Twilio + Virgil IP Messaging FAQ

This document answers Frequently Asked Questions regarding integration of Twilio IP Messaging with Virgil's end-to-end encryption and verification.

What does Virgil provide to enable end-to-end encryption, authentication, and verification of data?

Virgil consists of an open-source encryption library, which implements Cryptographic Message Syntax (CMS) and Elliptic Curve Integrated Encryption Scheme (ECIES) (including RSA schema), a Key Management API, and a cloud-based Key Management Service (Virgil Keys). The Virgil Keys Service consists of a public key service and a private key escrow service.

See our Technical Specifications for the up-to-date list of programming languages and platforms supported by our library. Generally all modern platforms and programming languages are supported.

If I send a message to a channel that contains many recipients does it mean that the message will be encrypted as many times as there are recipients in the channel? How does group chat work?

No. Only the AES encryption key used to secure the payload (default AES-256) will be re-encrypted for each individual recipient. The payload itself will not be re-encrypted.

Who controls private keys?

Developers have full control over how private keys are generated, stored, and synchronized on end-client devices. Virgil provides a Private Key Escrow Service that can help backup and synchronize private keys.

Most users are given 3 options:

Easy: In this mode the Virgil Private Key API is used to store private keys associated with the user/app combo. Virgil stores key in encrypted (not hashed) format; however, we also maintain a hashed "password" that is used to decrypt the private Key Bundle. This mode is the least secure and requires end-users to trust Virgil. It also allows Virgil the ability to reset a user's Key Bundle password.

Normal: End users are given an option to store an encrypted private Key Bundle for backup and device synchronization purposes. Virgil cannot reset this password and cannot recover the private key bundle should the user forget the string used to encrypt the bundle.

Enterprise: In this mode the developer runs their own Private Key Escrow instance or end-users manage their private keys manually. There is nothing stored by Virgil except the corresponding public key for each private key.

How many public/private key pairs can each user have?

At this time there is no limit. Depending on the application you can and sometimes should generate a new public/private key pair as often as "per session".

Does Virgil use standard encryption?

Yes. Additional details can be found in our Technical Specification.

How can I add history to my Twilio IP Messaging channel and maintain end-to-end encryption?

Virgil provides sample application services for Twilio IP Messaging that, at the developer's discretion, run a history service for each channel where this behavior is appropriate. This service effectively re-encrypts the history for each user who is authorized to see this information.

Do I have to pick a specific configuration? For example, if I pick NSA Suite B compliance, will this break compatibility with other users?

No. Virgil uses Cryptographic Agility, which allows different users, platforms, and even individual files or chat sessions to have individually selected cryptographic parameters. Our library automatically detects which parameters were used and uses appropriate settings to decrypt content. Developers have full control over this functionality; however, we recommend sticking with the defaults selected by Virgil as we continuously evaluate and update best practices. Current defaults can be found in our Technical Specifications.

How is the user recovered after their private key is lost from a device?

1. Use case: Web browser with no ability to store private keys between sessions.
- The user would generate a new private key "per authenticated session" and re-register public key with Virgil Public Key Service. Old public and private keys would still be maintained using the Virgil Private Key storage API. This is the recommended, default behavior for the IP Messaging use case.
- The user would retrieve the previously issued private key using the Virgil Private Key API. The key may be optionally encrypted using a passphrase or a passcode. The passcode can be delivered using two-factor authentication or be a part of a multi-factor authentication process.

2. Use case: iOS or Android based mobile device.
- The Virgil Private Key API allows developers to easily implement private key synchronization across multiple devices and/or provide a simple recovery mechanism for keys. This service can be run "offline" or behind company firewall.

How are the user keys revoked?

1. Developers can use the API method for key revocation provided by Virgil. This method signs the given public key with an application specific "revoked" key. Developers can validate revocation key signatures by signing them with their own key. Revocation requests must be signed by the developer or by the end user.

2. Developers can use a key of their own choosing and define it as "revocation" key. By signing any public record with this key developers can be certain that the key has been revoked.

How are keys synced between multiple devices?

Private keys can use the Virgil Private Keys Service to securely synchronize their application keys across multiple user devices (it works similar to the iCloud Keychain service).

Public Keys and the associated signatures are always available using the Virgil Public Keys Service API.