**Projects in Electrical & Computer Engineering 2007-2008**

# Key Generation For Sensitive But Unclassified Communication

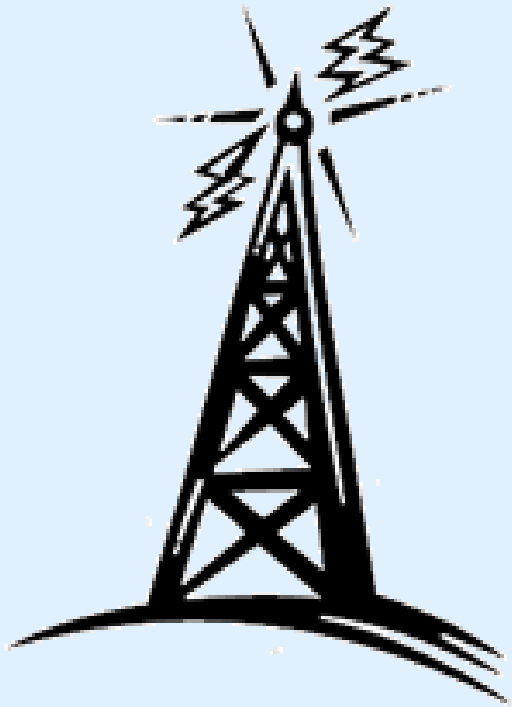

1/c Rebecca Doyle and 1/c Josh Gaidos

Project Advisor: Mr. Herb Holland

Sponsor: LT Lars McCarter, CG-6

# Outline

- Introduction
- Background
- Objectives
- Technical Approach
- Results
- Conclusion



Tower image: http://www.nationalcynical.com/images/radio_tower.gif

# Introduction

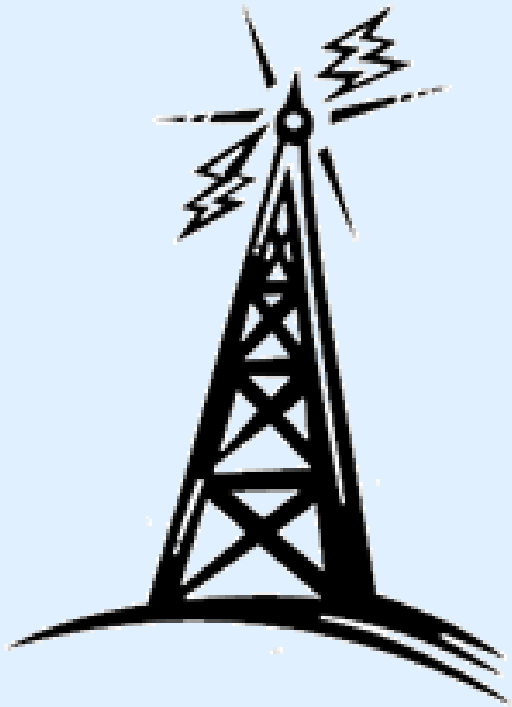
- What – Key generator for encryption
- Why – Two communities in need
  - Encrypted AIS
  - Small Boat comms

# Intro. (cont)

- Impact on the Coast Guard
  - OPSEC
  - Privacy
  - federal standards
- Relevance – applicable fleet-wide

# SBU Communication

- What is it?
- Why encrypt it?
- What do we need?



# Background

- Physical security analogy
- Random Number Generator (RNG)
- National Institute for Standards and Technology (NIST)
- Federal Information Processing Standards (FIPS) 140-2

# Background (cont.)

- Advanced Encryption Standard (AES)
- Current Hardware
- Current Software

NIST Logo: http://www.nist.gov/

# Objectives

Create a program that uses Graphical User Interface (GUI)

- Generates random key
- User-specified length
- Encrypts/Decrypts the key
- Key is output to:
  - GUI for copy/paste
  - Text file for saving/transmission

# Technical Approach

- Acquire the RNG module
- Write a GUI to interface with the module
- Test and export

# Automatic Key Generation Toolkit

Key Size 128

Select Key Size
Between 64 and 4096

Generate Key

Copy and
Paste Me
Into File

Generate Text File

Key Generation Toolkit

# Key Generation Toolkit

About

Generate the Key | Decrypt the Key

Key Length

256

Generate Key

Make Text File

Key for Transmission

E85CB02CE42818944CEE70504BC28EEB
BD957259EBB7F91C434BEAB330943AA6

Encryption Key

CEC9A05E60190A63F1EC2205581FB77E

Initialization Value

9649C47515683041BCB3F08C93F5952E

Key Generation Toolkit

# Key Generation Toolkit

About

Generate the Key | Decrypt the Key

Read From Text File

Input By Hand

Key Length

256

Read Key From Text File

Decrypt Key for Device Input

Make Text File

Encrypted Key or Key For Device Input

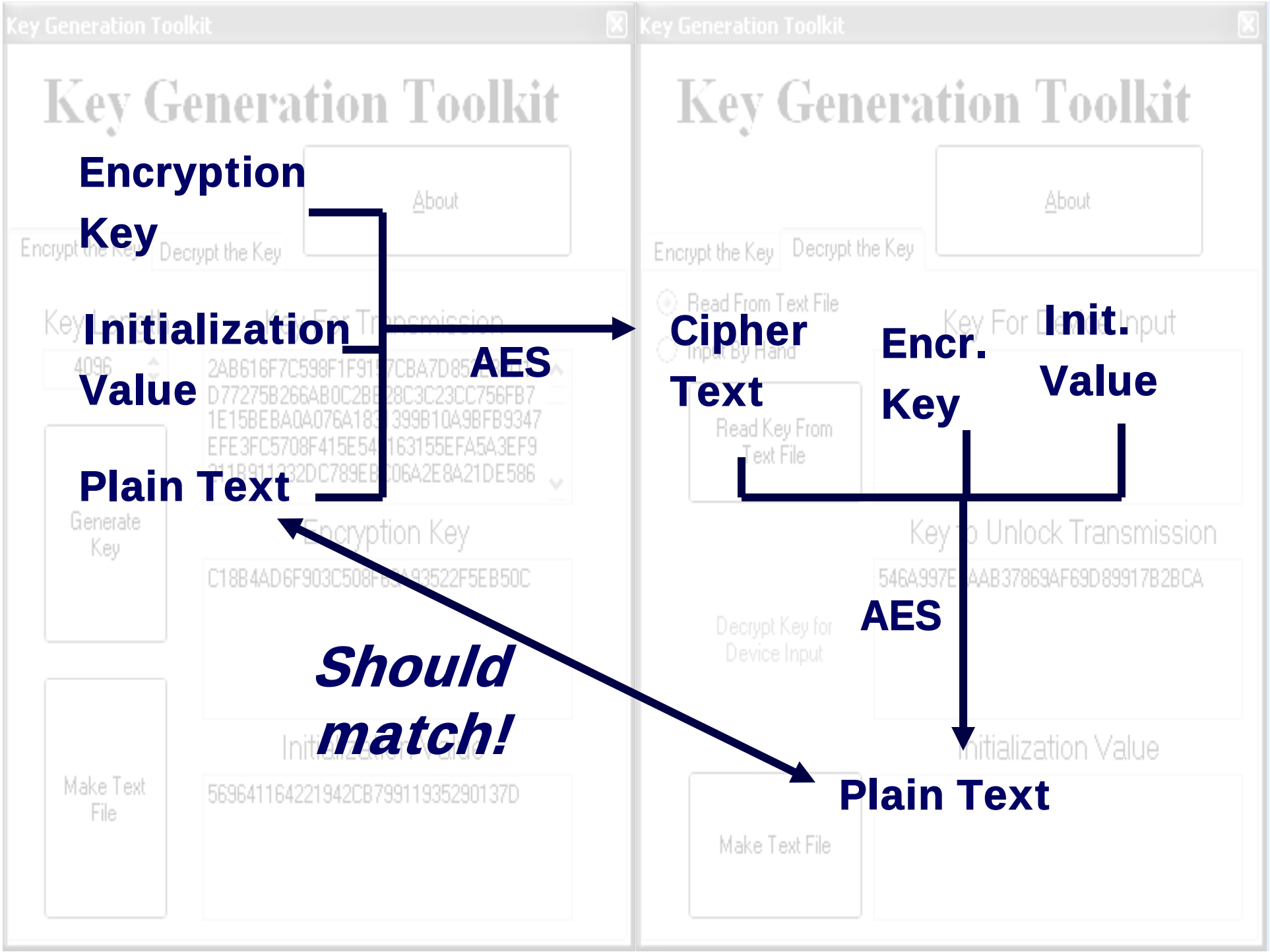
0F54ECE97E898B1D70CA56FC79356F41E
18E1D3E1013D85B9C683AA638B61C17

Key to Unlock Transmission

CEC9A05E60190A63F1EC2205581FB77E

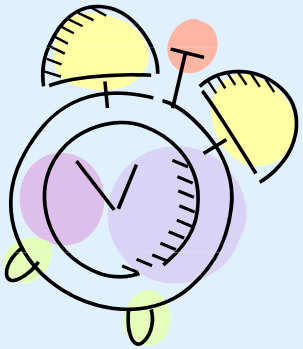
Initialization Value

9649C47515683041BCB3F08C93F5952E


Key Generation Toolkit
Key Generation Toolkit
Encryption Key
Initialization Value
Plain Text
AES
Cipher Text
Encr. Key
Init. Value
AES
Should match!
Plain Text
About
Encrypt the Key
Decrypt the Key
Read From Text File
Read Key From Text File
Generate Key
Make Text File
Decrypt Key for Device Input
Key To Unlock Transmission
Encryption Key
Initialization Value
Make Text File

# Results

**Three Stages**



1: Early Success

2: Expansion

3: Contraction

# Stage 1: Early Success

- Found CryptoSys API™
- Appeared to meet all requirements
- Generated random keys in GUI
- Fully working in about 1 month

# Stage 2: Expansion

**Requested:**

- Encrypt/Decrypt keys

**Delivered:**

- Chose AES128
- Fully working in about 1 week

# Stage 3: Contraction

**Problem:**

- Discovered CryptoSys API™ was “written to comply with FIPS 140-2”
- Complete programs have to be validated, not just component modules

**Solution:**

- Working deliverable -> Working certification submission

# Conclusions

**Summary**

- Need good key generation
- NIST and FIPS 140-2
- CryptoSys API™
- CG-6 will submit for certification testing