

MobiWear: A Plausibly Deniable Encryption System for Wearable Mobile Devices

Presenter: Niusen Chen
Spring 2021 MTU CS Cybersecurity Reading Group

Outline

- Background Introduction
- Motivation
- Design of MobiWear
- Evaluation
- Discussion

Mobile Devices is Ubiquitous



Smartphone



Tablet



Smartwatch

Wearable Devices

- For the purpose of general computing, fitness tracker
- Embedded a few sensors like accelerometers, gyroscopes, and heart ratesensors
- Small in size and limited powerful hardware (e.g., small screen, less powerful RAM)



The Architecture of Wearable Devices



Applications



Operating system (e.g., Wear OS)

Block-based access interface

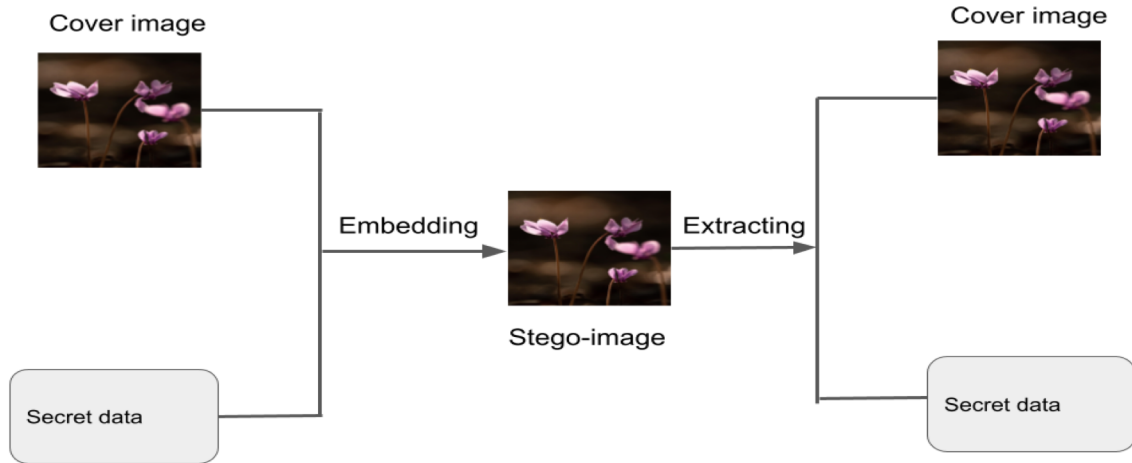
Flash translation layer (FTL)

Flash memory

Flash-based block device (e.g., microSD)

Image Steganography

- It is used to hide information in a cover image
- It can be performed in two domains:
 - Spatial domain
 - Transform domain







Spatial Domain

Least Significant Bits (LSB):





- Each pixel can be represented by 4 bytes in an ARGB image
 - R:red
 - G:green
 - B:blue
 - A:transparency
- Secret data are hidden in the least significant bit(s) of each pixel
- LSB has minimal effects on the image quality

Spatial Domain (cont.)

“Hi”: 1 0 0 1 0 1

A		11001011
R		11011011
G		10001011
B		10101000





Pixel 1

A		10011011
R		11111010
G		10000010
B		10101001





Pixel 2

Spatial Domain (cont.)

“Hi”: 1 0 0 1 0 1

A		1100101 ¹
R		1101101 ⁰
G		1000101 ⁰
B		1010100 ¹

Pixel 1

A		1001101 ⁰
R		1111101 ¹
G		10000010
B		10101001

Pixel 2

Watermarking

- It is typically used to identify ownership of the copyright
- It includes visible watermarking and invisible watermarking
- It can be achieved by image steganography technique



Original Image



Encoded Image

Motivation

How to Protect Confidentiality

- Full Disk Encryption (FDE)
 - Everything on disk is encrypted
 - Totally transparent to users
 - Can **not** defend against coercive attack

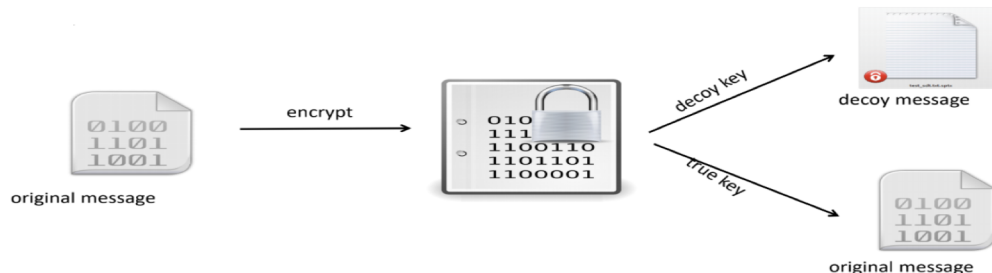
Coercive Attack:

An attacker forces the device owner to disclose the decryption key



Plausibly Deniable Encryption (PDE):

- A crypto primitive designed for mitigating coercive attacks
- Plain text is encrypted by a true key and a decoy key such that:
 - Decrypt with decoy key ➡ Decoy message
 - Decrypt with true key ➡ True message
- Upon being coerced: disclose decoy key, keep true key
- PDE is hard to be achieved in crypto
- Two techniques to simulate PDE
 - Hidden volume technique
 - Steganography technique



PDE Technique

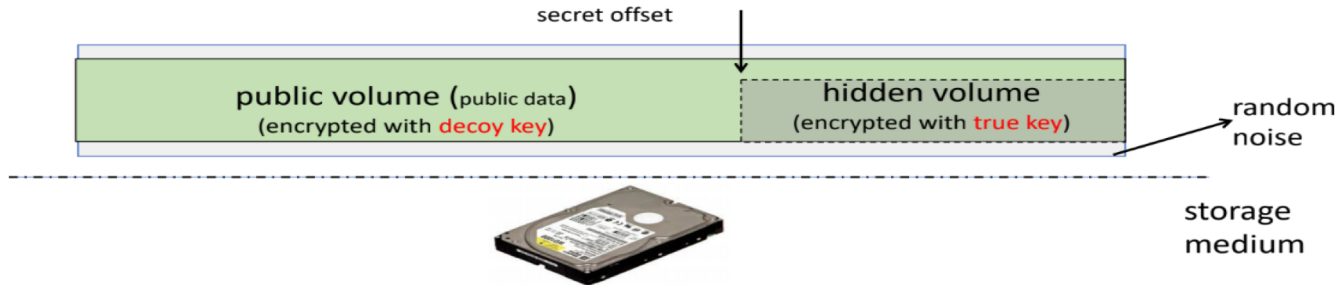
Steganography Technique

- Storage is filled with randomness
- The secret data are encrypted with secrets keys and stored at secret random locations of the entire disk
- Hidden secret data may be overwritten. Therefore, system should maintain several copies of secret data

PDE Technique (cont.)

Hidden Volume Technique

- Whole disk is initialized with randomness
- Two volumes: public volume and hidden volume
 - Public volume: encrypted with a **decoy** key; store non-sensitive data
 - Hidden volume: encrypted with **true** key; store sensitive data
- Disclosing the decoy key upon being coerced by attacker



Existing PDE Works

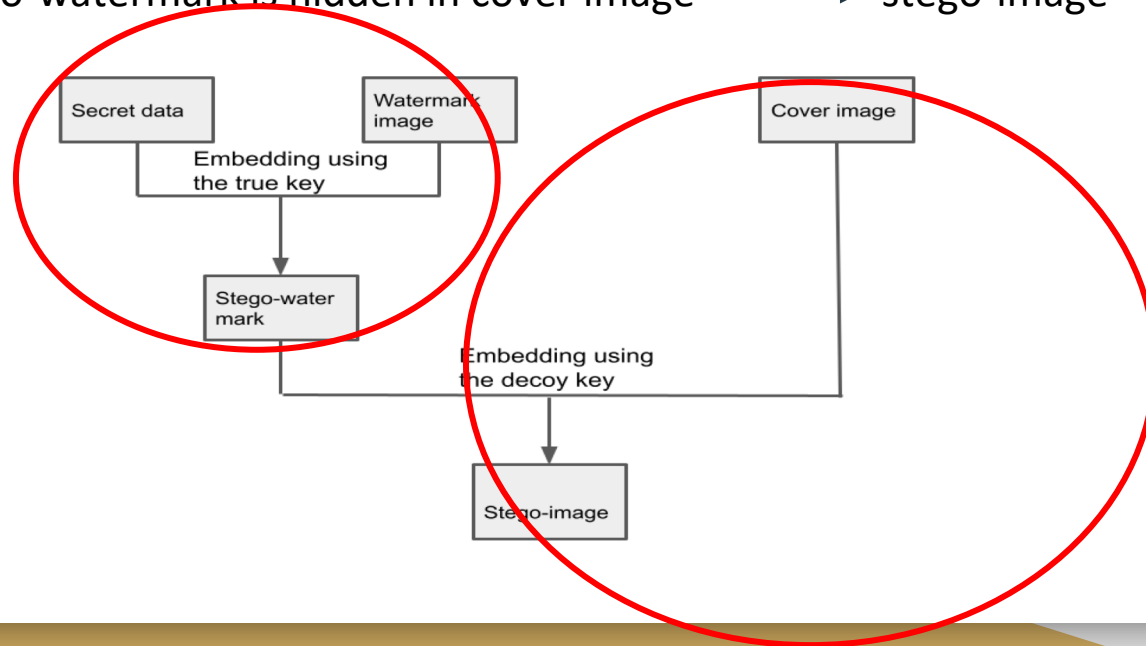
Name	Platform
TrueCrypt	Desktop computer
Veracrypt	Desktop computer
StegFS (Pang et al., 2003)	Desktop computer
EDS	Smartphone
Mobiflage (Skillen et al., 2013)	Smartphone
MobiHydra (Yu et al., 2014)	Smartphone
MobiGyges (Feng et al., 2020)	Smartphone
-	Wearable device

Adversarial Model

- The adversary is rational and will stop coercing the victim once convinced that the decryption keys have been disclosed
- The adversary cannot capture a victim at the point that he/she is working in the PDE mode
- We do not handle code security and therefore, the system itself should be malware-free
- The adversary is able to obtain both the original cover image and the stego-image. The adversary is **not** able to obtain the original watermark image
- We do not consider other attacks such as image cutting or cropping

Design Overview

- Using image steganography (LSB) to achieve PDE
 - Wearable devices often have limited computing resource
 - Do not need to explain why filling randomness initially
- Secret data are hidden in watermark \longrightarrow stego-watermark
- Stego-watermark is hidden in cover image \longrightarrow stego-image



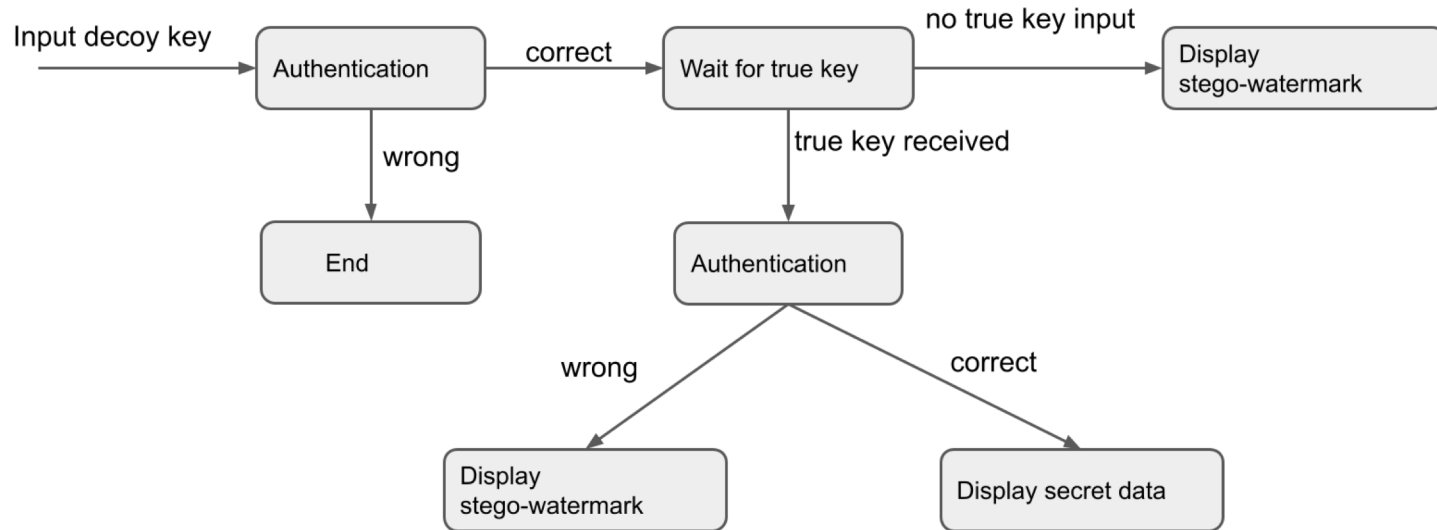
Design Details

Input of keys:

- Due to the small size of a wearable mobile device, using a keyboard or a touchscreen to input keys is inconvenient
- We use gyroscope sensor to input the keys
 - Convenient for users to enter the key (e.g., rotate the wrist)
 - Different rotation degrees represent different keys

Design Details (cont.)

User authentication:



Design Details (cont.)

Data hiding:

- Hiding secret data

- Sensitive data are encrypted by the **true key**
- Encrypted sensitive data are hidden to the watermark → Stego-watermark

- Hiding stego-watermark

- Stego-watermark is encrypted by the **decoy key**
- Stego-watermark is then embedded to the cover image → Stego-image

Evaluation

Evaluation

- MobiWear is implemented in a LG G watch
 - 512MB RAM, 4GB storage
 - OS: Android Wear 2.0
- Peak signal-to-noise-ratio (PSNR)
- Processing time

Evaluation (cont.)

PSNR: Represents the ratio between the maximum possible power of a signal and the power of the noise. Higher PSNR value indicates good image quality

$$PSNR = 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE)$$

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

m,n: the size of the original image is mxn

I: original image

K: noisy approximation of original image

MAX_I: maximum pixel value of the image

Evaluation (cont.)

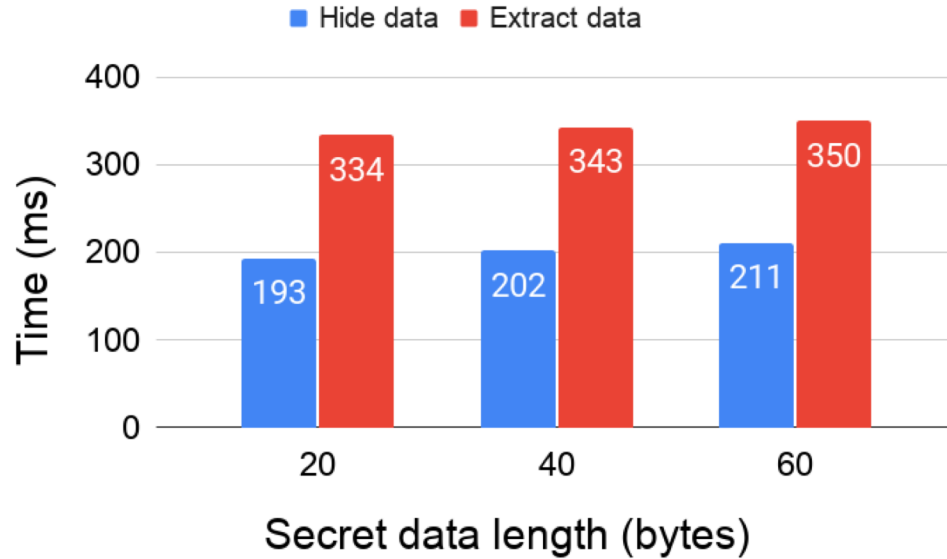
Length of secret data (bytes)	20	40	60
PSNR	30.6257	30.6153	30.5500

PSNR of stego-images under different lengths of secret data

Length of secret data (bytes)	20	40	60
PSNR	33.8356	33.7800	30.7600

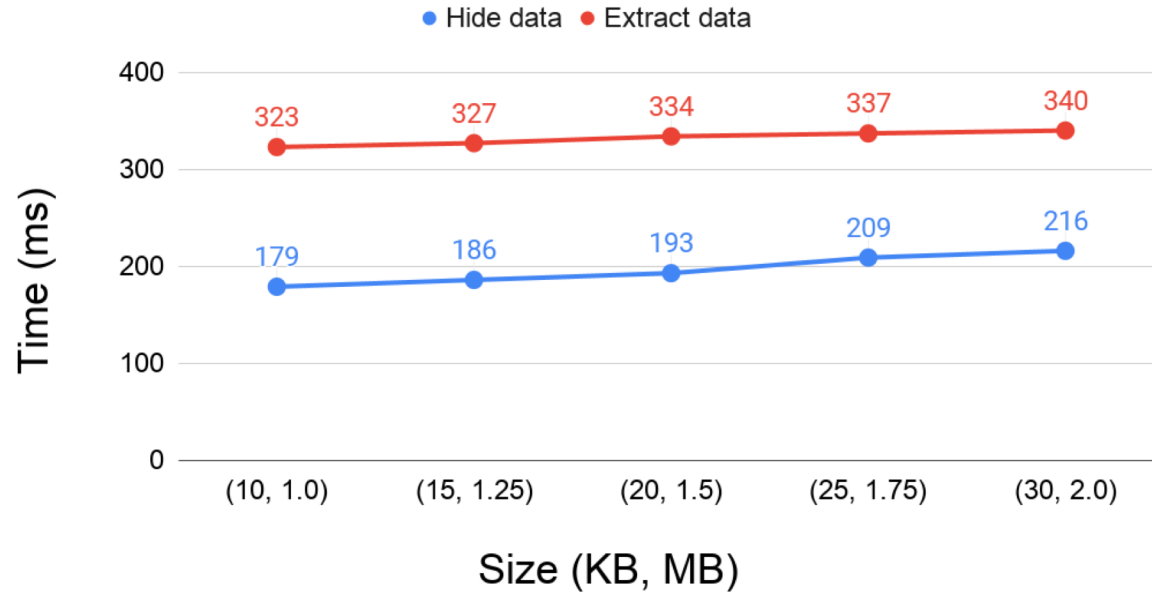
PSNR of stego-watermarks under different lengths of secret data

Evaluation (cont.)



Computational time for hiding and extracting sensitive data under different length of secret data

Evaluation (cont.)



Computational time for hiding and extracting sensitive data under different size of watermark and cover image. (a,b) represents (watermark size, cover image size)

Discussion

- Length of sensitive data which can be hidden
 - Suppose cover image is N pixels, each pixel consists of 4 bytes (ARGB)
 - Maximal size of watermark: $N/2$ bytes
 - Maximal length of secret data: $N/16$ bytes
 - Using more LSB to hide sensitive data
- Deniability compromise in memory
 - Secret data may leave traces in memory
 - Power-off the device
 - Utilizing the hardware isolation technique (e.g., ARM TrustZone)

Discussion (cont.)

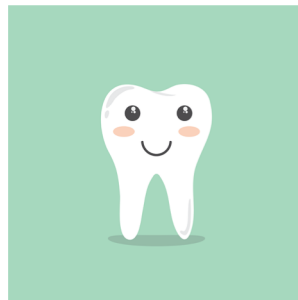
- Defend against multi-snapshot adversary
 - Deniability is compromised if attacker can have multiple access to the device
 - Each time when sensitive data are modified, new version of data will be embedded into a new cover image and watermark
- Mitigating data corruptions
 - LSB technique is vulnerable to image cutting and cropping attack
 - Back up the data periodically

Demo

Cover image



Watermark

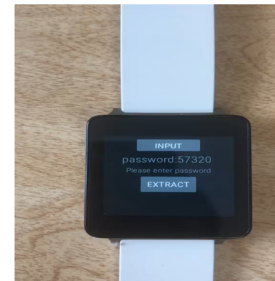
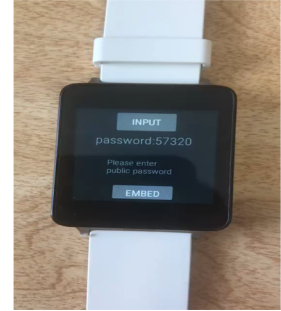


Secret message: hello

Hide secret data:

Extract secret data:

Upon being coerced:



Thanks