Team ISE

Image Selective Encryption

Team ISE

Image Selective Encryption

Joe Jarchow

Joseph Kadhim

Geoffrey Griffith

Shinya Daigaku

Andrew Pouzeshi

Presentation Overview:

- Statement of problem
- Initial research into compressed files
- Target Selection Process
- JPEG Statistical Analysis
- JPEG Manipulator Design
- JPEG Manipulator Demonstration
- Encryption Algorithm Selection
- JPEG Selective Encryption Algorithms
- ISE Production Code Design
- ISE Web Site Design
- Future Considerations

Presentation Overview:

- **Statement of problem**
- **Initial research into compressed files**
- **Target Selection Process**
- **JPEG Statistical Analysis**
- **JPEG Manipulator Design**
- **JPEG Manipulator Demonstration**
- **Encryption Algorithm Selection**
- **JPEG Selective Encryption Algorithms**
- **ISE Production Code Design**
- **ISE Web Site Design**
- **Future Considerations**

Problem:

- **Multimedia files are very large**
- **Encryption is expensive**
  - **Processing time**
  - **File size**
- **No widely accepted solutions**
  - **Encrypt entire file**
  - **No encryption**

Problem Statement

Affected User Scenarios:

- Images on websites
- File sharing
- Cable TV

Solution:

- **Selective Encryption**

Definition from MPEG paper:

Selective encryption applies encryption to a subset of a file with the expectation that the entire file will be rendered useless to anyone who cannot decrypt that subset.

Presentation Overview:

- Statement of problem
- Initial research into compressed files
- Target Selection Process
- JPEG Statistical Analysis
- JPEG Manipulator Design
- JPEG Manipulator Demonstration
- Encryption Algorithm Selection
- JPEG Selective Encryption Algorithms
- ISE Production Code Design
- ISE Web Site Design
- Future Considerations

Selective Encryption Requirements:

- Perceivable degradation of file
- Encryption of less than 10 %
- Minimize required computation
- Minimize increase in file size
- Cryptanalytic approach

Encryption of Compressed File Types:

- Independent of time (JPEG)
  - Must affect image related target
  - Can use a block or stream cipher

- Synchronous (MPEG)
  - Target could affect the image
  - Target could affect time components
  - Might require stream cipher

Structure of Compressed File Types:

- **Published international standards**
- **Partitioned into standard components**
  - ***Descriptive***
  - ***Mathematical***

JPEG Standard:

Standard implementation of JPEG compression

http://www.ijg.org

JPEG Structure:

- Markers, headers and data
- Example:

```
ff e0
00 10
4a 46 49 46 00 01 01 01 00 48 00 48 00 00
```

Marker:

- Indicates which component
- Example marker:

ff e0 (indicates Application Data)

Header:

- Indicates size of parameters to follow
- Example header:

00 10 -- (16 bytes of data will follow)

Data:

- **The information itself**
- **Example data:**

```
4a 46 49 46 00 01 01 01 00 48 00 48 00 00
```

(16 bytes of information indicating what application created the file.)

Encrypting *During* Compression:

- Would not produce standard file
- Requires reimplementation

Encrypting *After* Compression:

- Layered approach
- Creates intermediate file
  - Allows different extension
- Algorithm can be easily reviewed
- Applicable to non-synchronous files

General Development Approach:

- Study Compression Standard
- Study earlier approaches
- Create a testing toolkit
- Evaluate each target:
  - Percentage of file
  - Perceivable damage
- Design selective encryption algorithm
- Cryptanalytic approach

Cryptanalytic Approach:

- White hat
- Black hat
- Review by crypto community
- Correction of algorithm

Presentation Overview:

- Statement of problem
- Initial research into compressed files
- Target Selection Process
- JPEG Statistical Analysis
- JPEG Manipulator Design
- JPEG Manipulator Demonstration
- Encryption Algorithm Selection
- JPEG Selective Encryption Algorithms
- ISE Production Code Design
- ISE Web Site Design
- Future Considerations

Criteria For Bad Targets:

- Optional markers
- Not used in Baseline JPEG images
- Does not affect visibility of the image
- Easily guessed or forged by a hacker

Determining Initial Bad Targets:

- Resources:
  - JPEG Still Image Data Compression Standard
  - Compressed Image File Formats
  - ISO DIS 80918-1 Requirements and Guidelines
  - ISO DIS 80918-2 Compliance Testing
  - http://www.funducode.com/freec/fileformats/format3/format3b.htm

- APP - Application
  - No affect to visibility
- COM - Comments
  - No affect to visibility
- DAC - Define Arithmetic Conditioning Tables
  - Not part of Baseline Compression
- DHP - Define Hierarchical Progression
  - Not part of Baseline Compression
- DNL - Define Number of Lines
  - Easily forged (set size)

- DRI - Define Restart Interval
  - Easily forged (set size)
- EOI - End of Image
  - Easily forged (always last marker)
- EXP - Expand
  - Not part of Baseline Compression
- JPG - Reserved for Future Extensions
  - Not used in Baseline Compression

- RES - Reserved
  - Not used in Baseline Compression
- RST - Restart
  - Not part of Baseline Compression
- TEM - Temporary
  - Not used in Baseline Compression
- SOS - Start of Scan
  - Easily reconstructed
- Markers themselves are predictable

Remaining Targets for Selective Encryption:

- **Encoded Data Stream**
- **Quantizer Tables**
- **Huffman Tables**

Presentation Overview:

- Statement of problem
- Initial research into compressed files
- Target Selection Process
- **JPEG Statistical Analysis**
- JPEG Manipulator Design
- JPEG Manipulator Demonstration
- Encryption Algorithm Selection
- JPEG Selective Encryption Algorithms
- ISE Production Code Design
- ISE Web Site Design
- Future Considerations

JPEG Target Statistical Analysis:

- **Target Analysis Toolkit**
  - **Convert**
  - **Analyze**
  - **Manipulator**

Convert:

- C++ program
- Convert Binary to Hexadecimal
- File information for a single JPEG image

```
This is an ASCII representation (in hexadecimal) of the binary values found in the file
: Dust.jpg
Markers Found:================
ff d8 -- Start of Image
ff e0 -- Application Data -- 00 10 -- (16 bytes) -- 4a 46 49 46 00 01 01 01 00 48 00 48
00 00
ff db -- Define Quantization Table -- 00 43 -- (67 bytes) -- 00 06 04 05 06 05 04 06 06
05 06 07 07 06 08 0a 10 0a 0a 09 09 0a 14 0e 0f 0c 10 17 14 18 18 17 14 16 16 1a
1d 25 1f 1a 1b 23 1c 16 16 20 2c 20 23 26 27 29 2a 29 19 1f 2d 30 2d 28 30 25 28
29 28
ff db -- Define Quantization Table -- 00 43 -- (67 bytes) -- 01 07 07 07 0a 08 0a 13 0a
0a 13 28 1a 16 1a 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28
28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28
28 28
ff c0 -- Huffman Table -- Baseline DCT -- 00 11 -- (17 bytes) -- 08 01 cb 02 4a 03 01
22 00 02 11 01 03 11 01
ff c4 -- Huffman Table -- 00 1f -- (31 bytes) -- 00 00 01 05 01 01 01 01 01 01 00 00
00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 0a 0b
```

Statistical Analysis

Analyzer:

- File information for multiple JPEG’s
  - Average file size
  - Average number of Huffman tables
  - Average size of Huffman tables
  - Average number of Quantizer tables
  - Average size of Quantizer tables
  - Average size of the encoded stream
  - Average number of markers
  - Number of files processed

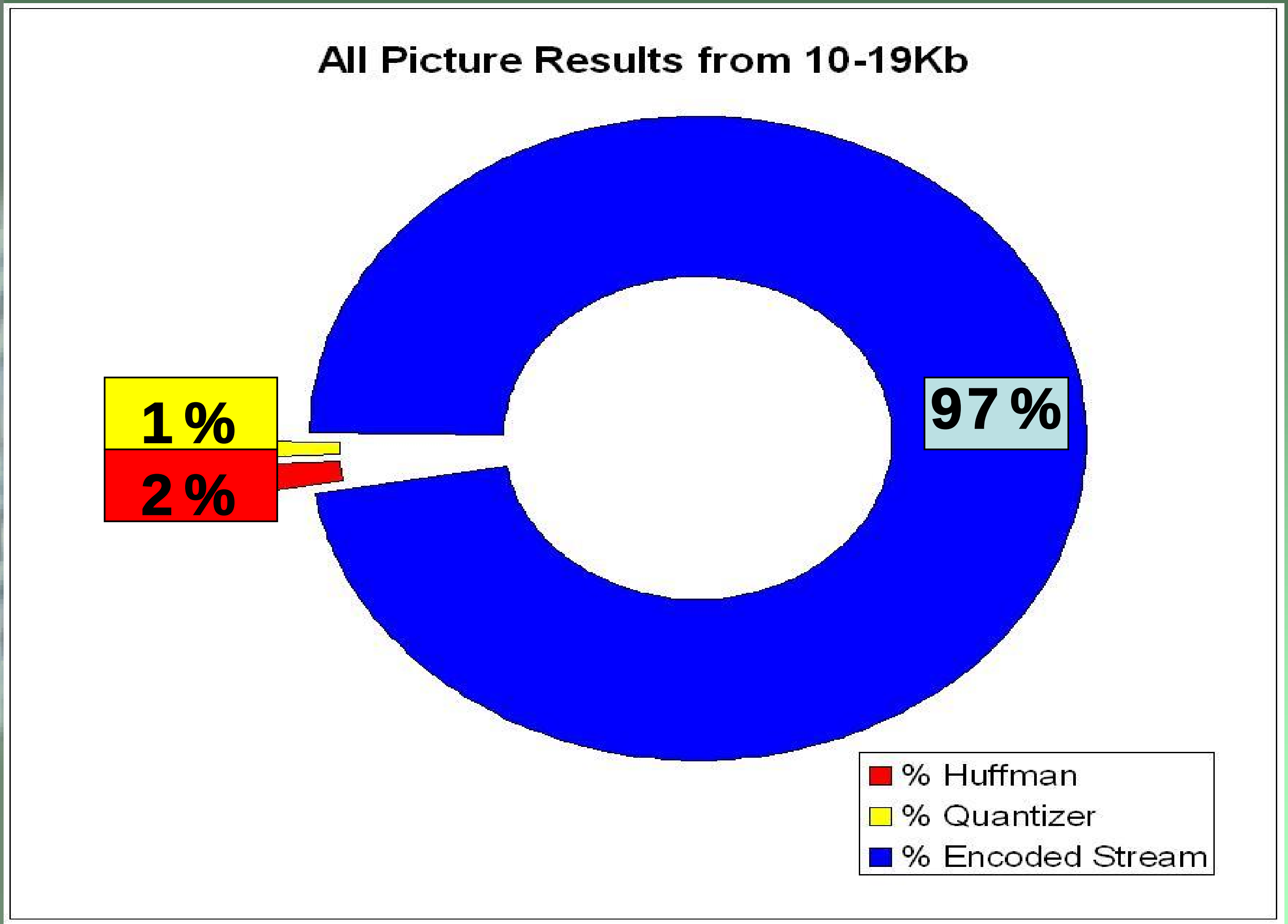
Statistical Analysis

Analyzer (cont):

- Percent of the file dedicated to:
  - Huffman tables
  - Quantizer tables
  - Encoded Stream

Test Cases for JPEG Analysis:

- Over 2500 JPEG images selected
  - Internet web sites
  - Digital photographs
  - Manmade images
- Size ranges:
  - 10-19KB, 100 KB, 1 MB, and larger
- Resolution Ranges:
  - 320x240, 640x480, and 800x640 pixels



Statistical Analysis

Joseph K

Encoded Data Stream:

- SOI (Start of Image) marker
- Compressed data stream
- Takes up a large portion of the file
- Averaged 90% of the file!

Quantizer Tables:

- DQT (Define Quantization Table) markers
- Defines Resolution
  - Luminance
  - Chrominance
- Averaged 0.88% of the file
- Unpredictable affects on image
- Might not visually damage the image!
- Can be replaced with another Quantizer!

Huffman Tables:

- DHT (Define Huffman Table) markers
- Used to encode/decode the image data
- Averaged 1.84% of the file
- Considerable damage to image
- Mathematically derived from the image
- This makes the Huffman Tables a perfect target for Selective Encryption

Presentation Overview:

- Statement of problem
- Initial research into compressed files
- Target Selection Process
- JPEG Statistical Analysis
- JPEG Manipulator Design
- JPEG Manipulator Demonstration
- Encryption Algorithm Selection
- JPEG Selective Encryption Algorithms
- ISE Production Code Design
- ISE Web Site Design
- Future Considerations

Requirements:

- **Testing tool**
- **Graphical user interface**
- **Displays each component**
- **Easy manipulation of JPEG files**
- **See changes side by side**

Modules:

- **Standard Windows methods**
- **Graphical User Interface**
- **Common methods**
- **Convert binary to ASCII**
- **Convert ASCII to binary**
- **Encrypt and Decrypt methods**

Standard Windows Methods:

- Required functions like main()
- Initialization functions
- Constructors and Destructors

Graphical User Interface:

- **Methods called during user interaction**
- **Event handlers**
  - **menus**
  - **buttons**
  - **text boxes**

Common Methods:

- **Create / Load / Save**
  - **project(s)**
  - **picture(s)**
- **Show warning(s)**
- **Clear interface data**
- **Updated manipulated picture**

Convert Binary to ASCII:

Convert ASCII to binary

- Methods to load images to interface
- Create images from interface

Encrypt and Decrypt methods

- Calls production code methods

Presentation Overview:

- Statement of problem
- Initial research into compressed files
- Target Selection Process
- JPEG Statistical Analysis
- JPEG Manipulator Design
- **JPEG Manipulator Demonstration**
- Encryption Algorithm Selection
- JPEG Selective Encryption Algorithms
- ISE Production Code Design
- ISE Web Site Design
- Future Considerations

JPEG Selective Encryption:

- Remove application data
- Remove comment data
- Leave initial Huffman marker
- Encrypt:
  - Huffman data (except initial marker)
  - Next non-Huffman marker and header

Presentation Overview:

- Statement of problem
- Initial research into compressed files
- Target Selection Process
- JPEG Statistical Analysis
- JPEG Manipulator Design
- JPEG Manipulator Demonstration
- Encryption Algorithm Selection
- JPEG Selective Encryption Algorithms
- ISE Production Code Design
- ISE Web Site Design
- Future Considerations

Requirements:

- Secure
- No increase in file size
- Recommendation from Prof. John Black

AES (Rijndael):

- **NIST selection of AES standard**
- **Block Cipher**
- **Rijmen and Daemen**
- **Open source optimized implementation**
- **Variable block length (128, 192, 256)**
- **Only whole byte operations**

Presentation Overview:

- Statement of problem
- Initial research into compressed files
- Target Selection Process
- JPEG Statistical Analysis
- JPEG Manipulator Design
- JPEG Manipulator Demonstration
- Encryption Algorithm Selection
- **JPEG Selective Encryption Algorithms**
- ISE Production Code Design
- ISE Web Site Design
- Future Considerations

Encryption Algorithm:

- Write file-type-byte to “.ise” file
  - ‘1’ for JPEG
- Read from input file
- Write unencrypted to “.ise” file

Remove App and Comment Data:

JPEG File

MARKER	HEADER	DATA
FFD8	NA	NA
FFE0	~~00 10~~	~~4A 46 49 ..~~
FFFE	~~4D 11~~	~~8C A2 12 ..~~

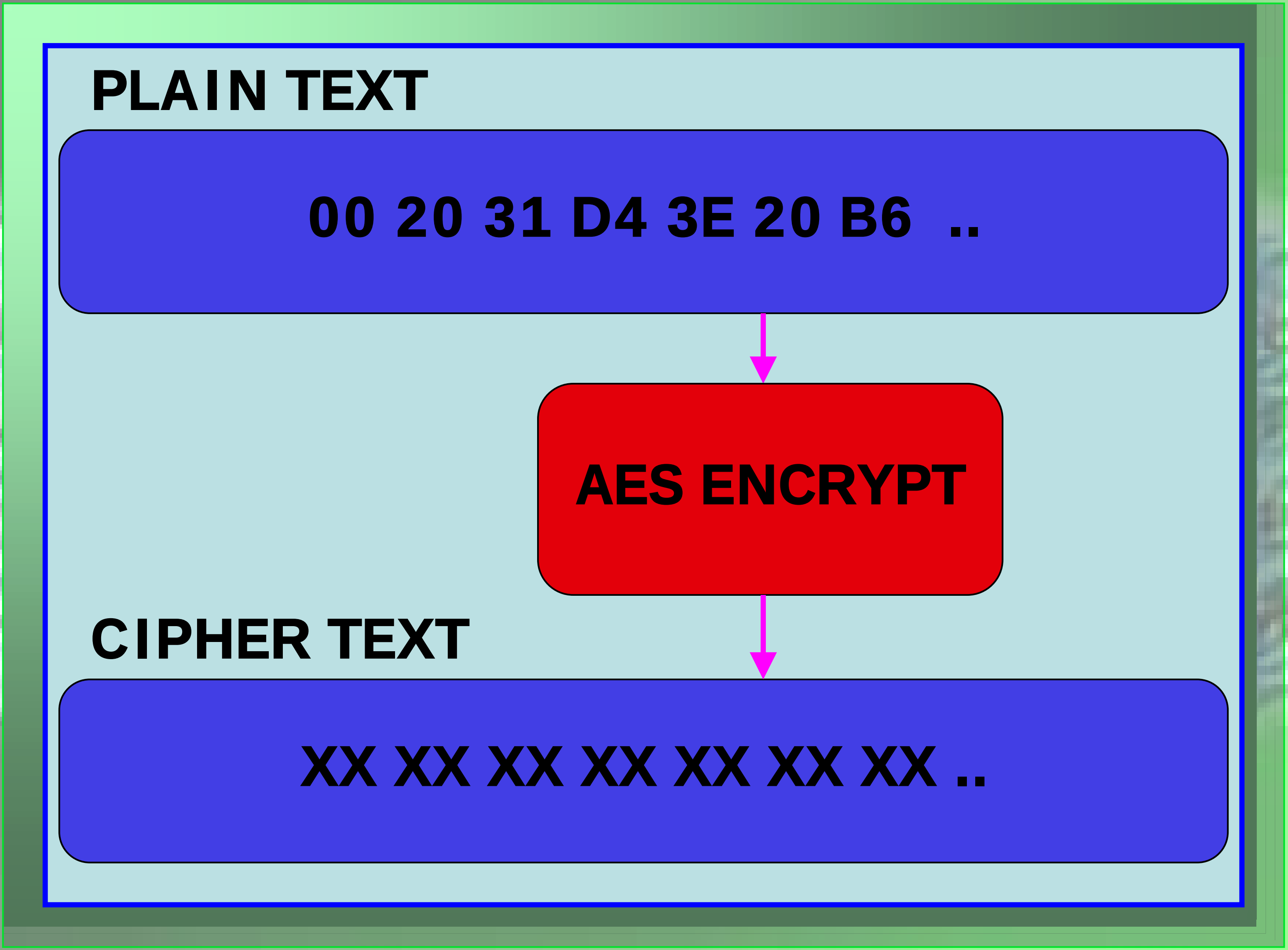
Encryption Algorithm (cont):

- Read / Write until marker [ffc0 - ffcf]
  - Indicates Huffman specification
    - ffc0 -- baseline frame
    - ffc4 -- Huffman table

Start Encrypting After FFC0:

JPEG file in hexadecimal

MARKER	HEADER	DATA
FFC0	Encrypted from here →	
	00 11	08 01 CB ..
FFC4	00 1F	00 00 01 ..



PLAIN TEXT

00 20 31 D4 3E 20 B6 ..

AES ENCRYPT

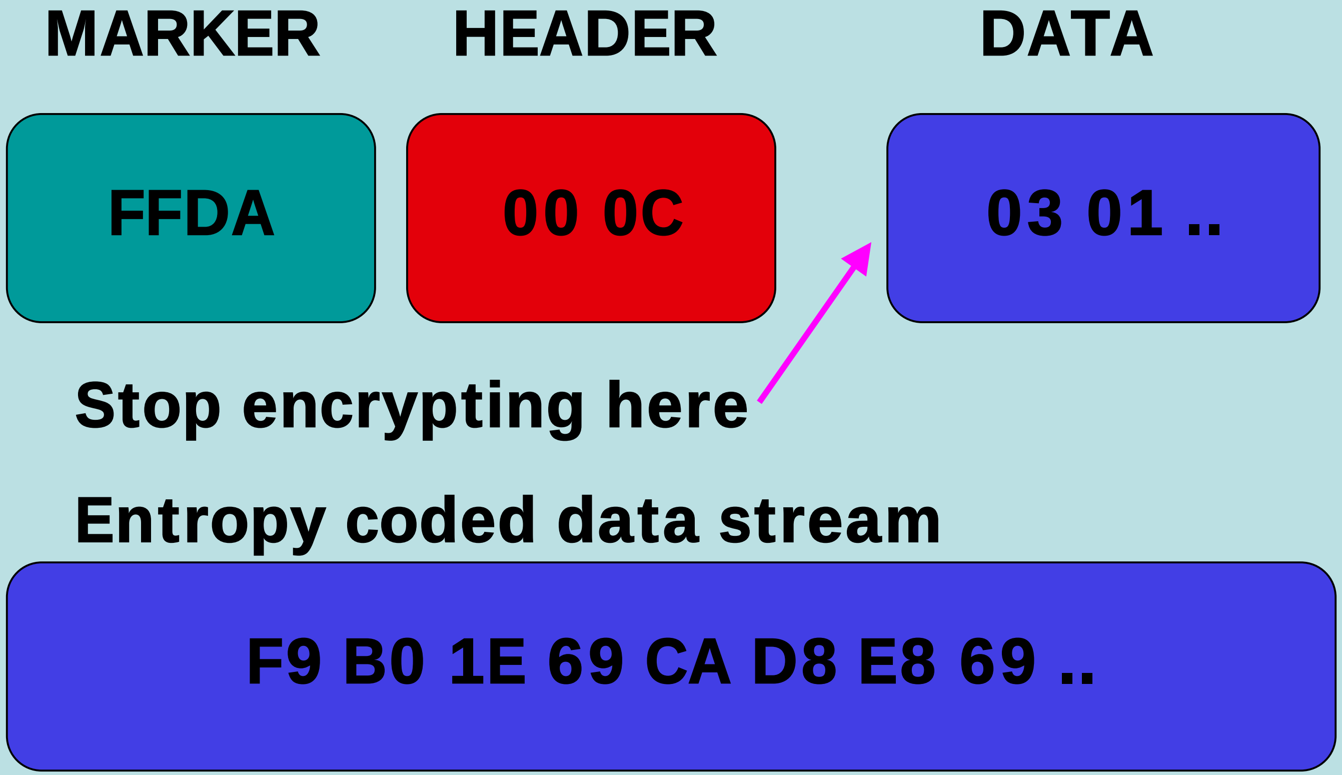
CIPHER TEXT

XX XX XX XX XX XX XX ..

Encryption Algorithm (cont):

- Write until non-Huffman marker
    - Below ffc0
    - Above ffcf

JPEG file in hexadecimal



MARKER

HEADER

DATA

FFDA

00 0C

03 01 ..

Stop encrypting here

Entropy coded data stream

F9 B0 1E 69 CA D8 E8 69 ..

Encryption Algorithm (cont):

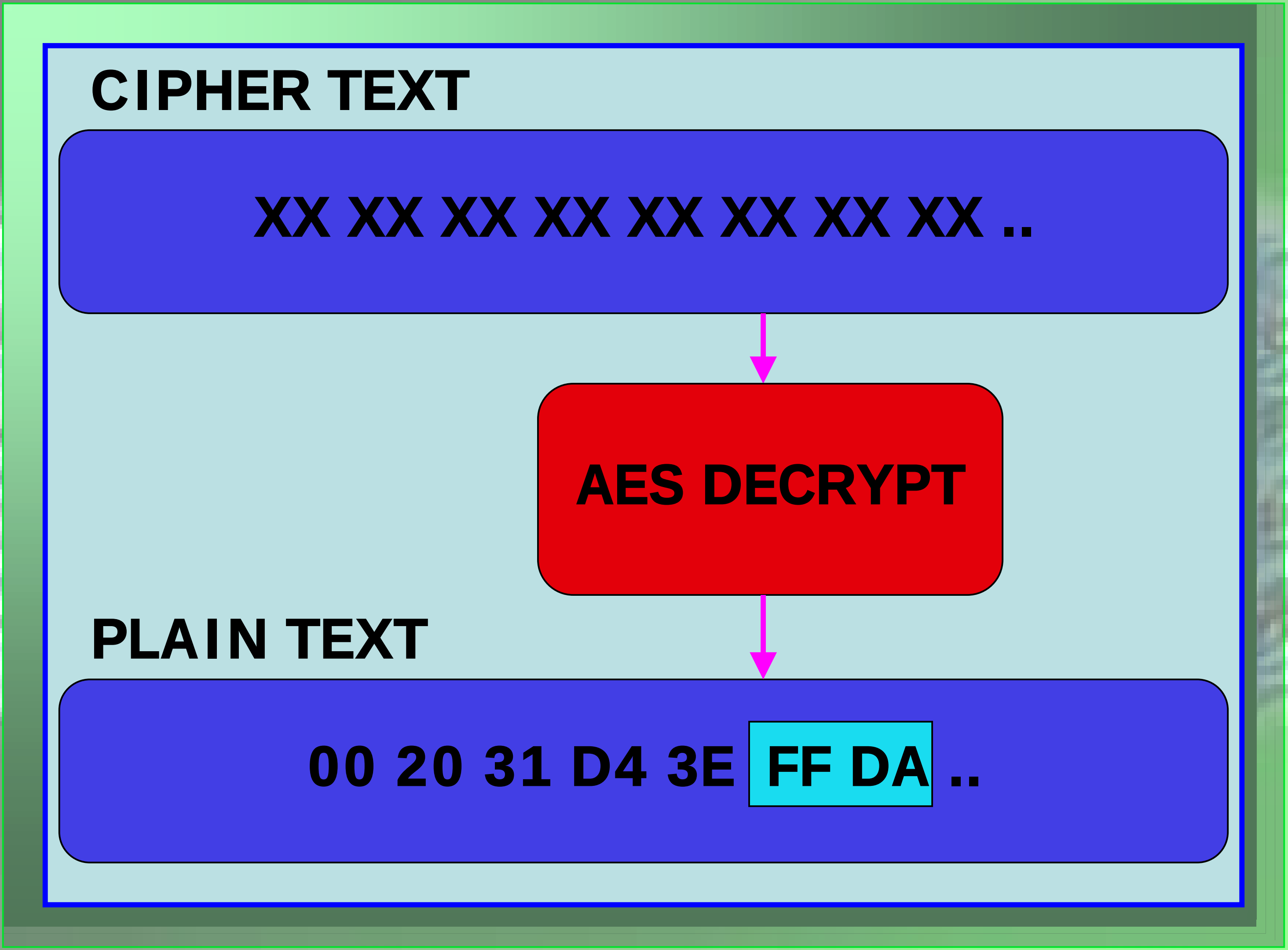
- Read/Write unencrypted
  - Until end of file (ffd9)
  - Unless another Huffman marker
- Efficiency
  - 97% evaluated by only a few if statements

Decryption Algorithm:

- Read file-type-byte from “.ise” file
  - ‘1’ for JPEG
- Read/Write until marker [ffc0 - ffcf]
  - Indicates start of encrypted data

ISE file in hexadecimal

MARKER	HEADER	DATA
FFC0	XX XX	XX XX ..
XX XX	XX XX	XX XX ..



CIPHER TEXT

XX XX XX XX XX XX XX XX ..

AES DECRYPT

PLAIN TEXT

00 20 31 D4 3E FF DA ..

Decryption Algorithm (cont):

- Write decrypted text to output file
- Read / Write unencrypted
    - Until end of file
    - Unless another Huffman marker

Presentation Overview:

- Statement of problem
- Initial research into compressed files
- Target Selection Process
- JPEG Statistical Analysis
- JPEG Manipulator Design
- JPEG Manipulator Demonstration
- Encryption Algorithm Selection
- JPEG Selective Encryption Algorithms
- ISE Production Code Design
- ISE Web Site Design
- Future Considerations

Object Oriented Outline:

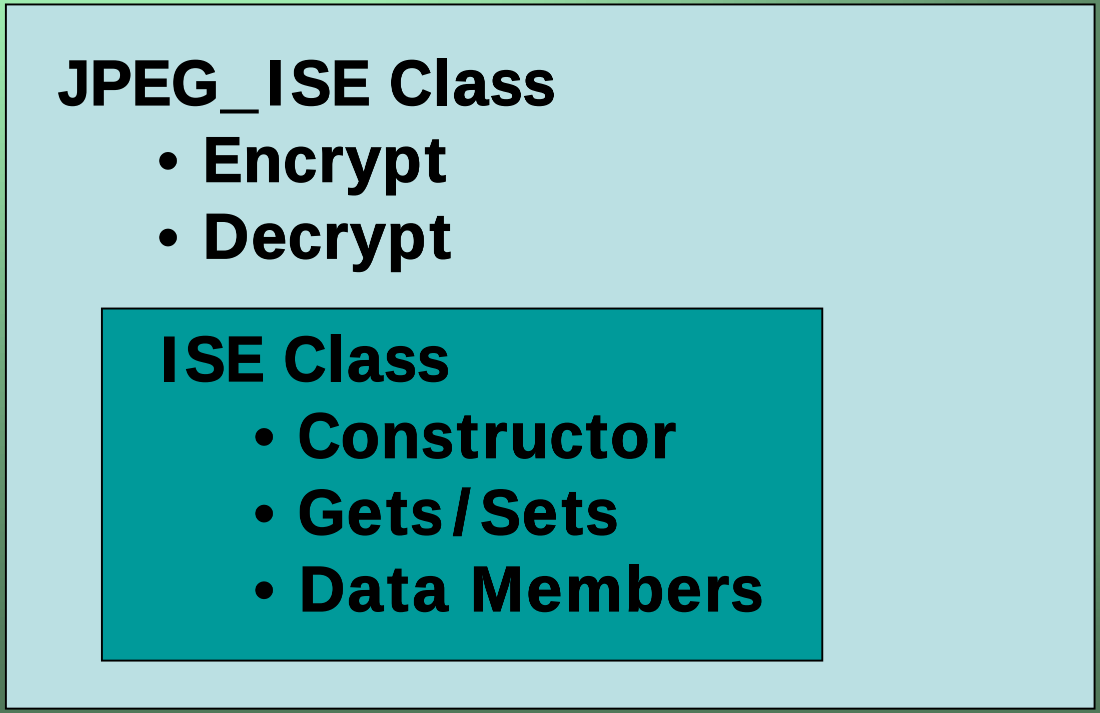
- Data Abstraction
  - ISE constructors
  - Virtual encrypt/decrypt methods
  - Data members and gets/sets
    - File names
    - Key
  - Make file name methods

Object Oriented Outline (cont):

- Information hiding
  - Data members
    - protected
  - Get/Set methods
    - File names
    - Key
    - File type

Object Oriented Outline (cont):

- Inheritance



JPEG_ISE Class

- Encrypt
- Decrypt

ISE Class

- Constructor
- Gets/Sets
- Data Members

Object Oriented Outline (cont):

- **Polymorphism**
  - **Constructors**
    - **ise()**
    - **ise (key, input_file_name, ise_file_name)**
      - **encrypting**
    - **ise(key, ise_file_name, output_file_name)**
      - **decrypting**

Object Oriented Outline (cont):

- Polymorphism
  - Encryption
    - encrypt_file()
    - encrypt_file(key, input_file_name, ise_file_name)
  - Decryption
    - decrypt_file()
    - decrypt_file(key, ise_file_name, output_file_name)

API Usage:

Encryption Scenario:

```
char[] myKey = “ISE_IS_THE_BEST”;
char[] myInputFile = “myImage.jpg”;
char[] myISEFile = “myImage.ise”;
jpeg_ise* myISE;
myISE = new jpeg_ise(myKey,myInputFile,MyISEFile);
myISE->encrypt_file();
delete myISE;
```

API Usage (cont):

Decryption Scenario:

```
    char[] myKey = “ISE_IS_THE_BEST”;
    char[] myISEFile = “myImage.ise”;
    char[] myOutputFile = “myImageDecrypt.jpg”;
    jpeg_ise* myISE;
    myISE = new jpeg_ise();
    myISE->set_key(myKey);
    myISE->set_ise_file(myISEFile);
    myISE->set_output_file(myOutputFile);
    myISE->decrypt_file();
    delete myISE;
```

OO Benefits:

- Objects easily extendable to other formats
- Clean, reliable code
- Apply what we’ve learned

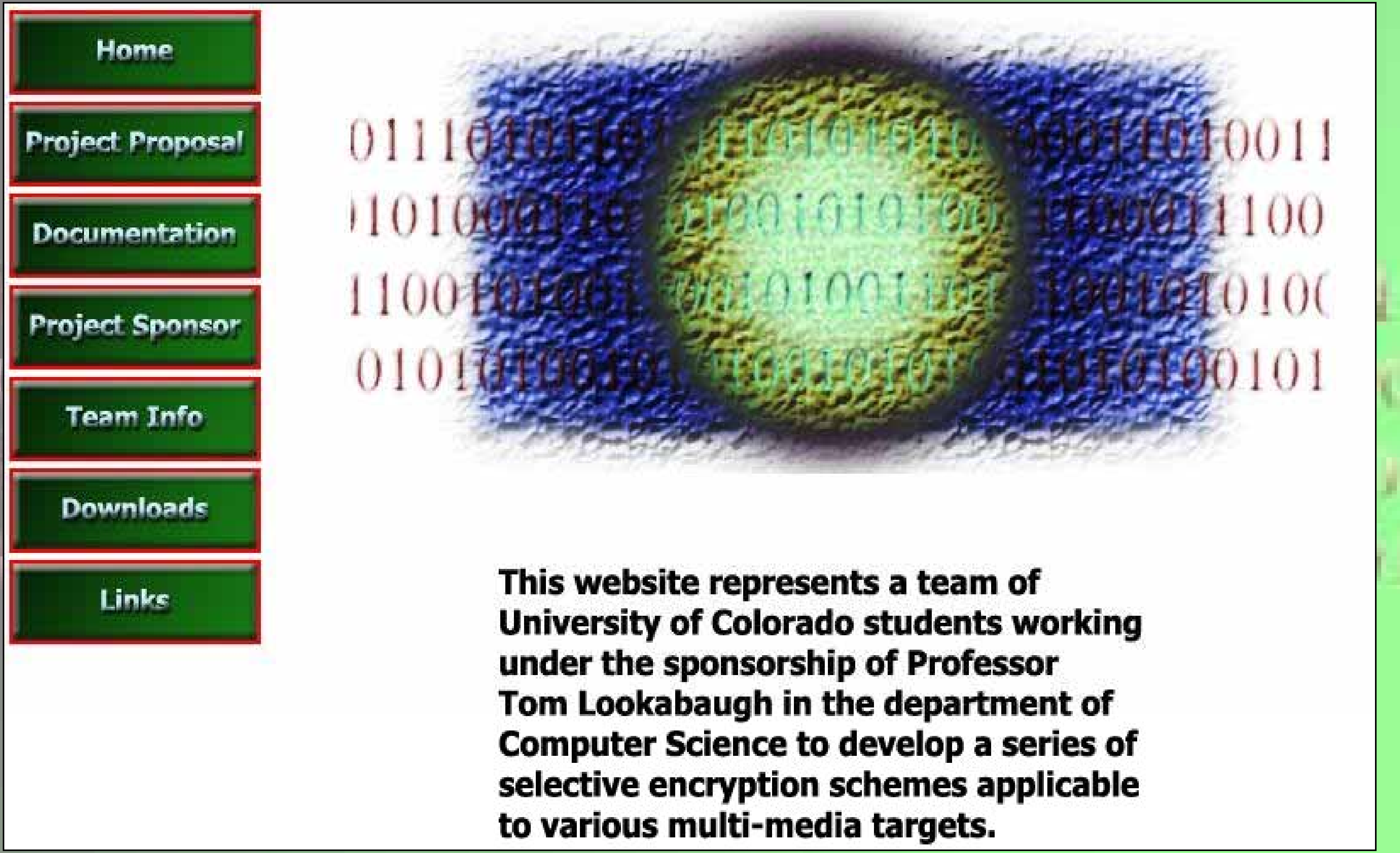
Presentation Overview:

- Statement of problem
- Initial research into compressed files
- Target Selection Process
- JPEG Statistical Analysis
- JPEG Manipulator Design
- JPEG Manipulator Demonstration
- Encryption Algorithm Selection
- JPEG Selective Encryption Algorithms
- ISE Production Code Design
- **ISE Web Site Design**
- Future Considerations

Requirements:

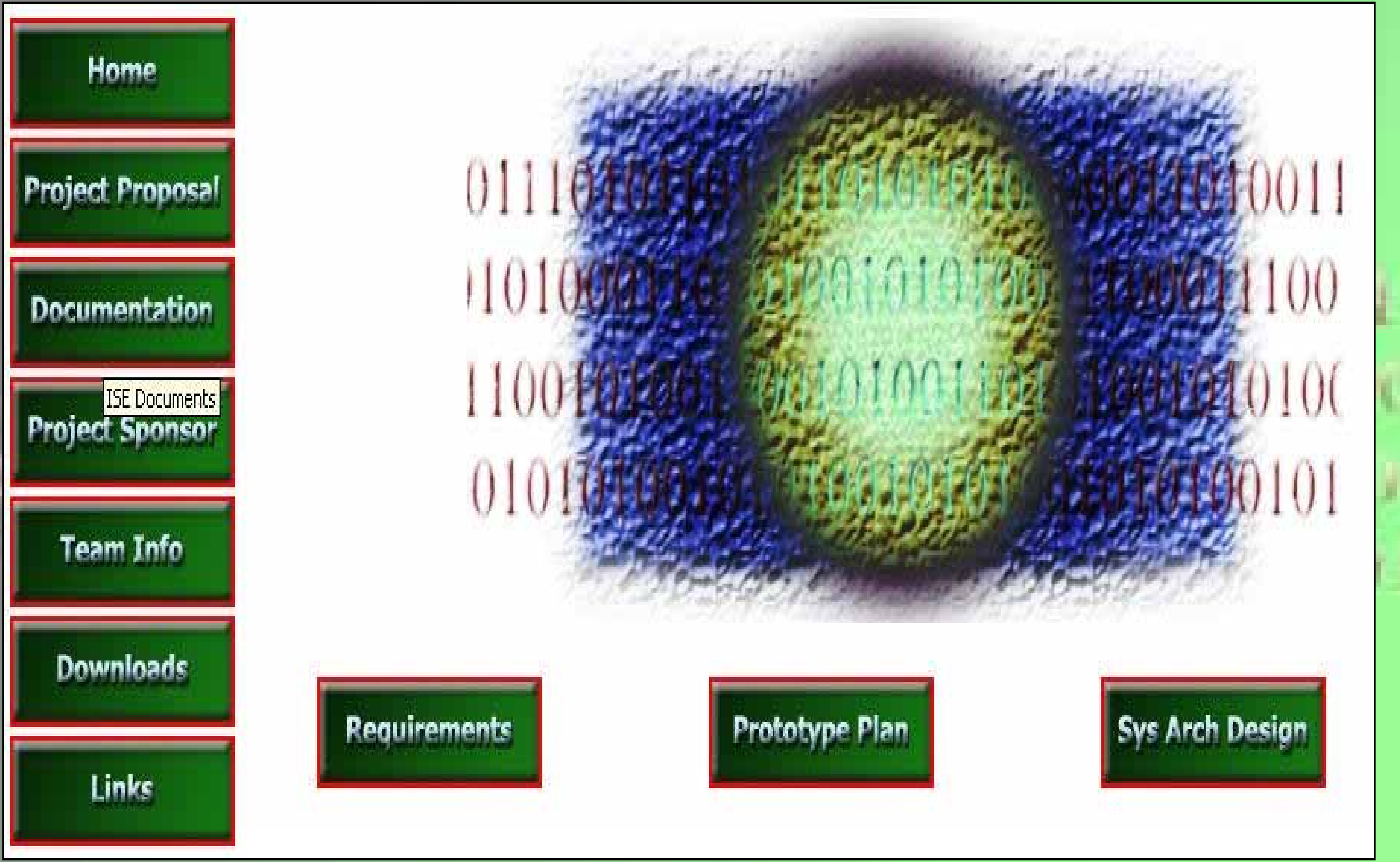
- **Easy to maintain**
- **Distribute products/documentation**
- **Create on existing computer in lab**
  - **http://128.138.75.184**

Home Page



This website represents a team of University of Colorado students working under the sponsorship of Professor Tom Lookabaugh in the department of Computer Science to develop a series of selective encryption schemes applicable to various multi-media targets.

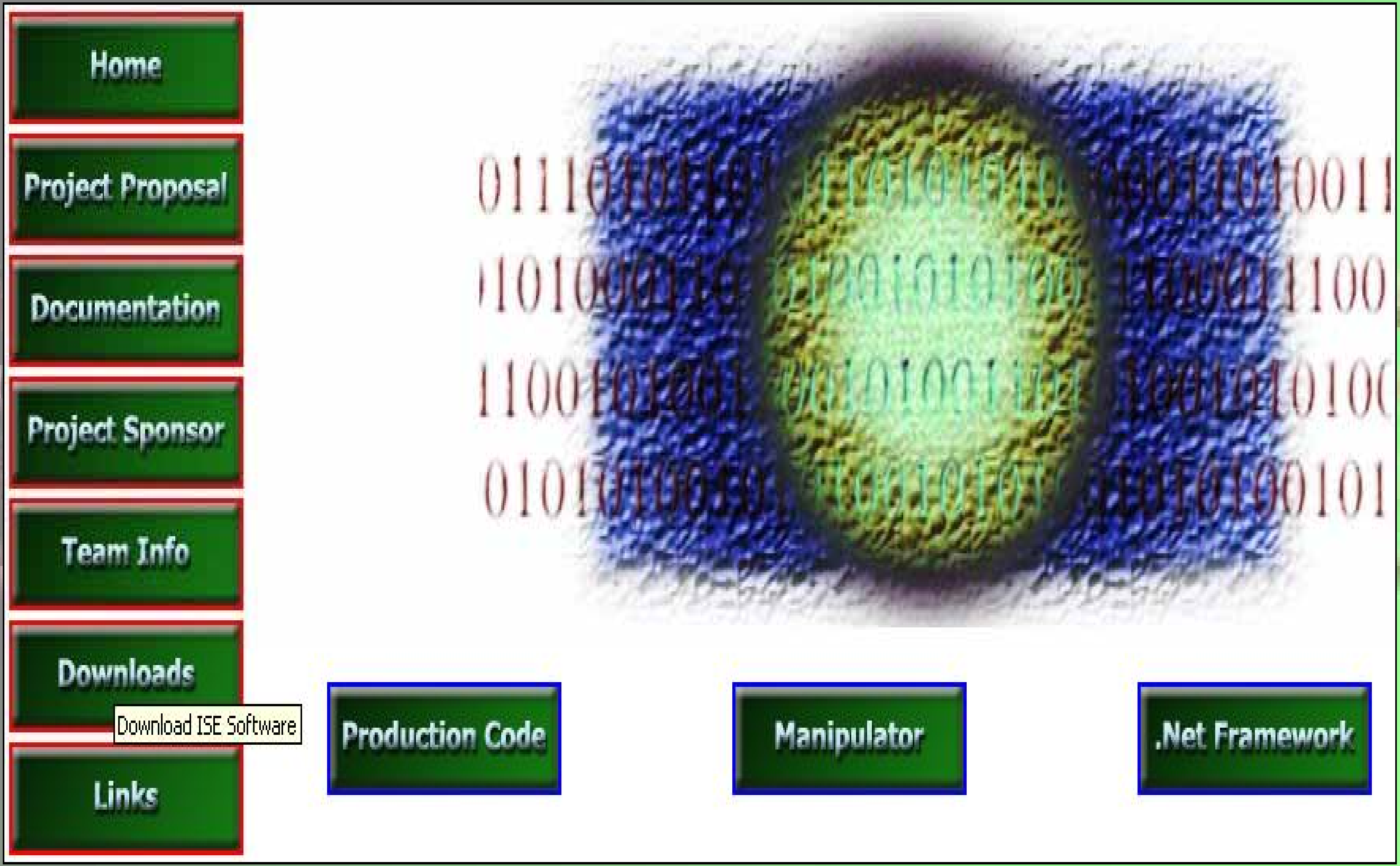
Documentation



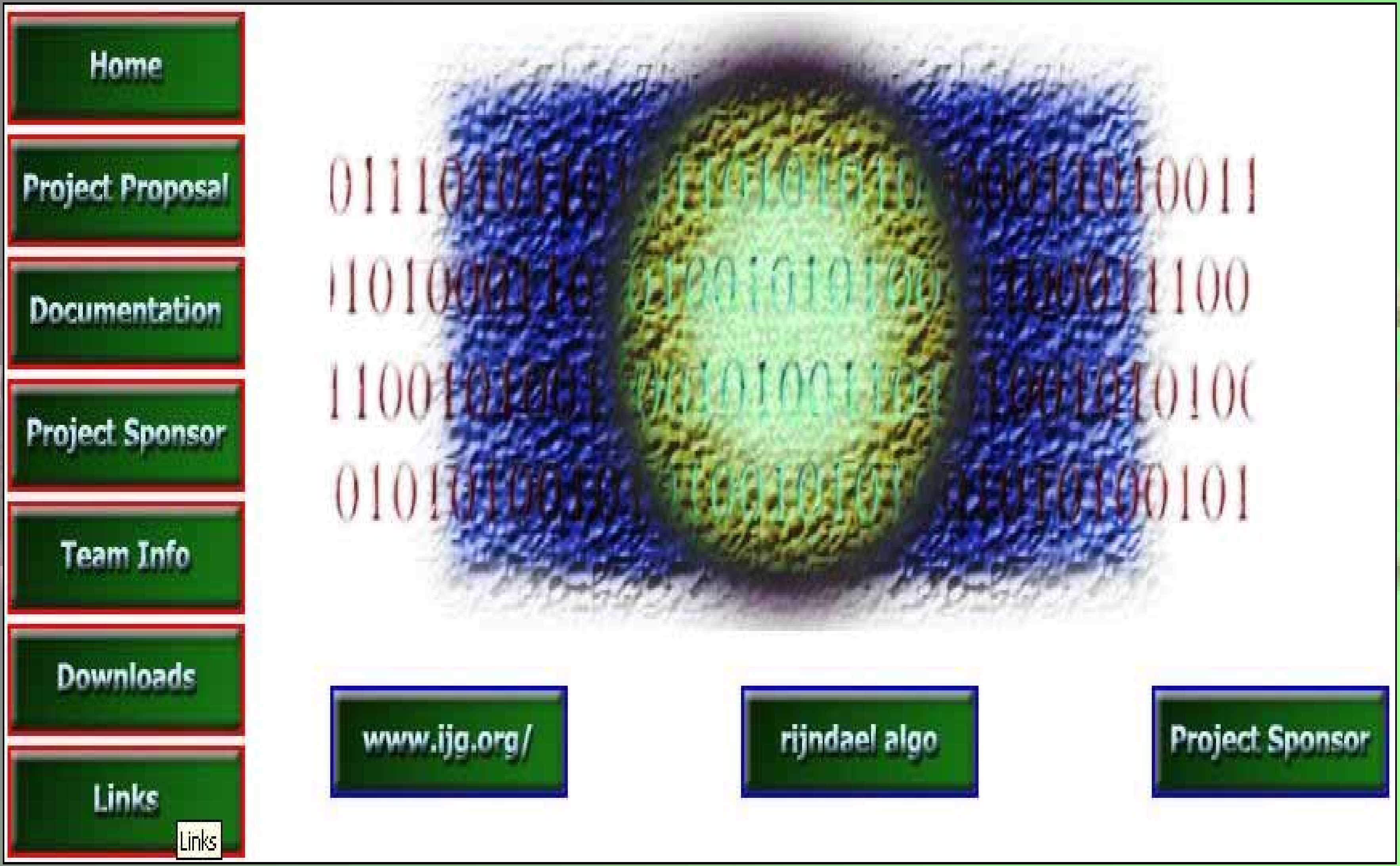
Web Site Design

Andrew

Downloads


Home
Project Proposal
Documentation
Project Sponsor
Team Info
Downloads
Download ISE Software
Links
Production Code
Manipulator
.Net Framework

Links



Web Site Design

Andrew

Presentation Overview:

- Statement of problem
- Initial research into compressed files
- Target Selection Process
- JPEG Statistical Analysis
- JPEG Manipulator Design
- JPEG Manipulator Demonstration
- Encryption Algorithm Selection
- JPEG Selective Encryption Algorithms
- ISE Production Code Design
- ISE Web Site Design
- **Future Considerations**

Future Considerations:

- **Black hat attacks**
  - **Huffman table**
    - **Replacement**
    - **Reconstruction**
      - **Based on Quantizer**
      - **Based on Application**
  - **Quantizer table**
- **Publish web site for community**
- **Corrections**

Questions