***Data in Transit:***

***Channel Encryption***

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. A channel encryption restricts overhearing and tampering in data communication. Most of the common channel encryption are HTTPS and TLS.

* **Transmission Layer Security 1.2**: The TLS is the most widely used protocol for implementing cryptography in web. A TLS provides secure communication channel to traditional TCP/IP protocol suite.
* **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.

Table: TCP/IP Protocol stack with TLS

|  |  |
| --- | --- |
| **TCP/IP Layer** | **Protocol** |
| Internet layer | IP |
| Transmission Control Protocol | TCP |
| Transport Security Layer | TLS |
| Application Layer | NNTP, HTTP, FTP & Telnet |

TLS uses public-key cryptography to provide authentication, and secret-key cryptography with hash functions to provide for privacy and data integrity.

The reason behind the TLS use is that various cryptographic algorithms are being involving in the protocol. TLS uses secret-key and public-key cryptography to provide privacy and data integrity and, authentication respectively.

**SECRET KEY ALGORITHM**

In secret key algorithm, sender and receiver should have same secret key to perform encryption and decryption. Before start sending message over a secure communication channel, secret key must be available at both parties to do encryption and decryption. The below section depicts sample encryption and decryption flows.

1. **Encryption Flow**
2. Obtain an ASCII values of the plain text
3. Generate binary for given text.
4. Calculate 1’s complement and reverse the number
5. Take any four digits as a key e.g., 1011 and then divide the reserved number with chosen key
6. Obtain and store quotient and remainder in first 5 and last 3 digits. If those are less than 5 and 3 digits, then add sufficient number of 0’s. The result would be the cipher text.
7. **Decryption Flow**
8. Obtain first 5 digits and multiply with key and add last 3 digits cipher to result
9. Reverse the output and then get the 1’s complement of it.
10. Identify the decimal equivalent to get the actual ASCII, which is plain text.

As the above algorithm uses same key for both encryption and decryption, it is called as symmetric algorithm. Well-known cryptographic algorithms are Advanced Encryption Standard (AES), Rivest Cipher 4 (RC4) and Triple Data Encryption Standard (3DES).

For more details on Experian Policy and TSB – [Experian policy and TSB](https://experian.sharepoint.com/sites/GlobalSecurityOffice/SitePages/Home.aspx) [🔗](https://experian.sharepoint.com/sites/GlobalSecurityOffice/SitePages/Home.aspx)

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

***PGP Encryption Algorithm***

**Encryption Flow**

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.

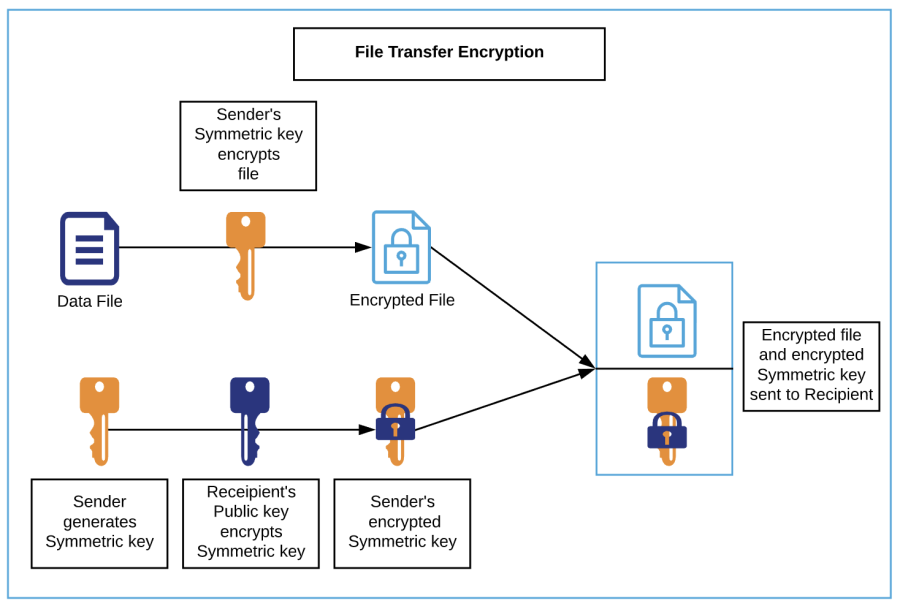
**Decryption Flow**

1. 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).

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

**Encryption flow**:

1. 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.



**Decryption flow**:

1. 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.

The data communication between client and Experian is always encrypted to secure the content using SSL/TLS, which is being transmitted across the networks. Alongside of this, some of the Experian services uses end-to-end payload encryption to secure the payload. For example, if some services using sensitive data like PII must adhere to the company standards and requirements.

**Encryption Flow**

Following are the steps for encryption:

1. A session key is generated along with encryption parameters
2. PII or sensitive data are encrypted using the key.
3. Session key encrypted using recipient’s public key.
4. Session key is sent along with payload and remaining parameters

Diagram

Description automatically generated

**Decryption Flow**

Following are the steps for decryption:

1. The session key will be decrypted using receiver’s private key
2. PII or sensitive data will be decrypted using session key.