

# CSCE315 Final Project Presentation

Sam Gwydir Chris Findeisen Martin Fracker Rafael Moreno Kyle  
Wilson

*<2015-05-10 Sun>*

# Outline

- 1 DickGrayson
- 2 Design
- 3 Demonstration
- 4 Conclusion
- 5 References

In the final project students were tasked with creating a collection of tools that allow a user to encrypt and decrypt messages using the RSA encryption algorithm and embed and extract messages (either plaintext or ciphertext) from BMP images and WAV audio files.

# DickGrayson

## Introduction

*DickGrayson* is a collection of tools that allow a user to encrypt and decrypt messages using the RSA encryption algorithm and embed and extract messages (either plaintext or ciphertext) from BMP images and WAV audio files.

## Tools

munchkincrypt (aka rsa-crypt) RSA Encryption

dorothy (aka rsa-attack) RSA Attacks

munchkinsteg (aka stego-crypt) Steganography

toto (aka stego-attack) Steganography Attacks

# Design Decisions

## DickGrayson

Target OS Linux x86\_64 (`build.tamu.edu`)

Compiler GCC 4.9.2

Language C++14

Build System CMake

Numerics GNU Multiple Precision Library

Unit Testing Google Test

Continuous Integration `travis-ci`

Code Coverage `coveralls`

# Implementation Decisions

## RSA Encryption

- Especially, explain what attacks you chose to make for both RSA and stego, and how your two stego schemes work
- Show proofs of Test-Driven Development (screenshots of failing / passing tests, GitHub revision history, running test code in the demo, etc)

## RSA Attacks

- Attacked using one general purpose attack, which would work on any rsa key, then created two more specific attacks that are efficient in some cases.

# Implementation Decisions (cont.)

Steganography

Steganography Attacks

# Build System & Tools

- Travis
- Coveralls



# rsa-crypt

- generate primes
- generate public and private keys
- encrypt and decrypt example messages

# rsa-attack

- factorization
- common modulus
- low exponent

# stego-crypt

## How it works

Our tool munchkinsteg supports image and audio least significant bit (LSB) steganography. The formats supported are Windows 8-bit BMP and PCM 16-bit WAV.

## Embedding message

The data is broken up into bits and each bit is stored consecutively in the LSB of the individual pixels and samples of the bmp and wav file respectively. A null byte is stored using 8 additional pixels/samples to be used during extraction as a termination symbol.

# stego-crypt (cont.)

## Extracting message

The LSB of the individual pixels and samples of the bmp and wav file respectively are extracted and concatenated into a string. This process terminates once a null byte is reached.

# stego-attack

## ■ Detection

## ■ Successes

# Links

GCC 4.9.2 <https://gcc.gnu.org>

GNU Multiple Precision Library <https://gmplib.org>

Google Test <https://code.google.com/p/googletest/>

travis-ci <https://travis-ci.org>

coveralls <https://coveralls.io>