Protocollo Signal

Elena Tonini, Matr.727382

Università degli Studi di Brescia

A.A. 2021/2022

Protocollo Signal

Elena Tonini, Matr.727382

Università degli Studi di Brescia

A.A. 2021/2022

Sommario I

1 Sommario
2 Applicazione Signal
Storia dell'Applicazione
L'Applicazione e il Protocollo Signal
3 Crittografia End-to-End
Applicazioni
Problematiche
4 Signal Protocol
Il protocollo
Proprietà
Attacchi possibili
Considerazioni
WhatsApp VS Signal VS Telegram
5 Bibliografia

Sommario I

1. Sommario
2. Applicazione Signal
   Storia dell'Applicazione
   L'Applicazione e il Protocollo Signal
3. Crittografia End-to-End
   Applicazioni
   Problematiche
4. Signal Protocol
   Il protocollo
   Proprietà
   Attacchi possibili
   Considerazioni
   WhatsApp VS Signal VS Telegram
5. Bibliografia

Applicazione Signal

Storia dell'Applicazione

L'applicazione Signal ha origine dall'unione dei due servizi di messaggistica **TextSecure** e **RedPhone**, sviluppati da **Moxie Marlinspike** e **Stuart Anderson**, che insieme fondarono la start-up **Whisper Systems** nel 2010.

Entrambe le applicazioni implementavano la crittografia end-to-end.

Applicazione Signal
Storia dell'Applicazione
L'applicazione Signal ha origine dall'unione dei due servizi di messaggistica **TextSecure** e **RedPhone**, sviluppati da **Moxie Marlinspike** e **Stuart Anderson**, che insieme fondarono la start-up **Whisper Systems** nel 2010.
Entrambe le applicazioni implementavano la crittografia end-to-end.

Applicazione Signal

Storia dell'Applicazione

L'applicazione Signal ha origine dall'unione dei due servizi di messaggistica **TextSecure** e **RedPhone**, sviluppati da **Moxie Marlinspike** e **Stuart Anderson**, che insieme fondarono la start-up **Whisper Systems** nel 2010.

Entrambe le applicazioni implementavano la crittografia end-to-end.

Applicazione Signal
Storia dell'Applicazione

L'applicazione Signal ha origine dall'unione dei due servizi di messaggistica **TextSecure** e **RedPhone**, sviluppati da **Moxie Marlinspike** e **Stuart Anderson**, che insieme fondarono la start-up **Whisper Systems** nel 2010.

Entrambe le applicazioni implementavano la crittografia end-to-end.

Applicazione Signal

Storia dell'Applicazione

A seguito del nuovo rilascio delle applicazioni nel 2011 i due servizi assumono la propria natura **open-source** che ancora oggi caratterizza l'applicazione Signal.

Nel 2013 Marlinspike fonda il progetto open-source **Open Whisper Systems**, grazie a cui rilascia la prima versione di Signal nel 2015 (anche per PC come applicazione Chrome), per poi rilasciarlo anche per Windows, Mac e Linux nel 2017.

Nel 2011 Twitter acquista Whisper Systems e Marlinspike diventa capo della cybersecurity del social media.
Nel 2013 Marlinspike abbandona Twitter e fonda la OWS.

Nello stesso anno inizia a lavorare al protocollo Signal insieme al fondatore di WhatsApp Trevor Perrin.

Applicazione Signal

Storia dell'Applicazione

A seguito del nuovo rilascio delle applicazioni nel 2011 i due servizi assumono la propria natura **open-source** che ancora oggi caratterizza l'applicazione Signal.

Nel 2013 Marlinspike fonda il progetto open-source **Open Whisper Systems**, grazie a cui rilascia la prima versione di Signal nel 2015 (anche per PC come applicazione Chrome), per poi rilasciarlo anche per Windows, Mac e Linux nel 2017.

Applicazione Signal
Storia dell'Applicazione

A seguito del nuovo rilascio delle applicazioni nel 2011 i due servizi assumono la propria natura **open-source** che ancora oggi caratterizza l'applicazione Signal.

Nel 2013 Marlinspike fonda il progetto open-source **Open Whisper Systems**, grazie a cui rilascia la prima versione di Signal nel 2015 (anche per PC come applicazione Chrome), per poi rilasciarlo anche per Windows, Mac e Linux nel 2017.

Nel 2011 Twitter acquista Whisper Systems e Marlinspike diventa capo della cybersecurity del social media.
Nel 2013 Marlinspike abbandona Twitter e fonda la OWS.

Nello stesso anno inizia a lavorare al protocollo Signal insieme al fondatore di WhatsApp Trevor Perrin.

Applicazione Signal

Storia dell'Applicazione

Nel febbraio 2018 Marlinspike e il co-fondatore di WhatsApp Brian Acton fondarono la **Signal Foundation**, il cui obiettivo è il supporto e l'accelerazione della diffusione della comunicazione privata e sicura. [Lumer]

Applicazione Signal
Storia dell'Applicazione
Nel febbraio 2018 Marlinspike e il co-fondatore di WhatsApp Brian Acton fondarono la **Signal Foundation**, il cui obiettivo è il supporto e l'accelerazione della diffusione della comunicazione privata e sicura. [Lumer]

Applicazione Signal

L'Applicazione e il Protocollo Signal

Nel 2013, dopo la fondazione di OWS, i fondatori Marlinspike e Trevor Perrin iniziarono a lavorare al **Protocollo Signal**.

Esso rendeva il metodo crittografico end-to-end utilizzato nell'applicazione Signal implementabile anche da altri servizi.

Ogni piattaforma di messaggistica che intraprese collaborazioni con OWS al fine di integrare il protocollo Signal al proprio interno lo implementò in modalità differenti e su scala/estensione diversa.

Applicazione Signal
L'Applicazione e il Protocollo Signal

Nel 2013, dopo la fondazione di OWS, i fondatori Marlinspike e Trevor Perrin iniziarono a lavorare al **Protocollo Signal**.

Esso rendeva il metodo crittografico end-to-end utilizzato nell'applicazione Signal implementabile anche da altri servizi.

Ogni piattaforma di messaggistica che intraprese collaborazioni con OWS al fine di integrare il protocollo Signal al proprio interno lo implementò in modalità differenti e su scala/estensione diversa.

Applicazione Signal

L'Applicazione e il Protocollo Signal

Nel 2013, dopo la fondazione di OWS, i fondatori Marlinspike e Trevor Perrin iniziarono a lavorare al **Protocollo Signal**.

Esso rendeva il metodo crittografico end-to-end utilizzato nell'applicazione Signal implementabile anche da altri servizi.

Ogni piattaforma di messaggistica che intraprese collaborazioni con OWS al fine di integrare il protocollo Signal al proprio interno lo implementò in modalità differenti e su scala/estensione diversa.

Applicazione Signal
L'Applicazione e il Protocollo Signal
Nel 2013, dopo la fondazione di OWS, i fondatori Marlinspike e Trevor Perrin iniziarono a lavorare al **Protocollo Signal**.
Esso rendeva il metodo crittografico end-to-end utilizzato nell'applicazione Signal implementabile anche da altri servizi.
Ogni piattaforma di messaggistica che intraprese collaborazioni con OWS al fine di integrare il protocollo Signal al proprio interno lo implementò in modalità differenti e su scala/estensione diversa.

Applicazione Signal

L'Applicazione e il Protocollo Signal

Nel 2013, dopo la fondazione di OWS, i fondatori Marlinspike e Trevor Perrin iniziarono a lavorare al **Protocollo Signal**.

Esso rendeva il metodo crittografico end-to-end utilizzato nell'applicazione Signal implementabile anche da altri servizi.

Ogni piattaforma di messaggistica che intraprese collaborazioni con OWS al fine di integrare il protocollo Signal al proprio interno lo implementò in modalità differenti e su scala/estensione diversa.

Applicazione Signal

L'Applicazione e il Protocollo Signal

Nel 2013, dopo la fondazione di OWS, i fondatori Marlinspike e Trevor Perrin iniziarono a lavorare al **Protocollo Signal**.

Esso rendeva il metodo crittografico end-to-end utilizzato nell'applicazione Signal implementabile anche da altri servizi.

Ogni piattaforma di messaggistica che intraprese collaborazioni con OWS al fine di integrare il protocollo Signal al proprio interno lo implementò in modalità differenti e su scala/estensione diversa.

Applicazione Signal

L'Applicazione e il Protocollo Signal

Tra le più note implementazioni (parziali) del Protocollo Signal troviamo:

- **Facebook**: introdusse la feature *Secret Conversations* per gli utenti di Facebook Messenger nel luglio 2016
- **Allo**: rilasciata nel settembre 2016, sfruttava il Protocollo Signal se utilizzata in modalità incognito
- **Duo**: protezione delle videochat
- **Skype**: conversazioni private dal 2018
- **WhatsApp**: tra le maggiori applicazioni che implementano Signal è l'unica che garantisce di default la crittografia end-to-end delle conversazioni (da aprile 2016)

[Gre29], [Lumer]

- Facebook: usa Signal solo nelle Secret Conversations
- Allo: applicazione mobile di messaggistica istantanea di Google, non esiste più dal 12 marzo 2019
- Duo: applicazione per videochiamate e chat mobile di Google
- Whatsapp: introdusse Signal per la prima volta nel 2014 per utenti Android, estendendolo a tutti gli utenti nel 2016
- Google: introduce Signal di default nell'applicazione di messaggi su Android

Applicazione Signal

L'Applicazione e il Protocollo Signal

Tra le più note implementazioni (parziali) del Protocollo Signal troviamo:

- **Facebook**: introdusse la feature *Secret Conversations* per gli utenti di Facebook Messenger nel luglio 2016
- **Allo**: rilasciata nel settembre 2016, sfruttava il Protocollo Signal se utilizzata in modalità incognito
- **Duo**: protezione delle videochat
- **Skype**: conversazioni private dal 2018
- **WhatsApp**: tra le maggiori applicazioni che implementano Signal è l'unica che garantisce di default la crittografia end-to-end delle conversazioni (da aprile 2016)

[Gre29], [Lumer]

- Facebook: usa Signal solo nelle Secret Conversations
- Allo: applicazione mobile di messaggistica istantanea di Google, non esiste più dal 12 marzo 2019
- Duo: applicazione per videochiamate e chat mobile di Google
- Whatsapp: introdusse Signal per la prima volta nel 2014 per utenti Android, estendendolo a tutti gli utenti nel 2016
- Google: introduce Signal di default nell'applicazione di messaggi su Android

Applicazione Signal

L'Applicazione e il Protocollo Signal

Tra le più note implementazioni (parziali) del Protocollo Signal troviamo:

- **Facebook**: introdusse la feature *Secret Conversations* per gli utenti di Facebook Messenger nel luglio 2016
- **Allo**: rilasciata nel settembre 2016, sfruttava il Protocollo Signal se utilizzata in modalità incognito
- **Duo**: protezione delle videochat
- **Skype**: conversazioni private dal 2018
- **WhatsApp**: tra le maggiori applicazioni che implementano Signal è l'unica che garantisce di default la crittografia end-to-end delle conversazioni (da aprile 2016)

[Gre29], [Lumer]

- Facebook: usa Signal solo nelle Secret Conversations
- Allo: applicazione mobile di messaggistica istantanea di Google, non esiste più dal 12 marzo 2019
- Duo: applicazione per videochiamate e chat mobile di Google
- Whatsapp: introdusse Signal per la prima volta nel 2014 per utenti Android, estendendolo a tutti gli utenti nel 2016
- Google: introduce Signal di default nell'applicazione di messaggi su Android

Applicazione Signal

L'Applicazione e il Protocollo Signal

Tra le più note implementazioni (parziali) del Protocollo Signal troviamo:

- **Facebook**: introdusse la feature *Secret Conversations* per gli utenti di Facebook Messenger nel luglio 2016
- **Allo**: rilasciata nel settembre 2016, sfruttava il Protocollo Signal se utilizzata in modalità incognito
- **Duo**: protezione delle videochat
- **Skype**: conversazioni private dal 2018
- **WhatsApp**: tra le maggiori applicazioni che implementano Signal è l'unica che garantisce di default la crittografia end-to-end delle conversazioni (da aprile 2016)

[Gre29], [Lumer]

- Facebook: usa Signal solo nelle Secret Conversations
- Allo: applicazione mobile di messaggistica istantanea di Google, non esiste più dal 12 marzo 2019
- Duo: applicazione per videochiamate e chat mobile di Google
- Whatsapp: introdusse Signal per la prima volta nel 2014 per utenti Android, estendendolo a tutti gli utenti nel 2016
- Google: introduce Signal di default nell'applicazione di messaggi su Android

Applicazione Signal

L'Applicazione e il Protocollo Signal

Tra le più note implementazioni (parziali) del Protocollo Signal troviamo:

- **Facebook**: introdusse la feature *Secret Conversations* per gli utenti di Facebook Messenger nel luglio 2016
- **Allo**: rilasciata nel settembre 2016, sfruttava il Protocollo Signal se utilizzata in modalità incognito
- **Duo**: protezione delle videochat
- **Skype**: conversazioni private dal 2018
- **WhatsApp**: tra le maggiori applicazioni che implementano Signal è l'unica che garantisce di default la crittografia end-to-end delle conversazioni (da aprile 2016)

[Gre29], [Lumer]

Applicazione Signal
L'Applicazione e il Protocollo Signal
Tra le più note implementazioni (parziali) del Protocollo Signal troviamo:
- **Facebook**: introdusse la feature *Secret Conversations* per gli utenti di Facebook Messenger nel luglio 2016
- **Allo**: rilasciata nel settembre 2016, sfruttava il Protocollo Signal se utilizzata in modalità incognito
- **Duo**: protezione delle videochat
- **Skype**: conversazioni private dal 2018
- **WhatsApp**: tra le maggiori applicazioni che implementano Signal è l'unica che garantisce di default la crittografia end-to-end delle conversazioni (da aprile 2016)
[Gre29], [Lumer]

- Facebook: usa Signal solo nelle Secret Conversations
- Allo: applicazione mobile di messaggistica istantanea di Google, non esiste più dal 12 marzo 2019
- Duo: applicazione per videochiamate e chat mobile di Google
- Whatsapp: introdusse Signal per la prima volta nel 2014 per utenti Android, estendendolo a tutti gli utenti nel 2016
- Google: introduce Signal di default nell'applicazione di messaggi su Android

Applicazione Signal

L'Applicazione e il Protocollo Signal

Tra le più note implementazioni (parziali) del Protocollo Signal troviamo:

- **Facebook**: introdusse la feature *Secret Conversations* per gli utenti di Facebook Messenger nel luglio 2016
- **Allo**: rilasciata nel settembre 2016, sfruttava il Protocollo Signal se utilizzata in modalità incognito
- **Duo**: protezione delle videochat
- **Skype**: conversazioni private dal 2018
- **WhatsApp**: tra le maggiori applicazioni che implementano Signal è l'unica che garantisce di default la crittografia end-to-end delle conversazioni (da aprile 2016)

[Gre29], [Lumer]

Applicazione Signal
L'Applicazione e il Protocollo Signal
Tra le più note implementazioni (parziali) del Protocollo Signal troviamo:
- **Facebook**: introdusse la feature *Secret Conversations* per gli utenti di Facebook Messenger nel luglio 2016
- **Allo**: rilasciata nel settembre 2016, sfruttava il Protocollo Signal se utilizzata in modalità incognito
- **Duo**: protezione delle videochat
- **Skype**: conversazioni private dal 2018
- **WhatsApp**: tra le maggiori applicazioni che implementano Signal è l'unica che garantisce di default la crittografia end-to-end delle conversazioni (da aprile 2016)

[Gre29], [Lumer]

- Facebook: usa Signal solo nelle Secret Conversations
- Allo: applicazione mobile di messaggistica istantanea di Google, non esiste più dal 12 marzo 2019
- Duo: applicazione per videochiamate e chat mobile di Google
- Whatsapp: introdusse Signal per la prima volta nel 2014 per utenti Android, estendendolo a tutti gli utenti nel 2016
- Google: introduce Signal di default nell'applicazione di messaggi su Android

Applicazione Signal

L'Applicazione e il Protocollo Signal

Ciascuna di queste *features* richiede che le conversazioni intraprese siano dichiarate "private" affinché sia possibile applicare la crittografia end-to-end su tutto il contenuto che viene scambiato

Inoltre, conversazioni già avvenute non possono essere protette applicando il protocollo ex post.
[Mar16]

Applicazione Signal
L'Applicazione e il Protocollo Signal

Ciascuna di queste features richiede che le conversazioni intraprese siano dichiarate "private" affinché sia possibile applicare la crittografia end-to-end su tutto il contenuto che viene scambiato

Inoltre, conversazioni già avvenute non possono essere protette applicando il protocollo ex post.
[Mar16]

La dichiarazione delle conversazioni come "private" avviene in genere per selezione esplicita da parte dell'utente e non di default.

WhatsApp implementa automaticamente la crittografia end-to-end sia per le chat private che per quelle di gruppo, tuttavia se si vuole verificare che le conversazioni siano private è necessario che entrambe le persone che partecipano alla conversazione selezionino la chat di interesse, clicchino sul nome del contatto, selezionino l'opzione "Crittografia" e scannerizzino il codice QR che viene presentato sul dispositivo dell'altro utente oppure confrontino i numeri a 60 cifre presentati.

Applicazione Signal

L'Applicazione e il Protocollo Signal

Ciascuna di queste *features* richiede che le conversazioni intraprese siano dichiarate "private" affinché sia possibile applicare la crittografia end-to-end su tutto il contenuto che viene scambiato

Inoltre, conversazioni già avvenute non possono essere protette applicando il protocollo ex post.
[Mar16]

Applicazione Signal
L'Applicazione e il Protocollo Signal

Ciascuna di queste features richiede che le conversazioni intraprese siano dichiarate "private" affinché sia possibile applicare la crittografia end-to-end su tutto il contenuto che viene scambiato

Inoltre, conversazioni già avvenute non possono essere protette applicando il protocollo ex post.
[Mar16]

La dichiarazione delle conversazioni come "private" avviene in genere per selezione esplicita da parte dell'utente e non di default.

WhatsApp implementa automaticamente la crittografia end-to-end sia per le chat private che per quelle di gruppo, tuttavia se si vuole verificare che le conversazioni siano private è necessario che entrambe le persone che partecipano alla conversazione selezionino la chat di interesse, clicchino sul nome del contatto, selezionino l'opzione "Crittografia" e scannerizzino il codice QR che viene presentato sul dispositivo dell'altro utente oppure confrontino i numeri a 60 cifre presentati.

Applicazione Signal

L'Applicazione e il Protocollo Signal

La sicurezza garantita dall'implementazione del protocollo è relativa al fatto che tutti i prodotti OWS sono incentrati sulla privacy degli utenti, infatti:

- Salvano solo le informazioni strettamente necessarie
- Rendono impossibile a terze parti accedere ai messaggi o ai file scambiati tra gli utenti (grazie alla crittografia end-to-end)

[Lumer]

Applicazione Signal

L'Applicazione e il Protocollo Signal

La sicurezza garantita dall'implementazione del protocollo è relativa al fatto che tutti i prodotti OWS sono incentrati sulla privacy degli utenti, infatti:

- Salvano solo le informazioni strettamente necessarie
- Rendono impossibile a terze parti accedere ai messaggi o ai file scambiati tra gli utenti (grazie alla crittografia end-to-end)

[Lumer]

Applicazione Signal

L'Applicazione e il Protocollo Signal

La sicurezza garantita dall'implementazione del protocollo è relativa al fatto che tutti i prodotti OWS sono incentrati sulla privacy degli utenti, infatti:

- Salvano solo le informazioni strettamente necessarie
- Rendono impossibile a terze parti accedere ai messaggi o ai file scambiati tra gli utenti (grazie alla crittografia end-to-end)

[Lumer]

Applicazione Signal

L'Applicazione e il Protocollo Signal

La sicurezza garantita dall'implementazione del protocollo è relativa al fatto che tutti i prodotti OWS sono incentrati sulla privacy degli utenti, infatti:

- Salvano solo le informazioni strettamente necessarie
- Rendono impossibile a terze parti accedere ai messaggi o ai file scambiati tra gli utenti (grazie alla crittografia end-to-end)

[Lumer]

End-to-End Encryption

La crittografia End-to-End (E2EE) è un processo di comunicazione sicura che impedisce a terze parti di accedere ai dati trasferiti da un utente a un altro.

Solamente gli utenti che sono in possesso della chiave segreta possono decifrare il testo cifrato e leggere il messaggio come *plaintext*.

In linea di massima E2EE garantisce che potenziali *eavesdroppers* non possano accedere alle chiavi necessarie per decifrare la conversazione. [Gre15]

Dati protetti da crittografia sono tali per cui solamente le persone autorizzate possono leggerne il contenuto in chiaro, mentre per tutti gli altri utenti si tratta di dati presentati in un formato non leggibile.

Grazie alla E2EE è possibile proteggere i dati trasmessi da terze parti malintenzionate che possono includere i provider dei servizi di telecomunicazione, gli Internet provider e utenti malevoli.

La E2EE si assicura inoltre che le comunicazioni tra due endpoint siano sicure.

End-to-End Encryption

La crittografia End-to-End (E2EE) è un processo di comunicazione sicura che impedisce a terze parti di accedere ai dati trasferiti da un utente a un altro.

Solamente gli utenti che sono in possesso della chiave segreta possono decifrare il testo cifrato e leggere il messaggio come *plaintext*.

In linea di massima E2EE garantisce che potenziali *eavesdroppers* non possano accedere alle chiavi necessarie per decifrare la conversazione. [Gre15]

End-to-End Encryption

La crittografia End-to-End (E2EE) è un processo di comunicazione sicura che impedisce a terze parti di accedere ai dati trasferiti da un utente a un altro.

Solamente gli utenti che sono in possesso della chiave segreta possono decifrare il testo cifrato e leggere il messaggio come *plaintext*.

In linea di massima E2EE garantisce che potenziali *eavesdroppers* non possano accedere alle chiavi necessarie per decifrare la conversazione. [Gre15]

Dati protetti da crittografia sono tali per cui solamente le persone autorizzate possono leggerne il contenuto in chiaro, mentre per tutti gli altri utenti si tratta di dati presentati in un formato non leggibile.

Grazie alla E2EE è possibile proteggere i dati trasmessi da terze parti malintenzionate che possono includere i provider dei servizi di telecomunicazione, gli Internet provider e utenti malevoli.

La E2EE si assicura inoltre che le comunicazioni tra due endpoint siano sicure.

End-to-End Encryption

La crittografia End-to-End (E2EE) è un processo di comunicazione sicura che impedisce a terze parti di accedere ai dati trasferiti da un utente a un altro.

Solamente gli utenti che sono in possesso della chiave segreta possono decifrare il testo cifrato e leggere il messaggio come *plaintext*.

In linea di massima E2EE garantisce che potenziali *eavesdroppers* non possano accedere alle chiavi necessarie per decifrare la conversazione. [Gre15]

End-to-End Encryption

La crittografia End-to-End (E2EE) è un processo di comunicazione sicura che impedisce a terze parti di accedere ai dati trasferiti da un utente a un altro.

Solamente gli utenti che sono in possesso della chiave segreta possono decifrare il testo cifrato e leggere il messaggio come *plaintext*.

In linea di massima E2EE garantisce che potenziali *eavesdroppers* non possano accedere alle chiavi necessarie per decifrare la conversazione. [Gre15]

Dati protetti da crittografia sono tali per cui solamente le persone autorizzate possono leggerne il contenuto in chiaro, mentre per tutti gli altri utenti si tratta di dati presentati in un formato non leggibile.

Grazie alla E2EE è possibile proteggere i dati trasmessi da terze parti malintenzionate che possono includere i provider dei servizi di telecomunicazione, gli Internet provider e utenti malevoli.

La E2EE si assicura inoltre che le comunicazioni tra due endpoint siano sicure.

End-to-End Encryption

E2EE si basa sulla crittografia *asimmetrica*.
La crittografia **asimmetrica**, o **a chiave pubblica**, cifra e decifra i dati usando due chiavi distinte:

- La chiave pubblica è usata per cifrare un messaggio e inviarlo al proprietario della chiave pubblica
- In seguito, il messaggio può essere decifrato solo utilizzando la corrispondente chiave privata.

Al contrario, la crittografia **simmetrica** utilizza una sola chiave privata per cifrare il *plaintext* e decifrare il *ciphertext*.

I messaggi vengono crittografati dal mittente, pertanto, anche se intercettati da una terza persona, essi non le saranno visibili in *plaintext* e saranno dunque conservabili solo in *ciphertext*.

Al contrario, il destinatario sarà in grado di ricevere i dati e decifrarli per sé.

End-to-End Encryption

E2EE si basa sulla crittografia *asimmetrica*.
La crittografia **asimmetrica**, o **a chiave pubblica**, cifra e decifra i dati usando due chiavi distinte:

- La chiave pubblica è usata per cifrare un messaggio e inviarlo al proprietario della chiave pubblica
- In seguito, il messaggio può essere decifrato solo utilizzando la corrispondente chiave privata.

Al contrario, la crittografia **simmetrica** utilizza una sola chiave privata per cifrare il *plaintext* e decifrare il *ciphertext*.

I messaggi vengono crittografati dal mittente, pertanto, anche se intercettati da una terza persona, essi non le saranno visibili in *plaintext* e saranno dunque conservabili solo in *ciphertext*.

Al contrario, il destinatario sarà in grado di ricevere i dati e decifrarli per sé.

End-to-End Encryption

E2EE si basa sulla crittografia *asimmetrica*.
La crittografia **asimmetrica**, o **a chiave pubblica**, cifra e decifra i dati usando due chiavi distinte:

- La chiave pubblica è usata per cifrare un messaggio e inviarlo al proprietario della chiave pubblica
- In seguito, il messaggio può essere decifrato solo utilizzando la corrispondente chiave privata.

Al contrario, la crittografia **simmetrica** utilizza una sola chiave privata per cifrare il *plaintext* e decifrare il *ciphertext*.

End-to-End Encryption

E2EE si basa sulla crittografia *asimmetrica*.
La crittografia **asimmetrica**, o **a chiave pubblica**, cifra e decifra i dati usando due chiavi distinte:

- La chiave pubblica è usata per cifrare un messaggio e inviarlo al proprietario della chiave pubblica
- In seguito, il messaggio può essere decifrato solo utilizzando la corrispondente chiave privata.

Al contrario, la crittografia **simmetrica** utilizza una sola chiave privata per cifrare il *plaintext* e decifrare il *ciphertext*.

I messaggi vengono crittografati dal mittente, pertanto, anche se intercettati da una terza persona, essi non le saranno visibili in *plaintext* e saranno dunque conservabili solo in *ciphertext*.

Al contrario, il destinatario sarà in grado di ricevere i dati e decifrarli per sé.

End-to-End Encryption

E2EE si basa sulla crittografia *asimmetrica*.
La crittografia **asimmetrica**, o **a chiave pubblica**, cifra e decifra i dati usando due chiavi distinte:

- La chiave pubblica è usata per cifrare un messaggio e inviarlo al proprietario della chiave pubblica
- In seguito, il messaggio può essere decifrato solo utilizzando la corrispondente chiave privata.

Al contrario, la crittografia **simmetrica** utilizza una sola chiave privata per cifrare il *plaintext* e decifrare il *ciphertext*.

End-to-End Encryption

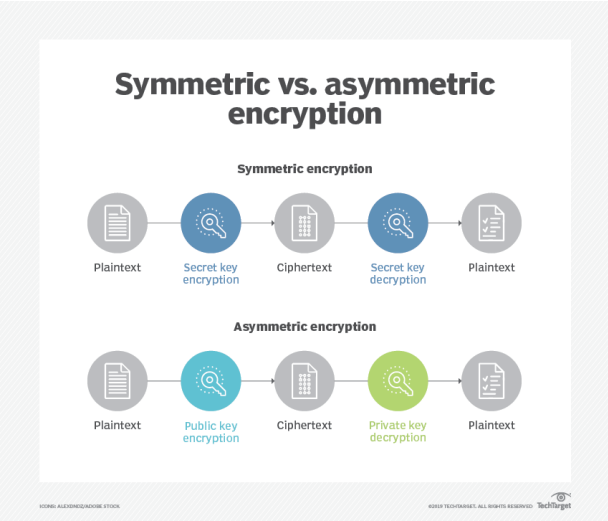
E2EE si basa sulla crittografia *asimmetrica*.
La crittografia **asimmetrica**, o **a chiave pubblica**, cifra e decifra i dati usando due chiavi distinte:
- La chiave pubblica è usata per cifrare un messaggio e inviarlo al proprietario della chiave pubblica
- In seguito, il messaggio può essere decifrato solo utilizzando la corrispondente chiave privata.

Al contrario, la crittografia **simmetrica** utilizza una sola chiave privata per cifrare il *plaintext* e decifrare il *ciphertext*.

I messaggi vengono crittografati dal mittente, pertanto, anche se intercettati da una terza persona, essi non le saranno visibili in *plaintext* e saranno dunque conservabili solo in *ciphertext*.

Al contrario, il destinatario sarà in grado di ricevere i dati e decifrarli per sé.

End-to-End Encryption



End-to-End Encryption

End-to-End Encryption

Applicazioni

- **Comunicazioni sicure**: applicazioni di messaggistica e posta elettronica per mantenere private le conversazioni degli utenti;
- **Gestione password**: in questo caso a entrambi gli endpoint della comunicazione si trova lo stesso utente, che è l'unica persona munita di chiave;
- **Data storage**: nei servizi di storage in cloud può anche essere garantita E2EE *in transit*, proteggendo i dati degli utenti anche dall'accesso da parte dei fornitori del servizio in cloud;

[IBM]

End-to-End Encryption
Applicazioni
- **Comunicazioni sicure**: applicazioni di messaggistica e posta elettronica per mantenere private le conversazioni degli utenti;
- **Gestione password**: in questo caso a entrambi gli endpoint della comunicazione si trova lo stesso utente, che è l'unica persona munita di chiave;
- **Data storage**: nei servizi di storage in cloud può anche essere garantita E2EE *in transit*, proteggendo i dati degli utenti anche dall'accesso da parte dei fornitori del servizio in cloud;

[IBM]

Alcuni sistemi, come ad esempio Lavabit e Hushmail, hanno in passato dichiarato di implementare la crittografia end-to-end nonostante ciò non fosse vero. [Gra 7]

Lavabit, servizio email in passato ritenuto sicuro e oggi non più attivo, nel 2014 consegnò al governo americano le chiavi che utilizzava per proteggere i dati dei propri utenti in occasione delle indagini sul caso Snowden. La controversia nacque dal fatto che la compagnia aveva in precedenza dichiarato che il proprio livello di sicurezza era tale che nemmeno gli amministratori della compagnia stessa avevano accesso al contenuto delle mail scambiate dai propri utenti. [Pou16], [GM13]

Hushmail, altro email provider dichiarato sicuro, violò la privacy dei propri utenti utilizzandone le password per decrittare le email e consegnarle al governo federale in *plaintext*. [Sin 7]

Altri sistemi, come per esempio Telegram, non implementano la crittografia end-to-end di default e sono pertanto stati criticati.

In modo particolare Telegram non la implementa né per le chat di gruppo né per i client desktop. Tra le altre critiche mosse all'applicazione c'è quella di utilizzare il protocollo di crittografia non standard MTProto. [EPP 1]

End-to-End Encryption

Applicazioni

- **Comunicazioni sicure**: applicazioni di messaggistica e posta elettronica per mantenere private le conversazioni degli utenti;
- **Gestione password**: in questo caso a entrambi gli endpoint della comunicazione si trova lo stesso utente, che è l'unica persona munita di chiave;
- **Data storage**: nei servizi di storage in cloud può anche essere garantita E2EE *in transit*, proteggendo i dati degli utenti anche dall'accesso da parte dei fornitori del servizio in cloud;

[IBM]

End-to-End Encryption
Applicazioni
- **Comunicazioni sicure**: applicazioni di messaggistica e posta elettronica per mantenere private le conversazioni degli utenti;
- **Gestione password**: in questo caso a entrambi gli endpoint della comunicazione si trova lo stesso utente, che è l'unica persona munita di chiave;
- **Data storage**: nei servizi di storage in cloud può anche essere garantita E2EE *in transit*, proteggendo i dati degli utenti anche dall'accesso da parte dei fornitori del servizio in cloud;

[IBM]

Alcuni sistemi, come ad esempio Lavabit e Hushmail, hanno in passato dichiarato di implementare la crittografia end-to-end nonostante ciò non fosse vero. [Gra 7]

Lavabit, servizio email in passato ritenuto sicuro e oggi non più attivo, nel 2014 consegnò al governo americano le chiavi che utilizzava per proteggere i dati dei propri utenti in occasione delle indagini sul caso Snowden. La controversia nacque dal fatto che la compagnia aveva in precedenza dichiarato che il proprio livello di sicurezza era tale che nemmeno gli amministratori della compagnia stessa avevano accesso al contenuto delle mail scambiate dai propri utenti. [Pou16], [GM13]

Hushmail, altro email provider dichiarato sicuro, violò la privacy dei propri utenti utilizzandone le password per decrittare le email e consegnarle al governo federale in *plaintext*. [Sin 7]

Altri sistemi, come per esempio Telegram, non implementano la crittografia end-to-end di default e sono pertanto stati criticati.

In modo particolare Telegram non la implementa né per le chat di gruppo né per i client desktop. Tra le altre critiche mosse all'applicazione c'è quella di utilizzare il protocollo di crittografia non standard MTProto. [EPP 1]

End-to-End Encryption

Applicazioni

- **Comunicazioni sicure**: applicazioni di messaggistica e posta elettronica per mantenere private le conversazioni degli utenti;
- **Gestione password**: in questo caso a entrambi gli endpoint della comunicazione si trova lo stesso utente, che è l'unica persona munita di chiave;
- **Data storage**: nei servizi di storage in cloud può anche essere garantita E2EE *in transit*, proteggendo i dati degli utenti anche dall'accesso da parte dei fornitori del servizio in cloud;

[IBM]

End-to-End Encryption
Applicazioni

- **Comunicazioni sicure**: applicazioni di messaggistica e posta elettronica per mantenere private le conversazioni degli utenti;
- **Gestione password**: in questo caso a entrambi gli endpoint della comunicazione si trova lo stesso utente, che è l'unica persona munita di chiave;
- **Data storage**: nei servizi di storage in cloud può anche essere garantita E2EE *in transit*, proteggendo i dati degli utenti anche dall'accesso da parte dei fornitori del servizio in cloud;

[IBM]

Alcuni sistemi, come ad esempio Lavabit e Hushmail, hanno in passato dichiarato di implementare la crittografia end-to-end nonostante ciò non fosse vero. [Gra 7]

Lavabit, servizio email in passato ritenuto sicuro e oggi non più attivo, nel 2014 consegnò al governo americano le chiavi che utilizzava per proteggere i dati dei propri utenti in occasione delle indagini sul caso Snowden. La controversia nacque dal fatto che la compagnia aveva in precedenza dichiarato che il proprio livello di sicurezza era tale che nemmeno gli amministratori della compagnia stessa avevano accesso al contenuto delle mail scambiate dai propri utenti. [Pou16], [GM13]

Hushmail, altro email provider dichiarato sicuro, violò la privacy dei propri utenti utilizzandone le password per decrittare le email e consegnarle al governo federale in *plaintext*. [Sin 7]

Altri sistemi, come per esempio Telegram, non implementano la crittografia end-to-end di default e sono pertanto stati criticati.

In modo particolare Telegram non la implementa né per le chat di gruppo né per i client desktop. Tra le altre critiche mosse all'applicazione c'è quella di utilizzare il protocollo di crittografia non standard MTProto. [EPP 1]

End-to-End Encryption

Problematiche

La E2EE non garantisce di per sé né la sicurezza né la privacy, in quanto i dati trasmessi potrebbero essere protetti in modo poco sicuro sui dispositivi endpoint. Tuttavia, l'implementazione della E2EE consente di applicare una protezione dei dati migliore della sola crittografia **"in transit"**.

Per molti sistemi di messaggistica i messaggi passano attraverso un intermediario che li conserva finché non vengono recuperati dal destinatario. Anche se protetti da crittografia, essi lo sono solamente in transito, quindi possono essere letti dai provider di servizi.
[int20], [IBM]

In questo modo è possibile monitorare il contenuto dei messaggi (per esempio in cerca di contenuti offensivi o pericolosi) ma si corre anche il rischio che utenti non autorizzati e/o malintenzionati aventi accesso allo storage dei messaggi possano fare un uso improprio dei contenuti.

Nella crittografia "in transit" è possibile o salvare direttamente i messaggi decrittati oppure salvare i dati crittografati e la chiave con cui decrittarli sullo stesso database.

End-to-End Encryption

Problematiche

La E2EE non garantisce di per sé né la sicurezza né la privacy, in quanto i dati trasmessi potrebbero essere protetti in modo poco sicuro sui dispositivi endpoint. Tuttavia, l'implementazione della E2EE consente di applicare una protezione dei dati migliore della sola crittografia **"in transit"**.

Per molti sistemi di messaggistica i messaggi passano attraverso un intermediario che li conserva finché non vengono recuperati dal destinatario. Anche se protetti da crittografia, essi lo sono solamente in transito, quindi possono essere letti dai provider di servizi.
[int20], [IBM]

End-to-End Encryption
Problematiche
La E2EE non garantisce di per sé né la sicurezza né la privacy, in quanto i dati trasmessi potrebbero essere protetti in modo poco sicuro sui dispositivi endpoint. Tuttavia, l'implementazione della E2EE consente di applicare una protezione dei dati migliore della sola crittografia "in transit".
Per molti sistemi di messaggistica i messaggi passano attraverso un intermediario che li conserva finché non vengono recuperati dal destinatario. Anche se protetti da crittografia, essi lo sono solamente in transito, quindi possono essere letti dai provider di servizi.
[int20], [IBM]

In questo modo è possibile monitorare il contenuto dei messaggi (per esempio in cerca di contenuti offensivi o pericolosi) ma si corre anche il rischio che utenti non autorizzati e/o malintenzionati aventi accesso allo storage dei messaggi possano fare un uso improprio dei contenuti.

Nella crittografia "in transit" è possibile o salvare direttamente i messaggi decrittati oppure salvare i dati crittografati e la chiave con cui decrittarli sullo stesso database.

End-to-End Encryption

Problematiche

Ulteriori problematiche:

- **Endpoint security**: gli endpoint sono vulnerabili se non protetti adeguatamente
- **Attacchi di tipo Man-in-the-Middle**: la conversazione può essere soggetta a *eavesdropping*
- **Backdoors**: metodi per bypassare l'autenticazione standard o la protezione crittografica di un dispositivo. Se non volute, possono essere introdotte tramite attacchi cyber e poi sfruttate per violare la sicurezza del sistema

[Gre15], [IBM]

- Endpoint security: E2EE protegge i dati solo tra i due endpoint; ciò significa che i due endpoint possono essere soggetti ad attacchi;
- Attacchi MITM: anziché forzare la crittografia dei dati, ci si può aspettare un tentativo da parte di terzi malintenzionati di impersonare il destinatario durante. Essi possono, per esempio, impersonare il destinatario durante lo scambio di chiavi con il mittente, decifrare il messaggio inviato e poi inoltrarlo al vero destinatario senza farsi notare. Una soluzione per questo tipo di attacchi è introdurre un metodo di autenticazione (per es. certification authorities, web of trust, fingerprint numeriche o come QR code)
- Backdoors: nonostante le *backdoors* non siano sempre implementate volutamente, esse possono essere introdotte grazie a *cyber-attacks* e poi essere utilizzate per la negoziazione delle chiavi o per oltrepassare la protezione crittografica.

End-to-End Encryption

Problematiche

Ulteriori problematiche:

- **Endpoint security**: gli endpoint sono vulnerabili se non protetti adeguatamente
- **Attacchi di tipo Man-in-the-Middle**: la conversazione può essere soggetta a *eavesdropping*
- **Backdoors**: metodi per bypassare l'autenticazione standard o la protezione crittografica di un dispositivo. Se non volute, possono essere introdotte tramite attacchi cyber e poi sfruttate per violare la sicurezza del sistema

[Gre15], [IBM]

- Endpoint security: E2EE protegge i dati solo tra i due endpoint; ciò significa che i due endpoint possono essere soggetti ad attacchi;
- Attacchi MITM: anziché forzare la crittografia dei dati, ci si può aspettare un tentativo da parte di terzi malintenzionati di impersonare il destinatario durante. Essi possono, per esempio, impersonare il destinatario durante lo scambio di chiavi con il mittente, decifrare il messaggio inviato e poi inoltrarlo al vero destinatario senza farsi notare. Una soluzione per questo tipo di attacchi è introdurre un metodo di autenticazione (per es. certification authorities, web of trust, fingerprint numeriche o come QR code)
- Backdoors: nonostante le *backdoors* non siano sempre implementate volutamente, esse possono essere introdotte grazie a *cyber-attacks* e poi essere utilizzate per la negoziazione delle chiavi o per oltrepassare la protezione crittografica.

End-to-End Encryption

Problematiche

Ulteriori problematiche:

- **Endpoint security**: gli endpoint sono vulnerabili se non protetti adeguatamente
- **Attacchi di tipo Man-in-the-Middle**: la conversazione può essere soggetta a *eavesdropping*
- **Backdoors**: metodi per bypassare l'autenticazione standard o la protezione crittografica di un dispositivo. Se non volute, possono essere introdotte tramite attacchi cyber e poi sfruttate per violare la sicurezza del sistema

[Gre15], [IBM]

End-to-End Encryption
Problematiche
Ulteriori problematiche:
- **Endpoint security**: gli endpoint sono vulnerabili se non protetti adeguatamente
- **Attacchi di tipo Man-in-the-Middle**: la conversazione può essere soggetta a eavesdropping
- **Backdoors**: metodi per bypassare l'autenticazione standard o la protezione crittografica di un dispositivo. Se non volute, possono essere introdotte tramite attacchi cyber e poi sfruttate per violare la sicurezza del sistema

[Gre15], [IBM]

- Endpoint security: E2EE protegge i dati solo tra i due endpoint; ciò significa che i due endpoint possono essere soggetti ad attacchi;
- Attacchi MITM: anziché forzare la crittografia dei dati, ci si può aspettare un tentativo da parte di terzi malintenzionati di impersonare il destinatario durante. Essi possono, per esempio, impersonare il destinatario durante lo scambio di chiavi con il mittente, decifrare il messaggio inviato e poi inoltrarlo al vero destinatario senza farsi notare. Una soluzione per questo tipo di attacchi è introdurre un metodo di autenticazione (per es. certification authorities, web of trust, fingerprint numeriche o come QR code)
- Backdoors: nonostante le *backdoors* non siano sempre implementate volutamente, esse possono essere introdotte grazie a *cyber-attacks* e poi essere utilizzate per la negoziazione delle chiavi o per oltrepassare la protezione crittografica.

End-to-End Encryption

Problematiche

- **Complessità nel definire gli endpoint**: alcune implementazioni consentono di decodificare e ricodificare i dati lungo il percorso, quindi è necessario definire accuratamente gli estremi della trasmissione
- **Privacy "eccessiva"**: enti governativi non hanno modo di verificare la natura dei contenuti trasmessi dagli utenti, pertanto non sono in grado di prendere misure adeguate in caso di illeciti
- **Metadati visibili**
- Non vi è certezza che E2EE possa funzionare altrettanto bene con l'eventuale introduzione di *quantum computer* che rendano la crittografia obsoleta

[LB21]

End-to-End Encryption
Problematiche
- Complessità nel definire gli endpoint: alcune implementazioni consentono di decodificare e ricodificare i dati lungo il percorso, quindi è necessario definire accuratamente gli estremi della trasmissione
- Privacy "eccessiva": enti governativi non hanno modo di verificare la natura dei contenuti trasmessi dagli utenti, pertanto non sono in grado di prendere misure adeguate in caso di illeciti
- Metadati visibili
- Non vi è certezza che E2EE possa funzionare altrettanto bene con l'eventuale introduzione di quantum computer che rendano la crittografia obsoleta

[LB21]

End-to-End Encryption

Problematiche

- **Complessità nel definire gli endpoint**: alcune implementazioni consentono di decodificare e ricodificare i dati lungo il percorso, quindi è necessario definire accuratamente gli estremi della trasmissione
- **Privacy "eccessiva"**: enti governativi non hanno modo di verificare la natura dei contenuti trasmessi dagli utenti, pertanto non sono in grado di prendere misure adeguate in caso di illeciti
- **Metadati visibili**
- Non vi è certezza che E2EE possa funzionare altrettanto bene con l'eventuale introduzione di *quantum computer* che rendano la crittografia obsoleta

[LB21]

End-to-End Encryption
Problematiche
- **Complessità nel definire gli endpoint**: alcune implementazioni consentono di decodificare e ricodificare i dati lungo il percorso, quindi è necessario definire accuratamente gli estremi della trasmissione
- **Privacy "eccessiva"**: enti governativi non hanno modo di verificare la natura dei contenuti trasmessi dagli utenti, pertanto non sono in grado di prendere misure adeguate in caso di illeciti
- Metadati visibili
- Non vi è certezza che E2EE possa funzionare altrettanto bene con l'eventuale introduzione di quantum computer che rendano la crittografia obsoleta
[LB21]

End-to-End Encryption

Problematiche

- **Complessità nel definire gli endpoint**: alcune implementazioni consentono di decodificare e ricodificare i dati lungo il percorso, quindi è necessario definire accuratamente gli estremi della trasmissione
- **Privacy "eccessiva"**: enti governativi non hanno modo di verificare la natura dei contenuti trasmessi dagli utenti, pertanto non sono in grado di prendere misure adeguate in caso di illeciti
- **Metadati visibili**
- Non vi è certezza che E2EE possa funzionare altrettanto bene con l'eventuale introduzione di *quantum computer* che rendano la crittografia obsoleta

[LB21]

End-to-End Encryption
Problematiche
- **Complessità nel definire gli endpoint**: alcune implementazioni consentono di decodificare e ricodificare i dati lungo il percorso, quindi è necessario definire accuratamente gli estremi della trasmissione
- **Privacy "eccessiva"**: enti governativi non hanno modo di verificare la natura dei contenuti trasmessi dagli utenti, pertanto non sono in grado di prendere misure adeguate in caso di illeciti
- **Metadati visibili**
- Non vi è certezza che E2EE possa funzionare altrettanto bene con l'eventuale introduzione di quantum computer che rendano la crittografia obsoleta
[LB21]

End-to-End Encryption

Problematiche

- **Complessità nel definire gli endpoint**: alcune implementazioni consentono di decodificare e ricodificare i dati lungo il percorso, quindi è necessario definire accuratamente gli estremi della trasmissione
- **Privacy "eccessiva"**: enti governativi non hanno modo di verificare la natura dei contenuti trasmessi dagli utenti, pertanto non sono in grado di prendere misure adeguate in caso di illeciti
- **Metadati visibili**
- Non vi è certezza che E2EE possa funzionare altrettanto bene con l'eventuale introduzione di *quantum computer* che rendano la crittografia obsoleta

[LB21]

End-to-End Encryption
Problematiche
- **Complessità nel definire gli endpoint**: alcune implementazioni consentono di decodificare e ricodificare i dati lungo il percorso, quindi è necessario definire accuratamente gli estremi della trasmissione
- **Privacy "eccessiva"**: enti governativi non hanno modo di verificare la natura dei contenuti trasmessi dagli utenti, pertanto non sono in grado di prendere misure adeguate in caso di illeciti
- **Metadati visibili**
- Non vi è certezza che E2EE possa funzionare altrettanto bene con l'eventuale introduzione di *quantum computer* che rendano la crittografia obsoleta

[LB21]

Signal Protocol

Il protocollo Signal fornisce crittografia end-to-end a sistemi di messaggistica istantanea e di chiamate vocali, combinando l'algoritmo **"Double Ratchet"**, pre-chiavi e un triplo handshake Elliptic-curve Diffie–Hellman (3-DH).

Signal Protocol

Il protocollo Signal fornisce crittografia end-to-end a sistemi di messaggistica istantanea e di chiamate vocali, combinando l'algoritmo **"Double Ratchet"**, pre-chiavi e un triplo handshake Elliptic-curve Diffie–Hellman (3-DH).

Signal Protocol

Le specifiche di riferimento sono infatti: [sig]

- **XEdDSA e VXEdDSA**: algoritmi per la creazione e verifica di *signatures* compatibili con EdDSA utilizzando formati di chiavi pubbliche e private inizialmente definiti per le funzioni X25519 e X448 di Diffie-Hellman su curve ellittiche. L'algoritmo VXEdDSA estende XEdDSA rendendolo verificabile.
- **Double Ratchet**: algoritmo utilizzato da due parti per lo scambio di messaggi basato su una chiave segreta condivisa.
- **X3DH**: protocollo di negoziazione delle chiavi Extended Triple Diffie-Hellman.
- **Sesame**: gestisce le sessioni crittografate in ambiente asincrono e multi-device.

Signal Protocol

Le specifiche di riferimento sono infatti: [sig]
- **XEdDSA e VXEdDSA**: algoritmi per la creazione e verifica di signatures compatibili con EdDSA utilizzando formati di chiavi pubbliche e private inizialmente definiti per le funzioni X25519 e X448 di Diffie-Hellman su curve ellittiche. L'algoritmo VXEdDSA estende XEdDSA rendendolo verificabile.
- Double Ratchet: algoritmo utilizzato da due parti per lo scambio di messaggi basato su una chiave segreta condivisa.
- X3DH: protocollo di negoziazione delle chiavi Extended Triple Diffie-Hellman.
- Sesame: gestisce le sessioni crittografate in ambiente asincrono e multi-device.

- Double Ratchet: le due parti derivano nuove chiavi per ogni messaggio in modo tale che chiavi usate in precedenza non possano essere ricavate dalle chiavi successive.
- X3DH: stabilisce una chiave segreta condivisa da due parti che si autenticano a vicenda basandosi su chiavi pubbliche. X3DH fornisce *forward secrecy* e *cryptographic deniability*

Forward secrecy: un sistema di crittografia possiede la proprietà di forward secrecy se l'analisi in *plaintext* dei dati scambiati durante la fase di negoziazione delle chiavi durante l'inizializzazione della sessione di comunicazione non rivela la chiave utilizzata per cifrare il resto della sessione.
Si ottiene generando nuove chiavi di sessione per ogni messaggio e assicura che i messaggi scambiati in passato non siano decifrabili ma che al più il messaggio corrente possa essere compromesso.

Cryptographic deniability: l'esistenza di un file cifrato o di un messaggio è rinnegabile, nel senso che un altro utente non può dimostrare che i dati in *plaintext* esistono. Gli utenti possono negare che dei dati siano cifrati o anche negare di essere in grado di decifrarli, indipendentemente dal fatto che ciò sia vero o meno.

Signal Protocol

Le specifiche di riferimento sono infatti: [sig]

- **XEdDSA e VXEdDSA**: algoritmi per la creazione e verifica di *signatures* compatibili con EdDSA utilizzando formati di chiavi pubbliche e private inizialmente definiti per le funzioni X25519 e X448 di Diffie-Hellman su curve ellittiche. L'algoritmo VXEdDSA estende XEdDSA rendendolo verificabile.
- **Double Ratchet**: algoritmo utilizzato da due parti per lo scambio di messaggi basato su una chiave segreta condivisa.
- **X3DH**: protocollo di negoziazione delle chiavi Extended Triple Diffie-Hellman.
- **Sesame**: gestisce le sessioni crittografate in ambiente asincrono e multi-device.

Signal Protocol

Le specifiche di riferimento sono infatti: [sig]
- **XEdDSA e VXEdDSA**: algoritmi per la creazione e verifica di signatures compatibili con EdDSA utilizzando formati di chiavi pubbliche e private inizialmente definiti per le funzioni X25519 e X448 di Diffie-Hellman su curve ellittiche. L'algoritmo VXEdDSA estende XEdDSA rendendolo verificabile.
- **Double Ratchet**: algoritmo utilizzato da due parti per lo scambio di messaggi basato su una chiave segreta condivisa.
- X3DH: protocollo di negoziazione delle chiavi Extended Triple Diffie-Hellman.
- Sesame: gestisce le sessioni crittografate in ambiente asincrono e multi-device.

- Double Ratchet: le due parti derivano nuove chiavi per ogni messaggio in modo tale che chiavi usate in precedenza non possano essere ricavate dalle chiavi successive.
- X3DH: stabilisce una chiave segreta condivisa da due parti che si autenticano a vicenda basandosi su chiavi pubbliche. X3DH fornisce *forward secrecy* e *cryptographic deniability*

Forward secrecy: un sistema di crittografia possiede la proprietà di forward secrecy se l'analisi in *plaintext* dei dati scambiati durante la fase di negoziazione delle chiavi durante l'inizializzazione della sessione di comunicazione non rivela la chiave utilizzata per cifrare il resto della sessione.
Si ottiene generando nuove chiavi di sessione per ogni messaggio e assicura che i messaggi scambiati in passato non siano decifrabili ma che al più il messaggio corrente possa essere compromesso.

Cryptographic deniability: l'esistenza di un file cifrato o di un messaggio è rinnegabile, nel senso che un altro utente non può dimostrare che i dati in *plaintext* esistono. Gli utenti possono negare che dei dati siano cifrati o anche negare di essere in grado di decifrarli, indipendentemente dal fatto che ciò sia vero o meno.

Signal Protocol

Le specifiche di riferimento sono infatti: [sig]

- **XEdDSA e VXEdDSA**: algoritmi per la creazione e verifica di *signatures* compatibili con EdDSA utilizzando formati di chiavi pubbliche e private inizialmente definiti per le funzioni X25519 e X448 di Diffie-Hellman su curve ellittiche. L'algoritmo VXEdDSA estende XEdDSA rendendolo verificabile.
- **Double Ratchet**: algoritmo utilizzato da due parti per lo scambio di messaggi basato su una chiave segreta condivisa.
- **X3DH**: protocollo di negoziazione delle chiavi Extended Triple Diffie-Hellman.
- **Sesame**: gestisce le sessioni crittografate in ambiente asincrono e multi-device.

Signal Protocol

Le specifiche di riferimento sono infatti: [sig]
- **XEdDSA e VXEdDSA**: algoritmi per la creazione e verifica di signatures compatibili con EdDSA utilizzando formati di chiavi pubbliche e private inizialmente definiti per le funzioni X25519 e X448 di Diffie-Hellman su curve ellittiche. L'algoritmo VXEdDSA estende XEdDSA rendendolo verificabile.
- **Double Ratchet**: algoritmo utilizzato da due parti per lo scambio di messaggi basato su una chiave segreta condivisa.
- **X3DH**: protocollo di negoziazione delle chiavi Extended Triple Diffie-Hellman.
- **Sesame**: gestisce le sessioni crittografate in ambiente asincrono e multi-device.

- Double Ratchet: le due parti derivano nuove chiavi per ogni messaggio in modo tale che chiavi usate in precedenza non possano essere ricavate dalle chiavi successive.
- X3DH: stabilisce una chiave segreta condivisa da due parti che si autenticano a vicenda basandosi su chiavi pubbliche. X3DH fornisce *forward secrecy* e *cryptographic deniability*

Forward secrecy: un sistema di crittografia possiede la proprietà di forward secrecy se l'analisi in *plaintext* dei dati scambiati durante la fase di negoziazione delle chiavi durante l'inizializzazione della sessione di comunicazione non rivela la chiave utilizzata per cifrare il resto della sessione.
Si ottiene generando nuove chiavi di sessione per ogni messaggio e assicura che i messaggi scambiati in passato non siano decifrabili ma che al più il messaggio corrente possa essere compromesso.

Cryptographic deniability: l'esistenza di un file cifrato o di un messaggio è rinnegabile, nel senso che un altro utente non può dimostrare che i dati in *plaintext* esistono. Gli utenti possono negare che dei dati siano cifrati o anche negare di essere in grado di decifrarli, indipendentemente dal fatto che ciò sia vero o meno.

Signal Protocol

Le specifiche di riferimento sono infatti: [sig]

- **XEdDSA e VXEdDSA**: algoritmi per la creazione e verifica di *signatures* compatibili con EdDSA utilizzando formati di chiavi pubbliche e private inizialmente definiti per le funzioni X25519 e X448 di Diffie-Hellman su curve ellittiche. L'algoritmo VXEdDSA estende XEdDSA rendendolo verificabile.
- **Double Ratchet**: algoritmo utilizzato da due parti per lo scambio di messaggi basato su una chiave segreta condivisa.
- **X3DH**: protocollo di negoziazione delle chiavi Extended Triple Diffie-Hellman.
- **Sesame**: gestisce le sessioni crittografate in ambiente asincrono e multi-device.

Signal Protocol

Le specifiche di riferimento sono infatti: [sig]
- **XEdDSA e VXEdDSA**: algoritmi per la creazione e verifica di signatures compatibili con EdDSA utilizzando formati di chiavi pubbliche e private inizialmente definiti per le funzioni X25519 e X448 di Diffie-Hellman su curve ellittiche. L'algoritmo VXEdDSA estende XEdDSA rendendolo verificabile.
- **Double Ratchet**: algoritmo utilizzato da due parti per lo scambio di messaggi basato su una chiave segreta condivisa.
- **X3DH**: protocollo di negoziazione delle chiavi Extended Triple Diffie-Hellman.
- **Sesame**: gestisce le sessioni crittografate in ambiente asincrono e multi-device.

- Double Ratchet: le due parti derivano nuove chiavi per ogni messaggio in modo tale che chiavi usate in precedenza non possano essere ricavate dalle chiavi successive.
- X3DH: stabilisce una chiave segreta condivisa da due parti che si autenticano a vicenda basandosi su chiavi pubbliche. X3DH fornisce *forward secrecy* e *cryptographic deniability*

Forward secrecy: un sistema di crittografia possiede la proprietà di forward secrecy se l'analisi in *plaintext* dei dati scambiati durante la fase di negoziazione delle chiavi durante l'inizializzazione della sessione di comunicazione non rivela la chiave utilizzata per cifrare il resto della sessione.
Si ottiene generando nuove chiavi di sessione per ogni messaggio e assicura che i messaggi scambiati in passato non siano decifrabili ma che al più il messaggio corrente possa essere compromesso.

Cryptographic deniability: l'esistenza di un file cifrato o di un messaggio è rinnegabile, nel senso che un altro utente non può dimostrare che i dati in *plaintext* esistono. Gli utenti possono negare che dei dati siano cifrati o anche negare di essere in grado di decifrarli, indipendentemente dal fatto che ciò sia vero o meno.

Signal Protocol
XEdDSA e VXEdDSA

Signal Protocol
XEdDSA e VXEdDSA

Signal Protocol
Double Ratchet

Signal Protocol
Double Ratchet

Signal Protocol
X3DH

Signal Protocol

Sesame

Signal Protocol

Sesame

Bibliografia I

CPJ Middle East, North Africa Program, and CPJ Technology Program.
Why telegram's security flaws may put iran's journalists at risk - committee to protect journalists.
2016, May 1.

Barton Gellman and Jerry Markon.
Edward snowden says motive behind leaks was to expose 'surveillance state'.
The Washington Post, June 10, 2013.

Yael Grauer.
Mr. robot uses protonmail, but it still isn't fully secure.
2015, October 7.

Bibliografia I

CPJ Middle East, North Africa Program, and CPJ Technology Program.
Why telegram's security flaws may put iran's journalists at risk - committee to protect journalists.
2016, May 1.

Barton Gellman and Jerry Markon.
Edward snowden says motive behind leaks was to expose 'surveillance state'.
The Washington Post, June 10, 2013.

Yael Grauer.
Mr. robot uses protonmail, but it still isn't fully secure.
2015, October 7.

Bibliografia II

Andy Greenberg.
Hacker lexicon: What is end-to-end encryption?
2014, November 15.

Andy Greenberg.
Hacker lexicon: What is the signal encryption protocol?
2020, November 29.

Cos'è la e2ee (end-to-end encryption)?

Cryptography concepts - fundamentals - e3kit — virgil security.
2020.

Ben Lutkevich and Madelyn Bacon.
end-to-end encryption (e2ee).
June 2021.

Bibliografia II

Andy Greenberg.
Hacker lexicon: What is end-to-end encryption?
2014, November 15.
Andy Greenberg.
Hacker lexicon: What is the signal encryption protocol?
2020, November 29.
Cos'è la e2ee (end-to-end encryption)?
Cryptography concepts - fundamentals - e3kit — virgil security.
2020.
Ben Lutkevich and Madelyn Bacon.
end-to-end encryption (e2ee).
June 2021.

Bibliografia III

David J Lumb.
The story of signal.
Increment, (7), 2018, October.

Moxie Marlinspike.
Whatsapp's signal protocol integration is now complete.
Apr. 5, 2016.

K Poulsen.
Snowden's email provider loses appeal over encryption keys.
2014, April 16.

Signal documentation.

Bibliografia III

David J Lumb.
The story of signal.
Increment, (7), 2018, October.

Moxie Marlinspike.
Whatsapp's signal protocol integration is now complete.
Apr. 5, 2016.

K Poulsen.
Snowden's email provider loses appeal over encryption keys.
2014, April 16.

Signal documentation.

Bibliografia IV


Ryan Singel.
Encrypted e-mail company hushmail spills to feds.
2007, November 7.

Bibliografia IV

Ryan Singel.
Encrypted e-mail company hushmail spills to feds.
2007, November 7.