SECURE MOBILE APPS BASED ON NTRU

Vikram Hegde, Hanqing Zhao, Kefeng Shi, and Yi Yang

Fontbonne University
St. Louis, Missouri
BACKGROUND

- The need for mobile security
- The inefficiency of traditional algorithms in mobile apps (e.g. RSA)
SERVER ARCHITECTURE

- Server generates the cryptographic key
- Users “purchase” the cryptographic key
- Slow and unsecure
WHAT WE DO

- We apply the peer-to-peer method
- Use NTRU algorithm
- Fast and secure
WHAT IS NTRU

- NTRU (N-th Degree Truncated Polynomial Ring)
- A patented, open source public-key cryptosystem
- First public-key cryptosystem not based on factorization (such as RSA) or discrete logarithmic problems (ECC)
- Resistant to attacks using Shor’s algorithm in a quantum computer (unlike RSA and ECC)
- Less resource usage (memory, CPU)
PREVIOUS RESULTS (PC)

![Graph showing performance comparisons between different cryptographic algorithms.

Implementations used:
- RSA: OpenSSL 1.0.2a
- ECC-NIST: OpenSSL 1.0.2a
- curve25519: curve25519-donna
- ntru-ref: https://github.com/NTRUOpenSourceProject/NTRUEncrypt @12c811fa
- libntru: libntru 0.4

Hardware: Intel Core i7-2600 @3.1GHz
OS: 64-bit Linux

METHODOLOGY

Download and Install:

- Java NTRU library (including Benchmark app source code) from https://tbuktu.github.io/ntru/
- Eclipse IDE, Android SDK and Android Tools for Eclipse (on PC)
- CmdConsole (on Android device)
METHODOLOGY

- Compile source code for benchmark app included with NTRU library
- Install CmdConsole on mobile device to run the benchmark app
- Install benchmark app on phone
- Execute benchmark app inside CmdConsole using “run” command
METHODOLOGY

How CmdConsole works

## Evaluation

- We tested our results on a Samsung Galaxy S6 edge+ and Samsung Galaxy Express

<table>
<thead>
<tr>
<th>Model</th>
<th>Released</th>
<th>Chipset</th>
<th>CPU</th>
<th>OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samsung Galaxy Express</td>
<td>2012, November</td>
<td>Qualcomm MSM8960 Snapdragon S4 Plus</td>
<td>Dual-core 1.5 GHz Krait</td>
<td>Android 4.1.2</td>
</tr>
<tr>
<td>Samsung Galaxy S6 Edge+</td>
<td>2015, August</td>
<td>Exynos 7420 Octa</td>
<td>Octa-core (4x2.1 GHz Cortex A57 &amp; 4x1.5 GHz Cortex-A53)</td>
<td>Android 6.0.1</td>
</tr>
</tbody>
</table>
**SAMSUNG GALAXY EXPRESS**

Key Generation Time (ms/op)

<table>
<thead>
<tr>
<th>Algorithm / Key Size (or NTRU Parameter Set)</th>
<th>Key Generation Time (ms/op)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA-2048</td>
<td>3700</td>
</tr>
<tr>
<td>Curve25519 (256 bits)</td>
<td>50</td>
</tr>
<tr>
<td>ECDH-256</td>
<td>15</td>
</tr>
<tr>
<td>ECDH-521</td>
<td>25</td>
</tr>
<tr>
<td>NTRU-439</td>
<td>50</td>
</tr>
<tr>
<td>NTRU-743</td>
<td>150</td>
</tr>
</tbody>
</table>
CIPHERTEXT SIZE (BYTES)

Algorithms
1. RSA-2048
2. RSA-4096
3. RSA-7680
4 – 6: Excluded (ECC)
7. NTRU-439
8. NTRU-743
## COMPARISON OF PERFORMANCE/SECURITY
(SAMSUNG GALAXY EXPRESS)

<table>
<thead>
<tr>
<th>Symmetric Security Level (bits) [3]</th>
<th>Algorithm/Key Size (or RSA Modulus, or NTRU Param. Set)</th>
<th>Key Generation Time (ms/op)</th>
<th>Encryption Time (ms/op)</th>
<th>Decryption Time (ms/op)</th>
<th>Ciphertext Length (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>112</td>
<td>RSA-2048</td>
<td>3750.22</td>
<td>1.66</td>
<td>47.9</td>
<td>256</td>
</tr>
<tr>
<td>128</td>
<td>RSA-4096</td>
<td>10943.42</td>
<td>5.56</td>
<td>305.92</td>
<td>512</td>
</tr>
<tr>
<td>192</td>
<td>RSA-7680</td>
<td>378871.61</td>
<td>18.48</td>
<td>1960.80</td>
<td>960</td>
</tr>
<tr>
<td>128</td>
<td>Curve25519 (256 bits)</td>
<td>9.41</td>
<td>18.91</td>
<td>9.86</td>
<td>N/A</td>
</tr>
<tr>
<td>128</td>
<td>ECDH-256</td>
<td>23.04</td>
<td>81.83</td>
<td>61.22</td>
<td>N/A</td>
</tr>
<tr>
<td>256</td>
<td>ECDH-521</td>
<td>100.72</td>
<td>405.55</td>
<td>307.17</td>
<td>N/A</td>
</tr>
<tr>
<td>128</td>
<td>NTRU-439</td>
<td>133.16</td>
<td>6.12</td>
<td>5.64</td>
<td>604</td>
</tr>
<tr>
<td>256</td>
<td>NTRU-743</td>
<td>346.27</td>
<td>11.38</td>
<td>11.22</td>
<td>1022</td>
</tr>
</tbody>
</table>
SAMSUNG GALAXY S6 EDGE+

Key Generation Time (ms/op)

ms/op

Algorithm / Key Size (or NTRU Parameter Set)

RSA-2048

Curve25519 (256 bits)

ECDH-256

ECDH-521

NTRU-439

NTRU-743
## COMPARISON OF PERFORMANCE/SECURITY (SAMSUNG GALAXY S6 EDGE+)

<table>
<thead>
<tr>
<th>Symmetric Security Level (bits) [3]</th>
<th>Algorithm/Key Size (or RSA Modulus, or NTRU Param. Set)</th>
<th>Key Generation Time (ms/op)</th>
<th>Encryption Time (ms/op)</th>
<th>Decryption Time (ms/op)</th>
<th>Ciphertext Length (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>112</td>
<td>RSA-2048</td>
<td>584.81</td>
<td>.44</td>
<td>13.52</td>
<td>256</td>
</tr>
<tr>
<td>128</td>
<td>RSA-4096</td>
<td>2173.47</td>
<td>1.33</td>
<td>81.22</td>
<td>512</td>
</tr>
<tr>
<td>192</td>
<td>RSA-7680</td>
<td>78632.23</td>
<td>4.54</td>
<td>507.51</td>
<td>960</td>
</tr>
<tr>
<td>128</td>
<td>Curve25519 (256 bits)</td>
<td>5.24</td>
<td>10.38</td>
<td>5.15</td>
<td>N/A</td>
</tr>
<tr>
<td>128</td>
<td>ECDH-256</td>
<td>1.15</td>
<td>3.54</td>
<td>2.36</td>
<td>N/A</td>
</tr>
<tr>
<td>256</td>
<td>ECDH-521</td>
<td>5.95</td>
<td>22.79</td>
<td>16.23</td>
<td>N/A</td>
</tr>
<tr>
<td>128</td>
<td>NTRU-439</td>
<td>20.05</td>
<td>0.53</td>
<td>0.54</td>
<td>604</td>
</tr>
<tr>
<td>256</td>
<td>NTRU-743</td>
<td>49</td>
<td>0.81</td>
<td>0.92</td>
<td>1022</td>
</tr>
</tbody>
</table>
NTRU had the best balance of performance and security. It had the fastest encryption and decryption times, the longest cipher text, and was the only algorithm we benchmarked that is resistant to attacks by quantum computers using Shor’s algorithm.

The ECC algorithms had the shortest key generation time, but they had the slowest encryption, and decryption was slower than NTRU at a given security level.

RSA had the slowest key generation, the slowest decryption, and encryption was pretty fast.
FUTURE WORK

- Build an app for Android and/or iOS that uses NTRU to encrypt peer-to-peer communication.
- Do benchmarks using other implementations of NTRU
REFERENCES


Thanks!