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>
<thead>
<tr>
<th>Table of Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Target Platform</td>
</tr>
<tr>
<td>• Elliptic Curve Cryptography</td>
</tr>
<tr>
<td>• Elliptic Curve Integrated Encryption Scheme</td>
</tr>
<tr>
<td>• Java Crypto Lib: Bouncy Castle</td>
</tr>
<tr>
<td>• Java Crypto Lib on Android: Spongy Castle</td>
</tr>
<tr>
<td>• ECIES Implementation</td>
</tr>
<tr>
<td>• Evaluation</td>
</tr>
<tr>
<td>• Demo</td>
</tr>
</tbody>
</table>
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 (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
  - \( p = 6277101735386680763835789423207666416083908700390324961279 \)
  - \( n = 6277101735386680763835789423176059013767194773182842284081 \)
  - \( a = fffffffe ffffffff ffffffff fffffffe ffffffff fffffffc \)
  - \( b = 64210519 e59c80e7 0fa7e9ab 72243049 feb8deec c146b9b1 \)
  - \( Gx = 188da80e b03090f6 7cbf20eb 43a18800 f4ff0afd 82ff1012 \)
  - \( Gy = 07192b95 ffc8da78 631011ed 6b24cdd5 73f977a1 1e794811 \)
  - \( P(Gx, Gy) \)
Elliptic Curve Integrated Encryption Scheme is a hybrid encryption 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 $k_1$ and $k_2$ from T.
  - uses SHA-256 to hash 192-bit x coordinate of Point T into 256 bit code
  - breaks it into 128-bit key $k_1$ and 128-bit key $k_2$. 


ECIES Encryption (cont.)

- Alice uses 128-bit AES encryption algorithm to encrypt her message \( m \) with key \( k_1 \), and obtain cipher text \( c \).

- Alice chooses HMAC-SHA256 to calculate a message authentication code (MAC) \( r \) with key \( k_2 \).

- 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 $m$. 
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 need 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.

<table>
<thead>
<tr>
<th>Device</th>
<th>Enc (ms)</th>
<th>Dec (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android Virtual Device</td>
<td>4536</td>
<td>1670</td>
</tr>
<tr>
<td>Samsung Nexus S</td>
<td>566</td>
<td>225</td>
</tr>
<tr>
<td>Android Mini TV Stick</td>
<td>337</td>
<td>181</td>
</tr>
</tbody>
</table>
Demo - Android Virtual Device

ECIES Demo

Author: Liang Xia. This demo shows how to use spongy castle to implement Elliptic Curve Integrated Encryption Scheme.

Message to encrypt:

Looking at the mean value for the S-ranking of Computer Science Departments, Computer Science at UCSB is tied for 5th with CMU and Berkeley, just behind Stanford, Princeton, MIT and U Penn. Alternatively, based on the mean value for the R-rankings, UCSB CS ranks 9th among all programs, after such traditional powerhouses as Stanford, MIT, Berkeley, Carnegie Mellon University, and the University of Illinois at Urbana-Champaign.

Encrypted message:

0d74eb39731551f8a1aed7a474a362df6c89db
c9b12f5254ee01ed433d75edbbdf5ec9217616
1c52e2f300bbffef81ce90c9b4d12bd9f0db86
7a48b353e9169ee4c2df4cd237c2063bd441c
55b21ab1769a3ae1814ef631386bc9c21dc5
861f33ea88ad68db02817c8c9ee3e85c78a7
2caee5a68267d86bca16a301220a9aa9fff05dc
79259022333ae0f49f9c2bcdc6594d69951ff7f30
Demo - Android Virtual Device (cont.)
Demo - Android Virtual Device (cont.)

ECIES Demo

ECIES Demo

Decrypted message:

Looking at the mean value for the S-ranking of Computer Science Departments, Computer Science at UCSB is tied for 5th with CMU and Berkeley, just behind Stanford, Princeton, MIT and U Penn. Alternatively, based on the mean value for the R-rankings, UCSB CS ranks 9th among all programs, after such traditional powerhouses as Stanford, MIT, Berkeley, Carnegie Mellon University, and the University of Illinois at Urbana-Champaign.

Encryption time in nanoseconds: 4536847073

Decryption time in nanoseconds: 1670906637

Decrypted message is equal to original message.
Decryption pass!
● Displayed on a TOSHIBA TV