ANDROID SECURE COMMUNICATION SYSTEM USING STEGANOGRAPHY

Kleona Binjaku
Elinda Kajo Meçë

DAAD Workshop "Cooperation at Academic Informatics Education across Balkan Countries and Beyond"
Primošten, Croatia, 2nd - 8th September 2018
Content

➢ Introduction

➢ Application Development Phases

➢ Demonstration

➢ Conclusions
Introduction

Why secure application for mobiles?
- we mostly use mobiles for communication
- information we share is no more secure

Why steganography?
- the third party doesn’t even know if something is hidden
- interesting way of hiding information.

Why Android?
- 87% of users prefer Android operating system in their mobiles
- easier to program
Steganography-Theoretical background

Facts
Since 1499
Used in World War

to hide the information in a digital carrier, so that it is undetectable by third party

Goal
Does it contain a secret message?

Cover object
Stego-object

ESCAPE PLAN

DAAD Workshop "Cooperation at Academic Informatics Education across Balkan Countries and Beyond"
Primošten, Croatia, 2nd - 8th September 2018
Types and methods for steganography

- **Text**
  - Line shift coding
  - Word shift coding
  - Feature coding

- **Image**
  - LSB
  - Masking & filtering
  - Transformation

- **Audio**
  - LSB Coding
  - Phase coding
  - Spread Spectrum

DAAD Workshop "Cooperation at Academic Informatics Education across Balkan Countries and Beyond"
Primošten, Croatia, 2nd - 8th September 2018
Software development phases

WATERFALL MODEL

Requirements
Analysis and Definition

Design

Implementation

Testing

DAAD Workshop "Cooperation at Academic Informatics Education across Balkan Countries and Beyond"
Primošten, Croatia, 2nd - 8th September 2018
Use case diagram

1. **Home**
   - Encode text to image
   - Decode text from image
   - Encode image to image
   - Decode image from image
   - Share
   - Save

**Requirements Analysis and Definition**
**Design**
**Implementation**
**Testing**
Class diagram

One package
11 Java classes
Why LSB2:
- high capacity
- high imperceptibility.
Application name: CryptoStego
Minimum Required SDK: API 23
Target SDK: API 25
Screen size: Adaptive
Screen mode: Portrait
Online/Offline work: App works perfectly in offline mode. But, internet is needed to share images.
Application structure in Android Studio

- Fragments for different screens
- Java classes
- UI Layouts

Requirements
Analysis and Definition
Design
Implementation
Testing
Full Encoding / Decoding Procedure

**Encoding**
- Secret Input text/image
- AES
- LSB2
- Cover image
- Encoded image in byte

**Decoding**
- Encoded image
- Inverse LSB2
- Inverse AES
- Output text/image

**Key Generator**
- Key generator
- Key
- Password

**Requirements**
- Analysis and Definition
- Design
- Implementation
- Testing
Encode text to image

Text input

Password (alternative)

Cover image uploaded

Share image

Button to load image

Button for encoding

Help

Demostration
Decode text from image

- Button to load image
- Place where the chosen image will be shown
- Button for decoding
- Place where the hidden message will show
- Help

Requirements
Analysis and Definition
Design
Implementation
Testing
Testing method

Functional testing

Focus: on functions of applications

Why?

- Easier to implement
- Ensures that the software works as expected
- Ensures the proper working of all the functionalities
Min. test cases = 7
Max. test cases = 1008
<table>
<thead>
<tr>
<th>Test case</th>
<th>Text</th>
<th>Password</th>
<th>Image type</th>
<th>Image time chosen</th>
<th>Encode button</th>
<th>Share button</th>
<th>Help button</th>
<th>Expected result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Encode without encryption</td>
<td>Chosen before image</td>
<td>Not set</td>
<td>png</td>
<td>$\geq 4 \times \text{TextLength} + 100$</td>
<td>pushed</td>
<td>unpushed</td>
<td>unpushed</td>
<td>Encoded image</td>
</tr>
<tr>
<td>2. Encode with encryption</td>
<td>Chosen before image</td>
<td>1 to 15 chars</td>
<td>jpg</td>
<td>$\geq 4 \times \text{TextLength} + 100$</td>
<td>pushed</td>
<td>unpushed</td>
<td>unpushed</td>
<td>Encoded image with encrypted info</td>
</tr>
<tr>
<td>3. Password longer than allowed</td>
<td>Chosen before image</td>
<td>Longer than 15 chars</td>
<td>png</td>
<td>$\geq 4 \times \text{TextLength} + 100$</td>
<td>pushed</td>
<td>unpushed</td>
<td>unpushed</td>
<td>“Maximum password length is 15 chars” ERROR MESSAGE</td>
</tr>
<tr>
<td>4. Non allowed format image</td>
<td>Chosen before image</td>
<td>Not set</td>
<td>Other</td>
<td>$\geq 4 \times \text{TextLength} + 100$</td>
<td>pushed</td>
<td>unpushed</td>
<td>unpushed</td>
<td>“Not allowed format type. Should be PNG or JPG” ERROR MESSAGE</td>
</tr>
<tr>
<td>5. Text after image, image large enough to hide text</td>
<td>$\leq (\text{ImageSize-100})/4$</td>
<td>1 to 15 chars</td>
<td>png</td>
<td>before text</td>
<td>pushed</td>
<td>unpushed</td>
<td>unpushed</td>
<td>Encoded image</td>
</tr>
<tr>
<td>6. Text after image, image not large enough to hide text</td>
<td>$&gt; (\text{ImageSize-100})/4$</td>
<td>1 to 15 chars</td>
<td>jpg</td>
<td>before text</td>
<td>pushed</td>
<td>unpushed</td>
<td>unpushed</td>
<td>“Image not large enough to hide text” ERROR MESSAGE</td>
</tr>
</tbody>
</table>
## Test cases (2)
(Encode text to image)

<table>
<thead>
<tr>
<th>Test case</th>
<th>Text</th>
<th>Password</th>
<th>Image type</th>
<th>Image time chosen</th>
<th>Encode button</th>
<th>Share button</th>
<th>Help button</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.Image not large enough to hide text</td>
<td>Chosen before image</td>
<td>Not set</td>
<td>png</td>
<td>&lt;4*TextLength+100</td>
<td>pushed</td>
<td>unpushed</td>
<td>unpushed</td>
<td>“Image not large enough to hide text”ERROR MESSAGE</td>
</tr>
<tr>
<td>8.Share button pushed after encoding</td>
<td>Chosen before image</td>
<td>1 to 15 chars</td>
<td>jpg</td>
<td>&gt;=4*TextLength+100</td>
<td>pushed</td>
<td>pushed</td>
<td>unpushed</td>
<td></td>
</tr>
<tr>
<td>9.Share button pushed before encoding</td>
<td>Chosen before image</td>
<td>Not set</td>
<td>png</td>
<td>&gt;=4*TextLength+100</td>
<td>unpushed</td>
<td>pushed</td>
<td>pushed</td>
<td>“No image encoded” ERROR MESSAGE</td>
</tr>
<tr>
<td>10.Help button pushed</td>
<td>Chosen before image</td>
<td>1 to 15 chars</td>
<td>jpg</td>
<td>&gt;=4*TextLength+100</td>
<td>pushed</td>
<td>unpushed</td>
<td>unpushed</td>
<td>Help screen displayed</td>
</tr>
<tr>
<td>11.Text not entered on encode</td>
<td>Not entered</td>
<td>Not set</td>
<td>png</td>
<td>Before text</td>
<td>pushed</td>
<td>Unpushed</td>
<td>Unpushed</td>
<td>“Insert text message”ERROR MESSAGE</td>
</tr>
<tr>
<td>12.Image not entered on encode</td>
<td>Chosen before image</td>
<td>Not set</td>
<td>Not chosen</td>
<td>Not chosen</td>
<td>pushed</td>
<td>unpushed</td>
<td>Unpushed</td>
<td>“Cover image not chosen”ERROR MESSAGE</td>
</tr>
</tbody>
</table>
**Demostration**

Encode

**ENCODE TEXT TO IMAGE**

Text Message

HELLO!

Password (alternative)

******

LOAD IMAGE

ENCODE MESSAGE

Decode

**DECODE TEXT FROM IMAGE**

LOAD IMAGE

Hidden message

HELLO!

GET HIDDEN MESSAGE

**Requirements**

Analysis and Definition

**Design**

**Implementation**

**Testing**
Error messages

- Cover image not big enough to hide secret image
- Share button pressed without encoding
- No image encoded
Conclusions

Developed new skills along with the learned ones

The application:
✓ Hides the information successfully
✓ Uses steganography in an effective way
✓ Retrieves the hidden information correctly
✓ Decreases the possibility that the information will be caught by a third party using cryptography
THANK YOU!