To enable SafetyNet protection for your android apps, you'll need to (1) create OTPs configured with SafetyNet turned on, and (2) make some changes in your app to provide your Google API key using the newly added api in the sdk

For any doubts in below steps, contact - support@vdocipher.com

Steps to integrate SafetyNet protection on your android app:

1. Create an API key to use Google's SafetyNet api

Follow instructions here:
https://developer.android.com/training/safetynet/attestation#obtain-api-key

2. Otp creation (on your backend)

To create an OTP enabled with SafetyNet protection, provide additional parameters **androidAttest** and **apkCertificateDigestSha256,** where

- **androidAttest** -- is the package name of the target apk, and
- **apkCertificateDigestSha256** -- is the sha256 digest of the certificate used to sign the apk in base64 encoding.

To get the value for **apkCertificateDigestSha256**:
(a) Obtain the sha256 digest of the certificate (debug or release) used to sign the apk before uploading to google play store.

If you are using your own key to sign the apk, use the keytool utility to see its sha256 digest (visible under Certificate fingerprints > SHA256)

Note: Commands below are macOS specific; for other OS, use similar commands.

```
keytool -list -v -keystore <path_to_keystore>
```

Instead, if you use App Signing by Google, the fingerprints can be obtained in the Play Console.

(b) Next, convert the sha256 fingerprint value obtained in step (a) to base64 encoding.
You may use the following command after copying the fingerprint to clipboard:

```
pbpaste | tr -d ':' | xxd -r -p | base64
```

3. Client side changes (android app)

Specify your API key created in step 1 when creating the **`VdoInitParams`**, using the **`VdoInitParams.Builder`**'s **`setSafetyNetApiKey`** method.

After finishing step 1, we recommend testing the SafetyNet implementation on the debug apk first, on your local test devices and emulators. For this, you need to use the debug signing certificate. In step 2, use the debug keystore's path in the keytool command (usually `.android/debug.keystore`).

For testing locally, you can create SafetyNet-enabled OTPs using `curl` with the additional headers `androidAttest` and `apkCertificateDigestSha256`:

```
curl -X POST https://www.vdocipher.com/api/videos/<video-id>/otp -H
'Accept: application/json' -H 'Authorization: Apisecret
<your-api-secret>' -H 'Content-Type: application/json' -d '{"ttl":
2592000, "androidAttest": "your.package.name",
"apkCertificateDigestSha256":
"<base64-encoding-of-sha256-digest-of-signing-certificate>"}'
```

For production, you must use the **release certificate**'s sha256 digest to create the OTP **on your backend**.