# Gallagher Command Centre

## Mobile End to End Encryption

Technical Information Paper

**Disclaimer**

This document gives certain information about products and/or services provided by Gallagher Group Limited or its related companies (referred to as "Gallagher Group").

The information is indicative only and is subject to change without notice meaning it may be out of date at any given time. Although every commercially reasonable effort has been taken to ensure the quality and accuracy of the information, Gallagher Group makes no representation as to its accuracy or completeness and it should not be relied on as such. To the extent permitted by law, all express or implied, or other representations or warranties in relation to the information are expressly excluded.

Neither Gallagher Group nor any of its directors, employees or other representatives shall be responsible for any loss that you may incur, either directly or indirectly, arising from any use or decisions based on the information provided.

Except where stated otherwise, the information is subject to copyright owned by Gallagher Group and you may not sell it without permission. Gallagher Group is the owner of all trademarks reproduced in this information. All trademarks which are not the property of Gallagher Group, are acknowledged.

**Important:** If you received this document along with your Command Centre installation media, or via another similar channel then it may be out of date with respect to the functionality/behaviour of the cloud, and of the Mobile Connect Apps, which are distributed through platform App Stores and may be more recent than your Command Centre installation.

It is recommended you refer to the latest revision of this document, which can be found here: https://gallaghersecurity.github.io/r/mobileconnect-end-to-end-encryption

## Contents

# 1 Introduction

## 1.1 Pre-requisites

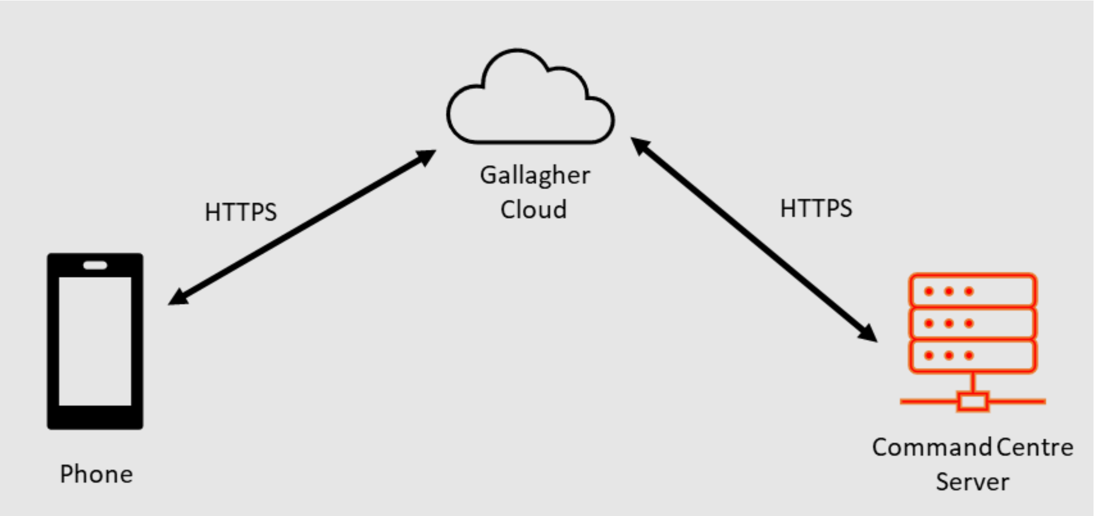
A pre-requisite to this document is the Mobile Connect Cloud and App Security technical information paper. If you are not familiar with that document, you should read it before proceeding.

The document is located here:
https://gallaghersecurity.github.io/r/mobileconnect-cloud-and-app-security

Alternatively, you received this document along with your Command Centre installation media, The Cloud and App Security TIP document should also have accompanied it.

## 1.2 Background

The communication architecture of Gallagher's Mobile Connect app and SDK is such that all data exchanged between Mobile Devices and a customer's on-premises Command Centre server is relayed through the cloud. This is necessary because either the server or mobile devices may be intermittently offline, making it very hard to establish a direct connection between the two. Furthermore, the security risk of allowing direct network traffic from mobile apps through to a Command Centre server is likely unpalatable to many customer IT departments.



As detailed in the Cloud and App Security TIP, Command Centre v8.40 and Mobile Connect for iOS and Android v15 adds support for Digital ID cards. Digital ID cards may contain personal information such as Photos, Names, identifying numbers (such as a Student ID) and more. Keeping this information safe and secure is critical.

Without end to end encryption, this would mean that:

a) Gallagher's Cloud Services would always have access to information associated with a Digital ID card as it passes through.

b) If the cloud services were ever to be compromised or suffer a data breach, attackers would gain access to that information.

Gallagher considers the security of our cloud services as paramount, and we take many steps (such as secure coding guidelines, external third-party security review and penetration testing, etc) to ensure this, however it is naïve in today's modern computing environment to claim that a cloud service could never be compromised, or that a data breach is impossible.

The only way to guarantee that our customers are protected from attacks like these is to have the edge devices (Mobile apps and Command Centre server) encrypt data in a way that cannot be decrypted by our cloud services. This is End to End encryption.

**Note:** In addition to securing Digital ID cards, end to end encryption will also be used to protect SALTO key data as it transits through the cloud, once a server is upgraded to v8.40 or later of Command Centre.

# 2 End to End Encryption Overview

There are a variety of protocols/schemes for implementing an end to end encryption system. Fundamentally they all depend on the core concept of **Asymmetric Encryption**.

**Disclaimer: This section is intended to introduce the concepts employed for End to End encryption and may not be strictly correct at all levels of detail. Consider this a high-level overview rather than a technical reference.**

With asymmetric encryption, the key used to encrypt data consists of two parts - A "public" key and a "private" key. Together these form a "key pair". The underlying mathematics of cryptography are such that data encrypted by one side can be decrypted by the other, and vice versa.

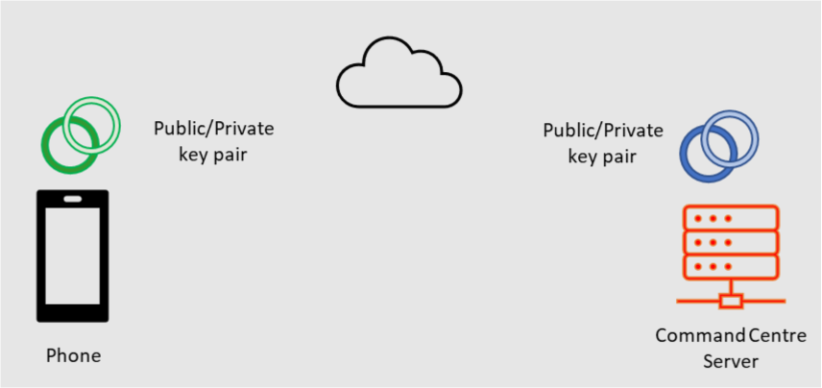
If a party wants to communicate with another using asymmetric encryption, it will generate a key pair, and send the public key to the remote device. It will keep the private key which will never be transmitted over a network. The remote device can use the public key to encrypt a message. Only the private key can successfully decrypt this message, so the remote device knows that it is safe to send across the network.
For two-way communications, both ends can generate a key pair, and exchange public keys.

The public key is called the "public" key because it is safe for anyone to have it. As such, we can safely transmit it over the network, via the cloud, or any other mechanism. The foundation of the security lies in the private key, which is never transmitted.

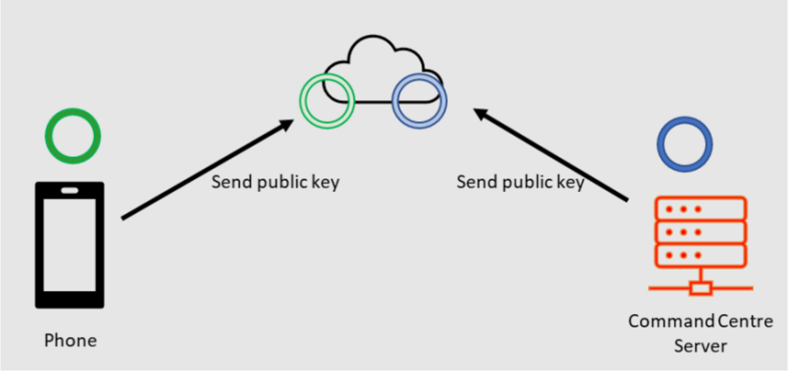
## 2.1 Key and Message Distribution

The Command Centre server and Mobile Connect apps exchange public keys using the Gallagher Cloud. The Gallagher cloud keeps a copy of the last-known public key for a server or phone; as above the public key is not sensitive and is safe to keep.

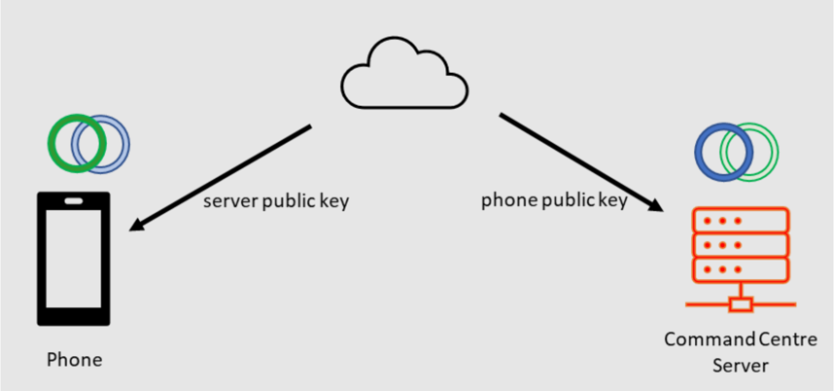
**Step 1: Both ends generate a key pair.**
In the following series of diagrams, light colored circles represent public keys, and dark colored circles represent private keys. A paired set of keys is indicated by sharing a color.



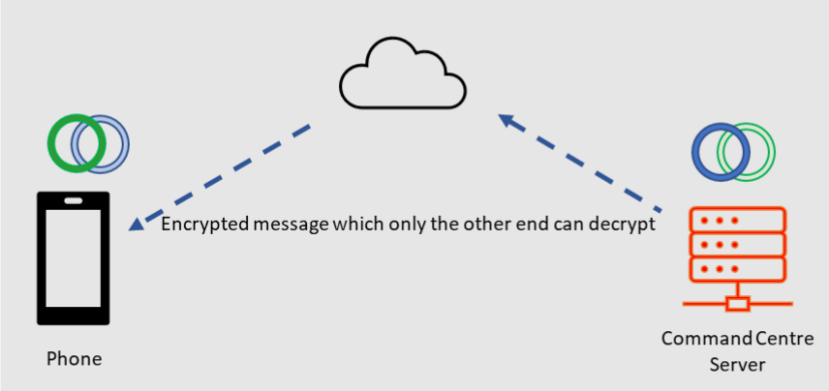
**Step 2: Both ends send their public key to the Cloud**



**Step 3: Cloud sends public keys to the other end**



**Step 4: The server can now encrypt messages that can only be decrypted by the private key of the target device, and vice versa**



# 3 Technical Implementation Details

The above sequence diagram represents an abstract view of what is happening. In practice the situation is much more complex; keys are used to derive other keys, message authentication codes are employed, and so forth.

Note: The following details refer to version 1 of the Gallagher mobile end to end encryption system. If flaws or weaknesses are ever discovered in this system, we will update it and publish a revised version.

This document is current as of January 2021.

## 3.1 Protocol

Gallagher implements the Elliptic Curve Integrated Encryption Scheme (ECIES) for end to end encryption.

ECIES is an industry standard and is most notably used by Apple for their implementation of end to end encryption for their iMessage service. We modelled our encryption off of Apple's and have attempted to provide at least as secure as iMessage, or more.

More information can be found online, such as at the Wikipedia Integrated Encryption Scheme page: https://en.wikipedia.org/wiki/Integrated_Encryption_Scheme

Or this general overview article:

https://www.nominet.uk/how-elliptic-curve-cryptography-encryption-works/

## 3.2 ECIES implementation details

Elliptic-curve P256 (secp256r1) is used for public and private keys

AES-128 is used for per-message derived keys

HMAC-SHA256 is used for message integrity

## 3.3 Key rotation and revocation

As private keys never leave the device, and ECIES generates unique keys per each message that is sent, we do not rotate the underlying elliptic curve keypair on a periodic basis

Our network protocol between the phone, cloud and server does allow for future key rotation, and if a key was deemed to be compromised, we could issue an update to either Command Centre or the Mobile Apps to rotate keys if necessary.

In the event of a phone being compromised, this would not be necessary; The Mobile Connect app ties E2E encryption keys to the end user's Mobile Credential, which can simply be revoked from the Command Centre server.

## 3.4 Key storage

On iOS devices, the end to end encryption private key is generated and stored in the Secure Enclave. This is a separate processor with its own isolated secure hardware key storage. More information is available from Apple: https://support.apple.com/en-nz/guide/security/sec59b0b31ff/web

On Android devices, the Android keystore is the most secure form of storage available and uses hardware-backed key storage on all devices which support it.

However, while the Android keystore can store Elliptic Curve P256 keys, those keys can only be used for signature verification, rather than encryption. As such, we store the Elliptic Curve private key on the device filesystem inside the app's sandbox, and this key is encrypted by a second key which is stored in the keystore itself.

For Android 5 devices, the second keystore key is RSA2048, and for Android 6 and newer, it is AES256.

More information is available from Google:
https://developer.android.com/training/articles/keystore