# A Framework For Concealed Data Transfer



PROJECT WORK DONE AT

*Kumaraguru College of Technology*

PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degree of

*Bachelor of Engineering -Information Technology*

Bharathiyar University, Coimbatore

Submitted by

*B.Anil Kumar*

*S.Damodhar Bhat*

*K.P.Rajesh*

*V.Shantha Kumar*

Guided by

*Mrs.S.Devaki B.E., M.S*

Assistant Professor

Department of Computer Science and Engineering

Kumaraguru College of Technology

*Department of Computer Science and Engineering*

*Kumaraguru College of Technology*

*Coimbatore-641 006*

*March 2002*

# CERTIFICATE

*Department of Computer Science and Engineering*
*Kumaraguru College of Technology*
*Coimbatore-641 006*

This is to certify that the project work entitled
*'A Frame Work for Concealed Data Transfer'*

Done by
*B.Anil Kumar*
*S.Damodhar Bhat*
*K.P.Rajesh*
*V.Shantha Kumar*

In partial fulfillment for the award of the degree of
*Bachelor of Engineering -Information Technology*
Bharathiyar University, Coimbatore
During the academic year 2001-2002

S.Thangasamy 18/3/'02
**Professor and HOD**
Dr.S.Thangasamy B.E., PhD

S.Devaki 18/3/2002
**Guide**
Mrs.S.Devaki B.E., M.S

Certified that the candidate has been examined by us in the project work.
Viva voce examination held on ... 18.03.2002 ..... and the university register number was .....

Rajesh 18.3.02
**Internal Examiner**

K.M.S 18/3/2002
**External Examiner**

# DEDICATED TO MY FAMILY AND WELL WISHERS

# ACKNOWLEDGEMENTS

I take this opportunity to express my sincere word of gratitude and thanks to **Dr.K.K.Padmanaban**, Principal, Kumaraguru College of Technology, Coimbatore for presenting me this opportunity and for extending constant support and valuable guidance through out the project work.

I am profoundly grateful to **Prof. S.Thangasamy**, Head, Department of Computer Science, Kumaraguru College of Technology, Coimbatore-06, for presenting me this opportunity and for extending constant support and valuable guidance through out the project work.

I am thankful to **Mrs. S.Devaki**, my guide and Assistant Professor of Department of Computer Science, Kumaraguru College of Technology, Coimbatore-06, for her valuable guidance and constant monitoring throughout the course of my project.

I am also thankful to all the faculty members of Department of Computer Science, Kumaraguru College of

Technology, Coimbatore-06, for their cooperation and encouragement during my study period.

I would like to thank all my class friends and family members who gave me valuable tips and encouragement to complete this project successfully.

# SYNOPSIS

Steganography is the practice of embedding secret messages in other messages in a way that prevents an observer from learning that anything unusual is taking place. Steganography simply takes one piece of information and hides it within another. Computer files (images, sounds recordings, even disks) contain unused or insignificant areas of data. Steganography takes advantage of these areas, replacing them with information. The files can then be exchanged without anyone knowing what really lies inside of them.

This project aims at attaining Steganography providing operations to load a picture, embed a text in to it and try to recover the text from the file or picture in which the text was embedded .The Application is authenticated by means of a password which gives additional security to the system. This additional security is provided for the application because the security of the system has to be based on the assumption that the "Eaves Dropper" has full knowledge of the design and implementation details of the steganographic system.

This program uses three pixels for a character. It uses eight components of the three pixels (last remains unused) to form a binary representation of the ASCII code of the character, but instead of using zeroes and ones, it uses odd and even numbers. If we need to store the character "a" in the following three pixels: (154, 73, 211), (98, 110,39) and (16,255,85). The code for "a" is 97(in decimal) and in binary is 11000001. The program transforms the first eight components in even number so the three pixels became: (154,72,210), (98,110,38), (16,254,85). Then, it makes a Correspondence between the components and the binary representation of the character. If a digit in the binary number is 1, program adds 1 to the correspondent component. The final values of the components are:(155,73,210),(98,110,38),(16,255,85).When noticed there will not be much difference between the initial and the final values.

A little bit about the password protection. Consider the ASCII character set as a deck of cards. The program creates a number of such "decks" which is equal to the number of the character in the password. Then it shuffles each "deck" in a way,

which depends on the entire password and on the correspondent character in the password. For example, if the character "b" is the fifth in the password, the shuffle will depend on 5 * Asc ("b") * a number which depend on the entire password. The total number of arrangements of a sequence of 256 elements is 1*2*3* ...*255*256, which is a number with more than 600 digits! Thus the longer is the password the stronger is the encryption.

# CONTENTS

**1. INTRODUCTION**

Synopsis

Existing system and its limitations

Proposed system and its advantages

**2. SYSTEM REQUIREMENTS**

Product Definition

Project Plan

**3. SOFTWARE REQUIREMENT SPECIFICATION**

**4. DESIGN DOCUMENT**

**5. PRODUCT TESTING**

**6. FUTURE ENHANCEMENT**

**7. CONCLUSION**

**8. REFERENCE**

**9. APPENDIX**

Input Screen

Output Screen

Sample Output

# EXISTING SYSTEM

***Steganography:***

Steganography is the art of hiding information in ways that prevent the detection of hidden messages. Steganography, derived from Greek, literally means, "Covered writing." It includes a vast array of secret communications methods that conceal the message's very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, Covert channels, and spread spectrum communications. Steganography and cryptography are cousins in the spy craft family. Cryptography scrambles a message so it cannot be understood. Steganography hides the message so it cannot be seen. A message in cipher text, for instance, might arouse suspicion on the part of the recipient while an "invisible" message created with steganographic methods will not.

***Ancient methods of concealing data:***

Throughout history, people have hidden information by a multitude of methods and variations. For example, ancient Greeks

wrote text on wax-covered tablets. To pass a hidden message, a person would scrape wax off a tablet, write a message on the underlying wood and again cover the tablet with wax to make it appear blank and unused. Another ingenious method was to shave the head of a messenger and tattoo a message or image on the messenger's head. After the hair grew back, the message would be undetected until the head was shaved again.

Invisible inks offered a common form of invisible writing. Early in World War II, steganographic technology consisted almost exclusively of these inks. With invisible ink, a seemingly innocent letter could contain a very different message written between the lines. Documents themselves can hide information document text can conceal a hidden message through the use of null ciphers (unencrypted messages), which camouflage the real message in an innocent sounding missive. Open coded messages, which are plain text passages, "sound" innocent because they purport to be about ordinary occurrences. Because many open-coded messages don't seem to be cause for suspicion, and therefore "sound" normal and

innocent, mail filters can detect the suspect communications while "innocent" messages are allowed to flow through.

***Technical Summary:***

To a computer, an image is an array of numbers that represent light intensities at various points (pixels). These pixels make up the image's raster data. A common image size is 640 x480 pixels and 256 colors (or 8 bits per pixel). Such an image could contain about 300 kilobits of data. Digital images are typically stored in either 24-bit or 8-bit files. A 24-bit image provides the most space for hiding information; however, it can be quite large (with the exception of JPEG images). All color variations for the pixels are derived from three primary colors: red, green, and blue. Each primary color is represented by 1 byte; 24-bit images use 3 bytes per pixel to represent a color value. These 3 bytes can be represented as hexadecimal, decimal, and binary values. In many Web pages, the background color is represented by a six-digit hexadecimal number—actually three pairs representing red, green, and blue. A white background would have the value FFFFFF: 100 per-cent red (FF), 100 percent green (FF),and 100 percent blue (FF).

three bytes making up white. This definition of a white background is analogous to the color definition of a single pixel in an image. Pixel representation contributes to file size. For example, suppose we have a 24-bit image 1,024 pixels wide by 768 pixels high—a common resolution for high-resolution graphics. Such an image has more than two million pixels, each having such a definition, which would produce a file exceeding 2 Mbytes.

***Implementation Summary:***

Embedding data, which is to be hidden, into an image requires two files. The first is the innocent looking image that will hold the hidden information, called the cover image. The second file is the message the information to be hidden. A message may be plain text, cipher text, other images, or anything that can be embedded in a bit stream. When combined, the cover image and the embedded message make a stego image. To a stegokey (a type of password) may also be used to hide, and then later decode, the message.

***Pitfalls in the existing system:***

The existing system doesn't provide the facility of authentication thus making it easier for the eavesdropper to hack the contents .A GUI interface should make the user comfortable with what he is using.

# PROPOSED SYSTEM

This system aims at achieving steganography with a GUI interface. The project is implemented by embedding a text file inside a picture file. We propose to achieve high security by means of authentication through password protection. Password protection provided is not the simplest by all means. Instead a process of shuffling among all characters of the password is done. The program creates a number of "decks" which is equal to the number of the character in the password. Then it shuffles each "deck" in a way, which depends on the entire password and on the correspondent character in the password. For example, if the character "b" is the fifth in the password, the shuffle will depend on 5 * Asc ("b") * a number which depend on the entire password. The total number of arangements of a sequence of 256 elements is 1*2*3* ...*255*256, which is a number with more than 600 digits! Thus the longer the password, the stronger is the encryption. When password is not input during the embedding of the text, there is no need to input a password while getting back the message.

The interface guides the user through all the steps of embedding the text file within the image. Initially the

picture is loaded in the provided picture box. Just to make the appearance fine enough the picture is aligned at the center. In case where the image is bigger than the allotted picture box, scroll bars are provided to view the entire picture file. The text to be hidden in the file is inserted in the text area provided. A button is provided to embed the message in the image. However before insertion the size of the text and the picture file is compared to make sure that the number of pixels available within the image is at least thrice equal to the number of characters within the text. When the message is inserted the password is read from the text box and the process of shuffling that was explained before is carried out. On completion of this process the new picture file is saved. It is to be noted that the two picture files show the least differences. Even the size of the two picture files remains the same. Once this is completed the new file is now loaded and the process of retrieval is now carried out. During the process of retrieval if the password provided is not correct the user is not requested again to enter the correct password but instead a message is displayed saying

'THIS PICTURE HAS NO SECRET MESSAGE'.

***Advantages:***

This project enhances the existing system by providing password protection for authentication. This helps in making it difficult for the eavesdropper to hack the content. A GUI interface is provided for the users to feel comfortable with what they are using.

# SYSTEM REQUIREMENTS

# PRODUCT DEFINITION

Steganography is the art and science of communicating in a way, which hides the existence of the communication. In contrast to cryptography, where the "enemy" is allowed to detect, intercept and modify message without being able to violate certain security premises guaranteed by a cryptosystem, the goal of steganography is to hide messages inside other "harmless" messages in a way that does not allow any "enemy" to even detect that there is a second secret message present. Steganography is in the (especially military) literature also referred to as transmission security or short TRANSEC.

A good steganography system should fulfill the same requirements posed by the "Kerckhoff principle" in cryptography. This means that the security of the system has to be based on the assumption that the "enemy" has full knowledge of the design and implementation details of the steganographic system. The only missing information for the “enemy" is a short easily exchangeable random number sequence, the secret key, and without the secret key, the "enemy" should not have the slightest

chance of even becoming suspicious that on an observed communication channel hidden communication might take place.

This idea has been implemented in this project and the results have proven to be successful.

# PROJECT PLAN

The basic plan in this project is to implement the art of steganography, which is vastly used by Osama Bin Laden and others. USA reported on February 7th 2001 that Osama and others "are hiding maps and photographs of terrorist targets and posting instructions for terrorist activities on sports chat rooms, pornographic bulletin boards and other websites."

While most of the steganographic tools available today are executed on the command line we made an attempt to implement it in GUI. This project is implemented using Microsoft Visual Basic 6.0. Moreover, since steganography as such is a means for secret message transfer the need for authentication was not emphasized. But we took in to consideration the need for security by password protection keeping in mind that the enemy might get to know about the existence of secret message transfer.

# SOFTWARE REQUIREMENTS SPECIFICATION

# SOFTWARE REQUIREMENTS SPECIFICATION

Steganography, the art of embedding one file into another so as to communicate in a secret manner is a very old technique. The requirement of this software is to enable a text file to be embedded into another picture file. The text file of any size should be able to be embedded into any picture file such as jpg, bmp, and gif.

The way embedding of text inside the picture should be in such a way that there remains no explicit difference between the actual picture file and the tampered picture file. Steganography as such is by means secure but taking into consideration the modern era, every possible condition is to be taken into account.

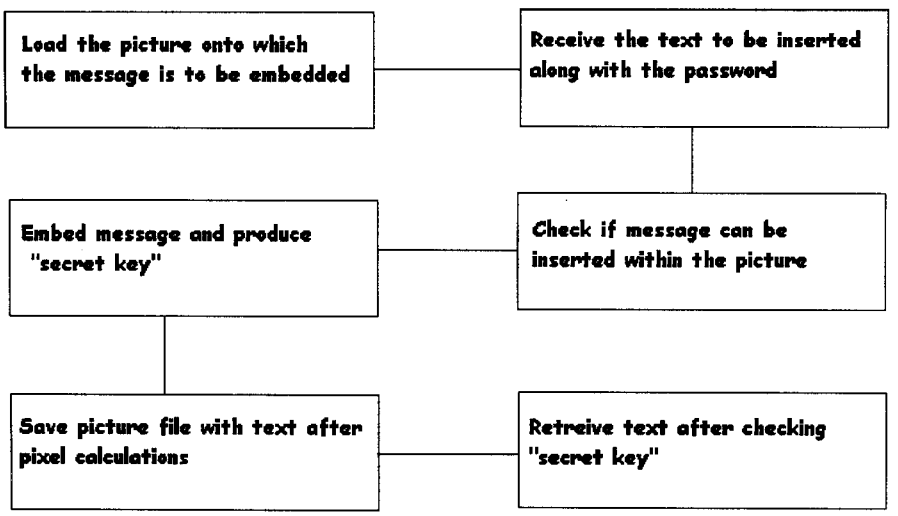
Assuming that the hacker or the eavesdropper knows the existence to steganography and its usage, novel methods are to be considered in order to make situations much better. Thus there arises a need for further security.

The additional security can be provided by means of password. The password used must be agreed between the two users in order to avoid to avoid confusion. This helps in providing additional security to the system.

The other requirement of software is GUI. GUI helps the user to use the software in the most efficient way. This also encourages the novel user.

# DESIGN DOCUMENT

# DESIGN DOCUMENT




This project deals with the most primitive form of steganography, embedding a text file within picture. This is the most widely accepted form of "hidden data transfer" and is being investigated in depth.

There are six phases involved in the process of embedding the message. We start with the initial step of loading the picture file into which the message is to be embedded. Once this is completed the message to be hidden is obtained from the

user. A password and a "secret key" is obtained. The pixel calculations are made such that the modifications are not visible to the human eye. This file is now saved under a new name. During retrieval the file is loaded and after he " secret key" is authenticated the message is retrieved. But however if the password doesn't match the user is not requested again to enter the password. But instead it is reported that the file doesn't contain any secret message.

# PRODUCT TESTING

# TESTING AND IMPLEMENTATION

The implementation phase is less creative than system design. No software project is assumed completed until it is successfully tested and implemented.

An elaborate test data is prepared and the system is tested using the test data. While testing, errors are noted and corrections are made. The corrections are also noted for future use. Both the hardware and software securities are made to run the developed system successfully in future.

***Objectives of Testing:***

- Testing is the process of executing a program with the indent of finding an error.
- A good test case is one that has the high probability of finding yet undiscovered error.
- A successful test is one that uncovers and yet undiscovered error.

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and effectively before the live operation commences. Testing is vital

to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

The candidate system is subjected to a variety of tests online response volume, stress, recovery, security and usability tests. A series of testing are performed before the system is read for the user acceptance testing.

***Types of Testing:***

Types of testing are

- Unit testing
- Integrated testing
- Black box testing
- User acceptance testing

***Unit testing:***

In this testing each sub module is tested individually and integrated with the overall system. Unit testing focuses verification effort on the smallest unit software design of the module. This is also known as module testing. The modules of

the system are tested separately. This testing was carried out during programming stage itself. In this testing step each module is found to be working satisfactorily as regard to the expected output from the module.

***Integrated testing:***

Data can be lost across an interface; one module can have an effect on another module, sub functions when combined may not produce the desired major functions. Integrated testing is a systematic testing for constructing the program structure. This testing was done with the sample data. The developed system was done successfully with the test data. The need for the integrated test is to find the overall system performance. The objective is to take unit-tested module and build a program structure. All the modules are combined and tested as a whole. Here correction is difficult because the vast expenses of the entire program complicate the isolation of cause. Thus in the integrated step all errors are uncovered, corrected for the next testing steps.

***Black box testing:***

Black box testing is done to find

- Incorrect or missing functions
- Interface error
- Errors in external device access
- Performance error
- Initialization and termination errors

The above testing was successfully carried out for this system

***Validation testing:***

At the culmination of the black box testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected and the final series of software tests begins. Validation succeeds when the software functions in a manner that can be reasonably expected by the customer. After validation test has been conducted, one of the two possible conditions exists.

- The function or performance characteristics confirmed to specification and are accepted.
- The deviation from specification is uncovered and deficiency list is created.

Proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

# FUTURE ENHANCEMENTS

# FUTURE ENHANCEMENTS

Future enhancements to this project could be dealt as below

- The message that was embedded within the picture file could be encrypted using modern encryption techniques.

- The field of steganography could be vastly used for military purposes.

# CONCLUSION

# CONCLUSION

A framework for concealed data transfer is a simple and secure process and ensures full performance. The implementation of this project was simple, making use of a user-friendly and GUI language, Visual Basic 6.0. The usage has been proved to be simple even allowing a person who is not familiar with Graphical Interfaces to work with this model.

# REFERENCE

# BIBLIOGRAPHY

## *References:*

### Books:


- Exploring Steganography: Seeing the Unseen

  Neil F. Johnson
  Sushil Jajodia
  (George Mason University)

- Visual Basic 6.0 Bible

  Kate Gregory

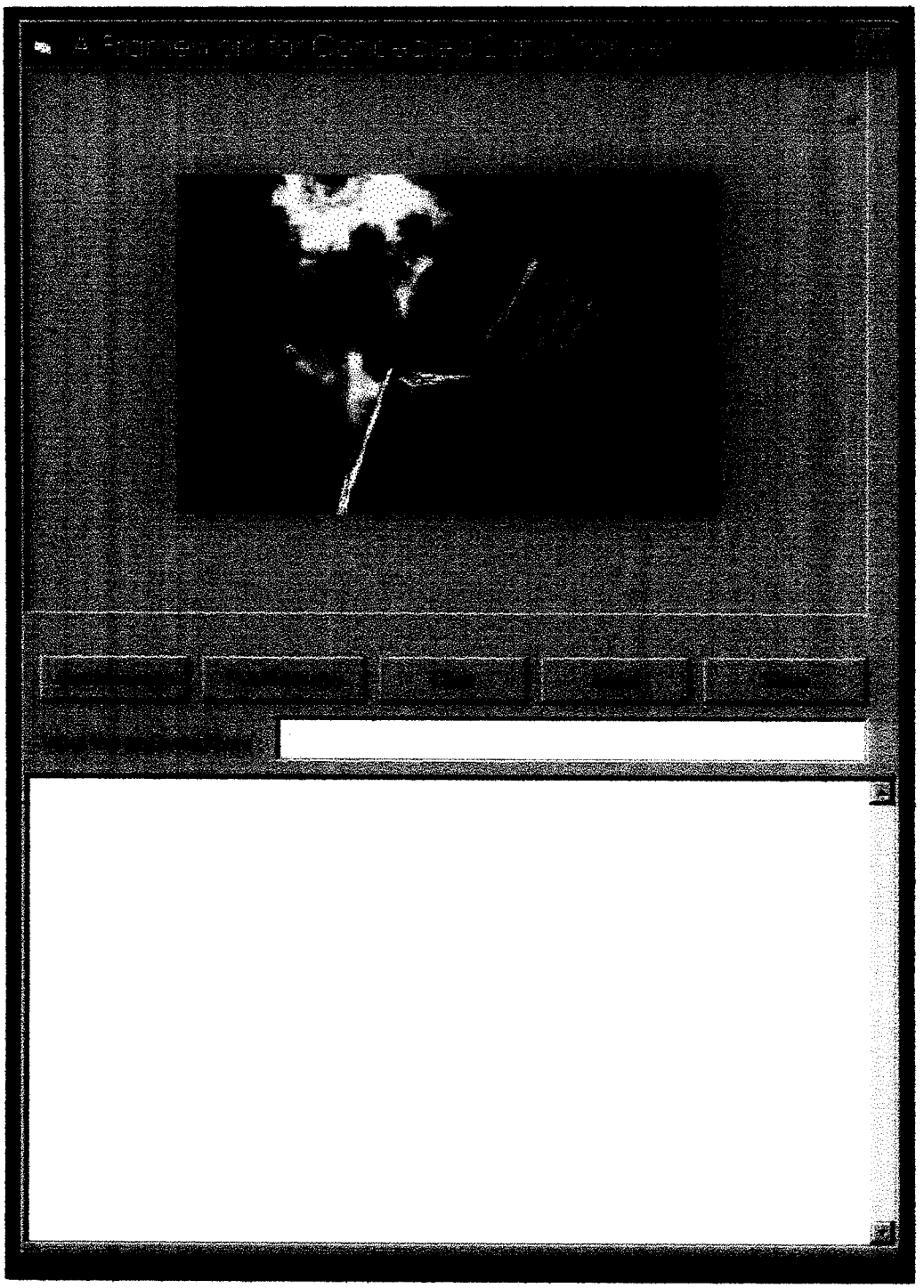

### Online Resources:


- **Newsgroups:** sci.crypt.research

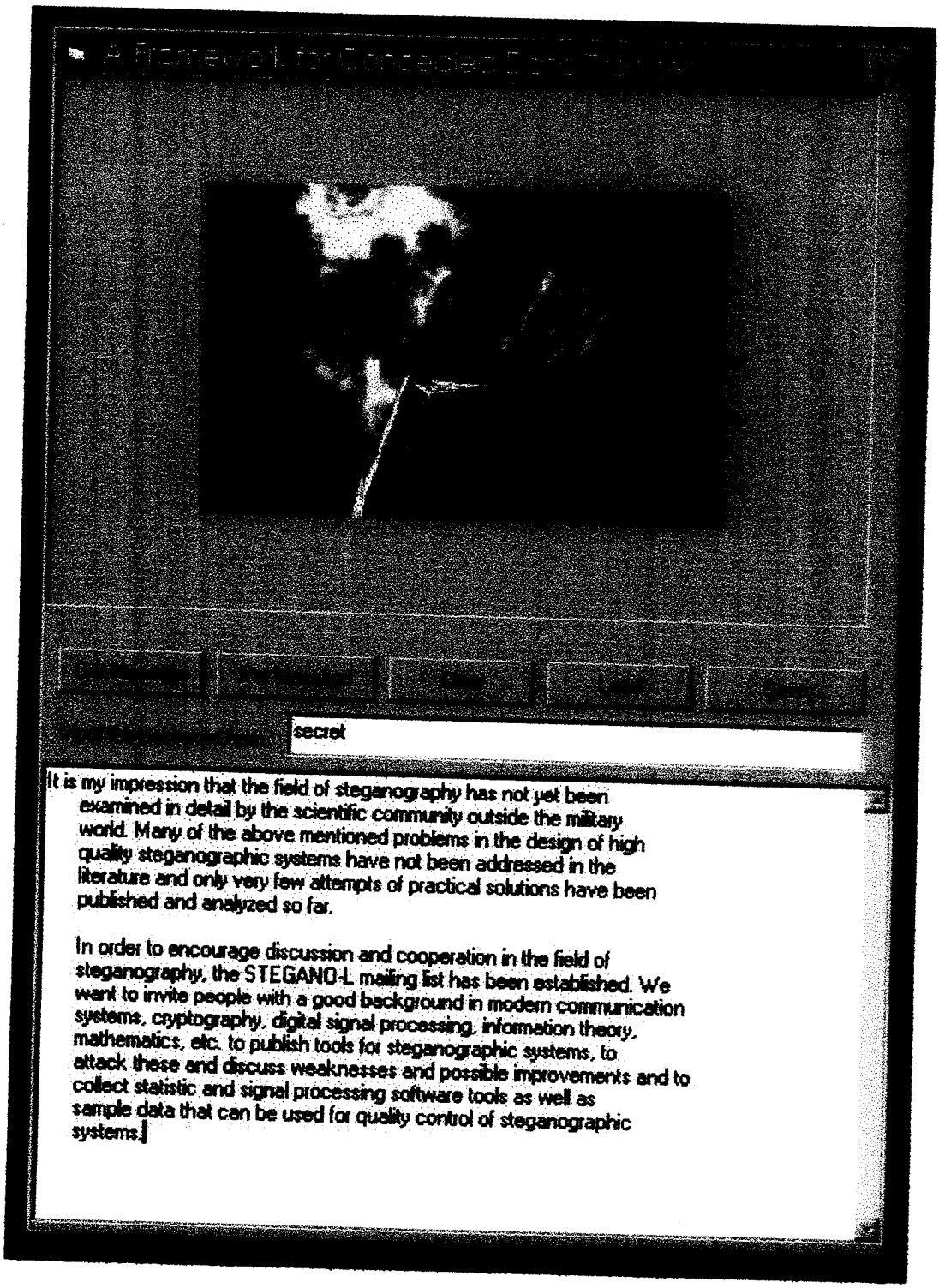
# APPENDIX

# INPUT SCREEN

# INPUT SCREEN



# OUTPUT SCREEN



# BEFORE EMBEDDING TEXT

# AFTER EMBEDDING TEXT