How Secure is End-to-End Encrypted Messaging? The problem with the trust establishment infrastructure

Diana A. Vasile and Alastair R. Beresford

June 14, 2018

Extended Abstract

Mobile apps today often use TLS to secure communication between a device and one or more servers in the cloud. Unfortunately, this approach does not guarantee data security on the servers since the TLS connection is terminated by the server the data is not encrypted on the servers themselves. Therefore data stored on servers might be read by a rogue employee, a hacker breaking into a server, or government-mandated interception.

For some apps, data must be processed on the server, and therefore either data needs to be decrypted by the server, or some form of homomorphic encryption scheme is required. However, for many apps, data ultimately flows from one mobile device to another, and the server does not need to process the data itself. Instead, the server is used to address limitations found in mobile and WiFi networks, which prevent direct connections between mobile devices, or to provide a data store-and-forward service when one device is not online due to network coverage issues, or to reduce energy consumption in the receiving device.

In order to reduce the trust placed in servers, messaging apps, such as as Facebook's WhatsApp and Apple's iMessage, have moved to an end-to-end encrypted model where data is encrypted on the sending device, and only decrypted on arrival at the receiving device.

End-to-end encryption is typically implemented with public-key cryptography: each mobile device generates a public-private key pair, and the public key together with a human-readable name representing the user, such as a phone number or email address, is shared with the app provider. The private key never leaves the device. The app provider then maintains Public-Key Infrastructure (PKI) in the cloud, which offers a service to map human-readable names to public keys. When Alice wishes to send a message to Bob, she first asks the PKI for the public keys of Bob's devices and then encrypts her message using the keys provided by the PKI. Therefore, provided Alice receives Bob's keys (and only his keys) from the PKI, end-to-end encryption prevents the rogue employee, hacker, or government agent from decrypting data stored on the server.

In this design, significant trust is placed in the PKI. A compromise in the PKI allows the attacker to add to, or modify, the public keys associated with a particular identifier. For example, an attacker may associate an additional mobile device, such as a tablet, with Bob's account. Then, when Alice encrypts her next message and sends it to Bob, her message will be readable by the tablet. Since all data sent between end-to-end encrypted apps is typically sent via the cloud, the tablet will then receive a readable copy of all future messages. Furthermore, Alice nor Bob are typically unaware that this has happened.

This presentation will discuss the current attempts at solving this problem, where the fallacies are, as well as consider an alternative approach: peer-to-peer sharing of key material over local WiFi networks through the gossiping of key-name graphs.