Cry – Project 2
(Software Requirements Specification): Presentation

2017-02-13

Daniel Dunning, Michael Degraw, Vu Phan

Abstract

- `Cry` is a cryptoframework targeted at cryptographers to allow them to easily benchmark their new cryptosystems
- It will also allow end-users to send and receive messages using any cryptosystem provided to `Cry`

Contents

Section 1

Introduction

Subsection 1

Scope

Scope

- `Cry` will allow cryptographers to quickly develop new cryptosystems
  - It will do so by making testing and benchmarking easier
- `Cry` will also allow the encryption/decryption of data

Subsection 2

Definitions, Acronyms, and Abbreviations

Definitions

- `Cry`: the cryptoframework under development
- `Team Crybabies`: the team responsible for the development of `Cry`
- Cryptographers: the target audience of `Cry`

Subsection 3

References

References

- **GMP (GNU Multiple Precision arithmetic library)**: `https://gmplib.org/`
- **Msieve (General Number Field Sieve integer factorization library)**: `https://github.com/radii/msieve`

Section 2

Overall Description

Subsection 1

Product Perspective

Product Perspective

- `Cry` will be implemented as a stand-alone framework, with built-in cryptosystems updated as needed
- User interface will start as command-line-based (possibility of implementing a GUI)

Subsection 2

Product Functions

Product Functions

Testing

- Develop new cryptosystems
- Test cryptosystems against cracking techniques and generate helpful output

Reporting

- Upon performing a test, a user will receive a report on the cryptosystem
- The report will indicate the security level of the cryptosystem

Subsection 3

User Characteristics

User Characteristics

Users will most likely have a medium to high level of experience with cryptosystems

Subsection 4

Constraints

Constraints

- Basic memory and CPU availability
- Further library implementations or updates may require parallel operation or interfacing with other applications

Subsection 5

Assumptions and Dependencies

Assumptions and Dependencies

The only assumption of `Cry` is that it has applicable administrative permissions at the command line

Section 3

Specific Requirements

Subsection 1

Interface

- `Alice` wants to send a confidential message to `Bob`
- `Eve` wants to eavesdrop that message
- These end-users invoke their downloaded `Cry` binaries using command-line shells

Subsection 2

Performance

Minimum hardware

RAM	4 GB
CPU	1.5 GHz

Key Generation by Bob

Input:

```
$ cry generatekeys -cryptosystem=<cryptosystem>
```

Output:

```
The public & private keys are <public key> & <private key>
(took <key-generation time>).
```

Requirements:

- <key-generation time> shall be less than 1 minute

Encryption by Alice

Input:

```
$ cry encrypt -cryptosystem=<cryptosystem> \
> -publickey=<public key> -plaintext=<plaintext>
```

Output:

```
The ciphertext is <ciphertext> (took <encryption time>).
```

Requirements:

- <plaintext> is an obviously meaningful string, such as ‘‘Eve is just a crybaby.’’
- <ciphertext> is an apparently meaningless string, such as ‘‘sdofAOVI29347asdjkADB234’’
- <encryption time> shall be less than 1 minute

Decryption by Bob

Input:

```
$ cry decrypt -cryptosystem=<cryptosystem> \
> -privatekey=<private key> -ciphertext=<ciphertext>
```

Output:

```
The plaintext is <plaintext> (took <decryption time>).
```

Requirements:

- `<decryption time>` shall be less than 1 minute

Cryptanalysis by Eve

Input:

```
$ cry cryptanalyze -cryptosystem=<cryptosystem> \
> -publickey=<public key> -ciphertext=<ciphertext>
```

Output:

```
The plaintext is <plaintext> (took <cryptanalysis time>).
```

Requirements:

- `<cryptanalysis time>` shall be more than 1 day

Subsection 3

Classes

cryptosystem.h

```
using IntPtr = mpz_t; // GNU Multiple Precision Integer Type
using Key = IntPtr;
using Text = IntPtr;
```

```
class Cryptosystem {
public:
  virtual void generateKeys(Key publicKey, Key privateKey);
    // set these

  virtual void encrypt(Text ciphertext, // set this
    const Text plaintext, const Key publicKey);

  virtual void decrypt(Text plaintext, // set this
    const Text ciphertext, const Key privateKey);

  virtual void cryptanalyze(Text plaintext, // set this
    const Text ciphertext, const Key publicKey);
};
```

Section 4

Interview Log

Subsection 1

Summary

Summary

The interviewees want:

- the cryptosystem reports to be as detailed as possible
- addition of other cryptographic algorithms, especially *AES*
- addition of a good pseudo-random number generator
- inclusion of certain libraries, like `msieve` for integer factorization

Section 5

Conclusion

Q & A

https://github.com/vuphan314/cry