**Data in Transit:**

1. Channel Encryption
   * + 1. Definition

Channel encryption is a process of encrypting transmitting data in client/server communications. This security layer lies between application and transport layer, most common TLS and HTTPS.

***TLS (server-side)***

Transport Layer Security (TLS) is an advanced version of Secure Socket Layer (SSL). It’s a cryptographic protocol, which is designed to provide secure communications over a computer network.

***mTLS (server/client mutual auth)***

Mutual Transport Layer Security (mTLS) is a process that establishes an encrypted TLS connection in which both parties use X.509 digital certificates to authenticate each other. MTLS can help mitigate the risk of moving services to the cloud and can help prevent malicious third parties from imitating genuine apps.

1. Experian Policy and TSB – link to EGSO

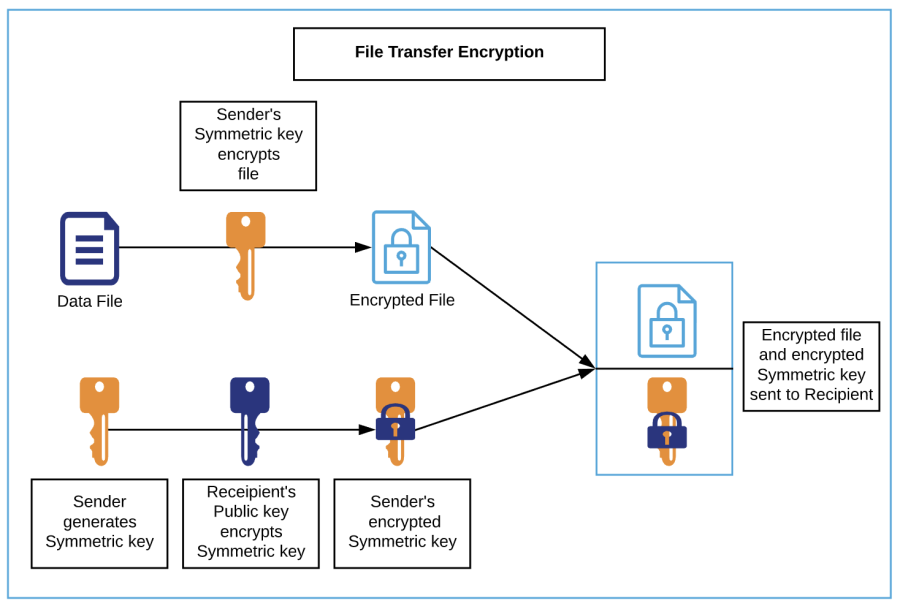
* [Experian policy and TSB](https://experian.sharepoint.com/sites/GlobalSecurityOffice/SitePages/Home.aspx)

1. Acceptable Cipher suites and algorithms

For data in transit, ascend shall leverage hybrid encryption (a combination of asymmetric key and symmetric key).  
The keys that shall be used for this purpose are

* PGP-based (for legacy support) OR
* Combination of asymmetric key and symmetric key

**Hybrid Encryption (combination of Asymmetric and Symmetric key)**



1. **Encryption flow**: For data being sent as files from Ascend to Client (external 3rd party or internal systems at Experian) OR Client to Ascend
   1. A secure channel, e.g., SFTP, is used for data transfer
   2. Recipient generates Asymmetric key pair, secures private key and sends their public key to the sender's authorized representative
   3. Sender generates a Symmetric key
   4. Sender encrypts file(s) with Symmetric key
   5. Sender encrypts Symmetric key with Recipient's public key
   6. The file(s) and the encrypted Symmetric key are sent to the Recipient to the pre-authorized location.
2. **Decryption flow**: For data received as files by Ascend from Client (external 3rd party or internal systems at Experian) OR Client from Ascend
   1. A secure channel, e.g., SFTP, is used for data transfer
   2. Recipient decrypts the encrypted Symmetric key with their private key to extract the Symmetric key
   3. Recipient decrypts file(s) with the Symmetric key.

**Payload Encryption (combination of Asymmetric and Symmetric key)**

This scenario shall apply when APIs are used for data exchange. The payload to be encrypted could be a record made up of multiple fields or select sensitive fields within the payload.

1. CIS
   1. CIS Security practice

This document defines the Ascend encryption and associated key management policies and practices. The purpose of this document is to ensure that data is secured while in transit and at rest. Ascend policies on encryption and key management ensures that key lifecycle and the levels of encryption are deployed in alignment conformance with Experian security principles and policies and adhere to regulatory and industry conformance standards. The policies and procedures stated in this document apply to internal and external handling of data within Ascend products and services. This also applies within Experian and 3rd parties including clients and partners who provide data to and access data within the Ascend Technology Platform. The procedures are designed to be prescriptive yet simple and clear to deploy in order to gain consistency for the Ascend use cases and workflows. In this way, they also offer transparency on how data security has been designed into our Ascend architecture. Additionally, as our Ascend platform is increasingly a cloud-based service, this document addresses data security aspects necessitated by the use of public clouds (AWS mainly, but also Azure, GCP and OCI). Finally, this document now embodies the Experian CIS 3-I Cloud Cybersec Framework *(create link to page)*, where the NIST cybersecurity components are incorporated in considering the security functions of Govern, Identify, Protect, Detect, Respond and Recover. It therefore supports the objective of continuously increasing the robustness of Ascend’s data protection capabilities.

* 1. What needs to be done?
  2. How to do it?
  3. Whom to contact?

1. Content Encryption
   1. Definition
   2. Experian Policy and TSB – link to EGSO
   3. Acceptable Cipher suites and algorithms
   4. CIS
      1. File-level encryption
         1. CIS Security practice – PGP
         2. ***What needs to be done?***

**PGP Encryption**

1. For data being sent as files from Ascend to Client (external 3rd party or internal systems at Experian)
   1. A secure channel, e.g. SFTP, is used for data transfer
   2. Client sends their PGP public key to the Ascend authorized representative
   3. The file(s) are encrypted with the shared Client-specific PGP public key
   4. The file(s) are either pushed to a pre-authorized Client location or pulled by the Client from an authorized Ascend location (S3 bucket) over the secure channel.
2. For data being sent as files from Client to Ascend
   1. A secure channel, e.g. SFTP, is used for data transfer
   2. Ascend generates PGP key pair and stores PGP private key in AWS Secrets Manager
   3. Ascend sends their PGP public key to the client authorized representative
   4. Client encrypts file(s) with Ascend PGP public key
   5. Client sends file(s) to the pre-authorized Ascend location in AWS.
   6. On receipt of encrypted file(s) at the authorized Ascend location, the file can be decrypted with the Ascend PGP private key (held in AWS Secrets Manager).
      * 1. How to do it? Sample code
        2. Whom to contact?

[Tallapaka Sai](mailto:Sai.Tallapaka@experian.com?subject=Ascend%20Encryption%20Process), Director, DevSecOps

* + 1. API (Payload encryption)
       1. CIS Security practice – JWE/JSE (JOSE Framework)

This addresses data exchanges through application and system APIs. The payload to be encrypted could be a record made up of multiple fields or select sensitive fields within the payload.

* + - * 1. Who generates the PGP key pair?
        2. Where is stored?
        3. How is access provided?
        4. How is it rotated?
      1. What needs to be done?
      2. How to do it? Sample code
      3. Whom to contact?