

Abstract

Introduction Mobile security is getting personal. Smartphones and tablets contain some of our most precious memories and personal details, including access to financial transactions and other sensitive information. Cybercriminals now develop five new threats every second. With the sharp growth in smartphones and tablets, the bad guys are aggressively attacking these devices and your information with a new range of insidious threats.

Here at Intel, we've begun creating small software applications to provide to independent software vendors (ISVs) as examples of how to use the security features on the platforms based on Intel® Architecture.

The purpose of this paper is to give developers a quick start by describing the basic Android* data encryption API and showing how to encrypt application data for the Android OS.

```

private void encryptVideo() {
    FileInputStream fis;
    try {
        fis = new FileInputStream(new
File("/sdcard/h264.mp4"));

        File outfile = new File("/sdcard/h264_enc.mp4");
        int read = 0;
        if (!outfile.exists())
            outfile.createNewFile();
        File decfile = new File("/sdcard/h264_dec.mp4");
        if (!decfile.exists())
            decfile.createNewFile();
        FileOutputStream fos = new FileOutputStream(outfile);
        FileInputStream encfis = new FileInputStream(outfile);
        FileOutputStream decfos = new
FileOutputStream(decfile);
        Cipher encipher = Cipher.getInstance("AES");
        Cipher decipher = Cipher.getInstance("AES");
        KeyGenerator kgen = KeyGenerator.getInstance("AES");

        SecretKey skey = kgen.generateKey();
        encipher.init(Cipher.ENCRYPT_MODE, skey);
        CipherInputStream cis = new CipherInputStream(fis,
encipher);

        decipher.init(Cipher.DECRYPT_MODE, skey);
        CipherOutputStream cos = new
CipherOutputStream(decfos, decipher);
        long start = System.nanoTime();
        Log.d("security", String.valueOf(start));
        while ((read = cis.read()) != -1) {
            fos.write((char) read);
            fos.flush();
        }
        long stop = System.nanoTime();
        Log.d("security", String.valueOf(stop));
        long seconds = (stop - start) / 1000000000; // for
seconds

        Log.d("security", String.valueOf(seconds));
        fos.close();
        mTimeTook = seconds;
        mDoneEncryption = true;

    } catch (FileNotFoundException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (NoSuchAlgorithmException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (NoSuchPaddingException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (InvalidKeyException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
}

```

