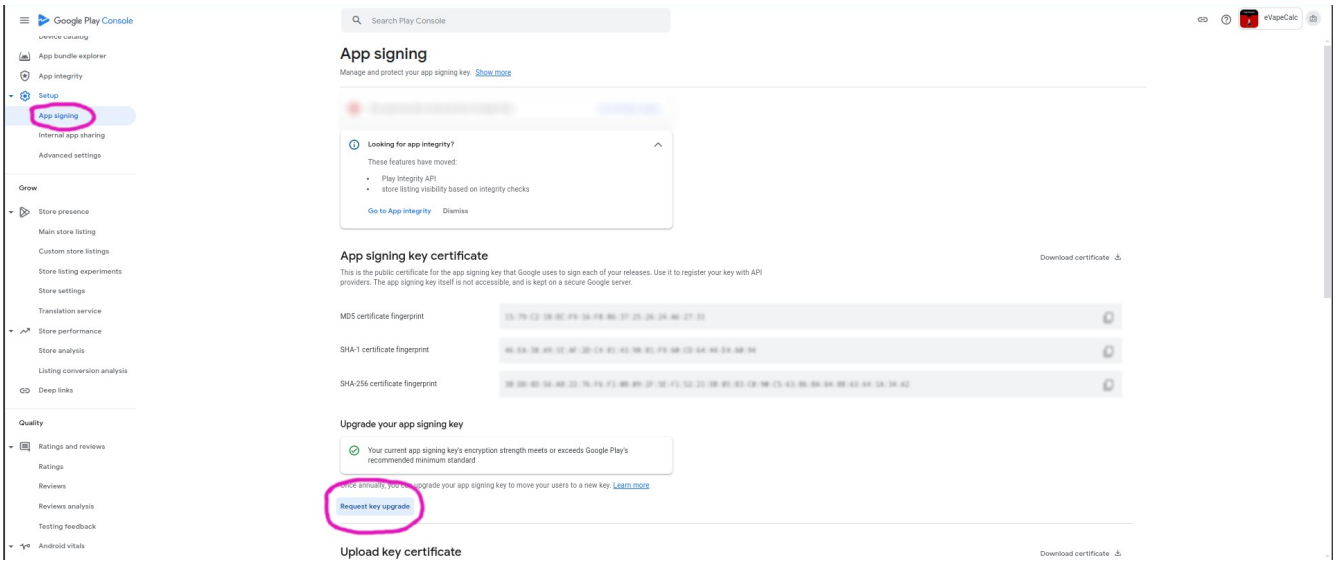

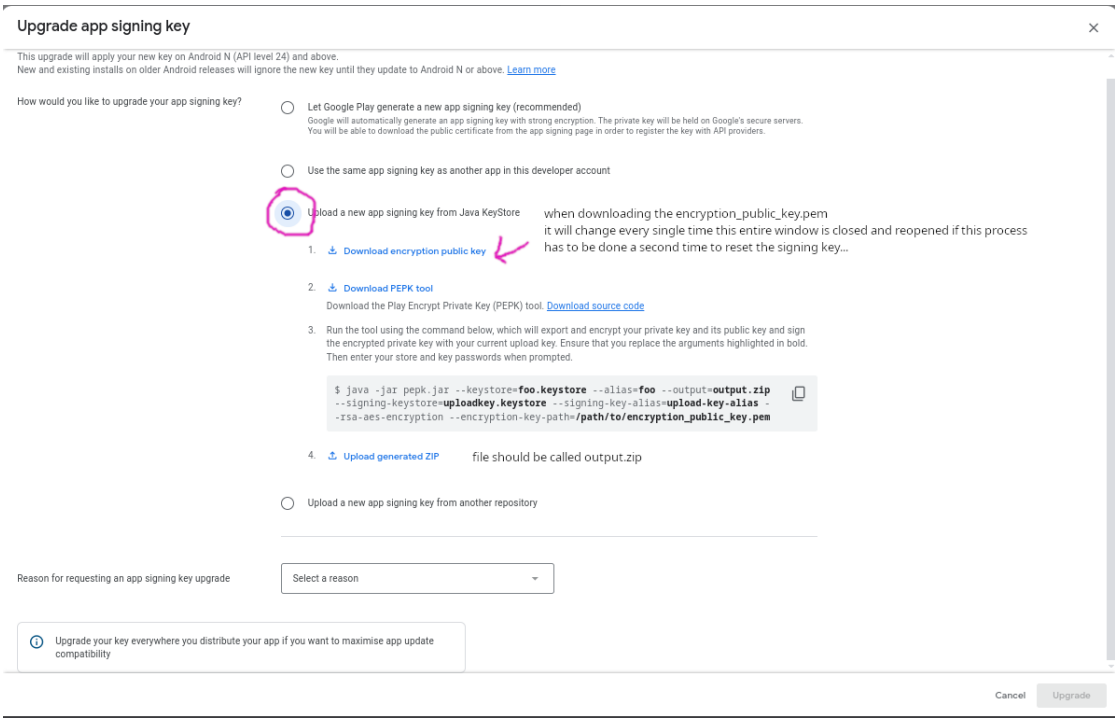

Suggest using “randomstring”
yum install curl -y
or
apt-get update && apt install curl -y
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
source ~/.bashrc

nvm install 21
nvm use 21
npm -g install randomstring

yum install java-17-openjdk-headless java-17-openjdk java-17-openjdk-devel

randomstring > google-key.txt

cp -rf ~/Downloads/pepk.jar
cp -rf ~/Downloads/encryption_public_key.pem

this string should be saved its your random password for the signing key
and my-release-key.jks should be kept safe

cat google-key.txt

keytool -genkey -v -keystore my-release-key.jks -alias alias_name -keyalg RSA -keysize 2048 -validity 10000

java -jar pepk.jar --keystore=my-release-key.jks --alias=alias_name --output=output.zip --include-cert --rsa-aes-encryption --encryption-key-path=encryption_public_key.pem

output.zip will be created here use this to upload to the request key update

after ./gradlew assembleRelease to create the .apk

creating appBundle .aab usually have to take place manually with the project folder as ./gradlew bundleRelease

acutally signing the appbundle .aab (using the exact same key that matches from my-release-key.jks google-key.txt)

jarsigner -verbose -sigalg SHA256withRSA -digestalg SHA-256 -keystore my-release-key.jks APP_NAME__arm64-v8a-release.aab alias_name

see also cordova for actually building html/js applications completely

npm -g install cordova serve