Implementing Elliptic Curve Integrated Encryption Scheme on Android Platforms

Liang Xia

Department of Computer Science
University of California Santa Barbara

liangxia@cs.ucsb.edu
June 2013

Table of Contents

Target Platform

- Android 4.2.2 API Level 17
- Development Environment: Eclipse SDK, Java SE 6
- Test devices
  - Android Virtual Device
    - ARM (armeabi-v7a)
  - Samsung Nexus S
    - ARM Cortex-A8 CPU @ 1GHz, 512MB RAM.
  - Android Mini TV Stick
    - Dual-Core A9 Processor @ 1GHz, 1GB RAM.

Elliptic Curve Cryptography

- Elliptic curve cryptography (ECC) is one of the strongest cryptography in terms of security level.

- A 160-bit ECC key is roughly equivalent to a 1024-bit RSA key.

Chosen Elliptic Curve

- Weierstrass form

- Curve P-192
  - p = 6277101735386680763835789423207666416083908700390324961279
  - n = 6277101735386680763835789423176059013767194773182842284081
  - a = ffffffff ffffffff ffffffff fffffffe ffffffff fffffffc
  - b = 64210519 e59c80e7 0fa7e9ab 72243049 feb8deec c146b9b1
  - Gx = 188da80e b03090f6 7cbf20eb 43a18800 f4ff0afd 82ff1012
  - Gy = 07192b95 ffc8da78 631011ed 6b24cdd5 73f977a1 1e794811
  - P(Gx, Gy)

ECIES

- Elliptic Curve Integrated Encryption Scheme is a hybrid encrytion scheme that works like static Diffie-Hellman followed by symmetric encryption.

- The scheme ECIES is composed of three algorithms: key generation, encryption and decryption.

ECIES Key Generation

- Choose a shared secret d as a private key. $d \in [1, p]$
- Generate a public key Q = [d]P
- Return key pair (Q, d).

ECIES Encryption

- Assume Alice wants to send Bob a message. Bob has public key Q. Alice and Bob both know the private key d.
- Alice generates a random number $k \in [1, p]$.
- Alice calculates U = [k]P.
- Alice calculates T = [k]Q.
- Alice uses a key derivation function (KDF) to compute two keys k1 and k2 from T.
  - uses SHA-256 to hash 192-bit x coordinate of Point T into 256 bit code
  - breaks it into 128-bit key k1 and 128-bit key k2.

ECIES Encryption (cont.)

- Alice uses 128-bit AES encryption algorithm to encrypt her message m with key k1, and obtain cipher text c.
- Alice chooses HMAC-SHA256 to calculate a message authentication code (MAC) r with key k2.
- Alice sends (U, c, r) as the cipher text to Bob.

ECIES Decryption

- Bob parses the cipher text e into (U, c, r).
- Bob has the secret private key d, so he can computes T = [d]U.
- Bob uses the same key derivation function to obtain k1, and k2.
- Bob computes MAC using k2 and compares the result with r.
- If they are different, he decides the message is invalid.
- Bob uses the 128-bit AES decryption algorithm to decrypt c. He obtains the original message m.

Java Crypto Lib: Bouncy Castle

- Bouncy Castle (BC) is a set of easy-to-use cryptography APIs.
- It is a provider for the Java Cryptography Extension and the Java Cryptography Architecture.
- It is implemented in both Java and C#.

Spongy Castle (Android Version)

- Spongy Castle is the stock Bouncy Castle libraries with a couple of small changes to make it work on Android.

- The following libraries needs to be downloaded and be set in java build path.
  - sc-light-jdk15on (jar) - Core lightweight API
  - scprov-jdk15on (jar) - JCE provider (requires sc-light-jdk15on)
  - scpkix-jdk15on (jar) - PKIX, CMS, EAC, TSP, PKCS, OCSP, CMP, and CRMF APIs (requires scprov-jdk15on)

Implementation - public class ECIES_Engine

- class CipherText - the encrypted message sent to Bob
- public String initialize() - initialize the curve and the point P
- CipherText ECIES_Encrypt(byte[] message)
  - the ECIES encryption function
  - Input: Alice's message
  - Output: encrypted cipher text
- byte[] ECIES_Decrypt(CipherText e)
  - the ECIES decryption function
  - Input: cipher text
  - Output: the original message

Implementation - public class ECIESDemoActivity

- Android activity class as user GUI
- shows the results of encryption and decryption

Evaluation

- I tested ECIES on three devices.
  - Android Virtual Device
  - Samsung Nexus S (ARM A8)
  - Android Mini TV Stick (ARM A9)

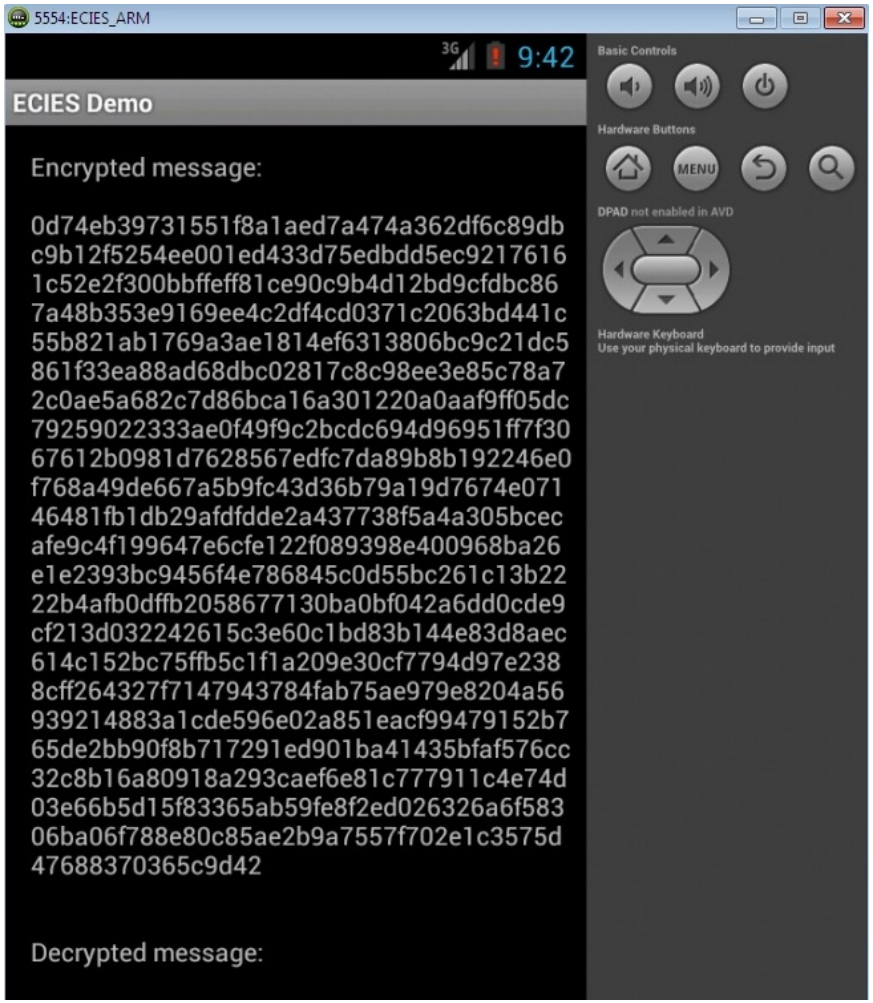
- Below are their completion time of encryption and decryption.

Table 1: ECIES Encryption and Decryption Time on Different Devices.

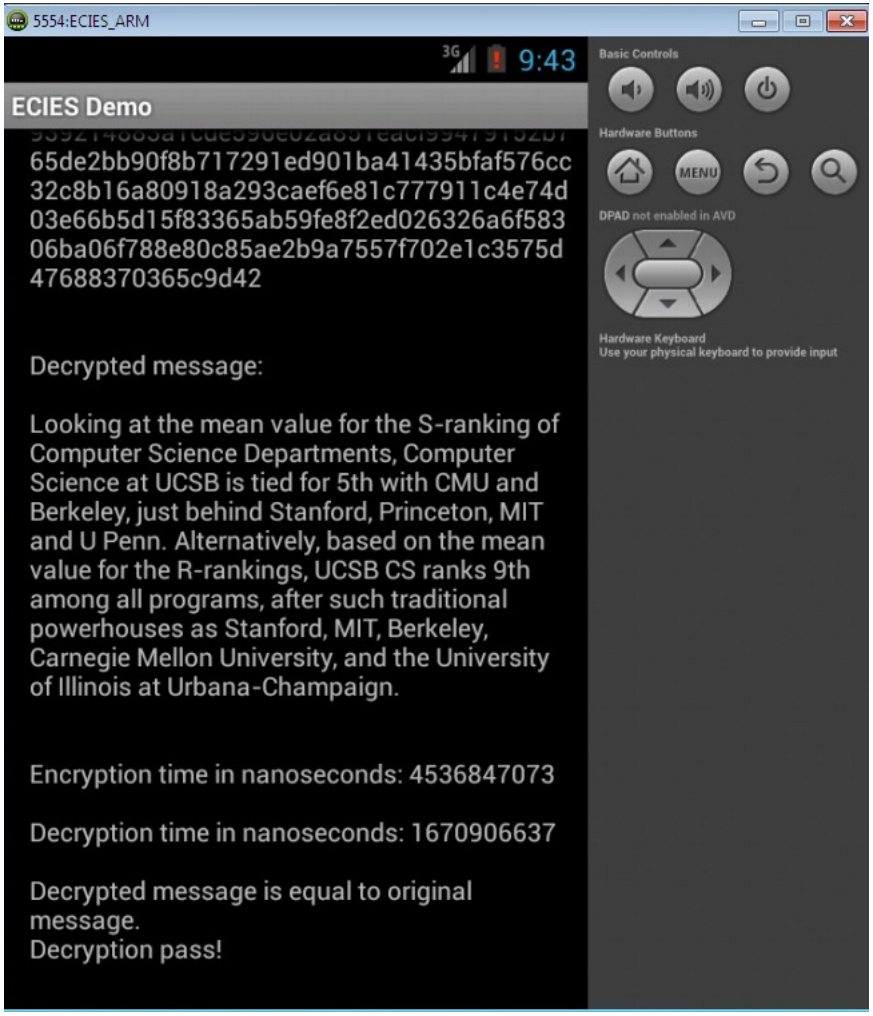
Device	Enc (ms)	Dec (ms)
Android Virtual Device	4536	1670
Samsung Nexus S	566	225
Android Mini TV Stick	337	181

Demo - Android Virtual Device

Demo - Android Virtual Device (cont.)



Demo - Android Virtual Device (cont.)



Demo - ARM A9 Processor

- Displayed on a TOSHIBA TV