *man in the middle*, ovvero un attacco crittografico nel quale l'intruso sia in grado di leggere, inserire o modificare a piacere i messaggi privati tra due comunicanti. Il secondo step è stato quello di isolare la conversazione in una stanza, rappresentata da una sessione PHP, e criptare i messaggi al suo interno con l'algoritmo di cifratura a blocchi *AES*, con una chiave da 256bit, creata lato client. Questo permette quindi una cifratura end-to-end,

ovvero i messaggi sono interpretati soltanto dagli utenti connessi alla chat che possiedono la giusta chiave di lettura. L'intera sessione viene poi memorizzata nella memoria RAM, con l'ausilio di memcached, un sistema di chaching distribuito, rendendo ancora più difficile la lettura, ma non impossibile. Quando la lista utenti è vuota, il file di sessione viene distrutto senza lasciare traccia.

L'utilizzo di sessioni memorizzate in RAM comporta 2 vantaggi importanti:

- Il salvataggio, la creazione e la lettura della sessione sono notevolmente più veloci
- La lettura della RAM richiede permessi di ROOT al server

Nel caso che un intruso venga poi in possesso di un id di una stanza, o ne componga uno casualmente, senza la chiave corretta i messaggi risulteranno incomprensibili, quindi la comunicazione resta comunque segreta. Inoltre in caso di login in una stanza, il nome utente non sarà mai nullo o vuoto, e verrà visualizzato nell'apposito contenitore degli utenti connessi alla stanza, in alto a destra. Quindi in caso di intrusione, si noterà un nuovo utente connesso alla stanza.

Il funzionamento generalizzato descritto in questo capitolo, viene schematizzato in Figura 1 qui sotto inserita.

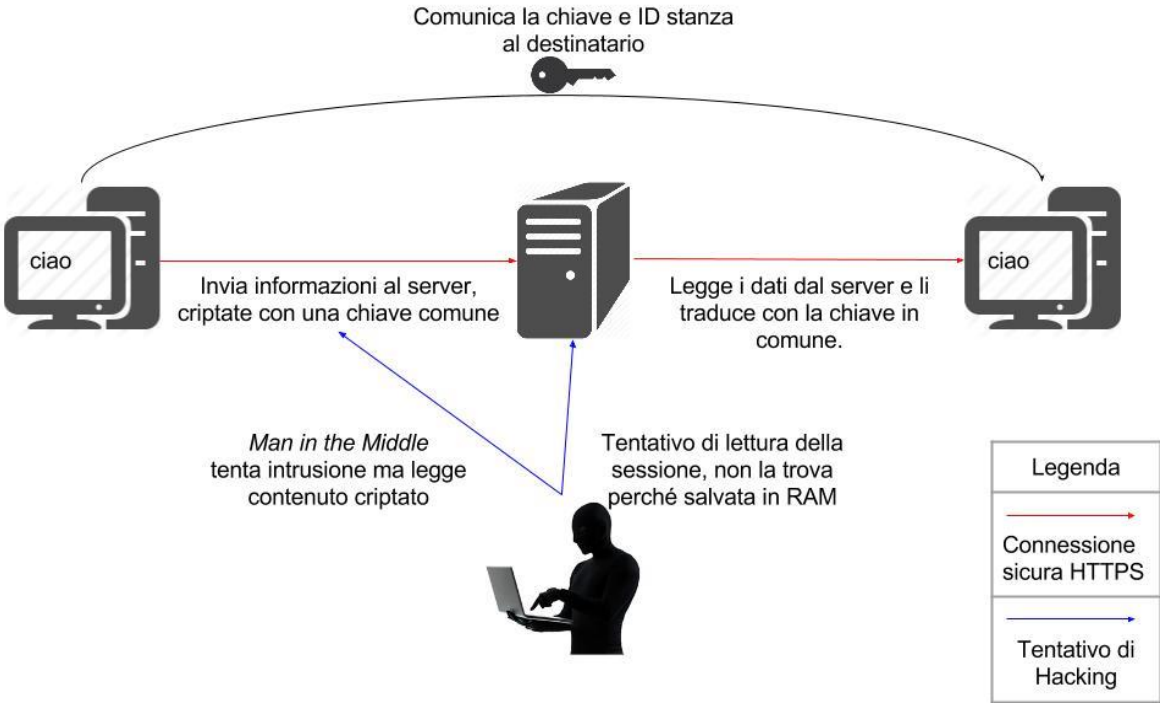

Figura 1 Schema di funzionamento semplificato

1.3. Obiettivi

L'obiettivo principale del progetto è stato quello di fornire all'utente uno strumento web sicuro e che gli garantisca la privacy. Per far ciò ho seguito dei punti che mi hanno permesso di ottenere un prodotto affidabile. Questi punti sono:

- Rendere la conversazione effettivamente privata, grazie alla creazione delle stanze che racchiudono tutto il contenuto e isolano una chatroom da un'altra.
- Rendere i dati che passano in rete indecifrabili, con l'utilizzo del protocollo HTTPS, appositamente sviluppato per cifrare i messaggi che passano in rete.
- Rendere i dati scritti nella sessione indecifrabili, con l'utilizzo dell'algoritmo di cifratura a blocchi *AES*
- Impedire la modifica dei file di sessione, grazie all'utilizzo del sistema di caching distribuito memcache integrato al file di configurazione di PHP.

1.4. Struttura della tesi

La struttura della tesi è la seguente:

1. Introduzione: una breve introduzione sul concetto di riservatezza in contrapposizione a quello di sicurezza. Un'analisi più ampia del funzionamento del software.

2. Informazioni di contesto:
   2.1. Informazioni di background sul contesto in cui si applica il software: approfondimento dell'obiettivo principale della tesi, ovvero il diritto alla segretezza delle informazioni scambiate, e di essere poi dimenticato dal sito, quindi il diritto all'oblio. Analisi del modo in cui questa chat assicuri questi diritti, garantendo segretezza ai messaggi scambiati, e cancellando il contenuto una volta conclusa la conversazione.
   2.2. Prodotti simili: analisi di prodotti simili presenti sul mercato, con descrizioni particolareggiate sulle modalità con cui si garantisce una maggiore privacy all'utente che li utilizza.

3. Progettazione di EASE
   3.1. Tecnologie selezionate: si presenta un elenco delle tecnologie utilizzate per lo sviluppo del progetto, seguito dalle motivazioni che hanno portato all'utilizzo di un dato linguaggio di programmazione e di una data tecnologia.
   3.2. Ingegnerizzazione del software: da titolo, in questo capitolo viene studiata la struttura del software, ingegnerizzando ogni parte di esso, attraverso diagrammi UML e use case.

4. Utilizzo del software
   4.1. Utilizzo: alcuni screenshot di quella che è la chat e del suo utilizzo, a partire dalla creazione di una stanza fino all'unione in una stanza esistente.
   4.2. Come il software garantisce la sicurezza delle informazioni scambiate: grafici ed immagini di come i dati fluiscono dal server al client e viceversa, e di come le informazioni fluiscano in rete in modo sicuro.

5. Conclusioni e Sviluppi Futuri: spiegazione delle migliorie da applicare al software, e futuri sviluppi per il progetto.

2. Informazioni di contesto

2.1. Informazioni di background sul contesto in cui si applica il software

Come accuratamente spiegato nel Capitolo 1, lo scopo finale della chat è rendere le informazioni scambiate tra gli utenti il più possibile sicure e inaccessibili, cancellando poi ogni traccia del loro passaggio a conversazione conclusa. EASE non necessita di registrazione, non richiede una e-mail e nemmeno nome e cognome dell'utente. Questo va a colmare il diritto all'oblio, vale a dire il diritto a poter cancellare completamente i propri dati personali, in questo caso contenuti solamente nella conversazione.

Per quanto riguarda la sicurezza delle informazioni scambiate, ogni messaggio che viene salvato all'interno di questa chat viene criptato con l'algoritmo AES, con una chiave di 256bit, per migliorarne l'efficienza e la robustezza, in caso di attacco brute force. Per quanto riguarda poi la comunicazione con il server, un'ulteriore sicurezza è rappresentata dall'utilizzo del protocollo HTTPS per garantire trasferimenti riservati di dati nel web, in modo da impedire intercettazioni dei contenuti che potrebbero essere effettuati tramite tecnica del man in the middle.

2.2. Algoritmo AES (Advanced Encryption Standard)

L'algoritmo in questione è un sistema di cifratura a blocchi che viene utilizzato per la crittografia di software e hardware in quanto è uno degli algoritmi più efficienti e sicuri attualmente sviluppati. Per questo motivo viene adottato come algoritmo standard dal governo degli Stati Uniti d'America (http://www.nist.gov/). È il successore dell'algoritmo DES. Accetta chiavi da 128, 192 e 256 bits. Citando Wikipedia "Data la sua sicurezza e le sue specifiche pubbliche si presume che in un prossimo futuro venga utilizzato in tutto il mondo come è successo al suo predecessore, il Data Encryption Standard (DES) che ha perso poi efficacia per vulnerabilità intrinseche. AES è stato adottato dalla National Institute of Standards and Technology (NIST) e dalla US FIPS (Federal Information Processing Standard) PUB nel novembre del 2001 dopo 5 anni di studi, standardizzazioni e selezione finale tra i vari algoritmi proposti".

2.3. Crittografia simmetrica

La crittografia simmetrica è uno schema caratterizzato dalla proprietà che, data una chiave di cifratura, sia facilmente calcolabile la chiave di decifratura. Un caso particolare, che è quello quasi sempre utilizzato nella pratica, è l'utilizzo di una stessa chiave condivisa sia per l'operazione di cifratura che per quella di decifratura, come sviluppato nel progetto. La forza della crittografia simmetrica è dunque riposta nella segretezza dell'unica chiave utilizzata dai due interlocutori che la usano, oltre che nella grandezza dello spazio delle chiavi, anche nella scelta di una buona chiave e nella resistenza dell'algoritmo agli attacchi di crittoanalisi. In questo caso, nel software sviluppato è stata utilizzata una chiave di cifratura a 256 bit, che viene creata all'avvio della stanza e resta privata tra gli utenti che si connettono ad essa, e l'algoritmo AES per la crittografia dei dati all'interno della chat.

2.4. Prodotti simili

Sono diverse le chat simili che possiamo trovare sulla rete e sugli store più famosi. Qui di seguito ne verranno elencate alcune, soffermandoci sull'aspetto della sicurezza delle comunicazioni.

Whatsapp: È la prima che andremo ad esaminare, non per importanza o per sicurezza, ma per il successo che questa applicazione ha avuto negli ultimi anni. Whatsapp è la nota applicazione di messaggistica istantanea multipiattaforma, che gran parte di noi possiede, acquistata il 19 Febbraio del 2014 dalla altrettanto nota compagnia statunitense Facebook inc. Per quanto riguarda la sicurezza dei messaggi scambiati, dal 2014 è stata integrata una funzionalità 'nascosta'. Infatti senza saperlo gli utenti utilizzano una crittografia end-to-end, basata sul protocollo TextSecure. La limitazione però sta nel fatto che questa tecnologia è applicata soltanto alle conversazioni one-to-one, quindi tra singoli utenti, e non in gruppi di chat.

Con il termine end-to-end si indica che i messaggi non possono essere decifrati nemmeno dal fornitore del servizio, anche nel caso di richieste da parte delle forze di polizia.

Skype: È un software proprietario freeware di messaggistica istantanea e VoIP. Per quanto riguarda la sicurezza delle informazioni, quindi in termini di crittografia, Skype utilizza due metodi, TLS (transport-level security) per crittografare i messaggi tra il client Skype e il servizio chat del cloud, oppure AES (Advanced Encryption Standard) se l'invio avviene

direttamente tra due client Skype. Il più utilizzato è quest'ultimo, con la crittografia complessa a 256bit. Le chiavi utente pubbliche sono certificate dal server Skype al momento del login.

Google Hangout: È il software di messaggistica istantanea e VoIP sviluppato da Google. È multipiattaforma, ed è integrato all'interno di gran parte delle Google apps. In G. Hangout tutti i segnali, ad esempio i messaggi, vengono crittografati e inviati su una connessione HTTPS a 128 bit, utilizzando TLS 1.2. La connessione è crittografata ed autenticata mediante AES a 128 bit. Il meccanismo utilizzato per lo scambio delle chiavi è ECDHE-ESCSA. Mentre per quanto riguarda audio e video viene adottata una connessione P2P quando possibile. In seguito ad una conversazione su Reddit, avvenuta a metà del 2015, alla quale hanno partecipato anche Richard Salgrado (Director for Law Enforcement and Information Security di Google) e David Lieber (Senior Privacy Policy Counsel di Google), è emerso un particolare contrastante con la descrizione del servizio, presente sul sito ufficiale di Hangout. La crittografia avviene esclusivamente "in transit", ovvero mentre i messaggi vengono trasmessi dal computer verso i server di Google. Una volta arrivati però, sono accessibili all'azienda, che dunque avrebbe la possibilità di archiviarli e fornirli alle autorità, in caso di formale richiesta relativa a un'indagine.

iMessage: Apple iMessage è un servizio di messaggistica per dispositivi iOS e Mac che supporta testo e allegati foto, contatti e posizioni. Il proprietario del servizio, la Apple inc. afferma di non tener traccia dei messaggi o degli allegati, e i loro contenuti sono protetti da una crittografia end-to-end, così da permettere l'accesso soltanto al mittente e il destinatario. Per quanto riguarda questa crittografia end-to-end, il dispositivo genera due coppie di chiavi, da utilizzare con il servizio: una chiave RSA a 1280bit per la codifica e una ECDSA a 245bit per la firma. Le chiavi private di entrambe le coppie sono salvate nel portachiavi del dispositivo, mentre le chiavi pubbliche vengono inviate al servizio di directory APPLE, dove sono associate al numero di telefono dell'utente. Il processo di invio e ricezione dei messaggi avviene codificando il messaggio in uscita. Le chiavi di codifica RSA pubbliche dei dispositivi riceventi vengono recuperate dall'IDS. Per ciascun dispositivo ricevente, il dispositivo mittente genere una chiave casuale a 128bit e la utilizza per codificare il messaggio con AES in modalità CTR.

Telegram: per ultimo ma non meno importante, Telegram è un servizio di messaggistica istantanea multipiattaforma erogato senza fini di lucro della società Telegram LLC. La peculiarità che rende noto e competitivo questo software è la possibilità di stabilire conversazioni cifrate punto-punto e scambiare file di dimensioni fino a 1,5GB. Nonostante l'app abbia preso piede negli ultimi tempi, l'aspetto della sicurezza non rispecchia le caratteristiche dichiarate dai produttori. Questo perché l'app offre una crittografia end-to-end, ma l'utente deve selezionare appositamente l'opzione "Secret Chat", altrimenti le conversazioni normali sono non criptate. Inoltre Telegram salva automaticamente tutti i contatti senza il consenso dell'utente o delle persone interessate.

Qui di seguito quindi, riassumiamo i più noti ed utilizzati software di messaggistica istantanea presenti attualmente sul mercato:

Nome	**Crypting Algorithm**	**Prezzo**	**Produttore**
Whatsapp	AES end-to-end	FREE	Facebook inc.
Skype	TLS – AES	FREE	Microsoft
Google Hangout	AES *in-transit*	FREE	Google
iMessage	AES end-to-end	FREE	Apple inc.
Telegram	AES (Secret Chat only)	FREE	Telegram LLC.

Tabella 1. Studio dei più noti software di messaggistica istantanea

3. Progettazione di EASE

In questa sezione si andrà a ad esaminare passo per passo la progettazione del software e le motivazioni per le quali si è deciso di utilizzare alcune tecnologie, fondamentali per lo sviluppo di alcuni strumenti.

Ho dovuto esaminare, passo per passo, le funzionalità che si era deciso di sviluppare, e le tecnologie esistenti per permettere un corretto funzionamento del software finale.

3.1. Tecnologie utilizzate

Per lo sviluppo del software sono state utilizzate le seguenti tecnologie e linguaggi di programmazione:

- HTML5 (HyperText Markup Language versione 5)
- HTTPS (HyperText Transfer Protocol over Secure Socket Layer)
- JQuery
- AJAX (Asynchronous JavaScript and XML**)**
- Libreria JS CryptoJS
- PHP (Hypertext Preprocessor)
- Memcache

Andiamo ora ad analizzare, uno per uno, i linguaggi di programmazione e le tecnologie utilizzate.

Per la formattazione e l'impaginazione della pagina web è stato utilizzato il linguaggio di markup HTML5. La traduzione letterale dell'acronimo (Hyper Text Markup Language) significa linguaggio a marcatori per ipertesti, e sta a significare che è il linguaggio markup usato per ipertesti disponibili nel Word Wide Web sotto forma di pagine web. Si tratta di un linguaggio di pubblico dominio, la cui sintassi è stabilita dal World Wide Web Consortium (W3C), un'organizzazione non governativa internazionale che ha come scopo lo sviluppo delle potenzialità del World Wide Web. Per il progetto è stata utilizzata la versione 5 di HTML perché è una versione recente e integra la gestione di diversi errori che prima andavano gestiti via JAVASCRIPT, oltre a una dozzina di novità che facilitano la vita dello sviluppatore web.

HTTPS è il risultato dell'unione di un protocollo di crittografia asimmetrica al protocollo di trasferimento di ipertesti HTTP. Viene tipicamente usato per garantire trasferimenti riservati di dati nel web, in modo da impedire intercettazioni che potrebbero essere effettuate tramite la tecnica del man in the middle. Questo protocollo utilizza la porta 443, integra un livello di autenticazione come il Secure Sockets Layer (SSL) tra il protocollo TCP e HTTP. Il flusso di dati tra client e server è collegato da un canale di comunicazione tramite uno scambio di certificati. Questo tipo di comunicazione garantisce che solamente il client e il server siano in grado di conoscere il contenuto della comunicazione.

JQuery è una libreria JavaScript per lo sviluppo di applicazioni web. Nasce con l'obiettivo di semplificare la selezione, la manipolazione, la gestione degli eventi e l'animazione di elementi DOM (Document Object Model) in pagine HTML, nonché implementare funzionalità AJAX.

AJAX è l'acronimo di Asynchronous JavaScript and XML. Questa è una tecnica di sviluppo software per la realizzazione di applicazioni web interattive, anche dette Rich Internet Application. Il funzionamento di AJAX si basa sullo scambio di dati in background fra browser e server, che consente l'aggiornamento dinamico di una pagina web senza esplicito ricaricamento da parte dell'utente. Infatti nel software sviluppato si è fatto largamente uso di questa tecnologia, basti pensare all'aggiornamento della chat, o degli utenti connessi alla stanza e alcuni controlli sincroni alla creazione delle stanze.

Al progetto inoltre è stata aggiunta la nota libreria JavaScript *CryptoJS*, la quale fornisce diversi algoritmi di crypting noti, tra cui SHA3, HMAC, tripleDES, Rabbit, AES eccetera. Nel caso del software sviluppato, si utilizza l'algoritmo AES (descritto ed analizzato nel capitolo 2.2) per criptare i dati end-to-end con una chiave randomica da 256bit generata da una funzione JavaScript presente all'interno delle pagine. La scelta di tale algoritmo, piuttosto che gli altri disponibili nella libreria inclusa al progetto, è dovuta al fatto che AES è un algoritmo veloce sia se implementato in software o in hardware, richiede poca memoria ed offre un buon livello di protezione.

Per la gestione dei dati lato server è stato utilizzato il noto linguaggio di scripting PHP (Hypertext Preprocessor), originariamente concepito per la programmazione di pagine web dinamiche. Attualmente si utilizza questo linguaggio per applicazioni web lato server, ovvero svolge le proprie operazioni dal server in un ambito client-server, contrapponendosi a tutto ciò che viene eseguito dal client. Viene anche utilizzato per scrivere script a riga di

comando o applicazioni stand-alone (applicazione in grado di funzionare in maniera autonoma, indipendentemente da altri oggetti o software, con cui potrebbe altrimenti interagire). La scelta di tale linguaggio deriva dalla documentazione completa disponibile su sito ufficiale in diverse lingue, oltre ad una enorme comunità di sviluppo, grazie alla quale è possibile trovare innumerevoli esempi, guide e script. Al file di configurazione PHP è stata inoltre aggiunta l'estensione memcache. Questo è un sistema di caching distribuito il quale ha una struttura client/server che permette di servire i dati più richiesti direttamente dalla RAM, riducendo allo stesso tempo il carico sul database. Nell'ambito del progetto è stato utilizzato con lo scopo di salvare appunto le sessioni in RAM, così da rendere la lettura e la scrittura ancora più veloci, e impedirne la lettura ai non autorizzati. In quanto si lavora su una memoria volatile e allo stesso tempo di dimensioni molto ridotte piuttosto che una memoria di massa molte volte più grande, si è deciso di salvare un massimo di 10 messaggi alla volta nella chat, così da contenere le dimensioni del file di sessione e rendere l'aggiornamento del testo ancora più rapido.

3.2. Ingegnerizzazione del software

Qui di seguito verrà fatto uno studio accurato del software e la sua ingegnerizzazione, quindi lista dei requisiti, Use Case diagram, UML e Activity Diagram.

Lista dei requisiti:

Requisiti funzionali:

- L'utente deve poter creare una stanza
- L'utente deve potersi unire ad una stanza
- L'utente deve poter scrivere messaggi nella chat
- L'utente può unirsi alla chat senza credenziali
- L'utente deve conoscere ID e chiave per unirsi ad una stanza
- L'utente può lasciare liberamente una stanza

Requisiti extra-funzionali:

- Il sistema software deve garantire che i messaggi non siano intercettati
- Il sistema software deve garantire che i messaggi restino comprensibili soltanto agli estremi durante il viaggio delle informazioni tra client e server

- Il sistema software deve validare i campi form all'interno della pagina login
- Il sistema software deve saper identificare quando un utente esce dal sito
- Il sistema software deve richiedere soltanto il nome all'utente che desidera aprire una nuova stanza
- Il sistema software deve richiedere nome, id stanza e chiave di cripting all'utente che desidera unirsi ad una stanza esistente
- Il sistema software deve garantire la privacy degli utenti
- Il sistema software deve garantire che le informazioni non possano essere lette da terzi

Use Case Diagram:

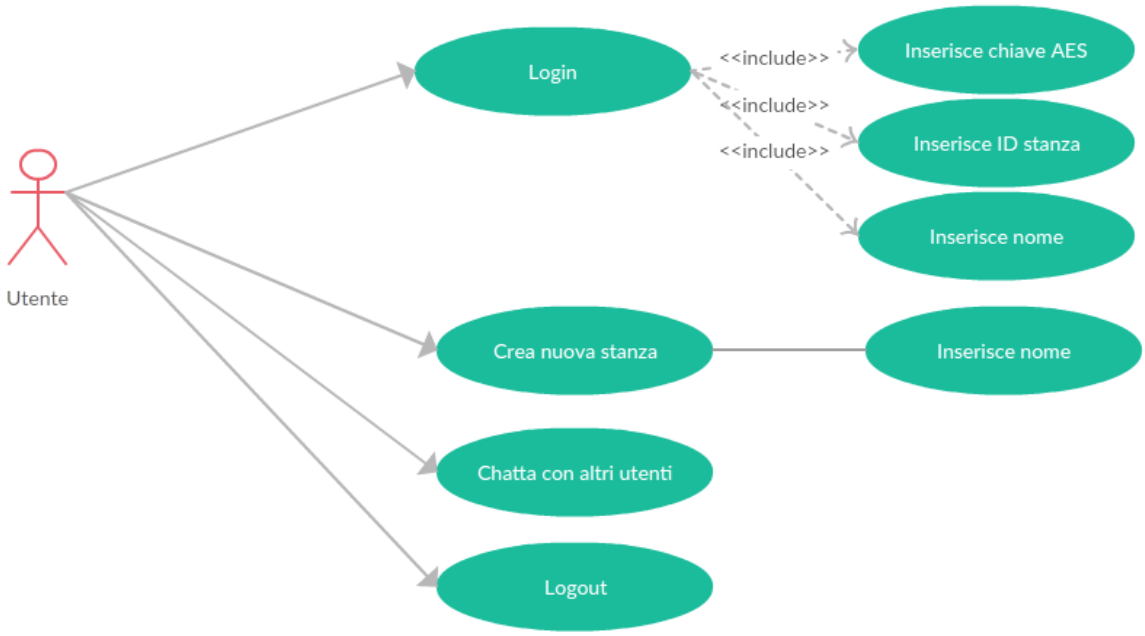

Figura 2. Use Case Diagram-Diagramma dei casi di utilizzo della webchat

Lo use case raffigurato nella Figura 2 indica le azioni che l'utente connesso al sito può svolgere.

L'utente ha modo di creare una stanza, la quale richiederà soltanto un nickname che l'utente inserirà per entrare nella stanza di chat.

Nel caso in cui l'utente voglia fare il login in una stanza esistente, gli verrà richiesto di inserire il nickname che intende utilizzare nella stanza, il codice identificatore della stanza e la chiave di accesso AES della stanza.

Una volta dentro la stanza l'utente ha modo di chattare con altri utenti connessi alla stanza corrente, scrivendo nella casella di testo presente nella pagina.

Quando l'utente chiude il browser o la scheda, il sito effettuerà il logout dell'utente stesso il quale non risulterà più nella pagina.

Activity Diagram:

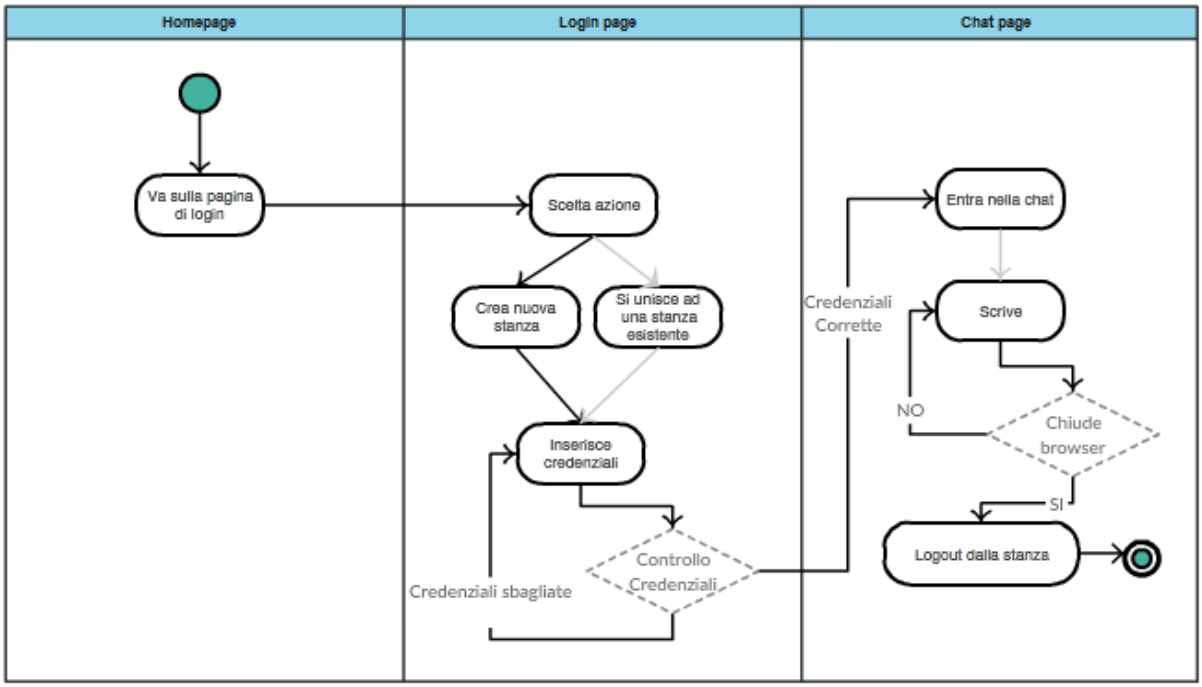

Figura 3. Activity Diagram-diagramma delle azioni operabili sul sito

L'activity Diagram raffigurato in Figura 3, indica le azioni che l'utente effettua dalla homepage fino alla pagina di chat.

L'utente che si connette al sito sarà indirizzato automaticamente alla homepage, ovvero la pagina di benvenuto. Qui l'utente si dirige alla pagina di login attraverso il collegamento ipertestuale posizionato nell'angolo superiore destro della pagina.

Nella pagina di login, l'utente ha modo di scegliere l'azione da svolgere, se creare una nuova stanza oppure unirsi ad una stanza esistente. Entrambe le azioni porteranno ad un inserimento delle credenziali e ad un controllo di esse da parte della pagina. Se le credenziali sono errate o non sono state inserite viene richiesto di reinserirle correttamente, altrimenti significa che le credenziali sono sufficienti e si passa alla pagina di chat.

L’utente viene inserito nella lista degli utenti connessi alla chat (entra nella chat) ed è libero di scrivere agli altri utenti connessi. Se la pagina di chat o il browser utilizzato non vengono chiusi, l’utente resta nella chat, altrimenti se questo avviene si effettua il logout dalla stanza e l’utente non risulterà più connesso.

4. Utilizzo del software

4.1. Utilizzo

Il software, come si evince dalla Figura 4 sottostante, si presenta con una homepage dallo stile minimale, che descrive il servizio che fornisce e i vantaggi nell'utilizzo di questa web chat in termini di privatezza ed affidabilità.

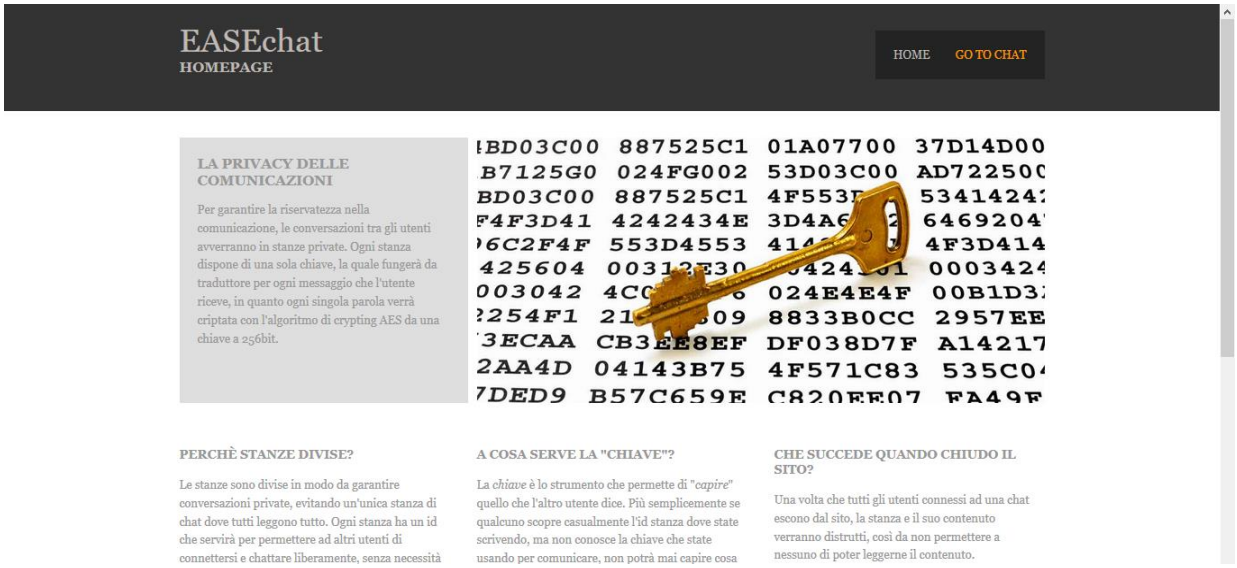

Figura 4. Homepage EASE

Da qui l'utente ha la possibilità di usufruire del servizio, direttamente cliccando sul link nel menu "go to chat". Questo porta alla pagina *login.html* che metterà l'utente di fronte alla scelta di:

- Creare una nuova stanza (Figura 5)

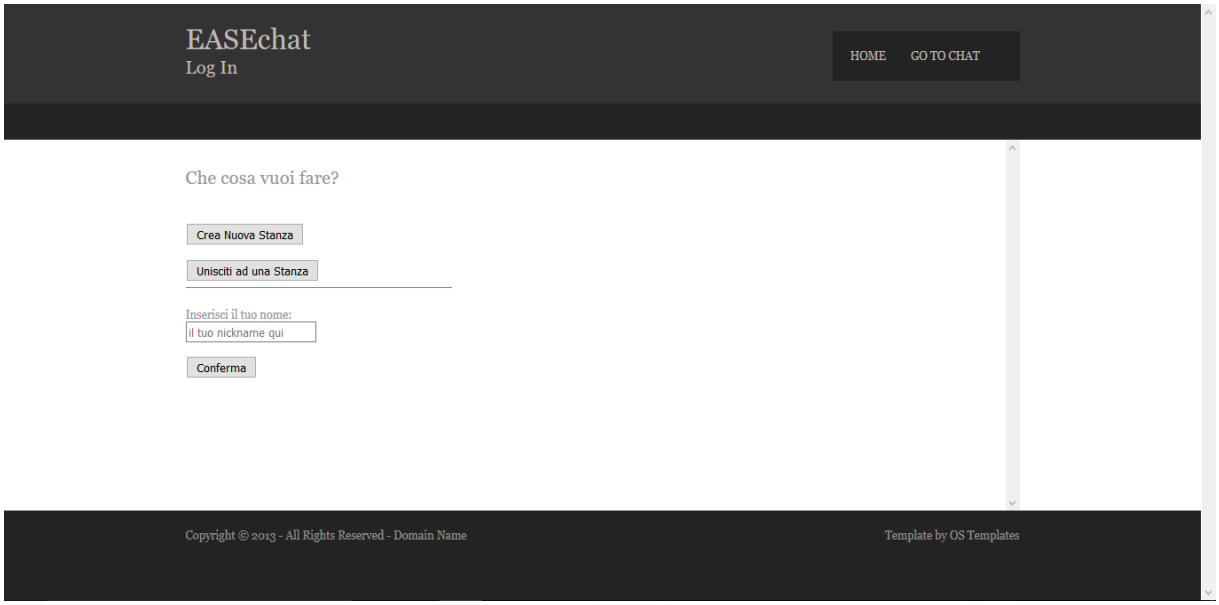

Figura 5. Creazione nuova stanza-pagina di login

– Unirsi ad una stanza esistente (Figura 6)

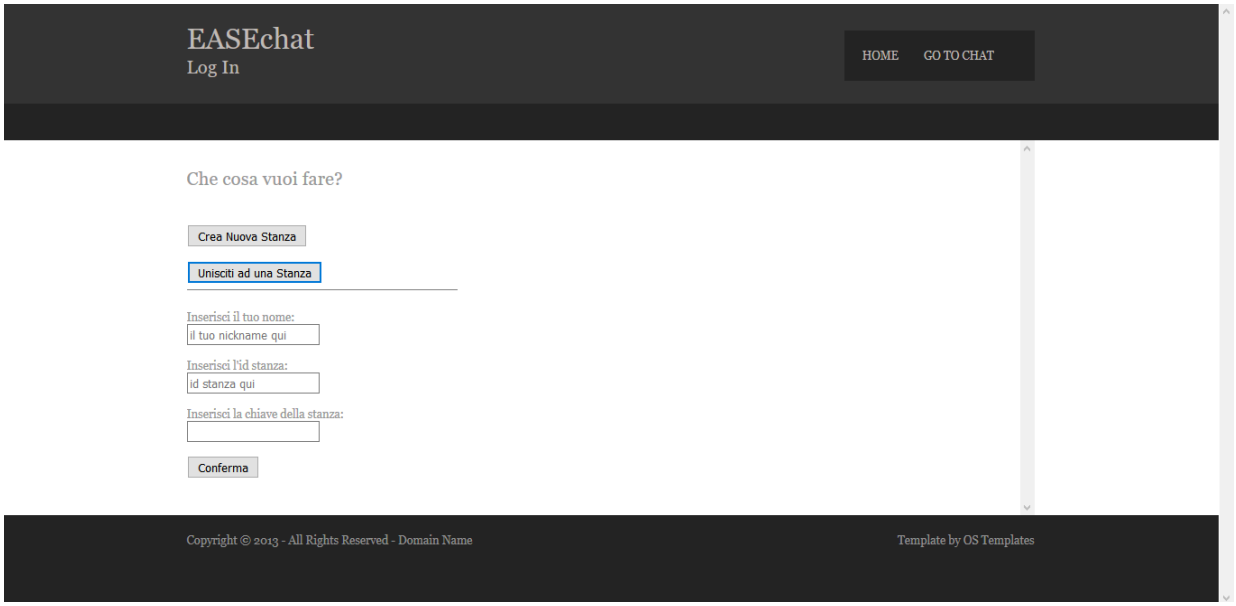

Figura 6. Login in una stanza esistente-pagina di login

La pagina di chat si presenterà, come in Figura 7, con la sezione di benvenuto, dove vengono specificati id della stanza e la chiave di codifica, entrambi indispensabili per permettere ad altri utenti di loggarsi nella chat room. A destra ci sono i nomi degli utenti connessi alla chat e al centro i messaggi scambiati. Il numero massimo di messaggi visualizzati sono 10, per via della memorizzazione dei messaggi in RAM.

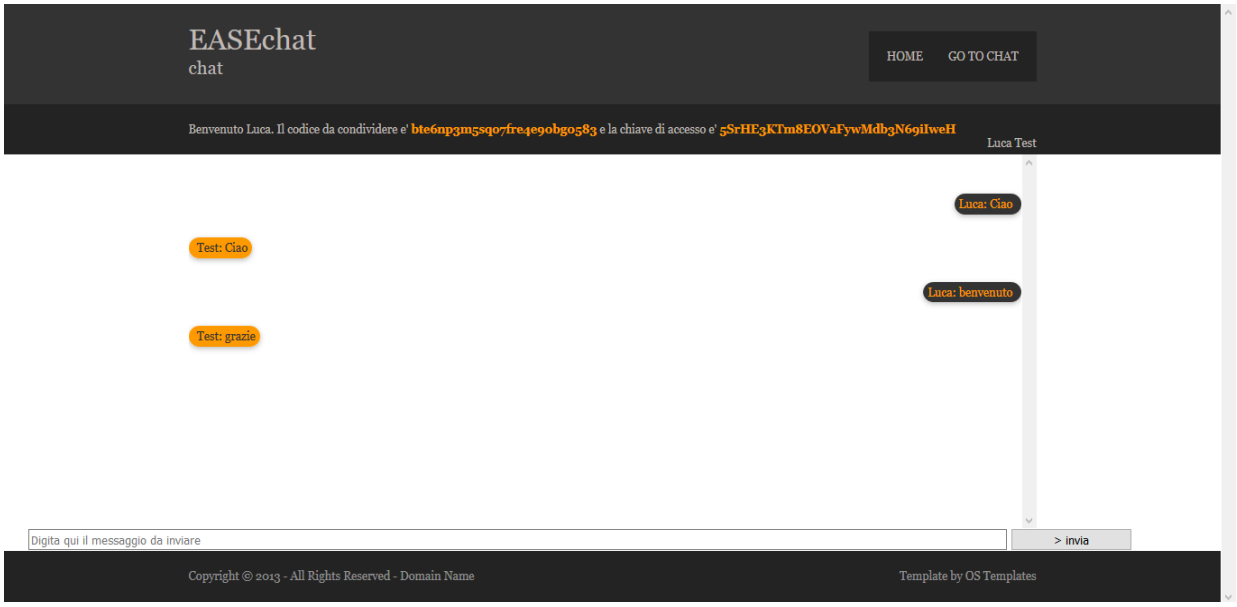

Figura 7. Pagina di chat di EASE

4.2. Come il software garantisce la sicurezza delle informazioni scambiate

Come analizzato nel Capitolo 2.4, e rappresentato schematicamente nella Tabella 1, gran parte dei software di messaggistica istantanea presenti oggi sul mercato, utilizzano la crittografia end-to-end per assicurare, a chi utilizza il software, la sicurezza che nessuno, eccetto lui e l'interlocutore, sappiano il contenuto della chat.

Questo metodo è stato utilizzato anche per EASE, così da impedire, a chi riesce a leggere la sessione lato server, di capirne il contenuto. Chi inoltre tentasse di utilizzare la tecnica MITM (Man in the Middle) su questo software, ovvero "sniffare" i pacchetti che fluiscono in una rete e leggerli, sarà ostacolato da HTTPS (descritto nel Capitolo 3.1), il quale invia dati criptati in rete, così da garantire che soltanto il client e il server siano in grado di conoscere il contenuto della comunicazione.

L'utente potrebbe tentare una modifica dei contenuti delle pagine attraverso gli strumenti di sviluppo integrati nei browser, o un'estensione come ad esempio Firebug, la quale permette la modifica e il monitoraggio di tutti gli aspetti di una pagina web. Questo viene previsto dal software, il quale gestisce queste eccezioni e ne fa continuare comunque il funzionamento.

Infine ma non meno importante, nel caso in cui un attaccante decida di leggere i file di sessione presenti nel server, troverebbe una serie di problemi ed ostacoli dovuti al fatto che queste sono salvate, grazie all'utilizzo di memcache (descritto nel capitolo 3.1), all'interno

della memoria RAM, la quale è difficilmente leggibile, e richiede tempi di attacco mediamente lunghi.

5. Conclusioni e Sviluppi Futuri

La web chat qui descritta è attualmente fine a se stessa, a dimostrazione che è possibile sviluppare un sistema di chat elementare, con tecnologie di base come html e JAVASCRIPT, ma allo stesso tempo sicura sotto ogni aspetto grazie ad algoritmi di crypting molto potenti e protocolli sicuri. Gli obiettivi inizialmente prefissati sono stati realizzati, ottenendo un servizio di messaggistica web quasi inattaccabile. Il sito è attualmente presente all'indirizzo http://easechat.altervista.org/. Purtroppo in quanto un hosting gratuito, non è stato possibile integrare memcache (Capitolo 3.1) al sito, ma è stato invece possibile richiedere l'utilizzo del protocollo HTTPS.

Il futuro di questo progetto può andare oltre una semplice tesi ed eguagliare il grado di sicurezza delle app di messaggistica più famose. Infatti uno sviluppo futuro di questa chat sarà un'applicazione Android con le stesse funzionalità, che vada ad utilizzare lo stesso metodo di comunicazione client server e le stesse tecnologie di crypting, oltre all'acquisto di un dominio privato per il servizio web.

Per finire, mi sembra doveroso riprendere l'attuale dibattito sulla contrapposizione privacy/sicurezza, con cui ho aperto l'introduzione. Non c'è dubbio che i pericoli, a cui oggi siamo esposti, debbono essere combattuti, tuttavia le chat vanno salvaguardate con motivazioni altrettanto valide. Innanzi tutto, rispetto alla varietà dei mezzi di comunicazione, oggi possibili e inimmaginabili fino a pochi decenni fa (si pensi alla posta elettronica), le chat hanno il pregio di offrire una comunicazione in tempo reale e consentono di dialogare a distanza, mantenendo vivo il rapporto anche tra persone che, per diversi motivi, sono impossibilitati a comunicare faccia a faccia. E' da sfatare anche il luogo comune per cui la chat sia frivola e futile e, come tale, destinata a non avere storia. Il discredito di cui godevano le chat fino a poco tempo fa, ha dimostrato tutta la sua inconsistenza, tanto che l'utenza che le usa è sempre più numerosa, comprendendo tutti i ceti socio-culturali e tutte le età.

6. Sitografia

- Sito web ufficiale linguaggio di scripting PHP

  http://php.net/

- Algoritmo AES Wikipedia

  https://it.wikipedia.org/wiki/Advanced_Encryption_Standard

- Crittografia Simmetrica (Wikipedia)

  https://it.wikipedia.org/wiki/Crittografia_simmetrica

- Documentazione libreria JavaScript CryptoJS
  https://code.google.com/p/crypto-js/

- Memcached official website
  http://memcached.org/about
- JQuery official website

  https://jquery.com/

- Ajax programming DMOZ.org

  https://www.dmoz.org/Computers/Programming/Languages/JavaScript/AJAX

- Ajax tutorial

  http://www.xul.fr/en-xml-ajax.html

- Installing memcached And The PHP5 memcache Module On Debian

  https://www.howtoforge.com/installing-memcached-and-the-php5-memcache-module-on-debian-etch-apache2

- Articoli riguardo le webchat esistenti

  WhatsApp:

  http://news.softonic.it/telegram-non-e-poi-cosi-sicura-come-afferma-whatsapp-meno-che-mai

http://enjoyphoneblog.it/33836/guide/come-leggere-i-messaggi-whatsapp-di-un-altra-persona-dal-pc.html

http://www.androidworld.it/2016/01/05/whatsapp-e-la-verifica-della-crittografia-end-to-end-364190/

http://www.androidworld.it/2014/11/18/whatsapp-adotta-crittografia-end-to-end-android-prima-ios-259179/

Telegram:

http://news.softonic.it/telegram-non-e-poi-cosi-sicura-come-afferma-whatsapp-meno-che-mai

Skype:

https://support.skype.com/it/faq/FA31/skype-usa-la-crittografia

iOS (iMessage) official documentation

https://www.apple.com/business/docs/iOS_Security_Guide.pdf

- Sito ufficiale Whatsapp

  https://www.whatsapp.com

- Sito ufficiale Skype

  https://skype.com

Da questo punto in poi vorrei fare dei ringraziamenti speciali a chi mi ha sostenuto durante questo percorso di studi, chi economicamente, chi con il suo prezioso tempo e chi regalandomi una compagnia insostituibile. Ringrazio la mia fantastica famiglia per il supporto economico e morale che mi ha permesso di sostenere gli studi in maniera più che serena. Ringrazio i miei coinquilini per l'amicizia speciale durante questi anni di studi, che spero durerà ancora a lungo. Ringrazio tutti i miei amici, di Osimo e di Camerino. Ringrazio chi mi è stato vicino durante la realizzazione della tesi, Tommaso, Fabio e Angela. Ringrazio i miei compagni di corso per aver intrapreso questa esperienza che è l'università con me. Infine ringrazio il mio correlatore, per la pazienza portata nella correzione del mio progetto e della tesi.

Grazie a tutti