點對點加密通訊協定之前瞻技術與探討---WhatsApp 的加密協定 探討

專案研究員 林漢洲

由於手機和可攜式裝置的普及，現在大眾隨時隨地都可輕易連上網，接踵而來，由手機和可攜式裝置分享和發布使用者所產生的內容, 呈現爆炸性成長。目前大眾對不法監聽敏感個人私密通訊有所警覺，也開始重視個資安全，但是國內欠缺如何有效地進行安全通訊的整體規畫與辦法。我們也需要一個可靠，可稽核(auditable)的機制來保護我國國內的通訊和重要機密。因為目前常用的網路平台，如 Facebook, Google, LINE, Apple, 其雲端伺服器都在國外，我們建議解決對策是將我們每天日常線上的對話，如同視為是公務性質，關注其資料安全與是否在地化。試論國內政府公務機關的即時通使用，最常被使用的應是 LINE, 是韓國的 Naver 集團旗下的 LINE 株式會社的即時通訊軟體。雖然，從 LINE 5.3.3 版開始提供點對點加密(end-to-end encryption, E2EE), 在 LINE 叫 letter sealing, 而且自動會以點以點加密傳輸。但是，我們發現這並沒有完全解決保護個人資料傳輸安全的問題。即使對話資料是以加密（連 LINE 也不能解密）方式存放在 LINE 的伺服器，金鑰是只有收發兩端才有，且不會轉送出這兩端以外。前提是我們必須「信任」LINE 在任何時候都會遵守，而且，由於是國外公司，須時時留意是否因股權所有權轉移，而改變其公司主要資安的政策或執行。（例如，因為擴大版圖到金融，如 LINE PAY, 網路銀行等）

相關文獻探討與比較

以下介紹市面上點對點加密的通訊軟體有不同的作法與特色。

Threema

是由一家瑞士公司 Threema GmbH 所開發，強調其伺服器地點都位於瑞士境內，因適用於瑞士的法律和瑞士在資料保護實際運作上，(資料中心有 ISO/IEC 27001 認證)會比其它相關競爭對手有優勢。[2,3]

Signal

有別於 Threema 點對點加密的方法是專有軟體(proprietary), Signal 採用開放源碼，並且公開所有伺服器和手機端的程式在 GITHUB (https://github.com/signalapp)。Signal 的通訊協定 (Signal Protocol)也是公開的，並且被 WhatsApp, Facebook Messenger, Skype 和 Google Allo 採用。除了 WhatsApp, 其它三個即時通並沒有內定自動使用 Signal Protocol 的點對點加密，這是使用者在決定通訊軟體的安全度需要仔細考量[4,5]。

Telegram

由俄羅斯人 Pavel Durov 和 Nikolai Durov 創立，公司位於倫敦。Telegram 的用戶端的程式是開放源碼，可是伺服器端是專有軟體。2018 年四月 Telegram 在俄羅斯被禁用，原因是俄羅斯聯邦安全局以反恐為由要 Telegram 繳出私鑰，以解密使用者的加密訊息。[6,7]

LINE

LINE 從 2016 年七月開始用點對點加密為所有 LINE 使用者傳送訊息。LINE 也公布它加密和金鑰交換和 1 對多群組訊

息加密的演算法。它使用橢圓曲線迪菲-赫爾曼金鑰交換(ECDH curve25519)。[8,9]

WhatsApp

WhatsApp 在 2014 年二月被臉書以美金 119 億收購。2016 年宣布 WhatsApp 使用點對點加密在任何一種通訊形式(文字/語音/影片等），且使用者可檢驗彼此的金鑰，以防止中間人攻擊(MITM, Man-in-the-middle attack)[11]，這並不是 WhatsApp 獨有的功能，如 Telegram 等也提供對應功能讓使用者能再驗證。因為和臉書的 Facebook Messenger 未來可能會整合, [10]目前臉書所有的即時通，WhatsApp 和 Facebook Messenger, 已經是全世界市佔的第一，二名。由此可知，臉書對即時通的隱私及作法，對全世界的用戶有巨大的影響和後續結果。

iMessage

iMessage 是蘋果公司開發，只能用在蘋果公司產品(iPhone, iPad, Mac), 它使用的通訊協定是 Apple Push Notification Service (APNs), 也提供點對點加密服務[12]。因為使用 iMessage 是整合在 iPhone(大部份的使用平台），它並沒有公布全球使用者數量，估計 iMessage 在全球使用者排名約在前五名。

Skype

Skype 在 2011 年被微軟公司以美金 85 億收購。Skype 使用專有軟體「基於 IP 的語音傳輸」（VoIP), 叫 Skype protocol。微軟收購後，新的叫 Microsoft Notification Protocol 24. 從 2018 年 8 月開始，Skype 在所有平台都有提供點對點加密[14]。

WhatsApp 的加密協定 探討

由於 WhatsApp 是目前全球前三大通訊軟體之一，加上它是臉書公司所有，它的加密協定是否安全是值得我們研究的議題。
WhatsApp 從 2016 年 3 月開始全部使用點對點加密協定，加密範圍包括一對一聊天，群組聊天，文，圖，影，音和附加檔。WhatsApp 的加密協定是利用 Signal Protocol, 這是由 Open Whisper Systems 所開發設計。所謂的點對點加密協定，它防止第三者（包括在通訊途中竊聽者，或 WhatsApp 的員工，或在 WhatsApp 用的雲端或其它相關伺服器）能讀取使用者的明文的文字或解密過語音訊息。

尤其是在當使用者的手機或終端設備（平板或電腦）裏的內容被破解竊取，其中的加密金鑰被偷，都不能用來解密過去已加密的訊息。這樣才能把訊息的安全性，進一步的保護其隱密性。

接下來，介紹本章使用的術語定義：

公鑰類型
Identity Key Pair -- 在安裝時產生的長期 Curve25519 金鑰組（橢圓曲線)

Signed Pre Key -- 在安裝時產生的中期 Curve25519 金鑰組, 用 Identify Key 簽章過, 在特定的周期下，定期交換循環使用
One-Time Pre Keys -- 在安裝時產生, 只一次性使用的 Curve25519 金鑰組，有很多組，用完時會再產生新的。

Session Key 類型
Root Key -- 用來產生 Chain Keys 有 32 位元組的值
Chain Key -- 用來產生 Message Keys 有 32 位元組的值
Message key -- 用來加密訊息內容，有 80 位元組的值。其中 32 位元用在 AES-256 金鑰，32 位元用在 HMAC-SHA256 金鑰, 16 位元用在 IV。

用戶端註冊

註冊時，從用戶端 WhatsApp 程式傳送這三個公開的金鑰至伺服器。

Identify Key,
Signed Pre Key (和其簽章）
One-Time Pre Keys

WhatsApp 將使用者的識別碼和這三公鑰儲存在伺服器。無論何時 WhatApp 伺服器都不會有使用者的任何私鑰，也無法讀取。

初始 Session 設定

Session 在這裏是指兩台電腦在通訊協定從開始連線到連線結束這段過程，因為中文沒有適切的詞，有人翻成"會話"，或"交談"，為了更清楚表達這個概念，這裡直接用英文。

在和另一個 WhatsApp 用戶建立交談，一開始要先啟始一個加密的 session。除非重新安裝 WhatsApp 或更換新機，都不用再另外建立新的 session。

下面是建立新 session 的步驟：

1. 想要開始建立交談的用戶(送件方）向伺服器請求對方（收件方）的 Identify Key, Signed Pre Key, 和一個 One-Time Pre Key。
2. 伺服器回傳這三個公鑰給用戶。One-Time Pre Key 因只被用一次，所以一回傳，這個 One-Time Pre Key 即被刪除。如果收件方的 One-Time Pre Key 都被用完，且沒有再補充，將不回傳 One-Time Pre Key。
3. 送件方將收到的 Identify Key 存成 I (recipient), Signed Pre Key 存成 S (recipient), One-Time Pre Key 存成 O（recipient)
4. 送件方產生一組暫時的 Curve25519 金鑰組，E (initiator)
5. 送件方將自己的 Identity Key 取出，為 I(initiator)
6. 送件方算出 master secret
   master_secret = ECDH(I initiator, S recipient) || ECDH (E initiator, I recipient) || ECDH(E initiator, S recipient) || ECDH(E initiator, O recipient). 如果沒有 One-Time Pre Key, 最後一個 ECDH 就可省略。

   筆者註，加上這個 One-Time Pre Key 的 ECDH, 可以防止惡意第三方用"重覆攻擊"，可提高此通訊協定的安全。因為 master_secret 會每次變成更不同，更不容易被攻擊。
7. 送件者用 HKDF (產生衍生金鑰）從 master_secret 來產用 Root Key, Chain Keys

接受端 Session 設定

當建立了可長久執行的加密 session, 送信方可以馬上開始傳送訊息給收信方，即使對方是離線狀態。送信方送出的資料包括 E initiator 和 I initiator , 這些資料是儲放在送出訊息的 header(表頭）。
當收信方收到這個訊息後，要執行下列步驟:

1. 收信方計算出 master_secret, 用自己的私鑰和儲放在訊息表頭的公鑰
2. 收信方刪除送信方用的 One-Time Pre Key
3. 送信方從 master_secret 用 HKDF 產生相對應的 Root Key，Chain Keys

傳送訊息

當建立了 session 之後，用戶可用 Message Key (AES256 加密法, CBC 模式）來加密訊息，並用 HMAC-SHA256 (雜湊函數是 SHA256 的訊息鑑別碼)來驗證其完整性（訊息內容在傳遞中沒有被更改）。為了防止 Message Key 加密過後，會被重覆用來作惡意攻擊（從 session state 重新再送一次，不需要知道 Message Key 的內容)。Message Key 需要是非常態，我們需要在傳完每個訊息後更換它。

Message Key 產生的方法是從發信方的 Chain Key 衍生出來的 (“ratchets” forward, 像齒輪轉動一格）。每次訊息發和送一次，用 ECDH 協定，產生新的 Chain Key。上述的作法，正式的術語為”hash ratchet” 和 round trip “DH ratchet” [15], 可以保證 forward secrecy, 也

就是私鑰被竊取只會讓未來的對話被偷聽，之前傳過的訊息都還是保持安全隱密。

從 Chain Key 計算出 Message Key

當發信方需要一個新的 Message Key，其運算方法如下：

1. Message Key = HMAC-SHA256(Chain Key, 0x01)
2. 我們也要跟著更改 Chain Key。
   Chain Key = HMAC-SHA256(Chain Key, 0x02)

如此一來，Chain Key 就會每次都會變動(利用和 0x02, 就是數值 2，作雜湊函數的運算）。而且，如果 Message Key 被私下保存下來，也不能回推算出現在或以前用過的 Chain Key, 以確保本通訊協定的安全。

從 Root Key 計算出 Chain Key

每次傳送出一個新的訊息，Curve25519 的公鑰會附在上面。當收到回應時，新的 Chain Key, Root Key 會以下面的公式算出：

1. Ephemeral_secret = ECDH(Ephemeral sender, Ephemeral recipient)
2. Chain Key, Root Key = HKDF(Root Key, ephemeral_secret)

根據上述的計算，每個用戶送出訊息時只會用一個特定的 Chain。也因此，由於 Message Keys 和 Chain Keys 的算出方法，即使在多個訊息送出時不按次序到達收件方，或甚至延遲或遺失都不會影響收信方解密。

傳送多媒體或其它附檔

點對點加密也會用在傳送較大的附加檔, 如影、音、圖或檔案等。

1. WhatsApp 發送方先產生一個 32 位元長的 AES256 公鑰（用完即刪）和 32 位元長的 HMAC_SHA265 金鑰（用完即刪）
2. 發送方用 AES256 金鑰（CBC 模式）和 (隨機亂數產生的）IV 加密附件，把密文用 HMAC-SHA256 產生的 MAC (訊息鑑別碼), 附在密文之後。
3. 發送方上傳此密文到 blob store。(兩位元大型物件儲存處）
4. 發送方傳送下列給收信方。加密金鑰，HMAC 金鑰，加密附件的 SHA256, 和一個指標指到 blob store 中此 blob 的位置。
5. 收信方解密訊息，取得加密過的 blob 從 blob store，驗證 SHA256 值（確定取得正確的加密過的 blob)，再驗證 MAC(確定此 blob 的完整性), 然後再解密取得訊息明文。

群組訊息

一般未加密的通訊軟體通常用"伺服器分送"(server-side fan-out)來實現群組聊天。使用者欲傳送訊息給群組，先送給其中一個使用者，伺服器再把它送 N 次給 N 個在群組裏的成員。

另一種方式是"用戶端分送"（client-side fan-out)。是由一個使用者獨自送 N 次到 N 個其它群組內的成員。

WhatsApp 內的群組訊息傳送的方式是用上述的一對一的加密方式來達到有效率的"伺服器分送"。這是採用 Signal Messaging Protocol 裏的"Sender Keys”的方式。

當一個 WhatsApp 的群組成員欲傳送訊息到群組：

1. 發送方先產生一個 32 位元的隨機亂數 Chain Key.
2. 發送方產生一個隨機亂數 Curve25519 Signature Key 的金鑰組(公鑰和私鑰）
3. 發送方將 32 位元的 Chain Key 和 Signature Key 的公鑰 Sender Key 的訊息.
4. 發送方一個一個把 Sender Key 送給群組裏的其它成員, 用前述的一對一通訊的方法。

後續的訊息用下列的步驟：

1. 發送方從 Chain Key 衍生而產生 Message Key 這是，並更新 Chain Key
2. 發送方用 AES256(CBC 模式）加密訊息
3. 發送方用 Signature Key 簽章加密訊息
4. 發送方傳送此密文到伺服器，然後伺服器再分送到群組其它成員。

Chain Key 的“Hash ratchet”確保了前向安全性(forward secrecy)。每當有成員離開群組，其它群組成員須刪除其 Sender Key, 從頭再產生(如上）。

結論

點對點加密用在 WhatsApp, 使得用戶間傳送的訊息只有真正的訊息發送方和收信方能讀取。不管是第三者想從中竊聽或是 WhatsApp 都無法讀取。而且，保護的訊息包括文字訊息、群組、圖、影音和附加檔。

WhatsApp 的伺服器並不能讀取用戶的私鑰。而且任何私鑰都不會存放在任何 WhatsApp 的伺服器裏。另外，提供額外的認證方式可供用戶手動驗證金鑰的正確性，以防止 Man-in-the-middle （惡意方假冒）攻擊。

之前我們探討 LINE 的加密協定，雖然是和 WhatsApp 有部分雷同，我們可以得知在 ratchet forward 的方式，WhatsApp 成功地讓每則訊息都用不同的金鑰加密，不但是 forward secrecy, 而且讓金鑰的衍出不佔記憶體，且不容易被破解（即使 Message Keys 都被儲存，試圖利用暴力破解）。也就是 open source 的 Whisper Systems 為基礎，讓更多資安專家和程式師能藉由對 open source 的找臭蟲，並以集思廣益的方式合作，產生對加密軟體的信任。

引用資料

1. WhatsApp Encryption Overview (Technical White Paper) https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf
2. Threema - Seriously secure messaging https://threema.ch/en
3. https://en.wikipedia.org/wiki/Threema
4. https://signal.org
5. https://github .com/whispersystems/libsignal-protocol-java/
6. https://en.wikipedia.org/wiki/Telegram_(software)
7. https://telegram.org
8. https://en.wikipedia.org/wiki/Line_(software)
9. https://line.me/en/
10. https://www.nytimes.com/2019/01/25/technology/facebook-instagram-whatsapp-messenger.html)
11. https://en.wikipedia.org/wiki/WhatsApp
12. https://en.wikipedia.org/wiki/IMessage
13. https://www.skype.com/en/
14. https://en.wikipedia.org/wiki/Skype
15. Double Ratchet Algorithm-Wikipedia https://en.wikipedia.org/wiki/Double_Ratchet_Algorithm