RPC Message Protection

iOS and Java Suite Alignment

APIs Alignment

All platforms should be aligned in term of APIs and behavior.

This will help developers creating apps for both platforms because they will not have to deal with different APIs between the platforms and different library behaviors. This reduces the learning curves for developers.

- iOS
- JavaSE
- Android (with or without managers)

API Naming Alignment

- Class, public function, and public variable names need to be the same (with some exceptions for naming conventions between iOS and Java)
- We recommend that Java Suite uses `EncryptionLifecycleManager` for the class name like iOS has already implemented.
- We recommend making the `EncryptionLifecycleManager` in JavaSE protected instead of public, inaccessible to the developer, and initialized from `LifecycleManager` not from `SdlManager`. In iOS this is already a private class.

Permission Manager

`PermissionManager` should have public methods that will let developers know if encryption is needed for an RPC.

iOS:
@property (assign, nonatomic, readonly) BOOL requiresEncryption ; //already implemented

(BOOL)rpcRequiresEncryption:(UInt32)functionID; (updated per meeting comment)

Android:
public boolean getRequiresEncryption()

public boolean getRPCRequiresEncryption(@NonNull FunctionID)

Sdl Manager

- The `SdlManager` should have a public method to start an encrypted service.

Java

This needs to be included in both the Android and JavaSE SdlManagers
public void startRPCEncryption()

iOS

-(void)startRPCEncryption() // Already implemented

Sdl Manager

- The `SdlManager`s should let the developer know when the encrypted service has started.

Java:

This needs to be included in both the Android and JavaSE SdlManagers. Also both SdlManagers need to deprecate the old setSdlSecurity()` method

builder.setSdlSecurity(secList, serviceEncryptionListener)

interface ServiceEncryptionListener{

void onEncryptionServiceUpdated(@NotNull SessionType type, boolean isServiceEncrypted, @Nullable String error);

}

iOS:

@property (copy, nonatomic, nullable) id<SDLServiceEncryptionDelegate> delegate; // already implemented

- (void)serviceEncryptionUpdatedOnService:(SDLServiceType)type encrypted:(BOOL)encrypted error:(NSError *__nullable)error; // already implemented

Behavior Alignment

- For all platforms, if any RPC request is sent before the encryption service is setup, the RPC should fail and the response handler/listener for that RPC should be called with the error message

Identified Issues

All of the following issues will be addressed in a future proposal for the next release:

- What should happen if an RPC is sent but the encrypted service has not yet been setup? The screen managers will fail if the policy table requires that any RPC needed to start the sub-managers be encrypted and the encrypted service has not yet setup.
- What should happen if an RPC is sent before the OnPermissionChanged notification is received?
- What should happen if the encrypted service failed to start (download certificate failed, no GPS signal…)? Should the library try to start the service again or should the security library handle that?