APP网络传输安全

https API

- HttpsURLConnection

```
URL url = new URL("https://google.com");
HttpsURLConnection urlConnection = url.openConnection();
InputStream in = urlConnection.getInputStream();
```

- SSLSocketFactory

```
private synchronized SSLSocketFactory getDefaultSSLSocketFactory() {
    try {
            SSLContext sslContext = SSLContext.getInstance("TLS");
            sslContext.init(null, null, null);
        return defaultSslSocketFactory = sslContext.getSocketFactory();
    } catch (GeneralSecurityException e) {
            throw new AssertionError();
    }
}
```

https API

- TrustManager

```
public interface X509TrustManager extends TrustManager {

    public void checkClientTrusted(X509Certificate[] chain, String authType)
            throws CertificateException;

    public void checkServerTrusted(X509Certificate[] chain, String authType)
            throws CertificateException;

    public X509Certificate[] getAcceptedIssuers();
}
```

自定义信任策略

```
// 取到证书的输入流
InputStream stream = getAssets().open(“server.crt");
KeyStore keystore = KeyStore.getInstance(KeyStore.getDefaultType());
keystore.load(null);
Certificate certificate =
        CertificateFactory.getInstance("X.509").generateCertificate(stream);
// 创建Keystore包含我们的证书
keystore.setCertificateEntry(“ca", certificate);

// 创建TrustManager，仅信任keyStore中的证书
String tmfAlgorithm = TrustManagerFactory.getDefaultAlgorithm();
TrustManagerFactory tmf = TrustManagerFactory.getInstance(tmfAlgorithm);
tmf.init(keyStore);

//用TrustManager初始化一个SSLContext
SSLContext context = SSLContext.getInstance("TLS");
context.init(null, tmf.getTrustManagers(), null);

URL url = new URL(path);
HttpsURLConnection conn = (HttpsURLConnection) url.openConnection();
conn.setSSLSocketFactory(context.getSocketFactory());
InputStream in = urlConnection.getInputStream();
```