**Please provide details how you would implement encryption at network, database and data transfer level**

When designing and implementing solutions in PwC we take security very seriously. Implementing encryption at the network, database, and data transfer levels is crucial for ensuring the security and privacy of your data. Beginning with a high level approach:

**Network Level Encryption**

To secure data as it travels over a network, preventing unauthorized access and eavesdropping.

* TLS/SSL (Transport Layer Security/Secure Sockets Layer): Implement TLS/SSL on network connections. This is commonly used for securing web traffic, email, and many other network services. Ensure that both the server and client support the latest and most secure versions of TLS/SSL.
* Utilizing VPNs to create a secure, encrypted tunnel for data to travel through. This is especially useful in this situation as MOD’s locations are in geographically distributed locations.

**Database Level Encryption:**

Encrypting data in the database ensures that even if an attacker gains access to the database, they cannot read the data without the decryption keys.

* Transparent Data Encryption (TDE): Database management systems, such as Oracle and Microsoft SQL Server offer TDE. We use TDE to encrypt the entire database, including data, log files, and backups.
* Column-level Encryption: For more granular control, you can encrypt specific columns or fields containing sensitive data, such as passwords or credit card numbers. This allows you to encrypt only the most critical data.
* Data at Rest Encryption: Encrypt data when it's stored on disk to protect against physical theft or unauthorized access to database files.

**Data Transfer Level Encryption:**

Securing data during its transfer between different systems is vital to protect it from interception or tampering.

* HTTPS: Use HTTPS for web applications and APIs. It combines HTTP with TLS/SSL, providing end-to-end encryption for data transmitted between clients and servers.
* SFTP (Secure File Transfer Protocol): Use SFTP for secure file transfers. It is an extension of the SSH protocol and encrypts data in transit.
* Encrypted Communication Protocols: For communication between different systems or devices, use encryption protocols like SSH, SCP, or secure email protocols (e.g., SMTPS).

In our solutions design processor always look to use the highest level of security possible and adhere to data Encryption Best Practices:

* Key Management: Implement strong key management practices to securely store and manage encryption keys.
* Secure Passwords: Ensure that encryption keys and passwords are strong, and never store them alongside the encrypted data.
* Regularly Update and Patch: Keep your encryption software and systems up to date to address known vulnerabilities.
* Monitoring and Logging: Implement monitoring and logging to detect and respond to security incidents.

**Oracle Database TDE Encryption**

Encryption is essential to secure data at rest and in transit, and the Oracle database provides tools to do both.

Transparent Data Encryption (TDE) encrypts data at rest within the Oracle database.

TDE encrypts the Oracle Data files at the OS level and stores the encryption keys externally in a keystore.

TDE encrypts data on write and decrypts on read. Both read and write happen transparently; therefore, no changes are required at the application level.

There are two types of TDE encryption:

* **Column:**- Used to encrypt individual columns within a table.  
  - AES 256 algorithm
* **Tablespace:**- Used to encrypt the entire tablespace and all associated data files.  
  - AES 256 algorithm

Both methods use data encryption keys, and both are encrypted by a master encryption key, which is stored outside the database, either in Oracle Key Vault or a wallet.

PwC would configure TDE to use the AES 256 algorithm and Tablespace encryption with Key Vault to store the Master encryption key.

All Oracle databases must have TDE within Oracle OCI; therefore, if there are unencrypted databases, we would convert them to TDE Tablespace encryption during the migration.

With any encryption, there is always a performance overheard as data is decrypted and encrypted.  
Column encryption stores data in the buffer cache encrypted.  
Tablespace encryption encrypts data on write and decrypts on read.  
  
With tablespace encryption, data in the buffer cache is not encrypted, and encryption is invisible to indexes and optimizers.

Suppose data in the buffer cache is encrypted. In that case, all queries using data in the buffer cache would incur a performance overhead, as column encryption affects indexes and can cause the optimizer to do full table scans, which can have a significant hit on performance.

Unless there is a specific need, we will ensure Tablespace encryption is in place across all tablespaces within the database.

We can improve the performance of tablespace encryption by making sure the SGA is large enough to service queries from the buffer cache.

We can tune performance further by using advanced compression to reduce the number of blocks. As the encrypt/decrypt operation happens at the block level, fewer blocks would mean fewer TDE operations, improving performance.