Τμήμα Πληροφορικής

Λαμία, 9 Ιουλίου 2018

Πτυχιακή εργασία με τίτλο :

Στεγανογραφία :
Το είναι μέσα στο φαίνεσθαι

Αναστασία Γ. Μακρή

ΑΜ 2113067

Επιβλέπων καθηγητής : Καθ. Γεώργιος Σταμούλης

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τους καθηγητές μου, για την πολύτιμη βοήθειά τους αυτά τα χρόνια και για τις γνώσεις που μου μεταλαμπάδευσαν.
Τέλος, ευχαριστώ την οικογένειά μου που ήταν δίπλα μου στον αγώνα των σπουδών μου όλα αυτά τα χρόνια δίνοντάς μου δύναμη και υποστήριξη.

Αφιερώνω την πτυχιακή αυτή

στον αδερφό μου Χρήστο…

Περιεχόμενα

Πίνακας εικόνων

Περίληψη

Η Στεγανογραφία είναι ένας όρος ο οποίος χρησιμοποιείται για να εξυπηρετεί την μυστική επικοινωνία. Κύριος στόχος της είναι η απόκρυψη δεδομένων-πληροφοριών-μηνυμάτων μέσα σε ένα οποιοδήποτε μέσο χωρίς αυτό να γίνει αντιληπτό από τρίτα άτομα. Την Στεγανογραφία, δεν την συναντήσαμε για πρώτη φορά τώρα, την γνωρίσαμε κιόλας από τα μέσα του 440 π.Χ. όπως και θα ανατρέξουμε και στο 1ο Κεφάλαιο που εμπεριέχει την ιστορική αναδρομή. Επίσης, θα αναφερθούν και θα αναλυθούν κάποιες τεχνικές στεγανογραφίας, καθώς και θα γίνει και αναφορά στην Κρυπτογραφία. Η αναφορά αυτή γίνεται τόσο για συγκριθούν οι δύο έννοιες αυτές όσο και για να ξεκαθαριστεί πως η καθεμία είναι ξεχωριστή (έννοια) και πως συμπληρώνει η μία την άλλη. Εμπεριέχει το Κεφάλαιο 1ο την σύγκριση μεταξύ αυτών των δύο, παραδείγματα αλλά και πλεονεκτήματα και μειονεκτήματα της χρήσης τους. Στο Κεφάλαιο 2ο υπάρχουν αναφορές σε εφαρμογές που προσφέρονται για στεγανογράφηση κειμένων, εικόνων, ήχου κλπ, που κάποιες από αυτές είναι δωρεάν και κάποιες επί πληρωμή ή και on line. Στο Κεφάλαιο 3ο αναπτύχθηκε μια εφαρμογή για την υλοποίηση της στεγανογραφίας

Λέξεις-Κλειδιά: Στεγανογραφία, Τεχνικές Στεγανογραφίας, Κρυπτογραφία, Εφαρμογή Στεγανογραφίας, MatLab, LSB

Summary

Steganography is a term used to serve secret communication. Its main goal is to hide data-information-messages in any staff without this being perceived by third person. We did not meet it for the first time now, we knew it from the middle of 440 BC. as we will also refer to the 1st Chapter that contains the historical retrospection. Also, some Steganography techniques will be reported and analyzed, as well as Cryptography. This reference is made both for comparing these two concepts and for clarifying that each is distinct (meaning) and how it complements one another. Chapter 1 contains a comparison of these two examples, but also advantages and disadvantages of their use. In Chapter 2, there are references to applications that are suitable for sealing of texts, pictures, sounds, etc., some of which are free of charge and some are paid or on-line. In Chapter 3 an application was developed, for the implementation of Steganography.

Key-Words: Steganography, Steganography Techniques, Cryptography, Steganography Application, MatLab, LSB

Εισαγωγή

Σκοπός της παρούσας πτυχιακής είναι η μελέτη και παρουσίαση των τεχνικών στεγανογραφίας και η ανάπτυξη προγράμματος για την εφαρμογής αυτής.

Στη σημερινή εποχή, θα παρατηρήσει κανείς πως η ανάπτυξη του διαδικτύου έχει φέρει μεγάλες αλλαγές στην ποσότητα αλλά και στην ποιότητα των διαθέσιμων περιεχομένων. Οι χρήστες ηλεκτρονικών υπολογιστών, και όχι μόνο, καθημερινά περιβάλλονται από ένα τεράστιο όγκο πληροφοριών, που μεταδίδονται με ταχύτατους ρυθμούς, όπως είναι τα κείμενα, εικόνες, ήχους, βίντεο κλπ. Το διαδίκτυο έχει εξαπλωθεί τόσο ώστε η ανάγκη για ακόμα περισσότερες πληροφορίες (κείμενα, εικόνες κλπ.) να έχει αυξηθεί. Επειδή ακριβώς ζούμε σε μια σύγχρονη εποχή, θα ήταν ιδανικό στον κόσμο να μπορούσαν οι χρήστες, να ανταλλάσσουν ανοιχτά πληροφορίες, email, κρυπτογραφημένα, χωρίς να έχουν αντίποινα και επιπτώσεις. Αυτό κάποιες φορές καθίσταται αδύνατο, είτε γιατί οι χρήστες δουλεύουν σε κάποια επιχείρηση που δεν εγκρίνει την διακίνηση κρυπτογραφημένης πληροφορίας, είτε γιατί το ίδιο το κράτος απαγορεύει γενικά την κρυπτογραφημένη επικοινωνία. Για αυτούς τους λόγους, αλλά και για κάποιους παραπάνω η Στεγανογραφία έρχεται για να βρει λύση σε αυτό το πρόβλημα.

Το αντικείμενο της Στεγανογραφίας είναι να μπορεί κρύψει ένα δεδομένο μέσα σε κάποιο άλλο με τέτοιο τρόπο ώστε να μην μπορεί να κινήσει υποψίες για στεγανογράφηση αντικειμένου. Τα αρχεία που διαχειρίζεται ένας χρήστης όπως προαναφέρθηκαν οι εικόνες, κείμενα κοκ περιέχουν αχρησιμοποίητους ή περιττούς τομείς δεδομένων. Με τη Στεγανογραφία ο χρήστης, μπορεί να εκμεταλλευτεί τον χώρο αυτό και να εντάξει ό,τι εκείνος επιθυμεί. Με την χρήση των στεγανογραφικών τεχνικών, αλλά και των διαθέσιμων εφαρμογών που προσφέρονται για αυτόν τον σκοπό, η διακίνηση και ανταλλαγή πληροφορίας ανάμεσα σε δύο πρόσωπα, ή και περισσότερα, επιτυγχάνεται χωρίς κανένας άλλος να γνωρίζει αν και τι μπορεί να κρύβουν (οι ανταλλασσόμενες πληροφορίες). Είναι λοιπόν, με άλλα λόγια το «Είναι μέσα στο φαίνεσθαι».

Αναπτύχθηκε εφαρμογή στο προγραμματιστικό περιβάλλον MatLab η οποία δίνει την δυνατότητα στον χρήστη να ενσωματώσει/αποκρύψει κείμενο ή εικόνα μέσα σε εικόνα, έτσι ώστε να είναι δυνατή η μεταφορά πληροφορίας χωρίς να γίνεται αντιληπτή από τρίτο πρόσωπο. Για να αυξηθεί η ασφάλεια της κρυμμένης πληροφορίας η εφαρμογή

κρυπτογραφεί τα δεδομένα. Επίσης προστέθηκε επιπλέον επίπεδο ασφαλείας εισάγοντας γεννήτρια τυχαίων αριθμών για την επιλογή και τοποθέτηση σε τυχαίες θέσεις της πληροφορίας.

Κεφάλαιο 1ο
Στεγανογραφία

Στεγανογραφία είναι η διαδικασία, η τέχνη, η επιστήμη κατά την οποία η πληροφορία-το μήνυμα του αποστολέα, είναι κρυμμένο μέσα σε ένα άλλο μέσο. Το μέσο που χρησιμοποιείται θα μπορούσε να είναι κάποιο υλικό αντικείμενο (πχ ένα στυλό) ή κάποιο άυλο αντικείμενο (πχ μια φωτογραφία σε ηλεκτρονική μορφή).

1.1 Ανάλυση λέξης

Στεγανογραφία είναι σύνθετη λέξη και προέρχεται από τις ελληνικές λέξεις:
Στεγανός= καλυμμένος, αδιαπέραστος, αδιάβροχος
Γραφή= αναπαράσταση του λόγου με γράμματα, σύμβολα, ακόμα και εικόνες

Δηλαδή είναι η γραφή που είναι ΄΄αδιαπέραστη΄΄, ΄΄καλυμμένη΄΄ ή αλλιώς ΄΄μυστική΄΄.

1.2 Στόχος Στεγανογραφίας

Στόχος της στεγανογραφίας είναι η επίτευξη ασφαλούς επικοινωνίας μεταξύ παραλήπτη-αποστολέα με έναν μη ανιχνεύσιμο τρόπο και χωρίς να κινεί υποψίες για μετάδοση κρυφών δεδομένων. Σκοπός δηλαδή δεν είναι μόνο το να μη γνωρίζει κάποιος την κρυφή πληροφορία, αλλά να μη μπορεί καν να την αντιληφθεί. Ο τρόπος με τον οποίο έγινε στεγανογράφηση της πληροφορίας αλλά και ο τρόπος με τον οποίο μπορεί να ανακτηθεί πρέπει να τον γνωρίζει μόνο ο παραλήπτης και ο αποστολέας. Έτσι επιτυγχάνεται και η μέγιστη ασφάλεια[1] μετάδοσης του μηνύματος.

1.3 Ιστορική Αναδρομή

Κάνοντας μια αναδρομή στο παρελθόν θα διαπιστωθεί πως ο άνθρωπος έβρισκε συνεχώς νέους τρόπους για να καταφέρει να αποκρύψει την πληροφορία που ήθελε να στείλει. Συγκεκριμένα, η Στεγανογραφία έκανε την «πρώτη της εμφάνιση» περίπου λίγο μετά του 440 π.Χ. με πρωταγωνιστή τον Ηρόδοτο που κατέγραψε 2 γεγονότα της στεγανογραφίας. Το πρώτο γεγονός είναι όταν ένας στρατιώτης που λεγόταν Δημήρατος έπρεπε να στείλει ένα μήνυμα στη Σπάρτη ότι ο Ξέρξης πρόκειται να

[1] Η ασφάλεια είναι ο όρος της προστασίας από τον κίνδυνο ή την απώλεια.

εισβάλει στην Ελλάδα. Τότε χρησιμοποίησε ένα πινάκιο που ήταν καλυμμένο με κερί το οποίο το αφαίρεσε και έγραψε το μυστικό μήνυμα επάνω στο ξύλο του πίνακα. Στη συνέχεια το κάλυψε ξανά με κερί ώστε να φαίνεται κενό και το έστειλε στη Σπάρτη χωρίς να γίνει αντιληπτό το μήνυμά του από τους Πέρσες.
Το δεύτερο γεγονός αναφέρεται στον βασιλιά Δαρείο από την Σούσα ο οποίος ξύρισε το κεφάλι ενός από τους φυλακισμένους του και έγραψε επάνω στο κρανίο του ένα μυστικό μήνυμα. Όταν τα μαλλιά του φυλακισμένου μεγάλωσαν, τον έστειλε στον βασιλιά της Μιλήτου Αρισταγόρα χωρίς να εντοπιστεί το μήνυμα από κανέναν και έτσι ξυρίζοντας πάλι το κεφάλι του φυλακισμένου έλαβε μόνο εκείνος το μήνυμα από τον Δαρείο.
Ο Στρατηγός Αινείας αναφέρει επίσης πώς πολλά μηνύματα τα οποία έπρεπε να μείνουν απόκρυφα μυστικά τα έκρυβαν είτε σε σόλες παπουτσιών των αγγελιοφόρων είτε σε γυναικεία κοσμήματα όπως ήταν σκουλαρίκια. Ακόμη πρότεινε κάποιους τρόπους σταγανογράφησης που μοιάζουν αρκετά με αυτά της απόκρυψης πληροφοριών σε έγγραφα κειμένου. Ένας από αυτούς τους τρόπους ήταν να μπορέσουν να αποκρύψουν ένα μικρό κείμενο σε ένα ορατό κείμενο. Αυτό πραγματοποιούνταν με πολύ μικρές οπές κάτω ή πάνω από τα γράμματα, ή τροποποιώντας απλώς το ύψος της γραμματοσειράς, όπου αυτό κιόλας θεωρείται ένα χαρακτηριστικό κωδικοποίησης. Μια άλλη αρκετά δημοφιλής μέθοδος στεγανογραφίας, που χρησιμοποιήθηκε αρχικά από τους Ρωμαίους και έφτασε μέχρι και τον 2ο Παγκόσμιο Πόλεμο, ήταν αυτή του αόρατου μελανιού. Το αόρατο μελάνι φτιαχνόταν από τέτοια συστατικά τα οποία όταν θερμαίνονταν, αυτά έπαιρναν σκούρο χρώμα. Το φτιάχνανε από φυσικά συστατικά όπως το γάλα, ξύδι, χυμοί φρούτων, και ούρα. Με τον πέρασμα των χρόνων τα αόρατα μελάνια εξελίχθηκαν και φτιάχνονταν από χημικά συστατικά τα οποία ήταν πιο πολύπλοκο στο να εμφανίσεις το μήνυμα που κρύβανε. Με την πάροδο των χρόνων έρχεται και η εξέλιξη της τεχνολογίας και έτσι επιτυγχάνεται η αποκωδικοποίηση των αόρατων μελανιών, τα οποία φτιάχτηκαν με τέτοιο τρόπο που αντιδρούσαν μόνο σε συγκεκριμένες χημικές ουσίες, όπως για παράδειγμα η εμφάνιση φωτογραφιών. Αναφέροντας και πιο πάνω τον 2ο Παγκόσμιο πόλεμο οι Γερμανοί κατάσκοποι χρησιμοποίησαν τα Microdots. Τις μικροτελείες δηλαδή οι οποίες ήταν λιλιπούτιες φωτογραφίες υψηλής ανάλυσης και το μέγεθος τους ήταν ασήμαντο. Ήταν ακριβώς τόσο μικρό ώστε να μην τραβήξει ποτέ τη παραμικρή προσοχή. Διαβιβάστηκαν πολλές και μεγάλες σε όγκο πληροφορίες οι οποίες συμπεριλάμβαναν τόσο μηνύματα και σχέδια, όσο και φωτογραφίες. Μάλιστα όταν ανακαλύφθηκε αυτή η τεχνική είχε

απαγορευτεί από τις ΗΠΑ διακίνηση σταυρολέξων και γρίφων γιατί πίστευαν ότι θα μπορούσαν να είναι καμουφλαρισμένα με τέτοιο τρόπο που να διέρρεαν πληροφορίες-προειδοποιήσεις.

Ακόμα μια μέθοδος μηνυμάτων είναι αυτή των Null Ciphers (στον 2ο Παγκόσμιο). Ήταν εκείνη η τεχνική κατά την οποία μπορούσες να κρύψεις ένα μήνυμα μέσα σε ένα άλλο μήνυμα (ενθυλάκωση μηνύματος). Για παράδειγμα :

Today Anastasia, Kate, Elena got up nine, sadly.

Το μήνυμα το οποίο προκύπτει από το παραπάνω παράδειγμα παίρνοντας μόνο τα αρχικά της κάθε λέξης είναι αυτό : TAKE GUNS. Κάπως έτσι λοιπόν αναπτύχθηκε αυτή η τεχνική και μάλιστα όσο περνούσε ο καιρός ανακάλυπταν και άλλους τρόπους να κρύψουν μηνύματα εκτός από το να ακολουθούν το πρότυπο, κάθε πρώτο γράμμα από την λέξη δίνει το πρώτο γράμμα του μηνύματος, αλλά αλλάζοντας κιόλας και την σειρά που επιλέγουν τα γράμματα. Για παράδειγμα :

Susan says Gail lies. Matt lets Susan feel jovial. Elated (or) angry?

Χρησιμοποιώντας το πρότυπο (1,2,3,1,2,3 [κάθε γράμμα σε κάθε λέξη]) δίνει το μήνυμα: SAIL AT SEVEN.

Είναι δύσκολο και χρονοβόρο να παραχθούν τέτοια μηνύματα ενθυλάκωσης που να φαίνονται απολύτως φυσικά και να μην δημιουργούν υποψίες. Υπήρχε τότε, όπως και σήμερα, η τεχνική της ανίχνευσης υπόπτων μηνυμάτων μέσω κάποιων ειδικών φίλτρων—μιας αυτοματοποιημένης διαδικασίας. Για αυτό τον λόγο αν έρθουμε στο σήμερα δεν θα δούμε αυτή τη μέθοδο να χρησιμοποιείται. Τουλάχιστον όχι τόσο συχνά. Μια άλλη έξυπνη μέθοδος για να ανταλλάσσουν μηνύματα μεταξύ τους χωρίς να γίνονται αντιληπτοί από τους φρουρούς ήταν και οι αιχμάλωτοι πολέμου. Εκείνοι με την σειρά τους είχαν μετατρέψει το αλφάβητο σε σήματα Morse όπως φαίνεται στην Εικόνα 1 :

A	.-	M	--	Y	-.--	6	-....
B	-...	N	-.	Z	--..	7	--...
C	-.-.	O	---	Ä	.-.-	8	---..
D	-..	P	.--.	Ö	---.	9	----.
E	.	Q	--.-	Ü	..--	.	.-.-.-
F	..-.	R	.-.	Ch	----	,	--..--
G	--.	S	...	0	-----	?	..--..
H	....	T	-	1	.----	!	..__.
I	..	U	..-	2	..---	:	---...
J	.---	V	...-	3	...--	“	.-..-.
K	-.-	W	.--	4	....-	‘	.----.
L	.-..	X	-..-	5	.....	=	-...-

Εικόνα 1 : Αναπαράσταση Αλφάβητου με σήματα Morse

Και κάπως έτσι με αυτή την μετατροπή του αλφάβητου οι αιχμάλωτοι κατάφερναν όχι μόνο να συνομιλούν μεταξύ τους αλλά σχεδίαζαν κιόλας πώς να ξεφύγουν. Σαφώς αυτή η μέθοδος δεν διήρκησε πολύ γιατί οι φρουροί είτε άλλαζαν τα σήματα για να τους μπερδέψουν είτε τους απαγόρευαν ακόμα και σκαλίζουν επάνω στους τοίχους. Πλέον η στεγανογραφία χρησιμοποιείται κυρίως στους υπολογιστές και στην απόκρυψη δεδομένων μέσα σε αρχεία. Πολλά εργαλεία και τεχνικές έχουν αναπτυχθεί βασιζόμενα στις ήδη υπάρχουσες τεχνικές της παλαιάς στεγανογραφίας.

1,4 Βασικές αρχές Στεγανογραφίας

Στην Στεγανογραφία υπάρχουν 3 (γενικές) βασικές αρχές που μπορούν να χρησιμοποιηθούν προκειμένου να αξιολογηθεί πόσο αποτελέσματική θα μπορούσε να είναι μια τεχνική :

1. Η ποσότητα των δεδομένων ή το μέγεθος του μηνύματος που μπορεί να ενσωματωθεί στο μέσο μεταφοράς. Όσα περισσότερα δεδομένα μπορούμε να κρύψουμε εφαρμόζοντας μια τεχνική, τόσο καλύτερη θεωρείται.

2. Η δυσκολία ανίχνευσης των κρυμμένων δεδομένων ή του μηνύματος, η οποία σχετίζεται ιδιαίτερα με το μέγεθός αυτών. Αυτό σημαίνει πως όσο αυξάνεται η

ποσότητα των δεδομένων που μπορούν να ενσωματωθούν τόσο αυξάνεται και η πιθανότητα ανίχνευσης του κρυμμένου μηνύματος.

3. Η δυσκολία εξαγωγής του κρυμμένου μηνύματος από κάποιο τρίτο πρόσωπο.

Όσον αφορά την Στεγανογραφία που χρησιμοποιούμε στους υπολογιστές είναι βασισμένη σε δύο αρχές.

1. Τα αρχεία που περιέχουν εικόνες ή ήχο μπορούν να αλλάξουν τόσο ώστε να μην αλλοιωθεί η λειτουργικότητά τους. Σε αντίθεση με τα προγράμματα τα οποία πρέπει να είναι ιδιαίτερα ακριβή για να καταφέρουν να λειτουργήσουν σωστά.
2. Ο άνθρωπος αδυνατεί να διακρίνει αν έχουν υπάρξει κάποιες αλλαγές στο χρώμα μιας εικόνας ή στην ποιότητα του ήχου. Ο ήχος καθίσταται εύκολος στο να χρησιμοποιηθεί και να εφαρμοστεί στα δεδομένα αυτά που εμπεριέχουν επιπλέον πληροφορίες, περιττές. Η τροποποίηση που γίνεται στις ψηφιακές εικόνες, αλλάζοντας την τιμή του λιγότερου σημαντικού bit (LSB) του χρώματος του εικονοκυττάρου (Pixel) δεν γίνεται αντιληπτή με γυμνό μάτι.

1.5 Τεχνικές Στεγανογραφίας

1.5.1 Στεγανάλυση

Η Στεγανάλυση χρησιμοποιείται για να ανιχνεύσει ένα κρυμμένο μήνυμα αλλά όχι να το εξάγει. Με λίγα λόγια είναι η διαδικασία που ανιχνεύει την στεγανογραφία, αν έχει χρησιμοποιηθεί. Αναγνωρίζει αν σε κάποιο μέσο έχει ενσωματωθεί κάποια κρυμμένη πληροφορία χωρίς να σημαίνει ότι θα ΄΄φανερώσει΄΄ το κρυμμένο μήνυμα. Εάν κάποιος έχει το αρχικό μη τροποποιημένο πχ αρχείο τότε μόνο μπορεί να το συγκρίνει με το ύποπτο σε στεναγογραφία αρχείο. Αν λοιπόν έχει το γνήσιο αρχείο μπορεί να συγκρίνει από τα μεγέθη τους μέχρι και την ημερομηνία τροποποίησής τους. Είναι η τεχνική που χρησιμοποιείται σαν απάντηση στην Στεγανογραφία όπως και η αποκρυπτογράφηση στην Κρυπτογραφία. Αυτές τις δύο έννοιες θα τις συγκρίνουμε και παρακάτω.

1.5.2 Εισαγωγή κειμένου σε εικόνα (LSB)

Μια εικόνα ισούται με χίλιες λέξεις. Ένα ρητό που χρησιμοποιείται μεταφορικά στην καθημερινότητα των ανθρώπων. Παρακάτω παρατίθεται, στην Εικόνα 2 ένα

χαρακτηριστικό παράδειγμα, ο πίνακας του Leonardo Da Vinci [2], η γνώστη σε όλον τον κόσμο Μόνα Λίζα [3] με το αινιγματικό της χαμόγελο. Δίχασε αρκετούς, στεγανογράφους, κρυπτογράφους στο αν τελικά ο ζωγράφος έχει κρύψει κάποιο μήνυμα μέσα στο χαμόγελό της. Μελέτες δείχνουν πως όντως ο καλλιτέχνης έχει κρύψει κάποιο μήνυμα αλλά ακόμα είναι σε εξέλιξη η εμφάνισή του (του μηνύματος).

Εικόνα 2 : Μόνα Λίζα

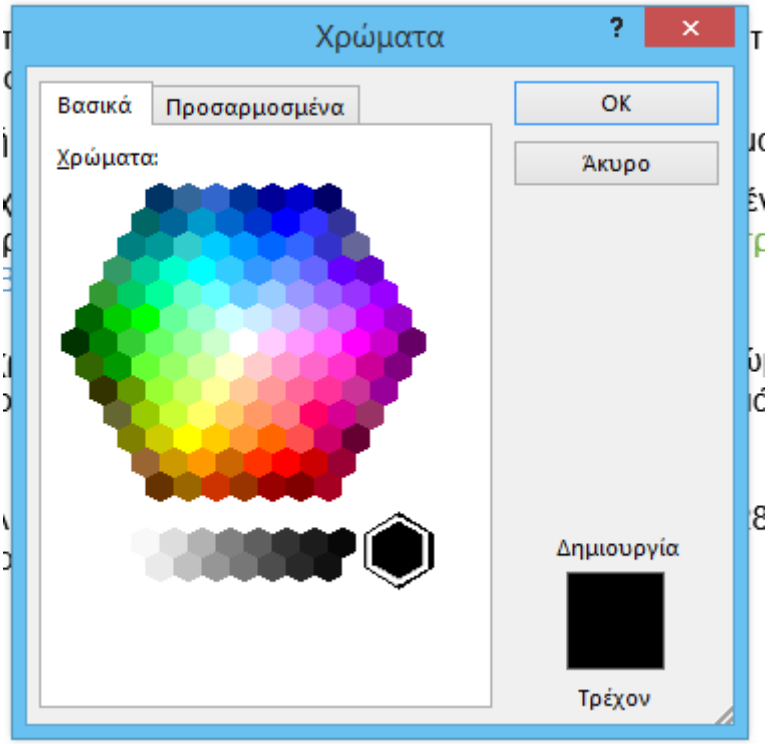
[2] Ο Λεονάρντο ντα Βίντσι ήταν Ιταλός αρχιτέκτονας, ζωγράφος, γλύπτης, μουσικός, εφευρέτης, μηχανικός, ανατόμος, γεωμέτρης, παλαιοντολόγος και γιατρός, που έζησε την περίοδο της Αναγέννησης.

[3] Η Μόνα Λίζα είναι προσωπογραφία που ζωγράφισε ο Ιταλός καλλιτέχνης Λεονάρντο ντα Βίντσι. Πρόκειται για ελαιογραφία σε ξύλο λεύκης, που ολοκληρώθηκε μέσα στη χρονική περίοδο 1503-1519

Από ότι φαίνεται με την Στεγανογραφία δεν είναι και τόσο μεταφορική η έννοια του ρητού διότι μέσα σε μια εικόνα μπορεί να κρύψει κανείς από ένα μήνυμα μέχρι και έναν ολόκληρο χάρτη. Ουσιαστικά μια εικόνα αποτελείται από αριθμούς.

Τεχνική απόκρυψης πληροφορίας-μηνύματος σε εικόνα με την μέθοδο λιγότερου σημαντικού ψηφίου. Η επιλογή γίνεται συνήθως μεταξύ 3 διαφορετικών ειδών χρωμάτων :

- 24-bit χρωματισμό: κάθε pixel περιγράφεται από 3 bytes, ένα byte για κάθε χρώμα από τα βασικά, που είναι το : κόκκινο (R), πράσινο (G), μπλε (B), που δίνονται από 8-bit (256 τιμές) το κάθε ένα
- 8-bit χρώμα: κάθε pixel μπορεί να έχει 256 (28) χρώματα, που επιλέγονται από μια παλέτα ή αλλιώς από ένα πίνακα χρωμάτων
- 8-bit κλίμακα του γκρίζου : κάθε pixel μπορεί να έχει 256 (28) σκιές της κλίμακας του γκρίζου



Εικόνα 3 : Όλα τα χρώματα, και η κλίμακα του γκρίζου

Εφόσον κάθε χρώμα μπορεί να περιγραφεί με 256 διαφορετικές τιμές, έτσι ένα μαύρο στοιχείο θα έχει τιμή (0,0,0) και το άσπρο θα έχει (255,255,255) όπου στο δυαδικό θα είναι ο αριθμός (11111111,11111111,11111111)

Παράδειγμα : **LSB (Least Signifigant Bit)**

Εχουμε μια εικόνα 24 bit όπου κάθε εικονοστοιχείο (pixel) αποτελείται από 3 αριθμούς των 8 bit οι οποίοι υποδηλώνουν τις τιμές των χρωμάτων του εικονοστοιχείου (RGB = Red Green Blue). Οι τιμές των LSB παρουσιάζονται **υπερτονισμένες** (**bold**). Έστω ότι θέλουμε να εισάγουμε τον χαρακτήρα 1 (ASCII) που στο δυαδικό έχει την τιμή **00110001**. Εισάγουμε τα δυαδικά ψηφία (bits) του χαρακτήρα αντικαθιστώντας τις ήδη υπάρχουσες τιμές των εικονοστοιχείων (pixel). Παρακάτω βλέπουμε τις τιμές τριών διαδοχικών εικονοστοιχείων.

Πριν την τροποποίηση :

- *1ο pixel : 1001010**1** . 0000110**1** . 1100100**1***
- *2ο pixel : 1001011**0** . 0000111**1** . 1110101**0***
- *3ο pixel : 1001111**1** . 0001000**0** . 1100101**1***

Μετά την τροποποίηση :

- *1ο pixel : 1001010**0** . 0000110**0** . 1100100**1***
- *2ο pixel : 1001011**1** . 0000111**0** . 1100101**0***
- *3ο pixel : 1001111**0** . 0001000**1** . 1100101**1***

Όπως βλέπουμε στον παραπάνω πίνακα τα λιγότερο σημαντικά δυαδικά ψηφία (LSB) έχουν αντικατασταθεί με τα δυαδικά ψηφία του χαρακτήρα ‘1’ και τυχόν πλεονασμοί μένουν αναλλοίωτοι (υπογραμμισμένο bit). Υποθετικά σε μια εικόνα διαστάσεων 1024x768 εάν κρύψουμε έναν χαρακτήρα σε κάθε τρία εικονοστοιχεία τότε μπορούμε να κρύψουμε έως και 262.144 χαρακτήρες. Η μέθοδος αυτή μπορεί να λειτουργήσει μόνο με μορφοποιήσεις εικόνων, των οποίων τα περιεχόμενα είναι οι τιμές των χρωμάτων στα εικονοστοιχεία (π.χ. PNG, BMP). Αντιθέτως τα περιεχόμενα μιας εικόνας JPEG αποτελούν αριθμούς συμπίεσης και γι’ αυτό δεν υποστηρίζεται η παραπάνω μέθοδος.

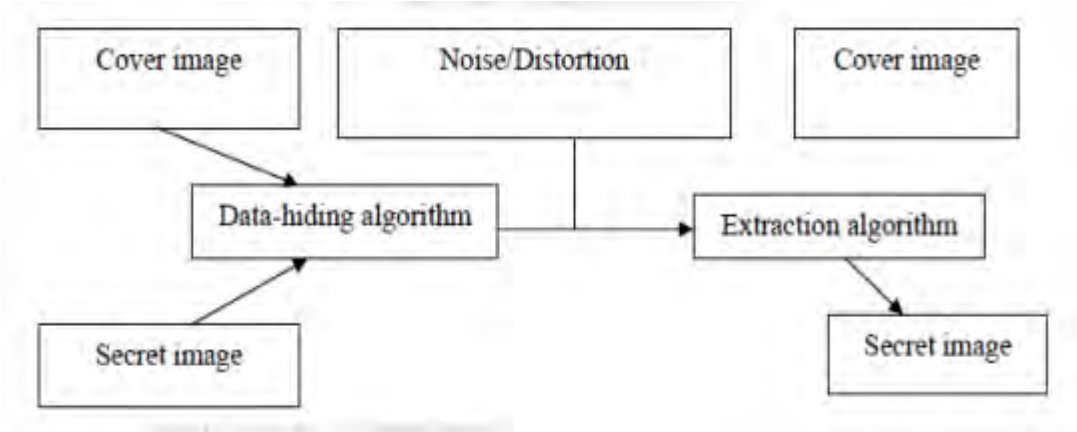
1.5.3 Εισαγωγή κειμένου σε εικόνα JPEG

Η μορφοποίηση εικόνας JPEG χρησιμοποιεί τον διακριτό μετασχηματισμό συνημίτονου (DCT) (discrete cosine transform) για τον εντοπισμό 64 συντελεστών DCT σε διαδοχικά 8x8 μπλοκ εικονοστοιχείων. Για την εισαγωγή κειμένου μέσα σε μια εικόνα JPEG εκμεταλλευόμαστε την διαδικασία αυτή αλλοιώνοντας τις τιμές των DCT συντελεστών.

1.5.4 Εισαγωγή κειμένου σε αρχείο

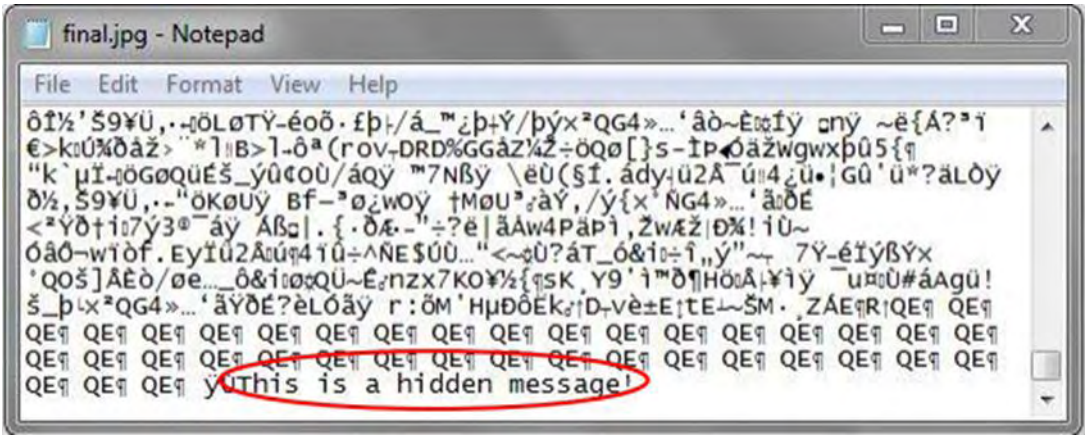
Στα λειτουργικά συστήματα Windows έχουμε την δυνατότητα να κρύψουμε ένα αρχείο κειμένου μέσα σε ένα οποιοδήποτε αρχείο (στην παρούσα περίπτωση εικόνα) γράφοντας στην γραμμή εντολών:

>copy C:\cover.jpg/b+ C:\message.txt/b C:\final.jpg



Εικόνα 4 : Αναπαράσταση Στεγανογραφικού γραφήματος

Όπου **cover.jpg** είναι το αρχείο εικόνας μέσα στο οποίο θα κρύψουμε το **message.txt** που είναι ένα απλό αρχείο κειμένου. Το **final.jpg** είναι η εικόνα-συνδυασμός αυτών των δύο. Στην συνέχεια για να ανασύρουμε το κρυμμένο κείμενο απλώς ανοίγουμε το final.jpg με έναν κειμενογράφο, όπως φαίνεται στην Εικόνα 4 :



Εικόνα 5 :Το final.jpg στον κειμενογράφο

Ο τρόπος αυτός εκμεταλλεύεται το γεγονός ότι το σύστημα παραβλέπει τυχόν δεδομένα που βρίσκονται μετά από το EOF (End Of File). Αυτό έχει ως αποτέλεσμα η εικόνα (στην παρούσα περίπτωση) να εκτελείται κανονικά και το κρυφό μήνυμα να ανασύρεται με έναν κειμενογράφο.

1.5.5 Στεγανογραφία ήχου

Οι περισσότερες από τις τεχνικές που χρησιμοποιούνται για την στεγανογράφηση εικόνας χρησιμοποιούνται και για την στεγανογράφηση ήχου. Κάποιες από αυτές είναι οι εξής :

- χαμηλά bit κωδικοποίησης τα οποία είναι παρόμοια με την μέθοδο LSB που χρησιμοποιούνται γενικά στις εικόνες. Αντικαθιστά το λιγότερο σημαντικό Bit (LSB) ενός ηχητικού σήματος. Το ανθρώπινο ακουστικό σύστημα δεν λειτουργεί πέρα από ένα δυναμικό εύρος συχνοτήτων. Τα χαμηλά bit κωδικοποίησης δεν είναι συνήθως αισθητά στο ανθρώπινο αυτί, γιατί δεν δημιουργούν σημαντικές αλλαγές στην ακουστική του ήχου. Αυτό συνεπάγεται την εκμετάλλευση των ανθρώπινων περιορισμών, χρησιμοποιώντας συχνότητες που δεν ακούγονται στο ανθρώπινο αυτί. Δουλεύοντας με κάθε συχνότητα πάνω από 20,000 Hz, τα μηνύματα που μπορεί να κρύβονται δεν θα εντοπιστούν από τους ελέγχους του ανθρώπου.
- Spread Spectrum : είναι μια άλλη μέθοδος που χρησιμοποιείται για να συγκαλύψει πληροφορίες στο εσωτερικό ενός αρχείο ήχου. Αυτή η μέθοδος λειτουργεί με την προσθήκη τυχαία θορύβου στο σήμα της πληροφορίας που είναι να κρύψουν μέσα σε ένα μεταφορέα και σε όλη την φάσμα συχνοτήτων.

- Echo : είναι τα δεδομένα που μπορούν να κρύβονται μέσα σε ένα αρχείο ήχου. Αυτή η μέθοδος χρησιμοποιεί την ηχώ σε αρχεία ήχου, προκειμένου να αποκρύψουμε πληροφορίες. Με την απλή προσθήκη επιπλέον ήχου μέσα σε ένα αρχείο ήχου. Το κομμάτι που καθιστά τη μέθοδο αυτή καλύτερη από άλλες μεθόδους είναι ότι μπορεί να βελτιώσει πραγματικά τον ήχο του ήχου μέσα στο αρχείο.
- Masking : Αυτή η τεχνική έχει τη μεγαλύτερη ικανότητα ενσωμάτωσης, αλλά και πάλι είναι λιγότερο ισχυρή. Αποθηκεύει δεδομένα σε ασήμαντες περιοχές του φάσματος.

Παράδειγμα

Έστω ότι υπάρχει ένα αρχείο ήχου σε .wav μορφή με τα εξής χαρακτηριστικά : 45200 Hz,16-bit stereo του ενός λεπτού, τότε έχουμε διαστάσεις αρχείου = (16-bit x 45200 Hz x 60sec) x 2 (είναι δικάναλο)= 84672000 bit

Έχουμε συνεπώς μέγεθος για να κρύψουμε στο αρχείο (χρησιμοποιώντας τα 2τελευταία LSB) = 84672000 bit / 16 x 2 = 86784000 bit. Δεν κρύβεται μια πληροφορία σε ένα wav και μετά συμπιέζεται, διότι υπάρχει μεγάλη πιθανότητα να χαθούν τα δεδομένα. Σε γενικές γραμμές είναι πολύ σημαντικό να μην αλλοιώνεται η ποιότητα του ήχου, ή για παράδειγμα ένα τραγούδι ακούγεται διαφορετικά από το αρχικό γιατί δημιουργούνται υποψίες ότι έχει πειραχτεί.

1.5.6 Στεγανογραφία βίντεο

Το βίντεο είναι γενικά μια συλλογή από εικόνες κι ήχους συνεπώς οι περισσότερες τεχνικές που εφαρμόζονται στην στεγανογραφία ήχου και στην στεγανογραφία εικόνας μπορούν να εφαρμοστούν και σε αρχεία video. Το video περιέχει μεγάλο όγκο πληροφορίας και δεδομένων, η συνεχής ροή κινούμενων εικόνων και ήχου αποτελεί πλεονέκτημα στη στεγανογραφία γιατί η οποιαδήποτε αλλαγή θα μπορούσε να περάσει απαρατήρητη. Η τεχνική στεγανογραφίας βίντεο συνδυάζει με άλλα λόγια την τεχνική στεγανογραφίας εικόνας (αφού πρόκειται για frames) και την τεχνική στεγανογραφίας ήχου.

1.5.7 Watermarking

Η ψηφιακή υδατογράφηση (Watermarking) είναι η διαδικασία κατά την οποία

ενσωματώνεται μέσα σε ένα αρχείο εικόνα, ήχου, βίντεο ένα ψηφιακό σήμα. Το ψηφιακό σήμα βρίσκεται μέσα σε κάθε αναλλοίωτο αντίγραφο του αρχικού αρχείου και με αυτόν τον τρόπο έχει το ρόλο της ψηφιακής υπογραφής στο αρχείο. Είναι δηλαδή σαν το αρχείο να μαρκάρεται και να αναγνωρίζεται από τον δημιουργό του. Κάποιες από τις τεχνικές του watermarking είναι οι εξής :

- Ανθεκτικότητα : ένα ψηφιακό σήμα ονομάζεται ανθεκτικό όταν παραμένει αναλλοίωτο μετά από διάφορες μορφοποιήσεις, όπως είναι για παράδειγμα περικοπή εικόνας κτλπ καθώς και σε διάφορους μετασχηματισμούς.
- Ικανότητα : Το μήκος του ενσωματωμένου μηνύματος προσδιορίζει δύο διαφορετικές κύριες κατηγορίες, η μία είναι που το υδατογράφημα έχει 0-bit, και η άλλη όταν έχει n-bit.
- Μέθοδος Ενσωμάτωσης : είναι η κβάντωση, ευρέως φάσματος, διαμόρφωση πλάτους.

1.6 Κρυπτογραφία

Η κρυπτογραφία ασχολείται με τη μελέτη, την ανάπτυξη και τη χρήση τεχνικών κρυπτογράφησης [4] και αποκρυπτογράφησης [5] με σκοπό την απόκρυψη του περιεχομένου των μηνυμάτων.

1.6.1 Στόχος Κρυπτογραφίας

Στόχος της κρυπτογραφίας είναι να δώσει τη δυνατότητα σε δύο πρόσωπα, να επικοινωνήσουν μέσα από ένα μη ασφαλές κανάλι με τέτοιο τρόπο ώστε ένα τρίτο πρόσωπο, μη εξουσιοδοτημένο (ένας αντίπαλος), να μην μπορεί να παρεμβληθεί στην επικοινωνία ή να κατανοήσει το περιεχόμενο των μηνυμάτων.

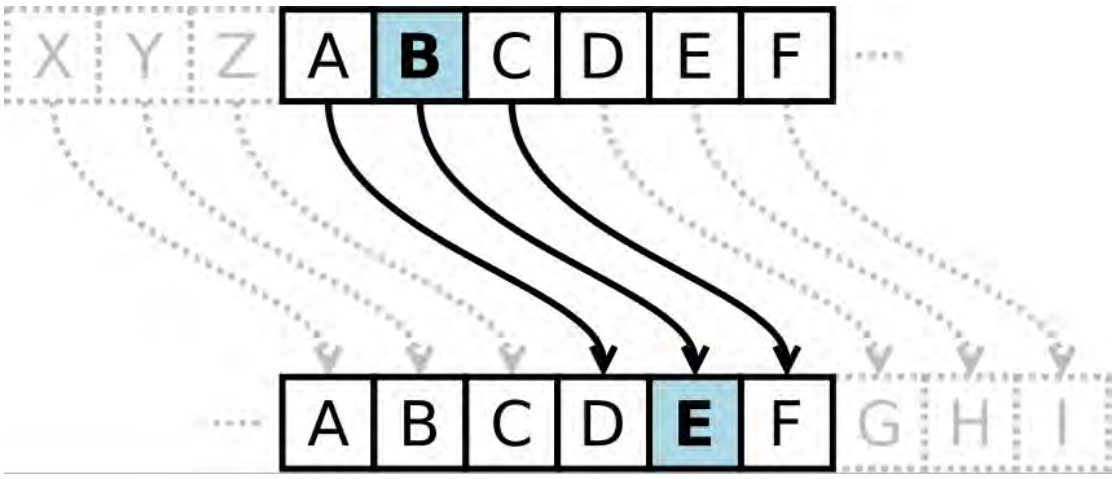
1.6.2 Ιστορική Αναδρομή

[4] είναι η διαδικασία της μετατροπής ενός μηνύματος, μεταξύ του αποστολέα και του παραλήπτη, σε μορφή που να μην είναι αναγνώσιμη από τρίτους.
[5] είναι η αντίστροφη διαδικασία όπου από το κρυπτογραφημένο κείμενο παράγεται το αρχικό μήνυμα.

Τα κρυπτογραφήματα εμφανίσθηκαν πρώτα στους Πέρσες και στους Έλληνες. Ο Άρπαγος ειδοποιεί τον Κύρο τον πρεσβύτερο μ' ένα σκοτωμένο λαγό, που στην κοιλιά του υπήρχε κάποιο μήνυμα. Χρησιμοποιήθηκαν διάφορες μέθοδοι ώστε τα μηνύματα να μπορούν να διαβαστούν μόνο απ' τον παραλήπτη και να είναι ακατανόητα σε βαθμό που να γίνονται άχρηστα για οποιονδήποτε άλλο. Μερικές μέθοδοι απ' αυτές δείχνουν πολύ απλοϊκές σήμερα, αλλά κάποιες άλλες δεν έχουν αποκρυπτογραφηθεί ακόμα. Στην αρχαία Σπάρτη για να στείλουν στρατιωτικά μηνύματα χρησιμοποιούσαν κύλινδρο που γύρω του ήταν τυλιγμένη μια στενή δερμάτινη λωρίδα σε σειρές. Αφαιρώντας τον κύλινδρο
έμενε η λωρίδα που μπορούσε να ξαναδιαβαστεί μόνο αν τυλιγόταν με τον ίδιο τρόπο πάνω σε κύλινδρο ίδιας διαμέτρου. Η λεγόμενη Σπαρτιατική σκυτάλη Εικόνα 5.

Εικόνα 6: Σπαρτιατική σκυτάλη

Ο Ιούλιος Καίσαρας έγραφε αντικαθιστώντας τα γράμματα του κειμένου, με γράμματα, που βρίσκονται 3 θέσεις μετά. Ήταν η πρώτη μέθοδος υποκατάστασης γραμμάτων. Σήμερα, το σύστημα κρυπτογράφησης που στηρίζεται στην αντικατάσταση των γραμμάτων του αλφαβήτου με άλλα που βρίσκονται σε καθορισμένο αριθμό θέσης πριν ή μετά, λέγεται κρυπτοσύστημα αντικατάστασης του Καίσαρα (Εικόνα 6)



Εικόνα 7 : Αλγόριθμος του Καίσαρα

Η εφαρμογή του κώδικα Καίσαρα συνίσταται στην αντικατάσταση κάθε γράμματος του κειμένου με ένα άλλο το οποίο έχει σταθερή απόσταση από αυτό στο αλφάβητο. Στην Εικόνα 6 χρησιμοποιείται μετατόπιση τριών θέσεων, έτσι ώστε το Β του κειμένου να γίνεται Ε στο κρυπτογραφημένο κείμενο.

Στο τέλος του Α' Παγκοσμίου Πολέμου ανακαλύφθηκε η μηχανή Enigma από τον Γερμανό Arthur Scherbius. Πρόκειται για μια οικογένεια ηλεκτρομηχανολογικών μηχανών, που χρησιμοποιούνται για την κρυπτογράφηση και την αποκρυπτογράφηση μηνυμάτων. Τα πρώτα μοντέλα είχαν χρησιμοποιηθεί για εμπορικούς σκοπούς από τις αρχές του 1920 και αργότερα εγκρίθηκαν από τις στρατιωτικές και κυβερνητικές υπηρεσίες σε πολλές χώρες - κυρίως από τη ναζιστική Γερμανία - πριν και κατά τη διάρκεια του Β' Παγκοσμίου Πολέμου. Πολλά και διαφορετικά μοντέλα Enigma παρήχθησαν, αλλά τα γερμανικά στρατιωτικά μοντέλα είναι αυτά που συζητιούνται συχνότατα και συνήθως αντιπροσωπεύουν τον θρύλο των μηχανών αυτών. Τον Δεκέμβριο του 1932, η μυστική υπηρεσία της Πολωνίας "έσπασε" πρώτη τους αλγόριθμους κρυπτογράφησης της Γερμανίας. Πέντε εβδομάδες πριν από το ξέσπασμα του Β' Παγκοσμίου Πολέμου, οι Πολωνοί παρουσίασαν τις τεχνικές αποκρυπτογράφησης του Enigma και τον αντίστοιχο εξοπλισμό σε γαλλικές και βρετανικές μυστικές υπηρεσίες. Χάρη σε αυτό το γεγονός, κατά τη διάρκεια του πολέμου οι κρυπτογράφοι ήταν σε θέση να αποκρυπτογραφήσουν ένα τεράστιο αριθμό μηνυμάτων χρησιμοποιώντας τη μηχανή Enigma.

Εικόνα 8 : Μηχανή Enigma

1.6.3 Βασικές λειτουργίες Κρυπτογραφίας

Η κρυπτογραφία παρέχει 4 βασικές λειτουργίες :

1. Εμπιστευτικότητα: Η πληροφορία προς μετάδοση είναι προσβάσιμη μόνο στα εξουσιοδοτημένα μέλη. Η πληροφορία είναι ακατανόητη σε κάποιον τρίτο.
2. Ακεραιότητα: Η πληροφορία μπορεί να αλλοιωθεί μόνο από τα εξουσιοδοτημ ένα μέλη και δεν μπορεί να αλλοιώνεται χωρίς την ανίχνευση της αλλοίωσης.
3. Μη απάρνηση: Ο αποστολέας ή ο παραλήπτης της πληροφορίας δεν μπορεί να αρνηθεί την αυθεντικότητα της μετάδοσης ή της δημιουργίας της.
4. Πιστοποίηση: Οι αποστολέας και παραλήπτης μπορούν να εξακριβώνουν τις τ αυτότητές τους καθώς και την πηγή και τον προορισμό της πληροφορίας με δ ιαβεβαίωση ότι οι ταυτότητές τους δεν είναι πλαστές.

Γενικά στη σημερινή εποχή υπάρχει τεράστια ανάγκη να μπορούμε να προστατεύσουμε τα προσωπικά μας δεδομένα και όχι μόνο. Και αυτήν την ανάγκη δεν την έχουν μόνο οι επιχειρήσεις και οι υπηρεσίες (πχ Τράπεζες) αλλά και οι υπόλοιποι άνθρωποι στην καθημερινότητά τους. Αυτό επιτυγχάνετε και με την κρυπτογραφία αλλά και με την στεγανογραφία.

1.6.4 Σύγκριση Στεγανογραφίας-Κρυπτογραφίας

Η Στεγανογραφία με την Κρυπτογραφία θα μπορούσε να πει κάποιος ότι είναι δυο έννοιες ίδιες. Δεν είναι όμως. Ο στόχος τους είναι κοινός, ανταλλαγή μηνυμάτων χωρίς να γίνονται αντιληπτά από άτομα που δεν θέλουμε, αλλά διαφέρουν στο πώς λειτουργούν. Η στεγανογραφία είναι κλάδος της κρυπτογραφίας, την συμπληρώνει με άλλα λόγια. Μια μεγάλη διαφορά μεταξύ αυτών των 2 είναι πως ότι η στεγανογραφία αποκρύπτει την ύπαρξη του μηνύματος, ενώ η κρυπτογραφία μετασχηματίζει το μήνυμα ώστε να το καθιστά ακατανόητο σε οποιονδήποτε τρίτο. Για να αποστείλει κανείς κρυπτογραφημένες πληροφορίες σε ένα άλλο πρόσωπο, θα πρέπει να τις κωδικοποιήσει με ένα κλειδί και αυτό το κλειδί να είναι γνωστό και στους δύο προκειμένου να επιτευχθεί η αποκωδικοποίηση των πληροφοριών αυτών και να αναγνωστεί το μήνυμα. Η στεγανογραφία πάνω σε αυτό το κομμάτι προσθέτει ένα επιπλέον επίπεδο ενισχύοντας έτσι την ασφάλεια των ευαίσθητων δεδομένων που θα αποσταλούν. Στην περίπτωση που κάποιος τρίτος αποκτήσει πρόσβαση σε κάποιο αρχείο το οποίο προστατεύεται από κωδικό, η χρήση των κατάλληλων εργαλείων μπορεί να τον οδηγήσουν στην αποκάλυψη του κωδικού προστασίας. Αν χρησιμοποιηθούν ακόμα πιο πολύπλοκοι κωδικοί θα υπάρξει μεγαλύτερο χρονικό διάστημα εώς και χρόνια μέχρι να τους <<σπάσουν>>.

Αυτό που κάνει η στεγανογραφία ουσιαστικά είναι :

- Ενσωμάτωση μυστικής πληροφορίας σε ένα αρχικό αντικείμενο (cover object)
- Στεγανογραφικό κλειδί (stego key)
- Στεγανογραφημένο αντικείμενο (stego object)
- Στεγανογραφική χωρητικότητα(cover capacity)

Οι λόγοι που οδήγησαν στην ανάπτυξη εφαρμογών είναι:

- Να μπορούν να μεταδίδουν πληροφορίες χωρίς να γίνονται αντιληπτοί από τρίτους.
- Αν γίνουν αντιληπτοί να μην υπάρχει η δυνατότητα να ερμηνευθεί το μήνυμα που μετέδωσαν.
- Αν τελικά υποκλαπεί το μήνυμα, ο παραβάτης να υφίστανται τις συνέπειες του νόμου.

- Αν αλλοιωθούν τα δεδομένα, να υπάρχει η δυνατότητα επαναφοράς στην αρχική τους μορφή.
- Αν διεκδικηθεί η ιδιοκτησία τους, να μπορούν να αποδείξουν την κυριότητα τους.

Η ανθεκτικότητα [6], η αντοχή [7] και η ευρωστία [8] είναι τρία χαρακτηριστικά που έχουν επιπτώσεις στη στεγανογραφία και τη χρησιμότητα της.

Παράδειγμα : **Καίσαρα**

Παρακάτω υπάρχει ένα κρυπτογραφημένο μήνυμα, με ελληνικό αλφάβητο.

ΝΧΖΞΚΦ ΝΠΡΤΖΜ !

Το κλειδί που δίνεται για να το αποκρυπτογραφήσουμε είναι να μεταθέσουμε τα γράμματα στο αλφάβητο +3 θέσεις. Δηλαδή το Ν αν το δούμε στην Εικόνα 7 κάτω και πάμε 3 θέσεις μπροστά δίνει το Π.

[6] αναφέρεται στο κατά πόσο θα μπορέσουν να μείνουν ανέγγιχτα τα ενσωματωμένα στοιχεία, εάν υπάρχει κάποιος μετασχηματισμός στη στεγο-εικόνα.

[7] είναι ζωτικής σημασίας για προστασία της πνευματικής ιδιοκτησίας, επειδή κάποιοι θα προσπαθήσουν να φιλτράρουν και να καταστρέψουν κάθε πληροφορία ενσωματωμένη σε εικόνες. Το μόνο μειονέκτημα που προσφέρει είναι μία υψηλή επιβάρυνση για μικρή πληροφορία και αν η μέθοδος αποκαλυφθεί δεν παρέχεται καμία προστασία.

[8] Πέρα από ευρωστία της καταστροφής, η παραποίηση αντοχής αναφέρεται στην δυσκολία για έναν εισβολέα να μεταβάλει ή να σφυρηλατήσει ένα μήνυμα τη στιγμή που θα έχει ενσωματωθεί σε μία στέγο -εικόνα. Όπως ένα πειρατικό αντικαθιστά ένα σήμα πνευματικής ιδιοκτησίας με μία διεκδίκηση της νόμιμης ιδιοκτησίας.

Εικόνα 9 : Αλφάβητο

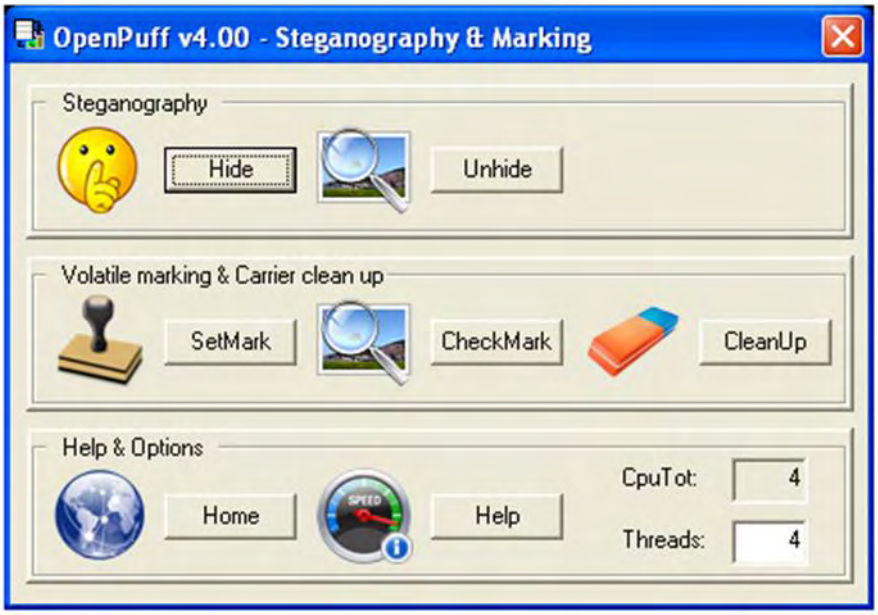
Άρα έχοντας και το κλειδί (+3 θέσεις) το μήνυμα που δίνεται αποκρυπτογραφημένο είναι το: ***ΠΑΙΡΝΩ ΠΤΥΧΙΟ !***

Κεφάλαιο 2ο
Εφαρμογές Στεγανογραφίας

Στο κεφάλαιο αυτό θα μελετηθούν κάποιες από τις πιο γνωστές εφαρμογές που χρησιμοποιούνται στην στεγανογραφία καθώς και οι ιστότοποι από πού μπορεί κάποιος να τα κατεβάσει.

2.1 OpenPuff 4.00

Εφαρμογή μέσω της οποίας μπορεί κάποιος να αποκρύψει αρχεία μέσα σε άλλα αρχεία. Με το σύστημα αυτό τα αρχεία που κρύβονται δεν είναι καν εμφανή αν δεν γνωρίζει κανείς που βρίσκονται καθώς τα μόνα αρχεία που φαίνονται είναι τα αρχεία στα οποία είναι μέσα κρυμμένα. Τα αρχεία, που περιέχουν τα κρυμμένα, είναι κανονικά (αρχεία) τα οποία λειτουργούν πλήρως ακόμα και όταν έχουν άλλα αρχεία κρυμμένα μέσα τους. Η διαδικασία υποστηρίζεται για ορισμένους τύπους αρχείων (στους οποίους μπορείτε να κρύφτούν άλλα μέσα) οι οποίοι όμως είναι αρκετοί και κοινής χρήσης (BMP, JPG, PCX, PNG, SWF, PDF μεταξύ άλλων).



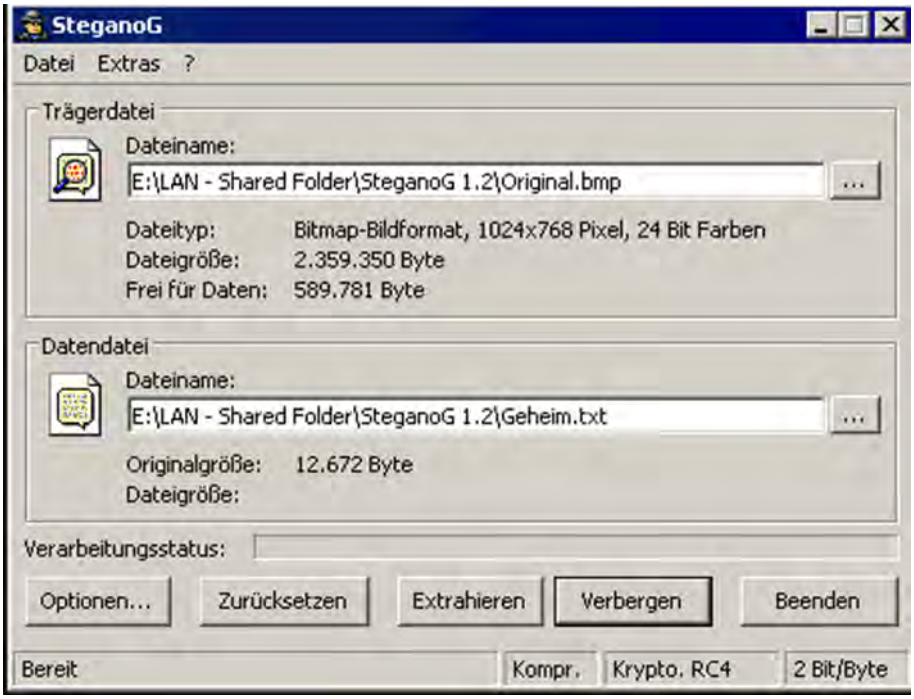
Εικόνα 10 : Αρχική οθόνη Open Puff

http://embeddedsw.net/OpenPuff_Steganography_Home.html

Υποστηρίζει Windows XP και νεότερα.

2.2 SteganoG

Το SteganoG είναι μια εφαρμογή που επιτρέπει στο χρήστη να αποκρύψει αρχεία μέσα σε αρχεία εικόνας τύπου BMP. Δίνεται η δυνατότητα κρυπτογράφησης τύπου RC4[9], Blofish[10]. Ιδανική εφαρμογή για αρχεία τα οποία δεν φαίνονται στον υπολογιστή του κατόχου.



Εικόνα 11 : Πρόγραμμα SteganoG

2.3 iStegano 2005

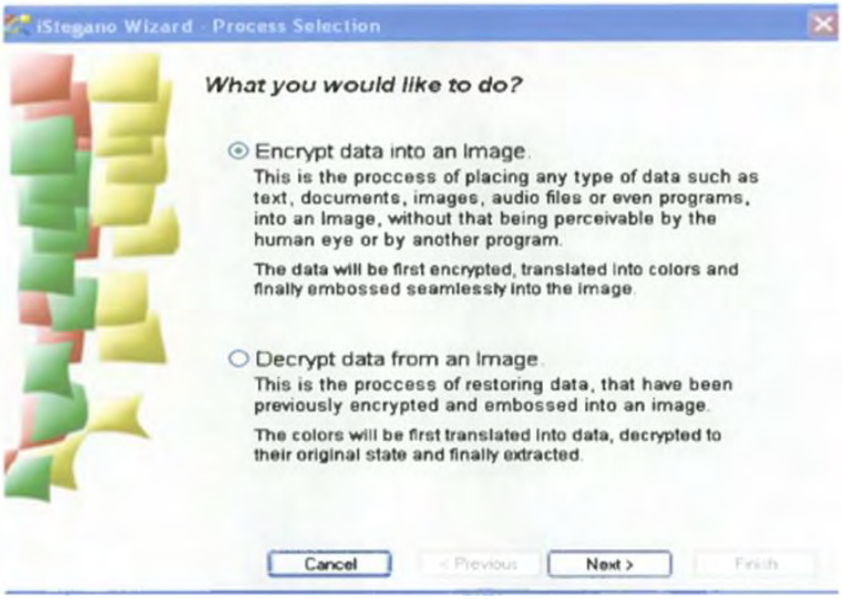
Αρκετά γνωστή εφαρμογή στεγανογραφίας. Μπορεί κάποιος να κρύψει δεδομένα (αρχεία εικόνας, βίντεο, κείμενα) μόνο μέσα σε εικόνες και στην συνέχεια να τα εξάγει. Παρέχει παράλληλα παράθυρα για ζωντανή προβολή και σύγκριση της αρχικής και της στεγανογραφημένης εικόνας. Περιλαμβάνει ενσωματωμένο συμπιεστή, ώστε να μειώνονται τα μεγέθη των εικόνων κατά 70%. Περιλαμβάνει επίσης ενσωματωμένο μετατροπέα εικόνας για μετατροπή εικόνων JPEG και GIF.

[9] έχει μεταβλητό μήκος κλειδιού και λειτουργεί στο επίπεδο του byte. Θεωρείται εξαιρετικά ασφαλής και οι υλοποιήσεις του σε λογισμικό τρέχουν πολύ γρήγορα

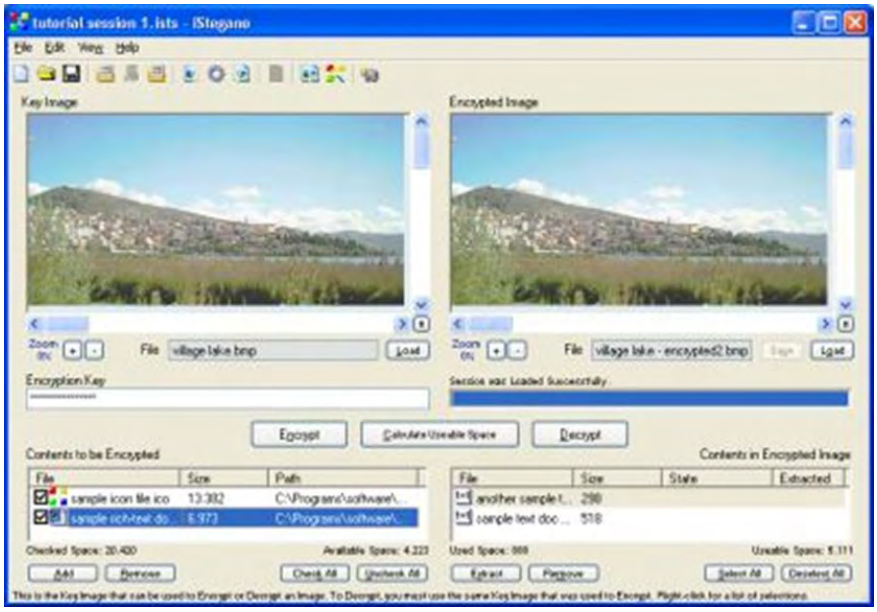
[10] έχει μέγεθος 64 bits κα
οι διεργασίες βασίζονται σε X-OR πράξεις και προσθέσεις λέξεων των 32 bits

Παρέχεται κατεβάζοντας και το πρόγραμμα βοηθός βήμα βήμα για να στεγανογραφηθούν τα δεδομένα.

Εικόνα 12 : Άνοιγμα iStegano



Εικόνα 13 : Χρήση του Προγράμματος

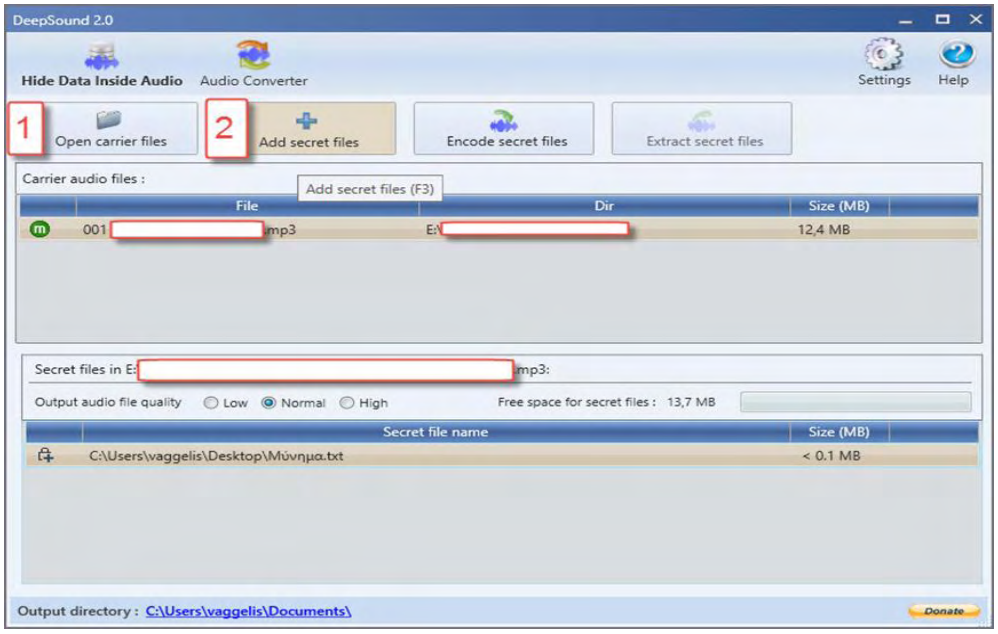


Εικόνα 14 : Διαδικασία Στεγανογράφησης

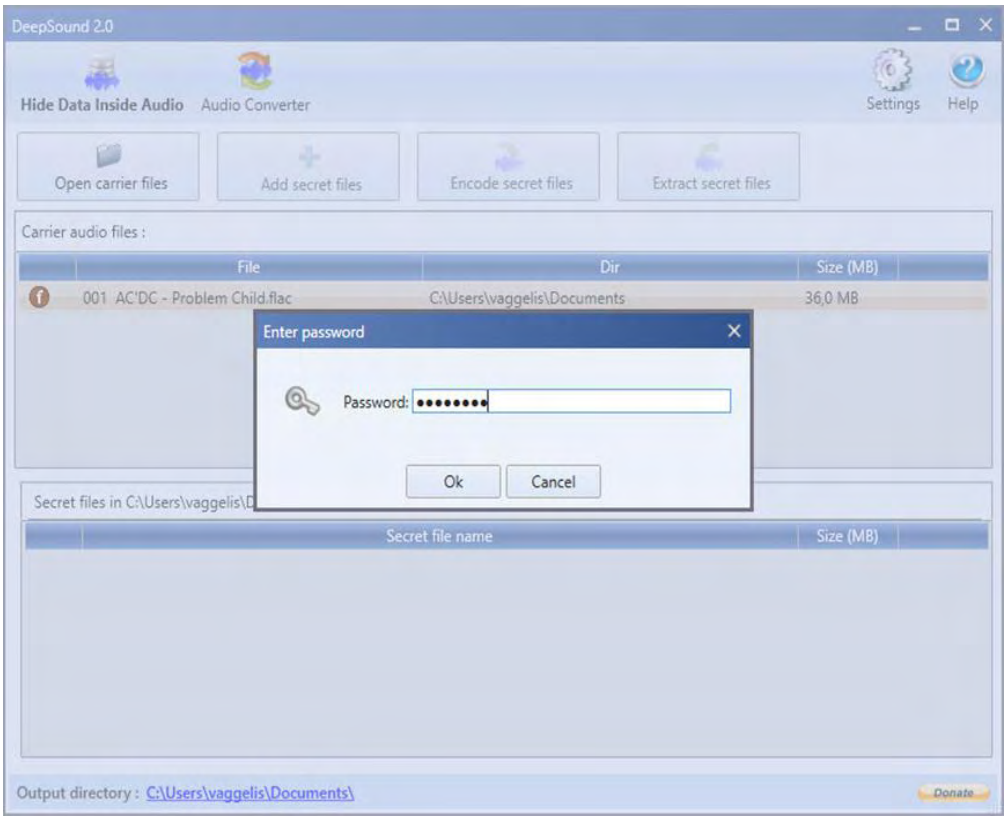
http://www.tucows.com/preview/395872/IStegano-2005

2.4 DeepSound

Το DeepSound είναι ουσιαστικά ένα λογισμικό στεγανογραφίας, μετατροπέας ήχου που ΄΄ξέρει΄΄ να κρύβει καλά τα δεδομένα σε αρχεία ήχου τα οποία μπορούν να προστατευτούν επιπλέον με κρυπτογράφηση. Η εφαρμογή επιτρέπει επίσης την εξαγωγή των κρυμμένων αρχείων απευθείας από τα αρχεία ήχου. Αυτή η ιδιαιτερότητα της εφαρμογής την καθιστά πολύ ελκυστική για πολλούς λόγους. Ένας από αυτούς ίσως και ο πιο βασικός είναι ότι θα μπορούσε να χρησιμοποιηθεί ως σήμανση πνευματικών δικαιωμάτων στα μουσικά αρχεία, ένας άλλος θα μπορούσε να είναι η δυνατότητα μεταφοράς κρυφών μηνυμάτων σε κάποιον ή κάποιους αποδέκτες χωρίς να μπορεί να φανταστεί κανείς ότι μέσα σ΄αυτά αρχεία βρίσκονται πολύτιμες πληροφορίες.



Εικόνα 15 : DeepSound σε χρήση

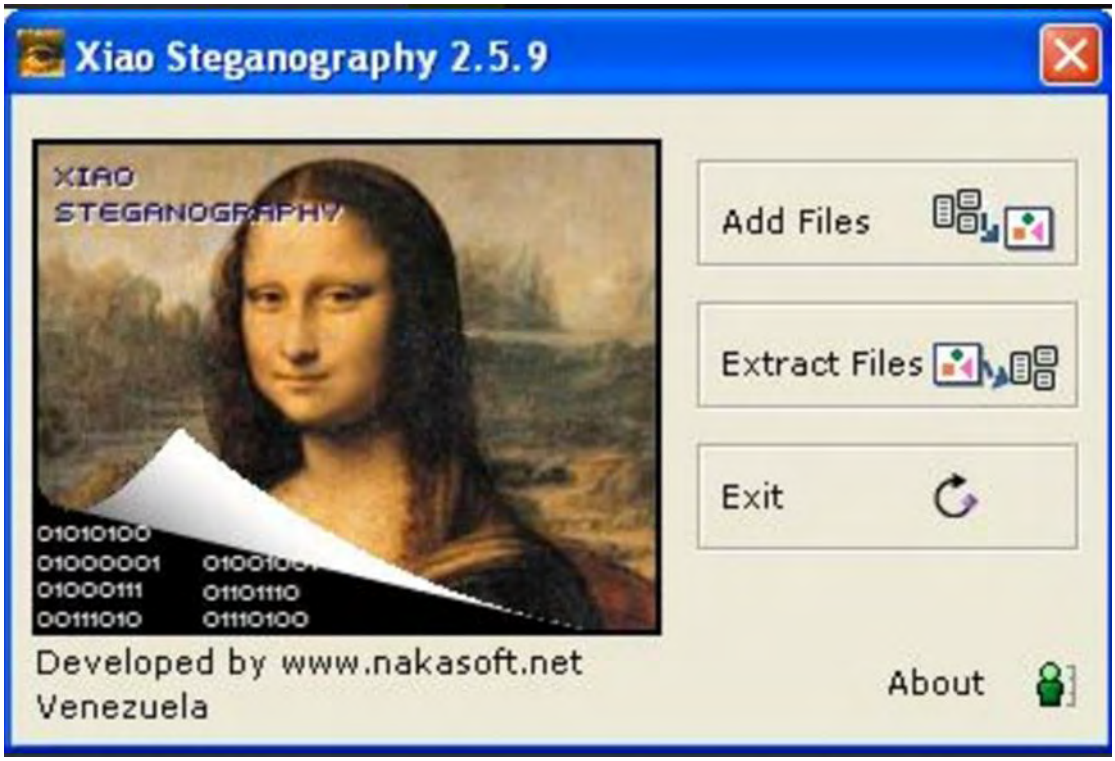


Εικόνα 16 : Προσθήκη κωδικού πρόσβασης

http://jpinsoft.net/deepsound/download.aspx

2.5 Xiao Steganography

Η διαδικασία που χρησιμοποιεί αυτή η εφαρμογή για την απόκρυψη εγγράφων χρησιμοποιείται συχνά από επίσημους και μυστικούς οργανισμούς για την αποστολή πληροφοριών σε ένα δίκτυο χωρίς να είναι ορατή. Μπορούν να κρυφτούν μηνύματα τόσο σε εικόνες όσο και ήχους. Υπάρχει και η δυνατότητα τροποποίησης του κώδικα κρυπτογράφησης (αλγόριθμος) και προστασία με κωδικό πρόσβασης. Είναι δωρεάν και στο κατέβασμά του δίνεται και βοηθητικό υλικό.



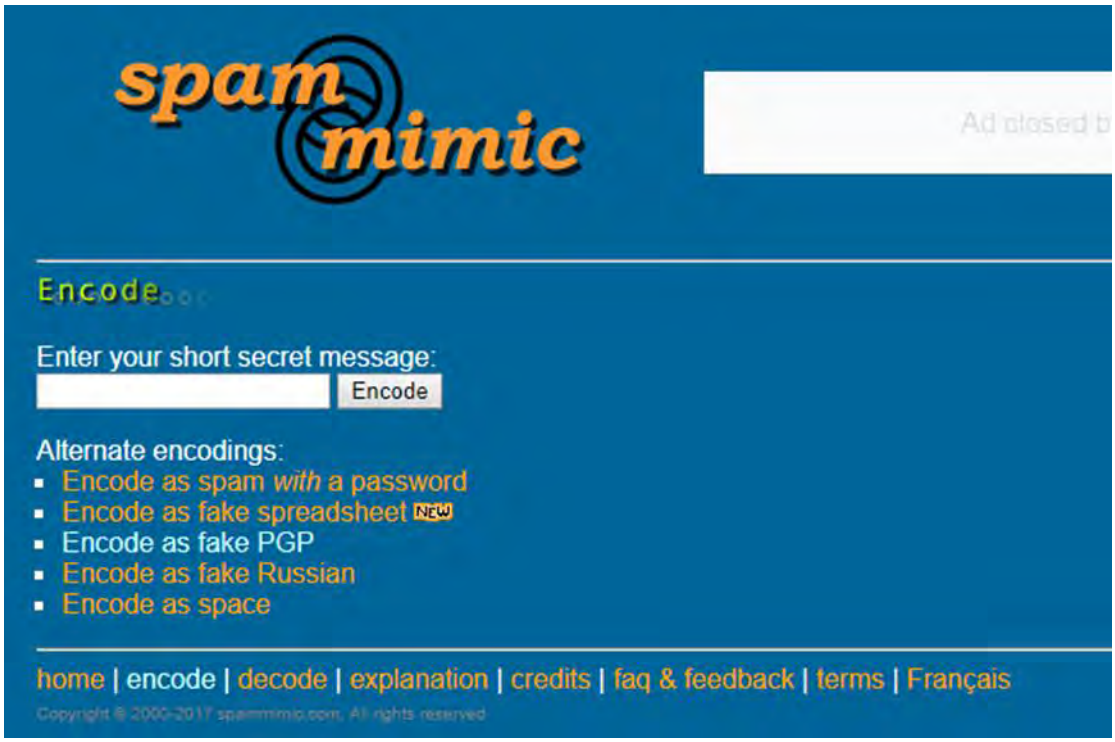
Εικόνα 17 : Άνοιγμα προγράμματος

https://xiao-steganography.en.uptodown.com/windows

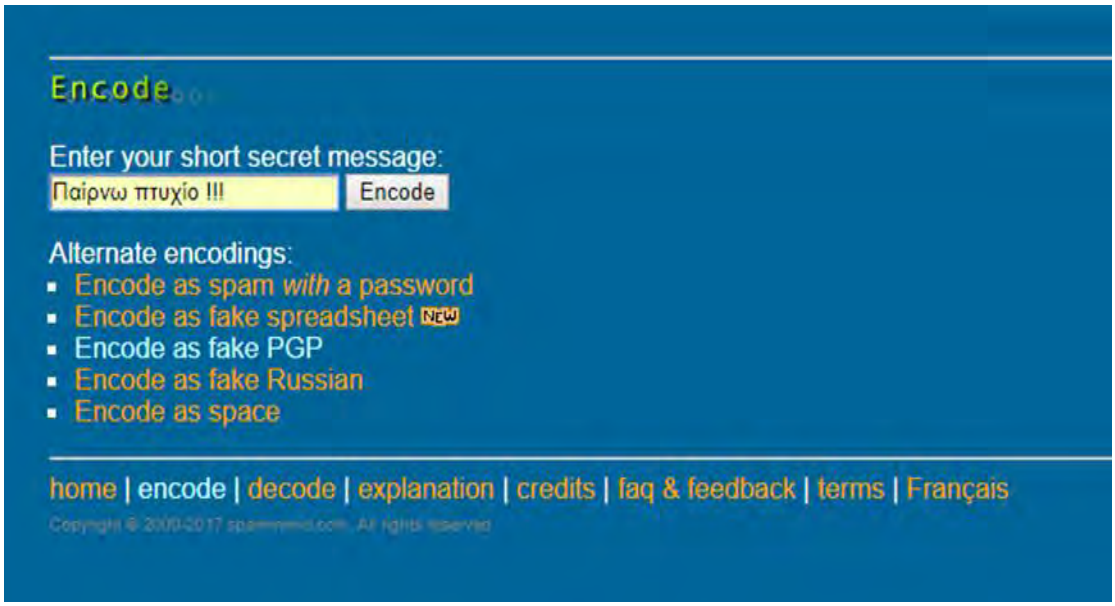
2.6 Spam Mimik

Πρόκειται για την πιο γρήγορη εφαρμογή, όπου είναι δωρεάν και online. Στεγανογραφεί μόνο μια φράση-κείμενο μέσα σε ένα άλλο κείμενο που το δημιουργεί επιτόπου. Δημιουργεί ένα spam email και στην συνέχεια αποστέλλεται στο άτομο που επιθυμεί ο χρήστης. Το spam email έχει πλεονεκτήματα για καμουφλάζ, για το λόγο

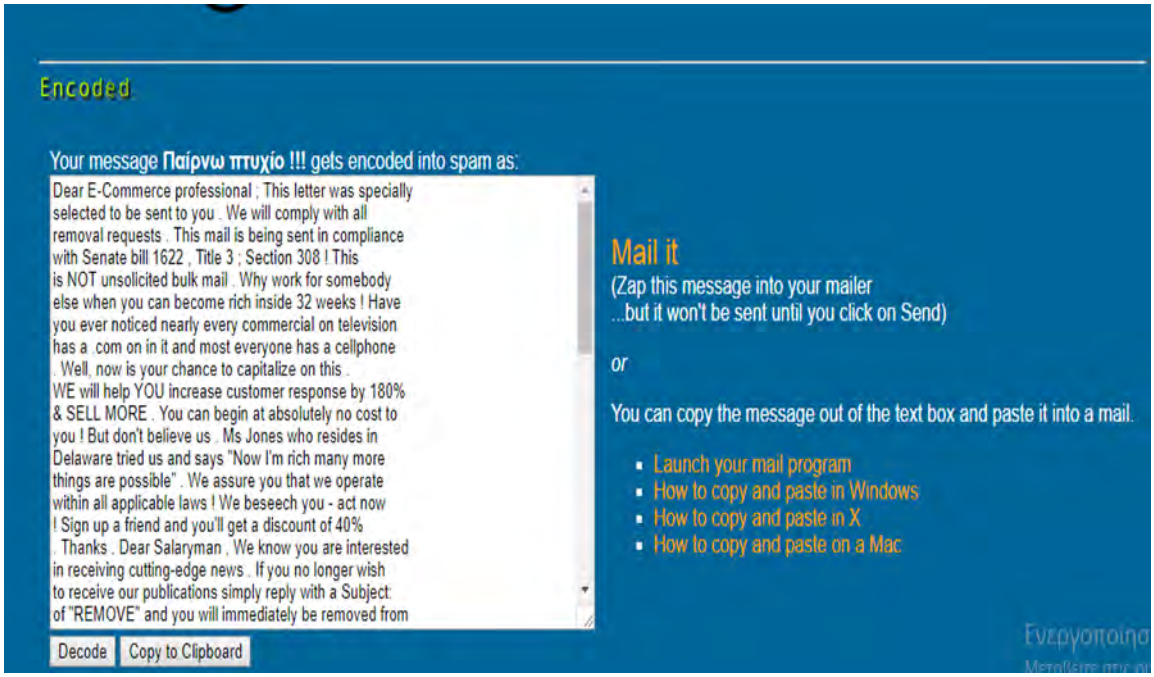
ότι οι περισσότεροι χρήστες αγνοούν τα spam email με αποτέλεσμα να στέλνεται υπεράνω πάσης υπ'οψίας.



Εικόνα 18 : Άνοιγμα Spam Mimic



Εικόνα 19 : Εισαγωγή της φράσης που θέλουμε να αποκρύψουμε



Εικόνα 20 : Η φράση στεγανοποιήθηκε μέσα σε αυτό το κείμενο

http://www.spammimic.com/

Κεφάλαιο 3ο Ανάπτυξη Εφαρμογής

3.1 Η εφαρμογή Στεγανογραφία 2018

Η ανάπτυξη της εφαρμογής έγινε στο περιβάλλον Matlab 2016a. Το Matlab είναι ένα ολοκληρωμένο περιβάλλον προγραμματισμού (η γλώσσα προγραμματισμού του μοιάζει πάρα πολύ με την γλώσσα C). Χρησιμοποιείται κυρίως για τεχνικό προγραμματισμό και διαθέτει μεγάλο σύνολο συναρτήσεων που βοηθούν τον προγραμματιστή στο να μην αναπτύσσει επιπλέον κώδικα παρά να εστιάζει περισσότερο στην υλοποίηση των αλγορίθμων του. Το βασικό του στοιχείο είναι η διαχείριση πινάκων. Οι πράξεις μεταξύ των πινάκων καθώς και όλες οι συναρτήσεις για υπολογισμούς είναι έτοιμες και με απλές εντολές ο προγραμματιστής μπορεί να πάρει τα αποτελέσματα που θέλει.

Η μέθοδος σταγανογραφίας που χρησιμοποιήθηκε είναι η LSB. Ως κάλυμμα (cover) χρησιμοποιούμε εικόνες οποιασδήποτε μορφής οι οποίες μετατρέπονται σε .bmp γιατί έτσι εξασφαλίζουμε ότι στην μορφή αρχείων δεν θα έχουμε συμπίεση και αλλοίωση των στοιχείων για να είναι δυνατή η ανάκτηση της πληροφορίας

Η πληροφορία που ενσωματώνουμε γίνεται σε όλα τα επίπεδα χρωμάτων του κάθε pixel. Συγκεκριμένα κάθε byte πληροφορίας που αντιστοιχεί σε 8 bit αλλοιώνει το χαμηλότερο bit του κάθε χρώματος 3 συνεχόμενων pixels της αρχικής φωτογραφίας. Εφαρμόζεται η συνάρτηση XOR για την αποθήκευση του κάθε bit.

Για να διασφαλιστεί η ασφάλεια στο μήνυμα που ενσωματώνεται επιλέχθηκε η υποχρεωτική κρυπτογράφηση (ο χρήστης πρέπει να πληκτρολογήσει το κλειδί κρυπτογράφησης). Το μήνυμα πρώτα κρυπτογραφείται και κατόπιν ενσωματώνεται στην εικόνα. Η κρυπτογράφηση είναι συμμετρική και για κάθε χαρακτήρα byte έχουμε: byte XOR enctyption_Key. Κατόπιν ο παραγόμενος χαρακτήρας αναλύεται στα bit (σε ASCII αναπαράσταση) και ενσωματώνεται στην εικόνα.

Για να αυξηθεί η ασφάλεια του συστήματος αναπτύχθηκε ξεχωριστά και συνάρτηση υπολογισμού των pixels (με τυχαίο τρόπο) χρησιμοποιώντας την ενσωματωμένη γεννήτρια τυχαίων αριθμών που διαθέτει το MatLab. Η συνάρτηση μας δίνει την ίδια ακολουθία τυχαίων αριθμών χρησιμοποιώντας τον ίδιο σπόρο (αρχή της ακολουθίας). Με αυτόν τον τρόπο διασφαλίζεται ότι δεν είναι δυνατή η ανάκτηση του μηνύματος εάν ο χρήστης δεν γνωρίζει την αρχή της ακολουθίας.

Για την υλοποίηση των παραπάνω αναπτύχθηκαν 4 συναρτήσεις στο Matlab:

```
function [image_with_message] = stegAMcode(img,msg,enc_key)
% stegAMcode:  Η συνάρτηση αυτή κρύβει ένα μήνυμα (msg) μέσα σε
% μία εικόνα msg με δεδομένο κάποιο κλειδί κρυπτογράφησης enc_key
% Είσοδοι:
% - img: Αρχική εικόνα στην οποία θα κρυφτεί το μήνυμα
% - msg: Μήνυμα (είναι κείμενο ή εικόνα η οποία μετατρέπεται σε gray
% - enc_key: Κλειδί κρυπτογράφησης
%
% Έξοδοι:
% - image_with_message: Το στεγανόγραμμα (εικόνα + μήνυμα)
```

Η συνάρτηση **stegAMcode** δέχεται ως παραμέτρους μία εικόνα, ένα μήνυμα το οποίο μπορεί να είναι είτε κείμενο είτε εικόνα και το κλειδί κρυπτογράφησης. Υπολογίζει το μήκος του μηνύματος που είναι να ενσωματωθεί (σε περίπτωση που είναι εικόνα μετατρέπεται σε κλίμακα του γκρι για να μειωθεί το μέγεθός της στο 1/3) κρυπτογραφείτε κάθε byte και μετά το κάθε byte ενσωματώνεται σε 3 διαδοχικά pixels της εικόνας. Βέβαια πριν την ενσωμάτωση αποθηκεύεται στην αρχή το μήκος του κειμένου καθώς και το κλειδί κρυπτογράφησης έτσι ώστε κατά την ανάκτηση να είναι να γνωρίζει το πρόγραμμα πιο είναι το κλειδί και να το συγκρίνει με αυτό που δίνει ο χρήστης καθώς και το μήκος το κειμένου που είναι να ανακτήσει. Το μήνυμα αποθηκεύεται σειριακά.

Η συνάρτηση επιστρέφει την εικόνα που έχει ενσωματώσει το μήνυμα. Η συνάρτηση είναι δυνατόν να χρησιμοποιηθεί από οποιαδήποτε εφαρμογή του Matlab

```
function [image_with_message] =
stegAMcodeRand(img,msg,enc_key,randSeed)
% stegAMcodeRand:   Η συνάρτηση αυτή κρύβει ένα μήνυμα (msg) μέσα σε
%         μία εικόνα msg με δεδομένο κάποιο κλειδί κρυπτογράφησης
%                   enc_key και ένα δεδομένο seed (randSeed)
% Είσοδοι:
% - img: Αρχική εικόνα στην οποία θα κρυφτεί το μήνυμα
% - msg: Μήνυμα (είναι κείμενο ή εικόνα η οποία μετατρέπεται σε gray
% - enc_key: Κλειδί κρυπτογράφησης
% - randSeed: seed της γεννήτριας τυχαίων αριθμών
%
% Έξοδοι:
% - image_with_message: Το στεγανόγραμμα (εικόνα + μήνυμα)
```

Η συνάρτηση **stegAMcodeRand** λειτουργεί όπως η **stegAMcode** με την μόνη διαφορά ότι δέχεται μία επιπλέον παράμετρο την **randSeed** η οποία καθορίζει τον πρώτο όρο της ακολουθίας των τυχαίων αριθμών οι οποίοι χρησιμεύουν στο να προσδιορίζουν το

πιο pixel θα επιλέγεται κάθε φορά για ενσωμάτωση, άρα δεν έχουμε σειριακή ενσωμάτωση αλλά τυχαία.

```
function [msg] = stegAMdecode(img,enc_key)
% stegAMdecode: Η συνάρτηση εξάγει ένα κρυμένο μήνυμα
%   από μία εικόνα (Το στεγανόγραμμα (εικόνα + μήνυμα))
% Είσοδοι:
% - img: Το στεγανόγραμμα (εικόνα + μήνυμα)
% - enc_key: Κλειδί κρυπτογράφησης
%
% Έξοδοι:
% - msg: Μήνυμα (είναι κείμενο ή εικόνα σε gray)
```

Η συνάρτηση **stegAMdecode** δέχεται ως εισόδους το Στεγανόγραμμα (εικόνα+μήνυμα) καθώς και το κλειδί κρυπτογράφησης ανακτά την κεφαλίδα με τα στοιχεία του μηνύματος καθώς και το ίδιο το μήνυμα και επιστρέφει το μήνυμα. Η ανάκτηση είναι εφικτή μόνο στην περίπτωση που το κλειδί που δίνει ο χρήστης είναι ίδιο με αυτό που είναι αποθηκευμένο στην κεφαλίδα των στοιχείων. Προφανώς η ανάκτηση γίνεται με σειριακό τρόπο.

```
function [msg] = stegAMdecodeRand(img,enc_key,randSeed)
% stegAMdecodeRand: Η συνάρτηση εξάγει ένα κρυμένο μήνυμα
%   από μία εικόνα (Το στεγανόγραμμα (εικόνα + μήνυμα))
% Είσοδοι:
% - img: Το στεγανόγραμμα (εικόνα + μήνυμα)
% - enc_key: Κλειδί κρυπτογράφησης
% - randSeed: seed της γεννήτριας τυχαίων αριθμών
%
% Έξοδοι:
% - msg: Μήνυμα (είναι κείμενο ή εικόνα σε gray)
```

Η συνάρτηση **stegAMdecodeRand** λειτουργεί όπως η **stegAMdecode** με την μόνη διαφορά ότι δέχεται μία επιπλέον παράμετρο την **randSeed** η οποία καθορίζει τον πρώτο όρο της ακολουθίας των τυχαίων αριθμών οι οποίοι χρησιμεύουν στο να προσδιορίζουν το πιο pixel θα επιλέγεται κάθε φορά για ανάκτηση, άρα δεν έχουμε σειριακή ανάκτηση αλλά τυχαία.

Για την υλοποίηση των παραπάνω συναρτήσεων χρησιμοποιήθηκε τμήμα κώδικα από “David Pipkorn and Preston Weisbrot, Project: Steganography - Hidden Messages in Images”.

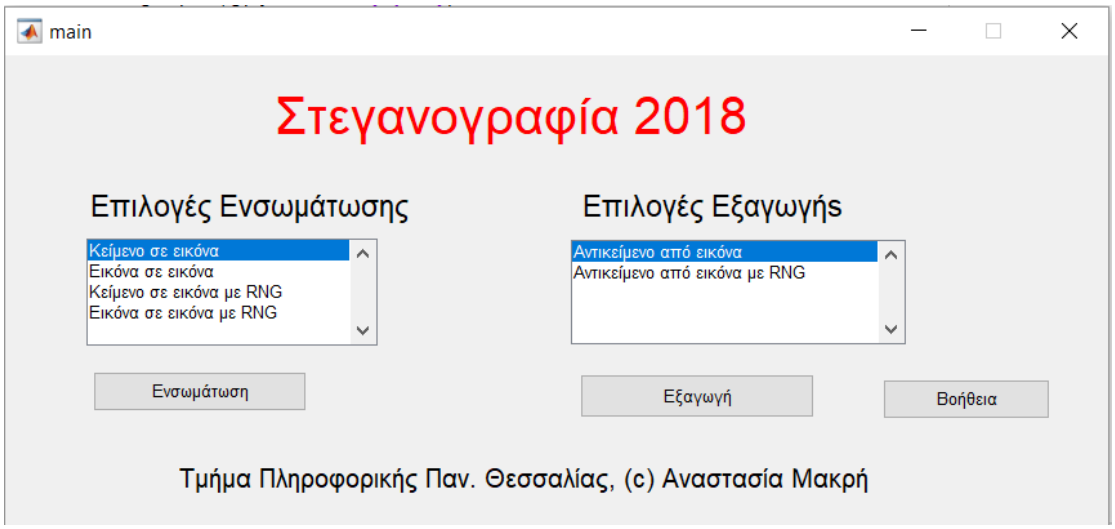
Κατόπιν αναπτύχθηκε γραφική διεπαφή χρήστη (gui) για την υλοποίηση της στεγανογραφίας και ενσωμάτωσης κειμένου ή εικόνας σε εικόνα.

Η εφαρμογή έχει μία βασική οθόνη όπου ο χρήστης μπορεί με απλά βήματα αλλά και με πλαίσια διαλόγου να επιλέξει την εργασία που επιθυμεί να εκτελέσει. Το εγχειρίδιο της εφαρμογής αναπτύσσετε στην επόμενη παράγραφο.

3.2 Εγχειρίδιο χρήσης της εφαρμογής

3.2.1 Αρχικό παράθυρο της εφαρμογής.

Από το κεντρικό παράθυρο ο χρήστης μπορεί να επιλέξει με απλά βήματα την εργασία που θέλει να κάνει και να πατήσει το αντίστοιχο κουμπί για να εκτελέσει την ενέργεια του.



Εικόνα 21 : Αρχικό παράθυρο εφαρμογής

Δυνατότητες που δίνει η εφαρμογή στον χρήση:

1. Ενσωμάτωση:
    a. Κείμενο σε εικόνα
    b. Εικόνα σε εικόνα
    c. Κείμενο σε εικόνα με χρήση γεννήτριας τυχαίων αριθμών
    d. Εικόνα σε εικόνα με χρήση γεννήτριας τυχαίων αριθμών
2. Εξαγωγή
    a. Αντικείμενο από εικόνα
    b. Αντικείμενο από εικόνα με χρήση γεννήτριας τυχαίων αριθμών

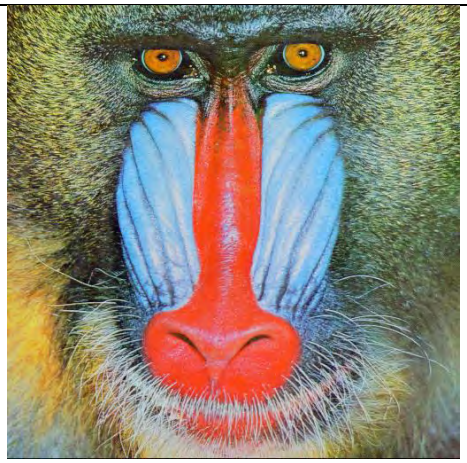
Σε όλες τις περιπτώσεις είναι απαραίτητη η εισαγωγή κλειδιού κρυπτογράφησης έτσι ώστε η μη γνώση του κλειδιού να καθιστά αδύνατη την εξαγωγή του αντικειμένου (είτε εικόνας είτε κειμένου) από το στεγανογράφημα.

Με το κουμπί Βοήθεια εμφανίζεται το παρόν εγχειρίδιο χρήσης.

3.2.2 Διαδικασία ενσωμάτωσης κειμένου σε εικόνα.

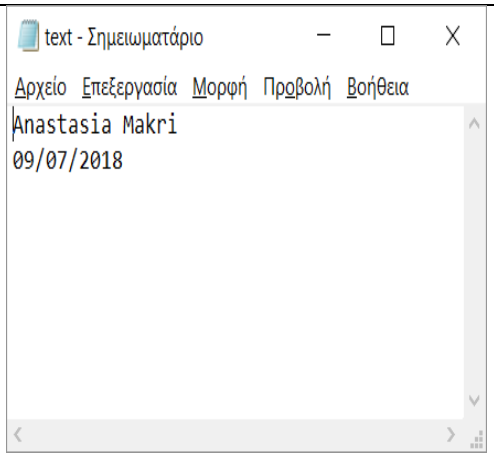
Ο χρήστης μπορεί να έχει μια οποιαδήποτε εικόνα σε οποιαδήποτε μορφή (.jpg, .png, .bmp, .tiff)

Πρέπει να γράφει σε ένα αρχείο το μήνυμά του και να το αποθηκεύσει.



Εικόνα 22 : Η εικόνα σε μορφή .png

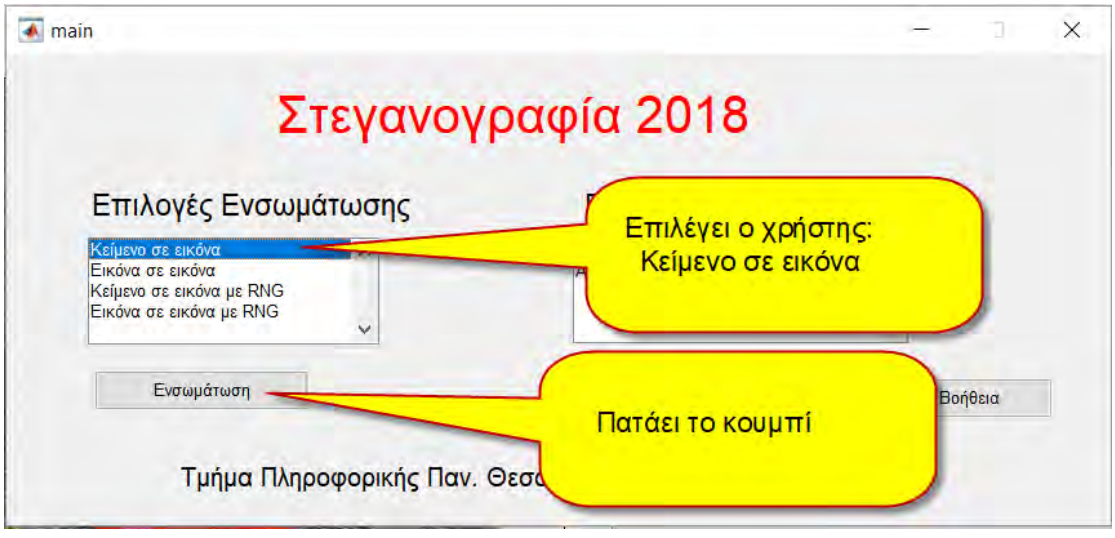
Baboon_big.png



Εικόνα 23 : Το κείμενο που θα κρυφτεί

text.txt

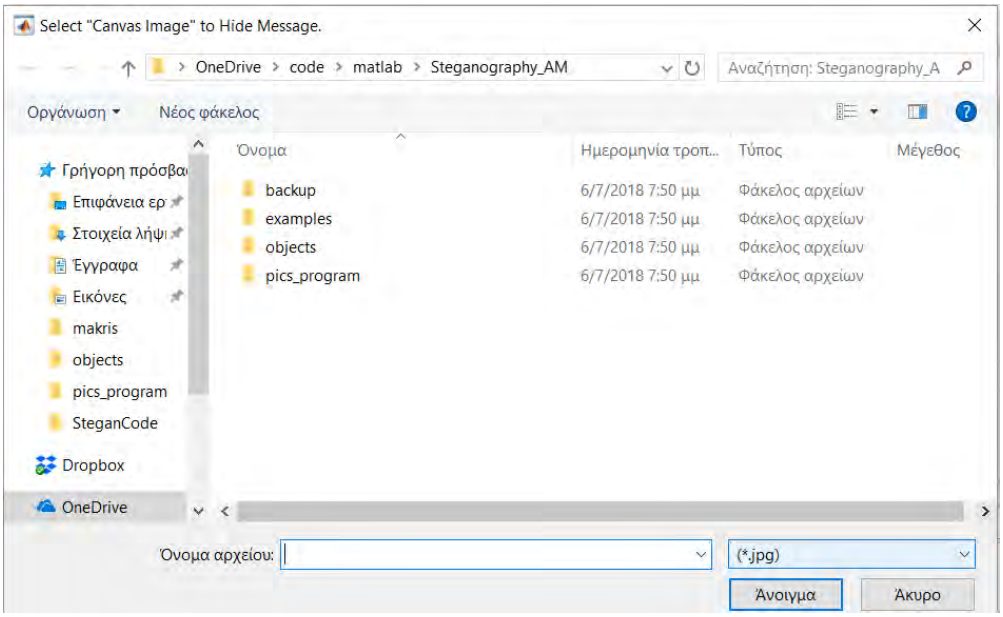
Ο χρήστης αφού επιλέξει από τις Επιλογές Ενσωμάτωσης «Κείμενο σε εικόνα» πατάει το κουμπί «Ενσωμάτωση»



Εικόνα 24 :Παράθυρο επιλογής ενδωμάτωσης

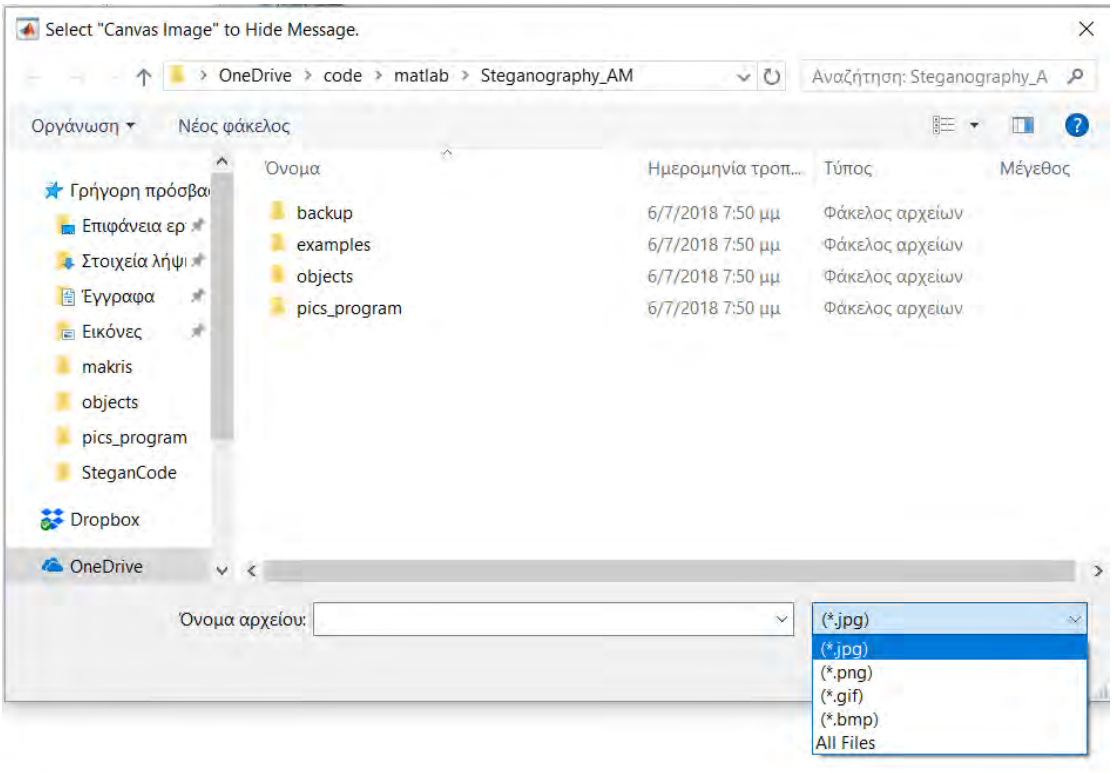
Η εφαρμογή θα ζητήσει από τον χρήστη να επιλέξει την εικόνα που θέλει να χρησιμοποιήσει μέσα στην οποία θα κρύψει το μήνυμά του.

Ο χρήστης έχει την δυνατότητα από οποιοδήποτε φάκελο του υπολογιστή να βρει την εικόνα που επιθυμεί.



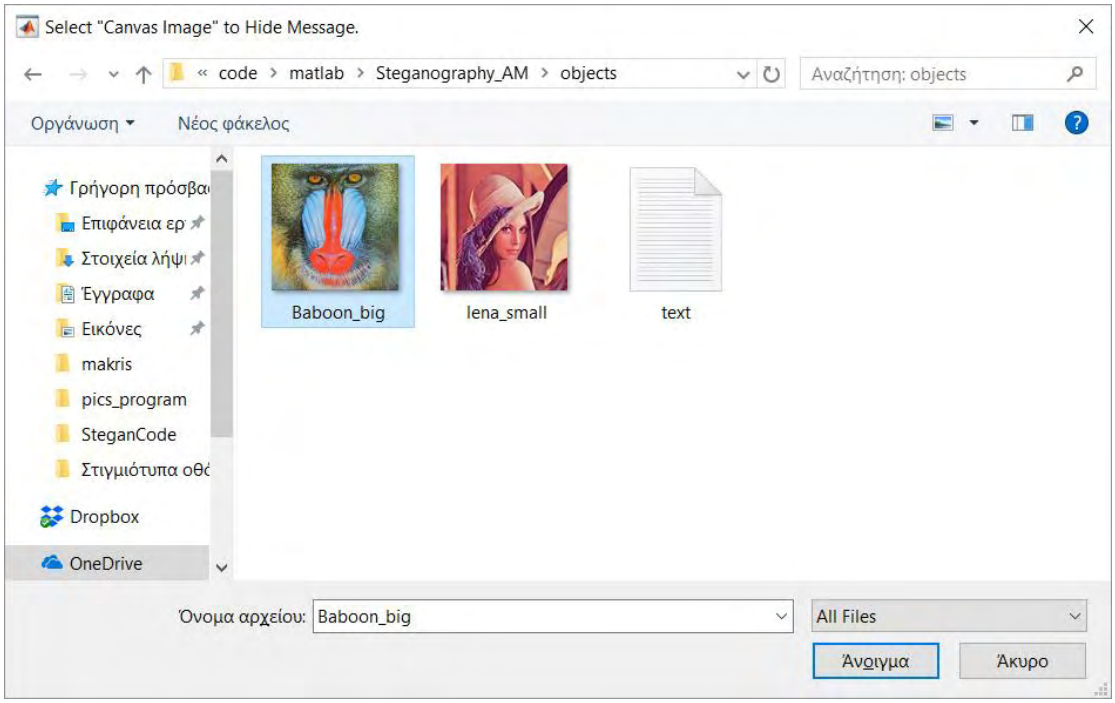
Εικόνα 25 : Αναζήτηση εικόνας από φάκελο

Ως προεπιλεγμένο τύπο αρχείο η εφαρμογή έχει jpg αλλά ο χρήστης μπορεί να επιλέξει άλλον τύπο αρχείου ακόμη και «All files»



Εικόνα 26 : Δυνατότητα επιλογής τύπου αρχείου

Και να επιλέξει την εικόνα που επιθυμεί από οποιοδήποτε φάκελο

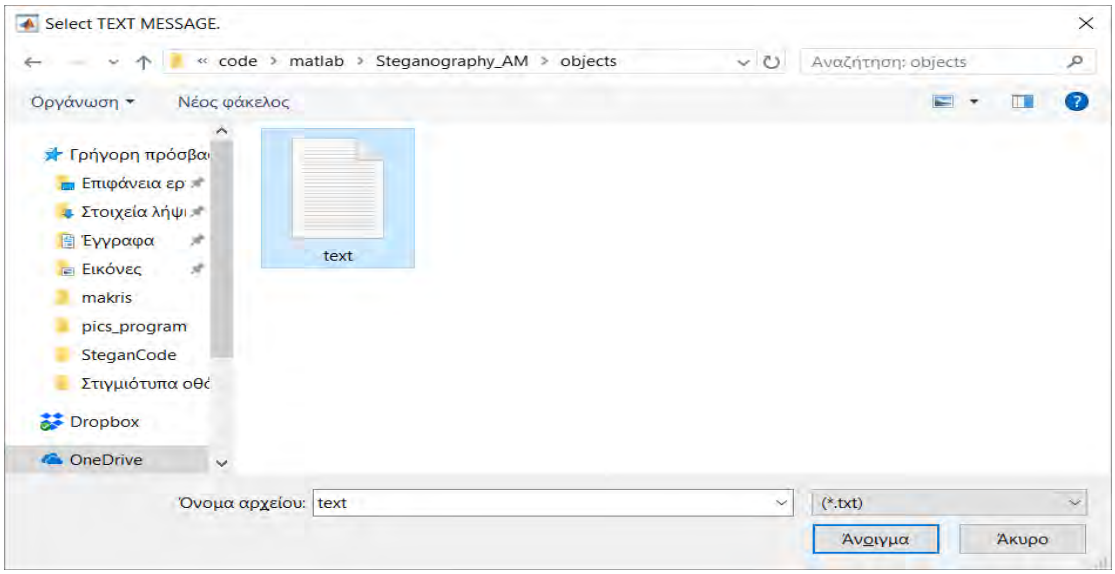


Εικόνα 27 : Επιλογή εικόνας

Εάν ο χρήστης πατήσει το κουμπί «**Άκυρο**» τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Πατώντας το κουμπί άνοιγμα τότε το πρόγραμμα ζητάει από τον χρήστη να επιλέξει το αρχείο κειμένου όπου περιέχει το κείμενο προς απόκρυψη.

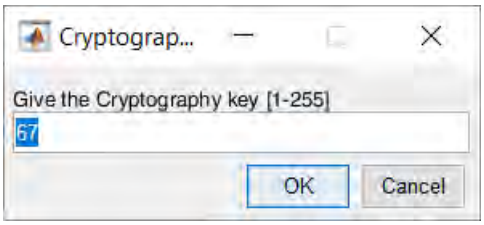
Η διαδικασία είναι η ίδια ακριβώς όπως και με την εικόνα.



Εικόνα 28 : Επιλογή αρχείου κειμένου

Εάν ο χρήστης πατήσει το κουμπί «**Άκυρο**» τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Πατώντας το κουμπί άνοιγμα τότε το πρόγραμμα ζητάει από τον χρήστη να εισάγει το κωδικό κρυπτογράφησης. Ο κωδικός κρυπτογράφησης είναι ένας αριθμός από το 1-255.

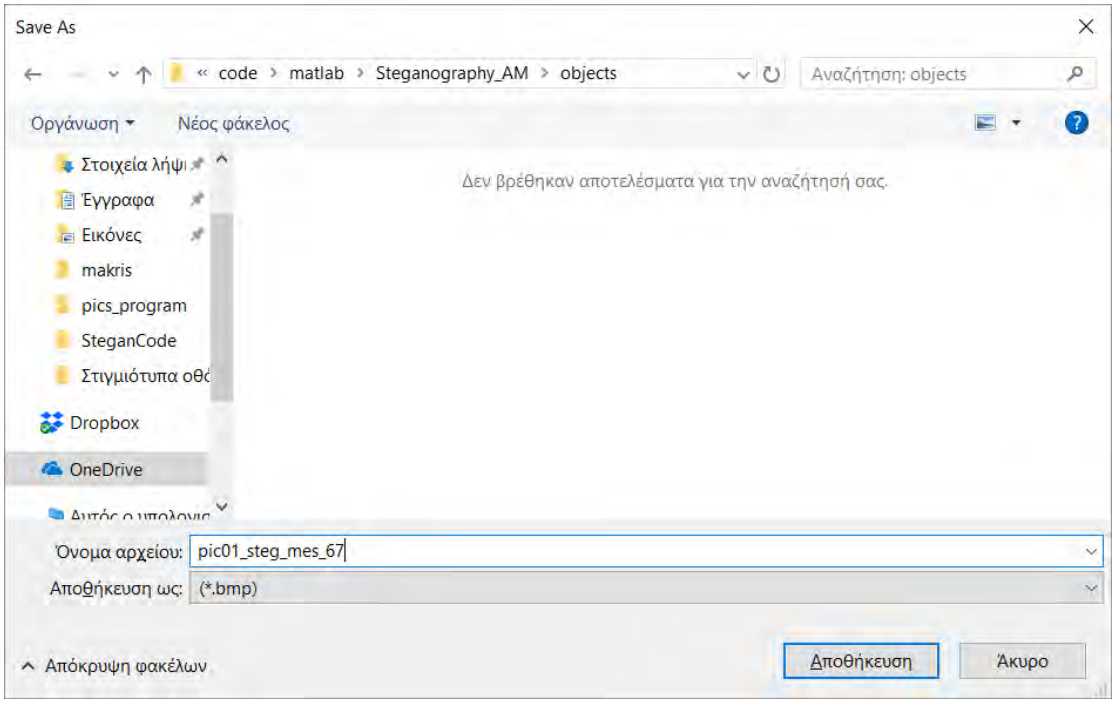


Εικόνα 29 : Κωδικός

Εάν ο χρήστης πατήσει το κουμπί «**Cancel**» ή δεν δώσει κωδικό μεταξύ του 1-255 τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Το πρόγραμμα προτείνει ως κωδικό τον αριθμό 67 (από τον ΑΜ της προγραμματίστριας στο τμήμα Πληροφορικής του Πανεπιστημίου Θεσσαλίας).

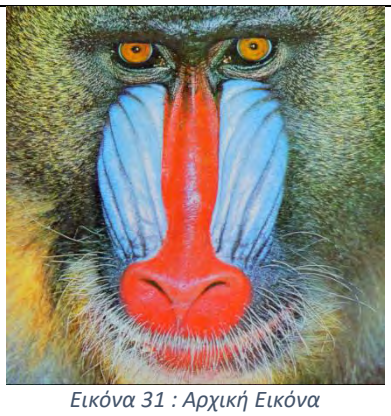
Η αποθήκευση του στεγανογράμματος (εικόνα + μήνυμα) γίνεται σε μορφή .**bmp** γιατί η μορφή **.jpg** διαθέτει αλγορίθμους συμπίεσης και θα αλλοίωνε το τελικό αποτέλεσμα και ίσως χαλούσε και τα δεδομένα της εικόνας.



Εικόνα 30 : Επιλογή φακέλου, όνομα και αποθήκευση

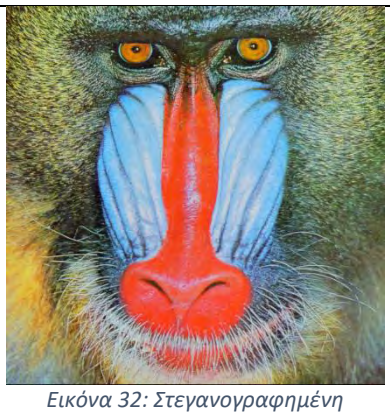
Ο χρήστης πληκτρολογεί ένα όνομα που επιθυμεί (προφανώς μπορεί να επιλέξει και τον φάκελο που θέλει) και μετά πατά αποθήκευση.

Εφόσον ολοκληρωθεί η διαδικασία τότε έχει δημιουργηθεί το στεγανόγραμμα που περιέχει την εικόνα και το κείμενο μαζί.



Εικόνα 31 : Αρχική Εικόνα

Baboon_big.png

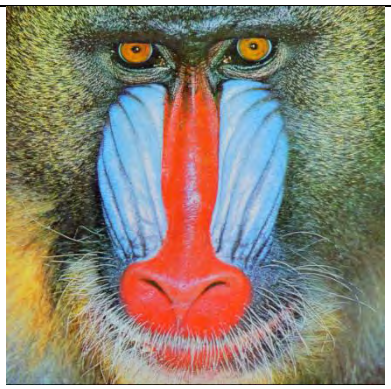


Εικόνα 32: Στεγανογραφημένη

pic01_steg_mes_67.bmp

3.2.3 Διαδικασία ενσωμάτωσης Εικόνα σε Εικόνα.

Ο χρήστης μπορεί να έχει δύο οποιεσδήποτε εικόνες σε οποιαδήποτε μορφή (.jpg, .png, .bmp, .tiff)



Εικόνα 33 : εικόνα κάλυμμα

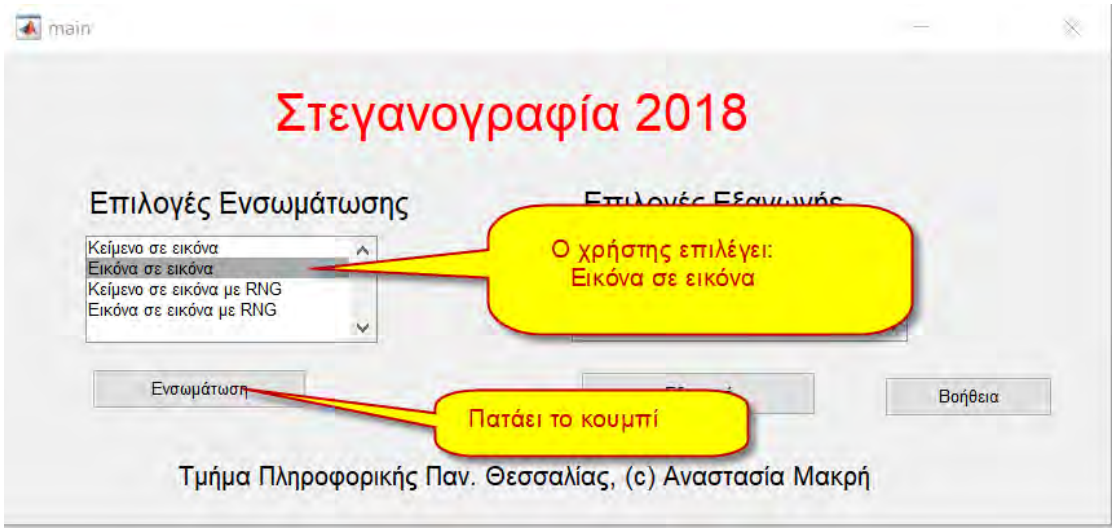
Baboon_big.png

Εικόνα 34 : το μήνυμα

lena_small.png

Η πρώτη εικόνα είναι το κάλυμμα(cover) και η δεύτερη εικόνα το μήνυμα που θέλουμε να κρύψουμε.

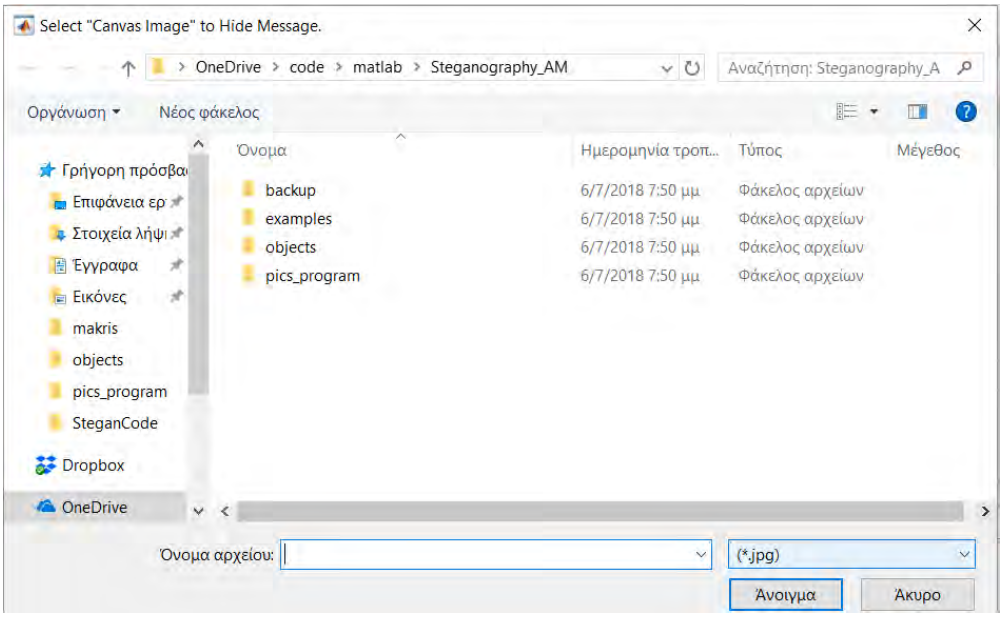
Ο χρήστης αφού επιλέξει από τις Επιλογές Ενσωμάτωσης «Εικόνα σε εικόνα» πατάει το κουμπί «Ενσωμάτωση»



Εικόνα 35 : Επιλογή ενσωμάτωσης

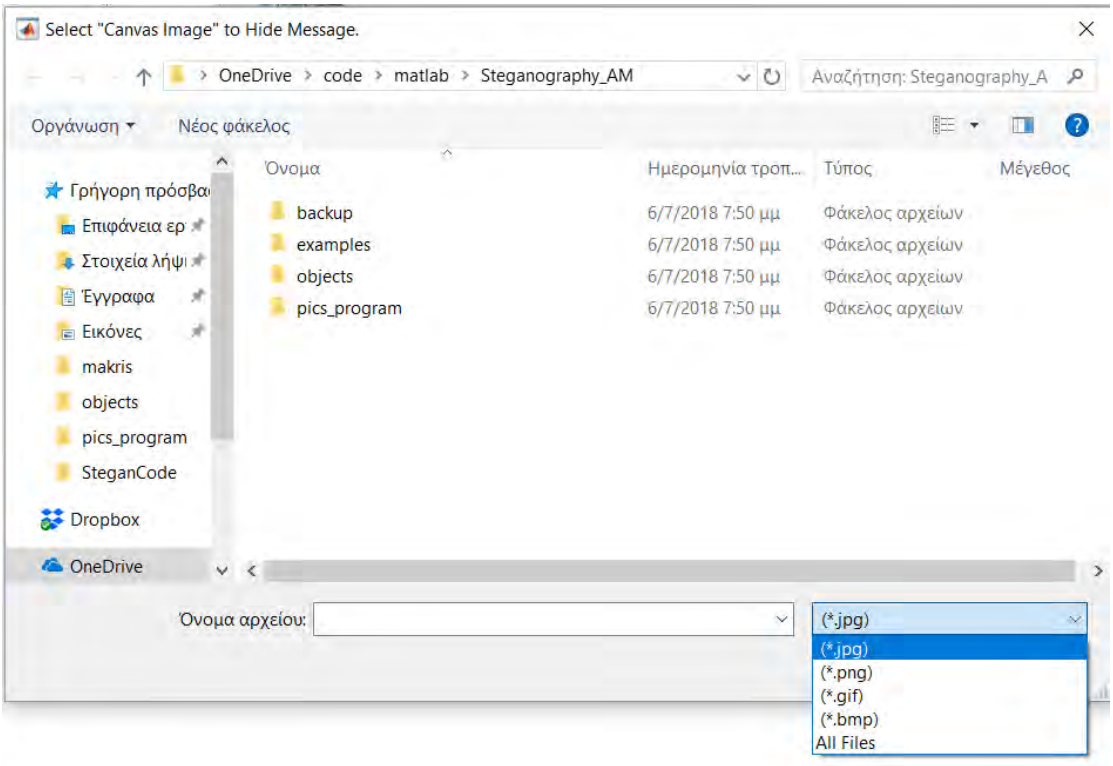
Η εφαρμογή θα ζητήσει από τον χρήστη να επιλέξει την εικόνα που θέλει να χρησιμοποιήσει μέσα στην οποία θα κρύψει το μήνυμά του.

Ο χρήστης έχει την δυνατότητα από οποιοδήποτε φάκελο του υπολογιστή να βρει την εικόνα που επιθυμεί.



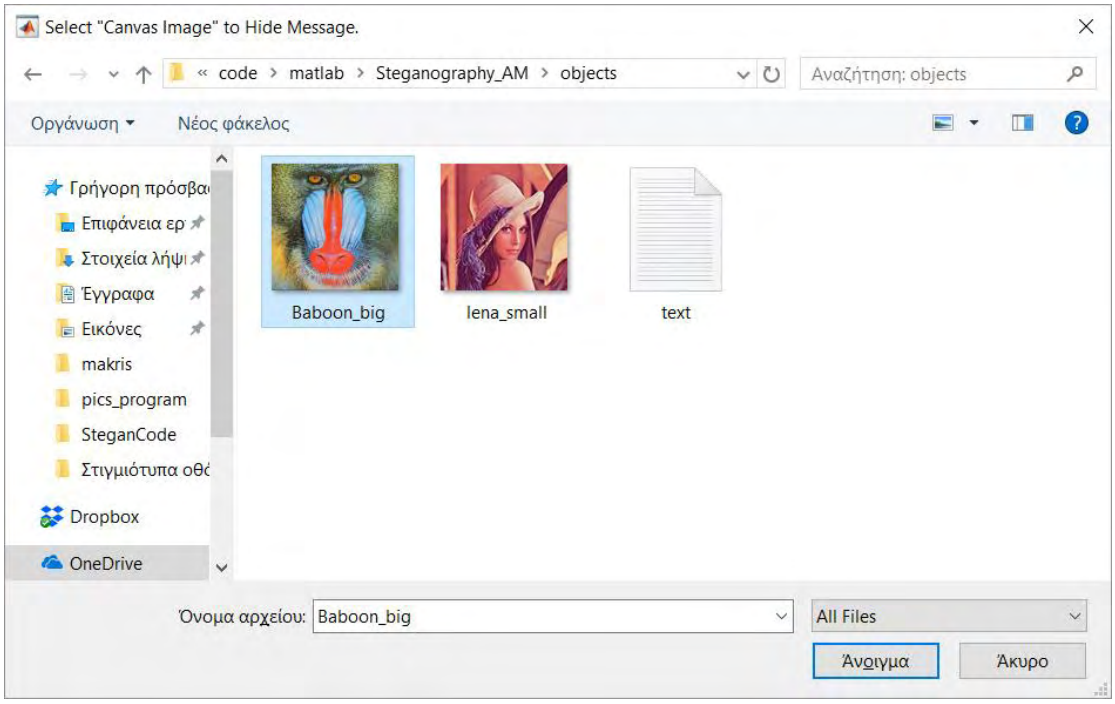
Εικόνα 36 : Αναζήτηση εικόνας

Ως προεπιλεγμένο τύπο αρχείο η εφαρμογή έχει jpg αλλά ο χρήστης μπορεί να επιλέξει άλλον τύπο αρχείου ακόμη και «All files»



Εικόνα 37 : Επιλογή τύπου αρχείου

Και να επιλέξει την εικόνα που επιθυμεί από οποιοδήποτε φάκελο

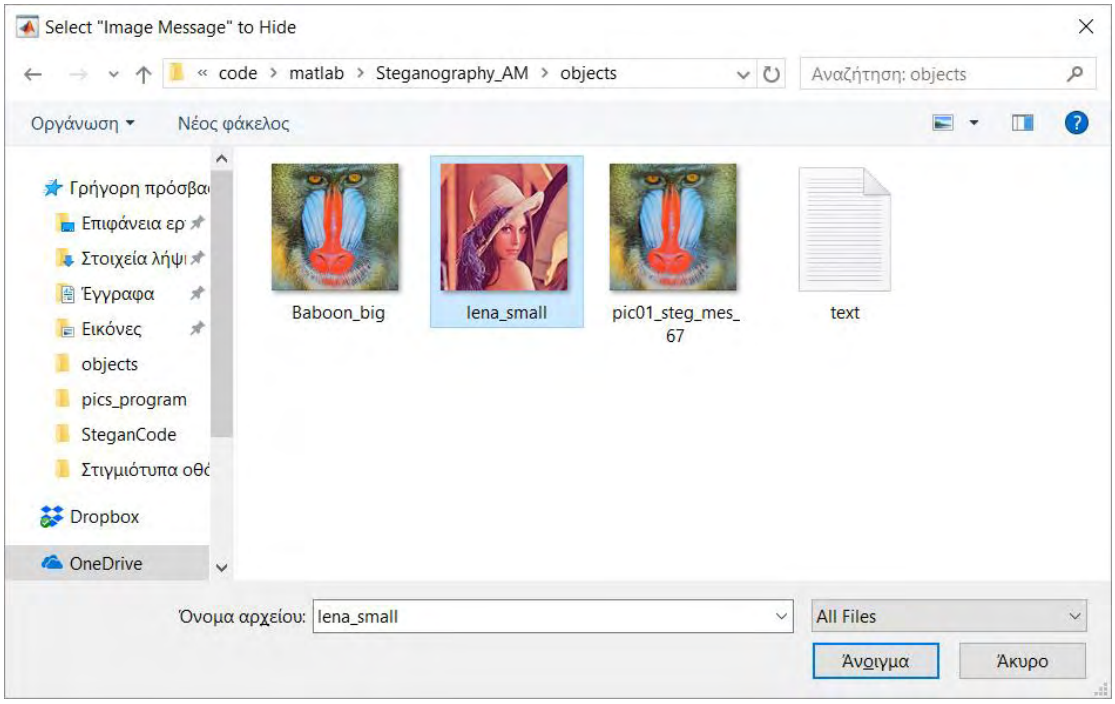


Εικόνα 38 : Επιλογή από φάκελο

Εάν ο χρήστης πατήσει το κουμπί «**Άκυρο**» τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Πατώντας το κουμπί άνοιγμα τότε το πρόγραμμα ζητάει από τον χρήστη να επιλέξει την εικόνα «μήνυμα» προς απόκρυψη.

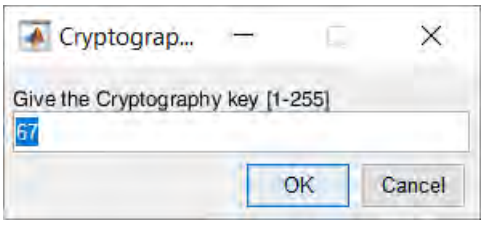
Η διαδικασία είναι η ίδια ακριβώς όπως και με την εικόνα cover.



Εικόνα 39 : εικόνα προς απόκρυψη

Εάν ο χρήστης πατήσει το κουμπί «**Άκυρο**» τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Πατώντας το κουμπί άνοιγμα τότε το πρόγραμμα ζητάει από τον χρήστη να εισάγει το κωδικό κρυπτογράφησης. Ο κωδικός κρυπτογράφησης είναι ένας αριθμός από το 1-255.

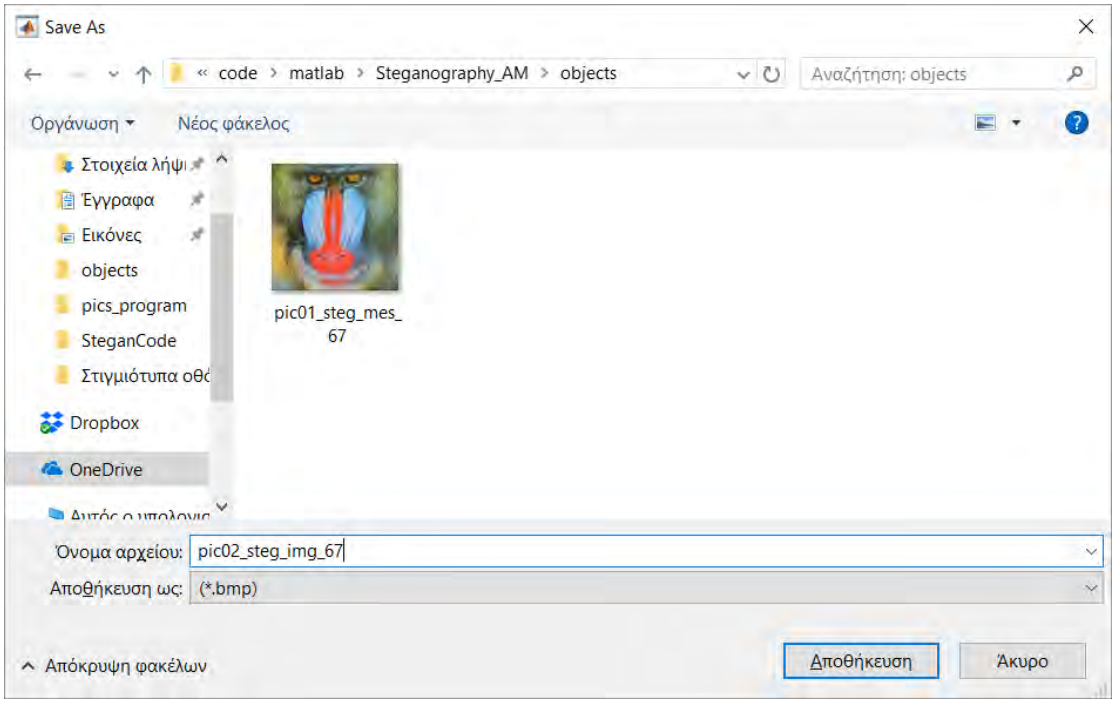


Εικόνα 40 : Κωδικός

Εάν ο χρήστης πατήσει το κουμπί «**Cancel**» ή δεν δώσει κωδικό μεταξύ του 1-255 τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Το πρόγραμμα προτείνει ως κωδικό τον αριθμό 67 (από τον ΑΜ της προγραμματίστριας στο τμήμα Πληροφορικής του Πανεπιστημίου Θεσσαλίας).

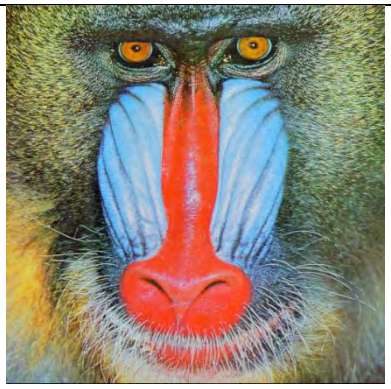
Η αποθήκευση του στεγανογράμματος (εικόνα + μήνυμα) γίνεται σε μορφή **.bmp** γιατί η μορφή **.jpg** διαθέτει αλγορίθμους συμπίεσης και θα αλλοίωνε το τελικό αποτέλεσμα και ίσως χαλούσε και τα δεδομένα της εικόνας.



Εικόνα 41 : Αποθήκευση

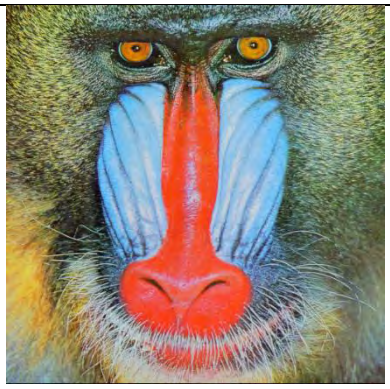
Ο χρήστης πληκτρολογεί ένα όνομα που επιθυμεί (προφανώς μπορεί να επιλέξει και τον φάκελο που θέλει) και μετά πατά αποθήκευση.

Εφόσον ολοκληρωθεί η διαδικασία τότε έχει δημιουργηθεί το στεγανόγραμμα που περιέχει την εικόνα και την εικόνα μήνυμα μαζί.



Εικόνα 42 : Αρχική εικόνα

Baboon_big.png



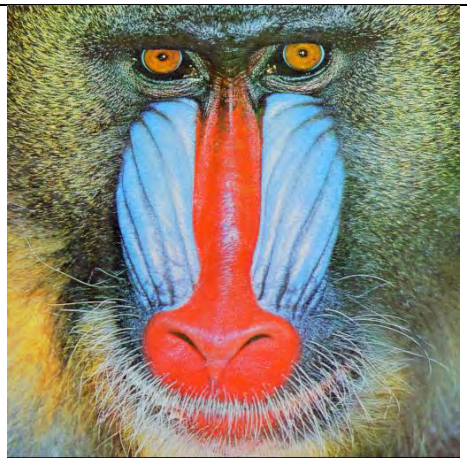
Εικόνα 43 : Στεγανογραφημένη

pic02_steg_img_67.bmp

3.2.4 Διαδικασία ενσωμάτωσης κειμένου σε εικόνα με χρήση RNG.

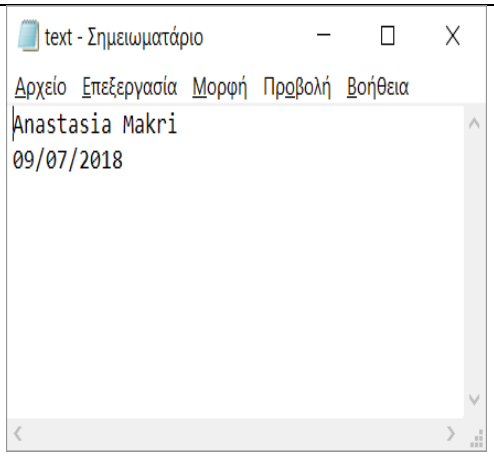
Ο χρήστης μπορεί να έχει μια οποιαδήποτε εικόνα σε οποιαδήποτε μορφή (.jpg, .png, .bmp, .tiff)

Πρέπει να γράφει σε ένα αρχείο το μήνυμά του και να το αποθηκεύσει.



Εικόνα 44 : Εικόνα επιλογής

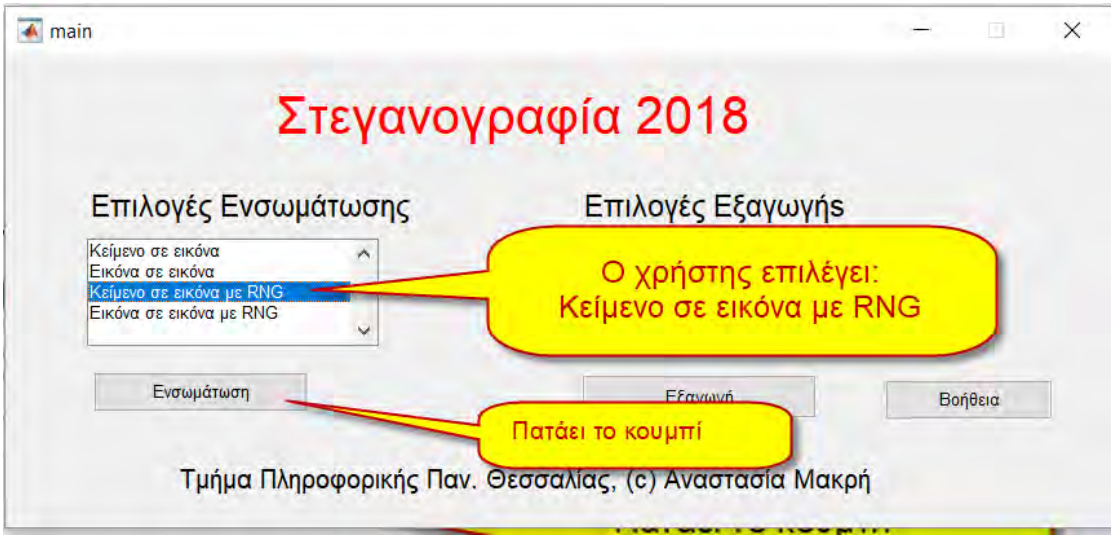
Baboon_big.png



Εικόνα 45 : Κείμενο για ενσωμάτωση

text.txt

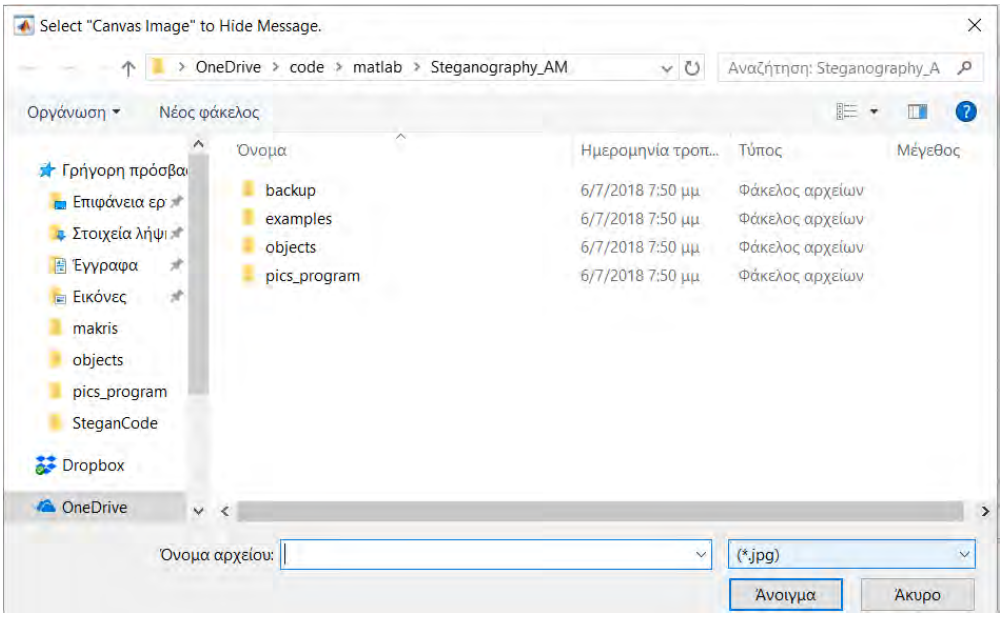
Ο χρήστης αφού επιλέξει από τις Επιλογές Ενσωμάτωσης «Κείμενο σε εικόνα με RNG» πατάει το κουμπί «Ενσωμάτωση»



Εικόνα 46 : Επιλογή ενσωμάτωσης

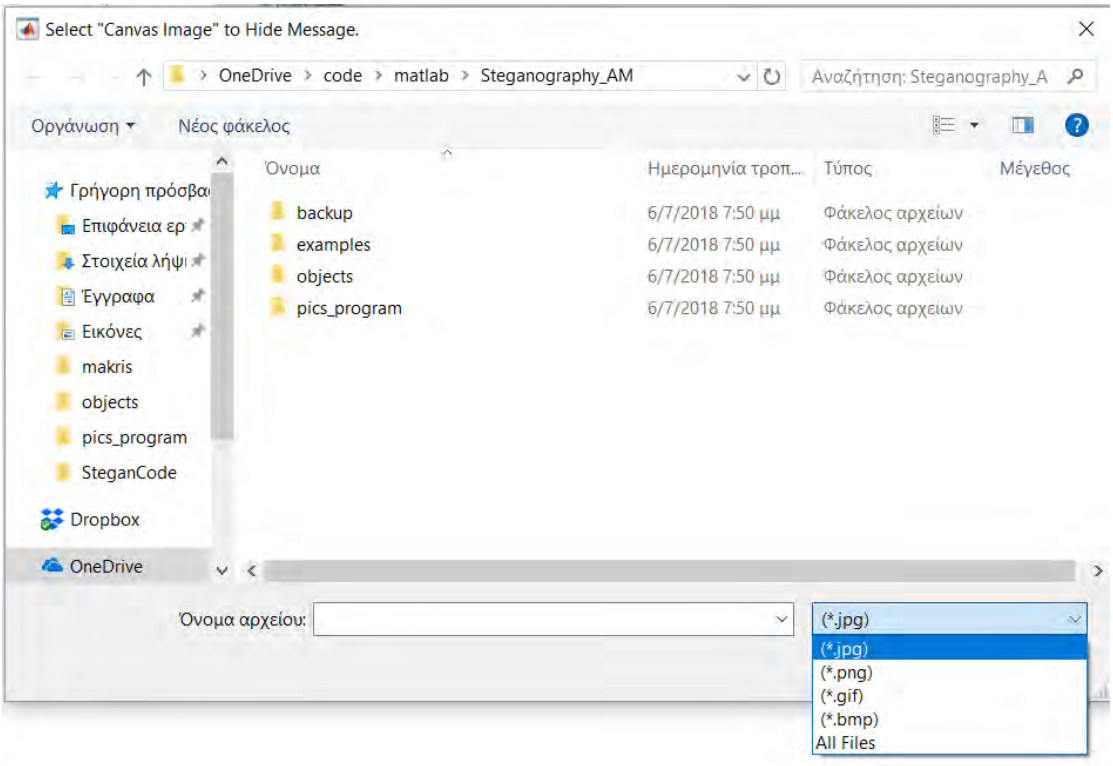
Η εφαρμογή θα ζητήσει από τον χρήστη να επιλέξει την εικόνα που θέλει να χρησιμοποιήσει μέσα στην οποία θα κρύψει το μήνυμά του.

Ο χρήστης έχει την δυνατότητα από οποιοδήποτε φάκελο του υπολογιστή να βρει την εικόνα που επιθυμεί.



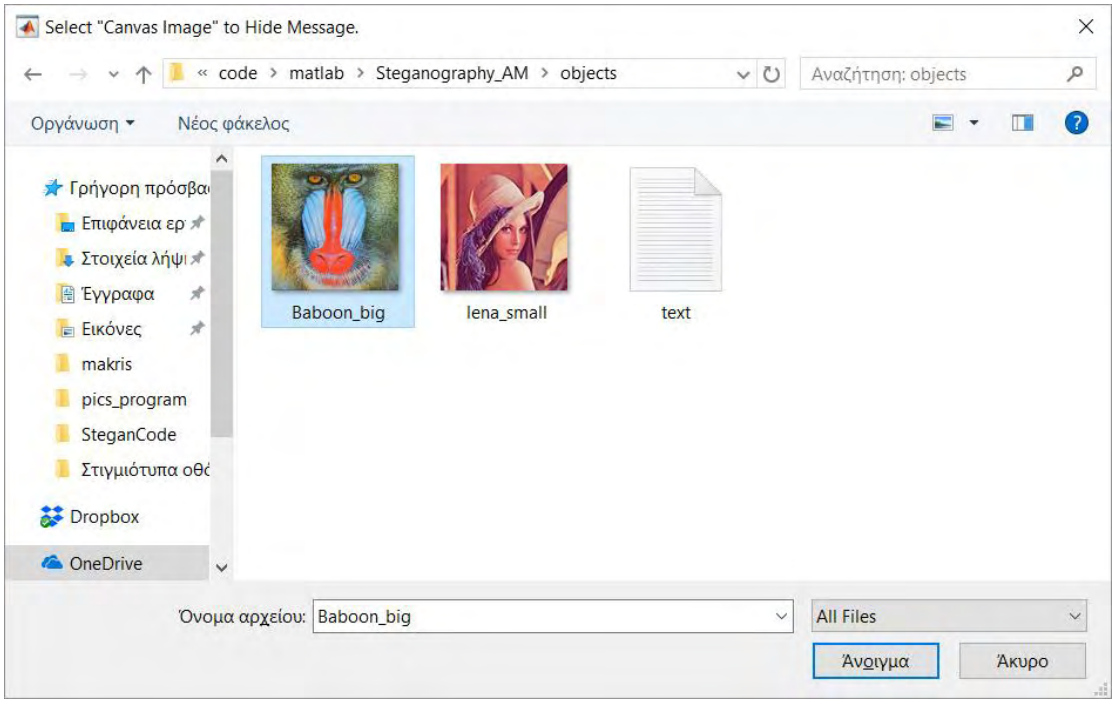
Εικόνα 47 : Αναζήτηση εικόνας

Ως προεπιλεγμένο τύπο αρχείο η εφαρμογή έχει jpg αλλά ο χρήστης μπορεί να επιλέξει άλλον τύπο αρχείου ακόμη και «All files»



Εικόνα 48 : Επιλογή τύπου αρχείου

Και να επιλέξει την εικόνα που επιθυμεί από οποιοδήποτε φάκελο

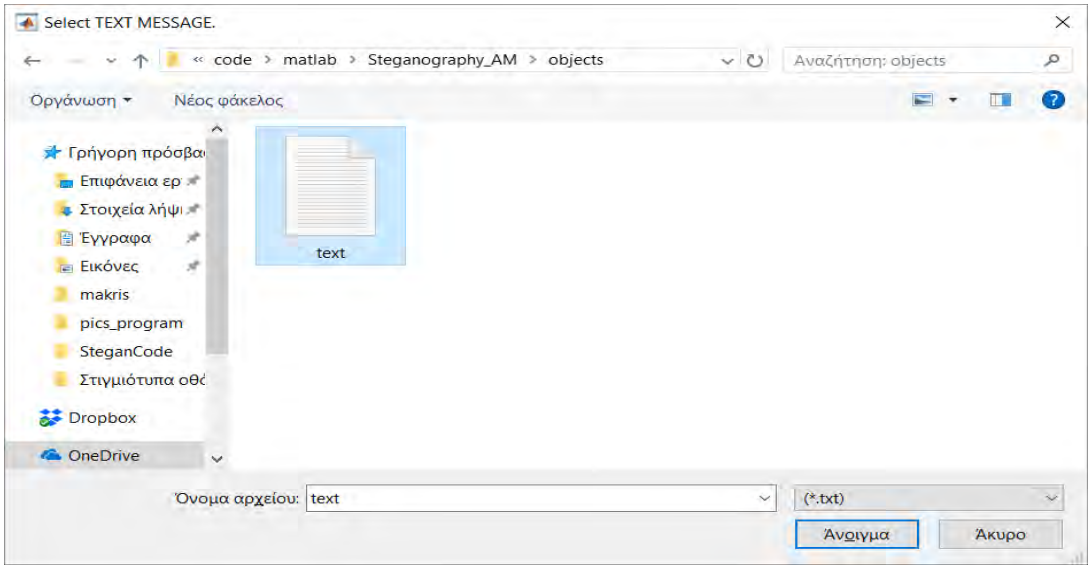


Εικόνα 49 : Επιλογή εικόνας

Εάν ο χρήστης πατήσει το κουμπί «**Άκυρο**» τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Πατώντας το κουμπί άνοιγμα τότε το πρόγραμμα ζητάει από τον χρήστη να επιλέξει το αρχείο κειμένου όπου περιέχει το κείμενο προς απόκρυψη.

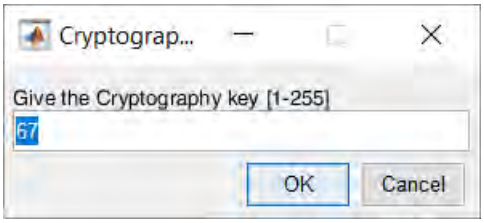
Η διαδικασία είναι η ίδια ακριβώς όπως και με την εικόνα.



Εικόνα 50 : επιλογή αρχείου

Εάν ο χρήστης πατήσει το κουμπί «**Άκυρο**» τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Πατώντας το κουμπί άνοιγμα τότε το πρόγραμμα ζητάει από τον χρήστη να εισάγει το κωδικό κρυπτογράφησης. Ο κωδικός κρυπτογράφησης είναι ένας αριθμός από το 1-255.

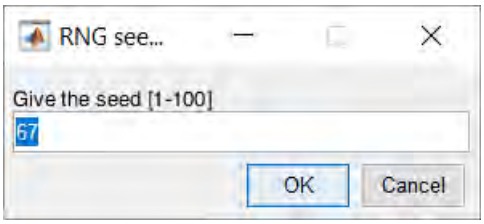


Εικόνα 51 : κωδικός

Εάν ο χρήστης πατήσει το κουμπί «**Cancel**» ή δεν δώσει κωδικό μεταξύ του 1-255 τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Το πρόγραμμα προτείνει ως κωδικό τον αριθμό 67 (από τον ΑΜ της προγραμματίστριας στο τμήμα Πληροφορικής του Πανεπιστημίου Θεσσαλίας).

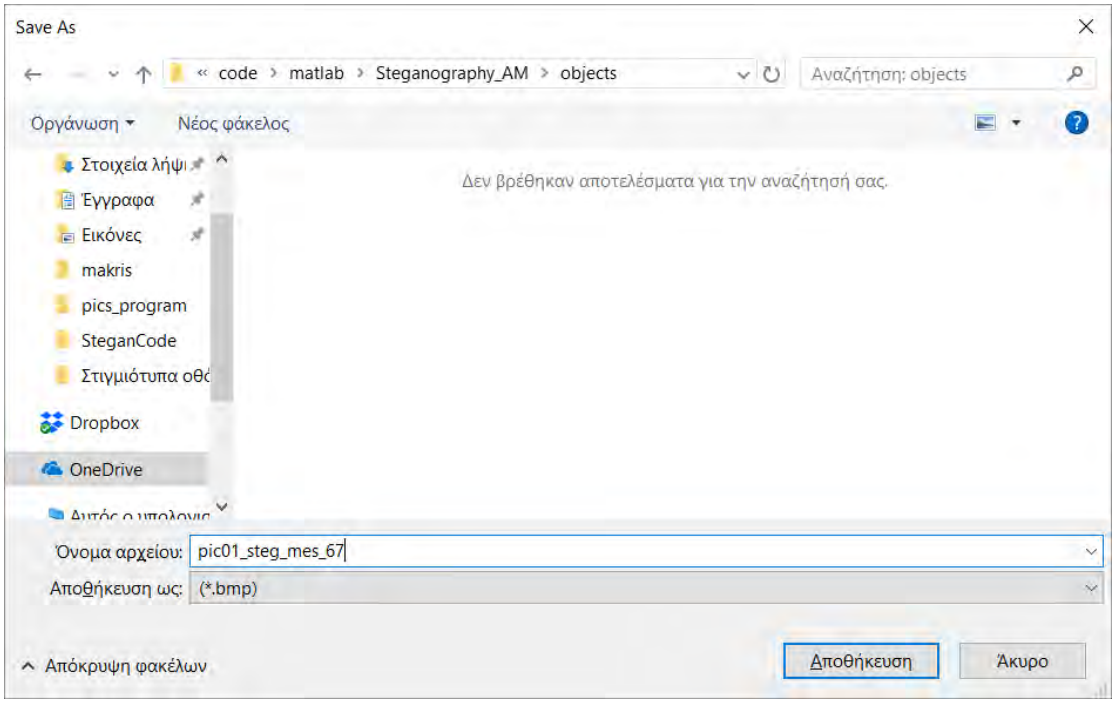
Το πρόγραμμα ζητάει από τον χρήστη να πληκτρολογήσει τον σπόρο (seed) της ακολουθίας τυχαίων αριθμών (για να αποθηκευτεί η πληροφορία σε τυχαίες και όχι συνεχόμενες θέσεις). Ο χρήστης πρέπει να πληκτρολογήσει έναν αριθμό μεταξύ του 1-100.



Εικόνα 52 : Εισαγωγή κωδικού

Εάν ο χρήστης πατήσει το κουμπί «**Cancel**» ή δεν δώσει κωδικό μεταξύ του 1-100 τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

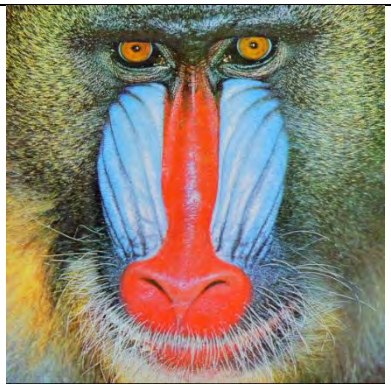
Η αποθήκευση του στεγανογράμματος (εικόνα + μήνυμα) γίνεται σε μορφή **.bmp** γιατί η μορφή **.jpg** διαθέτει αλγορίθμους συμπίεσης και θα αλλοίωνε το τελικό αποτέλεσμα και ίσως χαλούσε και τα δεδομένα της εικόνας.



Εικόνα 53 : Αποθήκευση

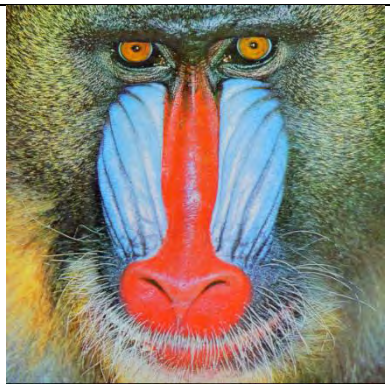
Ο χρήστης πληκτρολογεί ένα όνομα που επιθυμεί (προφανώς μπορεί να επιλέξει και τον φάκελο που θέλει) και μετά πατά αποθήκευση.

Εφόσον ολοκληρωθεί η διαδικασία τότε έχει δημιουργηθεί το στεγανόγραμμα που περιέχει την εικόνα και το κείμενο μαζί.



Εικόνα 54 : Αρχική εικόνα

Baboon_big.png

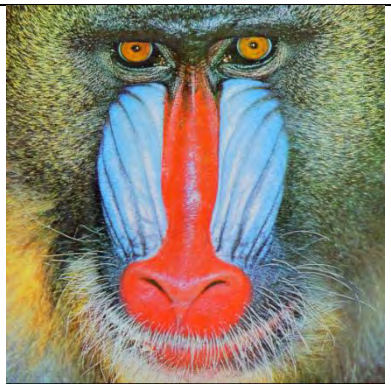


Εικόνα 55 : Στεγανογραφημένη εικόνα

pic03_steg_img_67_rng_67.bmp

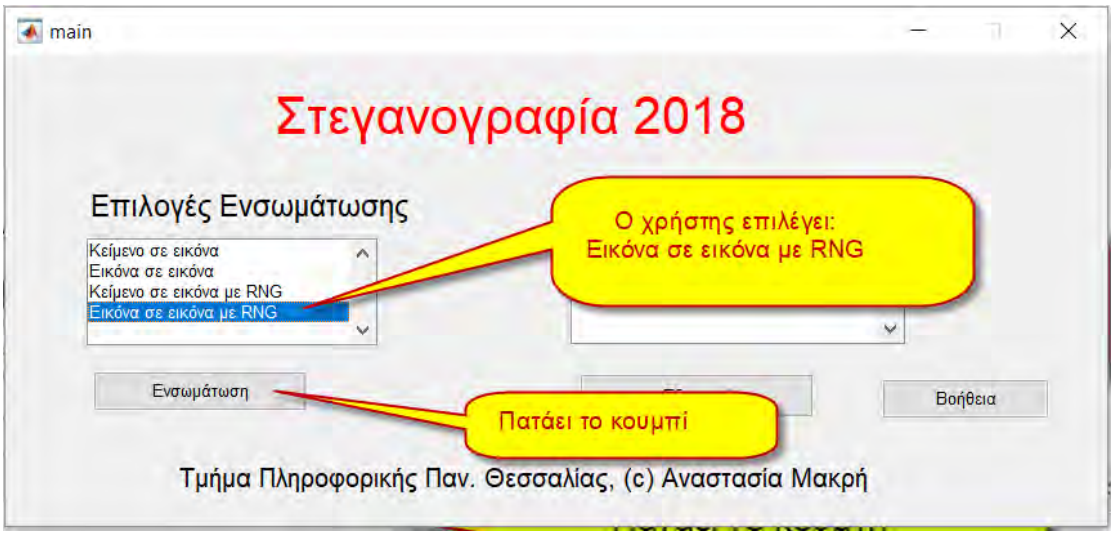
3.2.5 Διαδικασία ενσωμάτωσης Εικόνα σε Εικόνα με χρήση RNG

Ο χρήστης μπορεί να έχει δύο οποιεσδήποτε εικόνες σε οποιαδήποτε μορφή (.jpg, .png, .bmp, .tiff)

	
Εικόνα 56 : Επιλογή εικόνας ως κάλυμμα	*Εικόνα 57 : Επιλογή εικόνας για ενσωμάτωση*
Baboon_big.png	lena_small.png

Η πρώτη εικόνα είναι το κάλυμμα(cover) και η δεύτερη εικόνα το μήνυμα που θέλουμε να κρύψουμε.

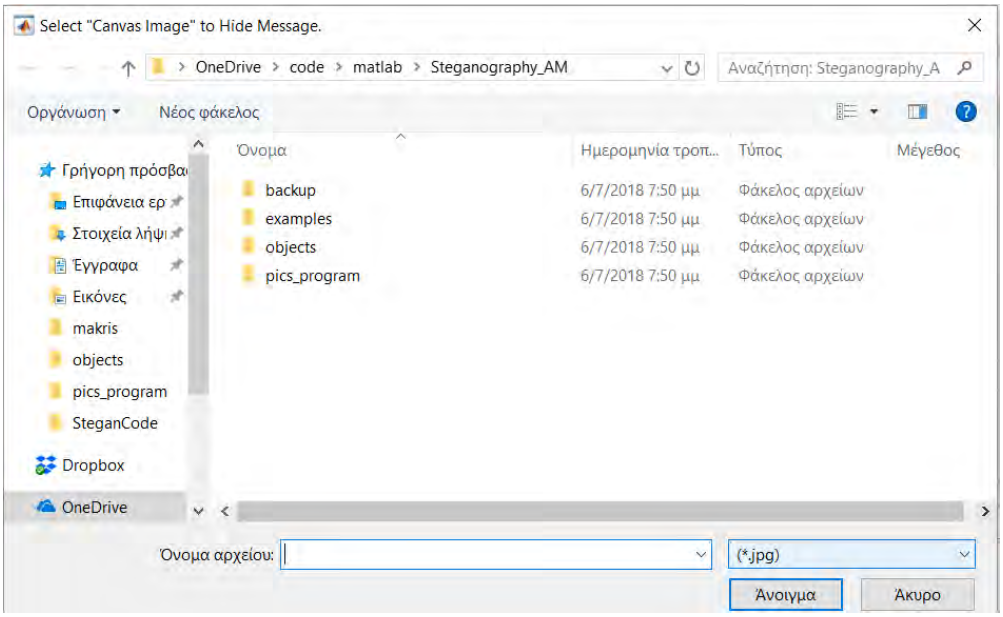
Ο χρήστης αφού επιλέξει από τις Επιλογές Ενσωμάτωσης «Εικόνα σε εικόνα με χρήση RNG» πατάει το κουμπί «Ενσωμάτωση»



Εικόνα 58 : Επιλογή Ενσωμάτωσης

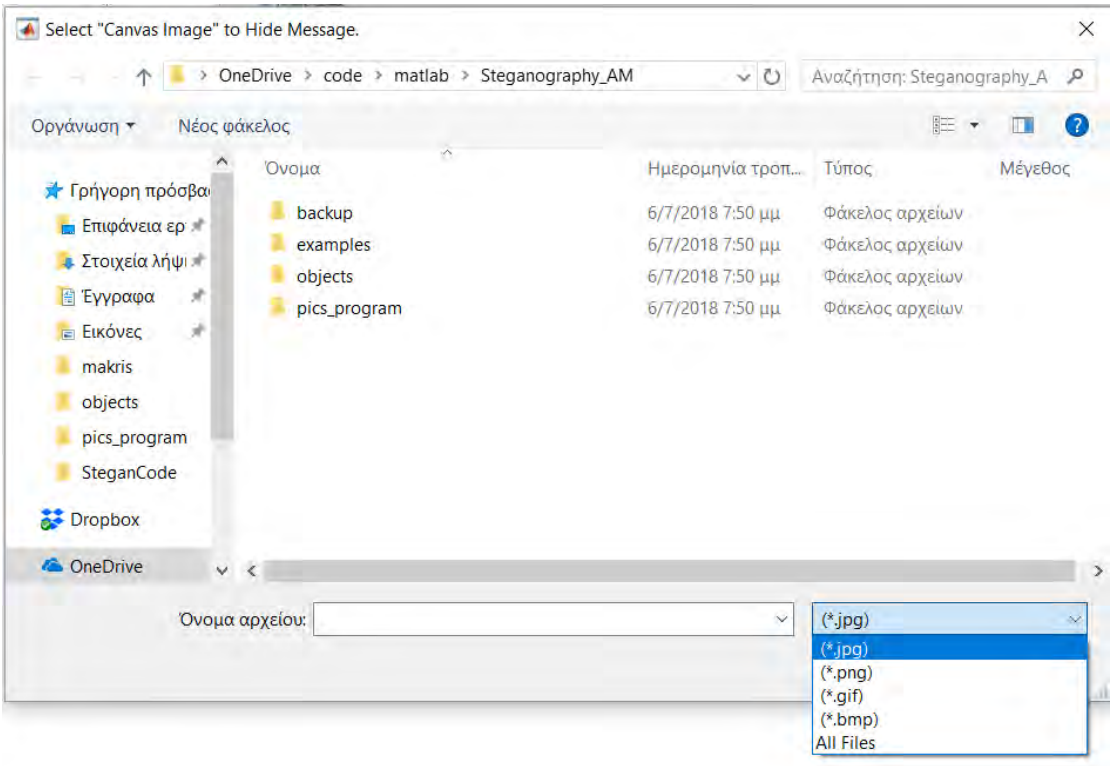
Η εφαρμογή θα ζητήσει από τον χρήστη να επιλέξει την εικόνα που θέλει να χρησιμοποιήσει μέσα στην οποία θα κρύψει το μήνυμά του.

Ο χρήστης έχει την δυνατότητα από οποιοδήποτε φάκελο του υπολογιστή να βρει την εικόνα που επιθυμεί.



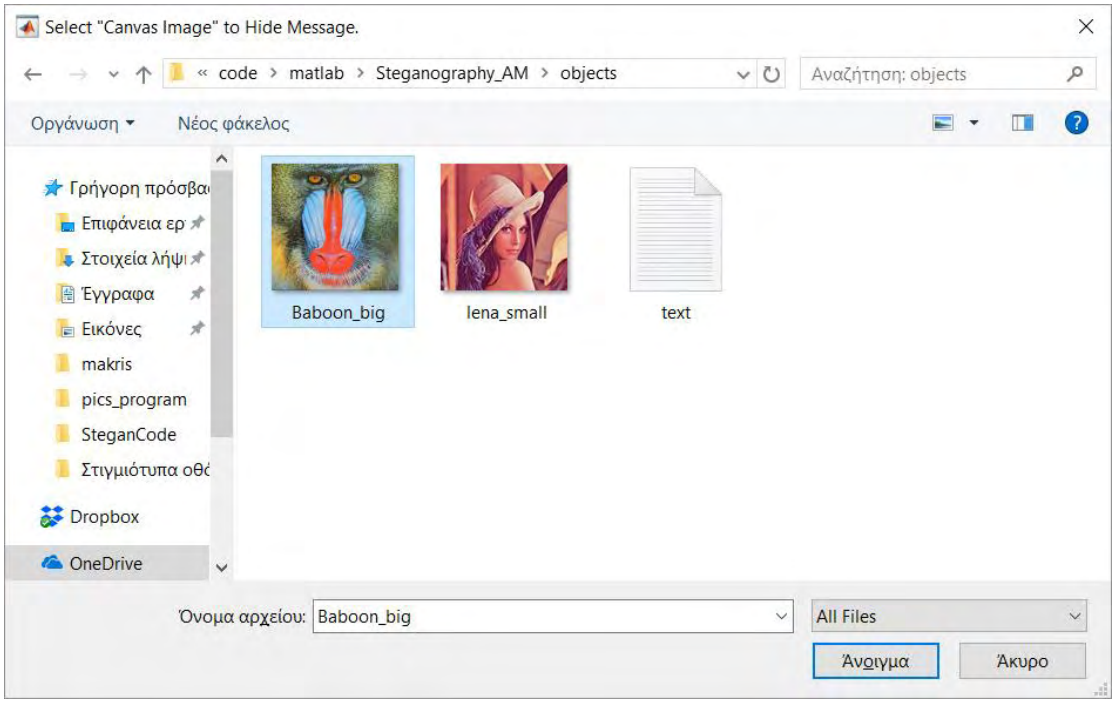
Εικόνα 59 : Ανοιγμα εικόνας cover

Ως προεπιλεγμένο τύπο αρχείο η εφαρμογή έχει jpg αλλά ο χρήστης μπορεί να επιλέξει άλλον τύπο αρχείου ακόμη και «All files»



Εικόνα 60 : Επιλογή τύπου All files

Και να επιλέξει την εικόνα που επιθυμεί από οποιοδήποτε φάκελο

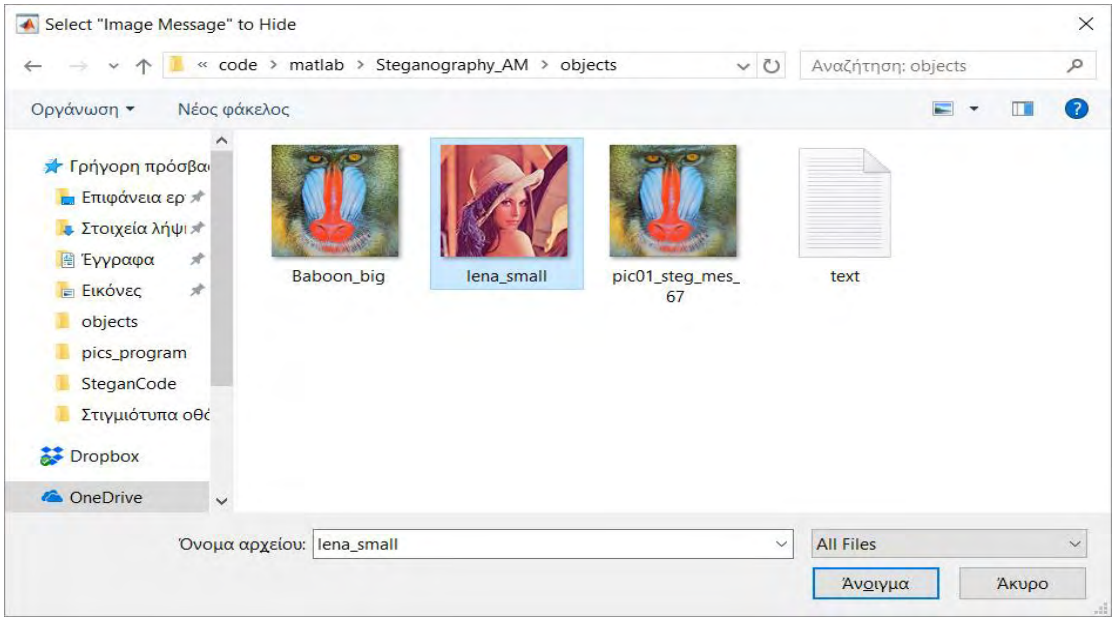


Εικόνα 61 : Επιλογή εικόνας

Εάν ο χρήστης πατήσει το κουμπί «**Άκυρο**» τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Πατώντας το κουμπί άνοιγμα τότε το πρόγραμμα ζητάει από τον χρήστη να επιλέξει την εικόνα «μήνυμα» προς απόκρυψη.

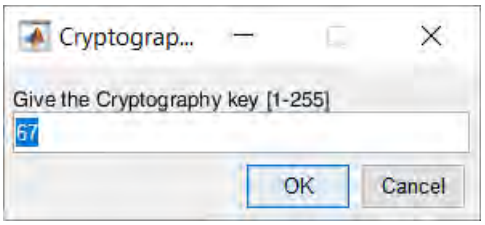
Η διαδικασία είναι η ίδια ακριβώς όπως και με την εικόνα cover.



Εικόνα 62 : Άνοιγμα εικόνας

Εάν ο χρήστης πατήσει το κουμπί «**Άκυρο**» τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Πατώντας το κουμπί άνοιγμα τότε το πρόγραμμα ζητάει από τον χρήστη να εισάγει το κωδικό κρυπτογράφησης. Ο κωδικός κρυπτογράφησης είναι ένας αριθμός από το 1-255.

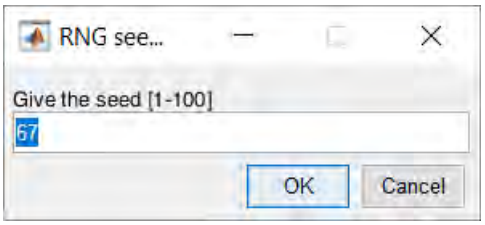


Εικόνα 63 : Κωδικός επιλογής

Εάν ο χρήστης πατήσει το κουμπί «**Cancel**» ή δεν δώσει κωδικό μεταξύ του 1-255 τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Το πρόγραμμα προτείνει ως κωδικό τον αριθμό 67 (από τον ΑΜ της προγραμματίστριας στο τμήμα Πληροφορικής του Πανεπιστημίου Θεσσαλίας).

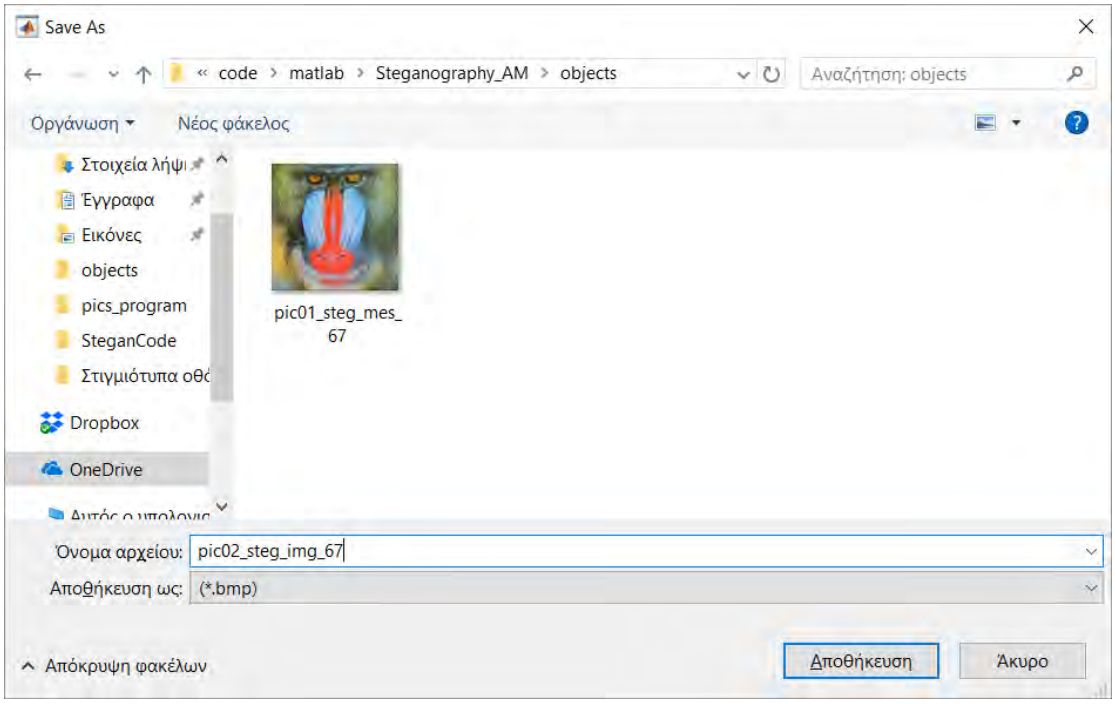
Το πρόγραμμα ζητάει από τον χρήστη να πληκτρολογήσει τον σπόρο (seed) της ακολουθίας τυχαίων αριθμών (για να αποθηκευτεί η πληροφορία σε τυχαίες και όχι συνεχόμενες θέσεις). Ο χρήστης πρέπει να πληκτρολογήσει έναν αριθμό μεταξύ του 1-100.



Εικόνα 64 : Εισαγωγή κωδικού

Εάν ο χρήστης πατήσει το κουμπί «**Cancel**» ή δεν δώσει κωδικό μεταξύ του 1-100 τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

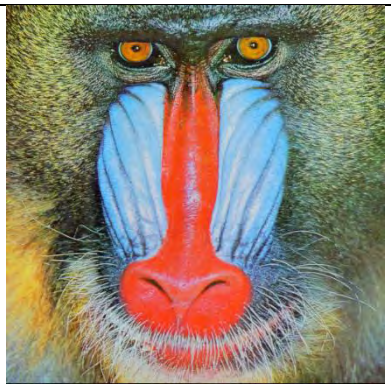
Η αποθήκευση του στεγανογράμματος (εικόνα + μήνυμα) γίνεται σε μορφή .**bmp** γιατί η μορφή **.jpg** διαθέτει αλγορίθμους συμπίεσης και θα αλλοίωνε το τελικό αποτέλεσμα και ίσως χαλούσε και τα δεδομένα της εικόνας.



Εικόνα 65 : Αποθήκευση

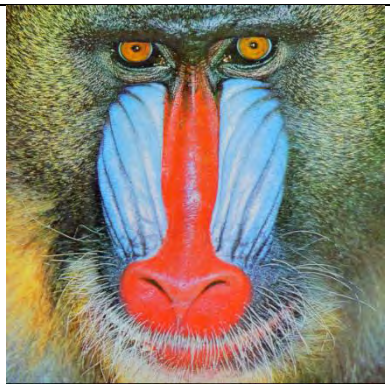
Ο χρήστης πληκτρολογεί ένα όνομα που επιθυμεί (προφανώς μπορεί να επιλέξει και τον φάκελο που θέλει) και μετά πατά αποθήκευση.

Εφόσον ολοκληρωθεί η διαδικασία τότε έχει δημιουργηθεί το στεγανόγραμμα που περιέχει την εικόνα και την εικόνα μήνυμα μαζί.



Εικόνα 66 : Αρχική εικόνα

Baboon_big.png



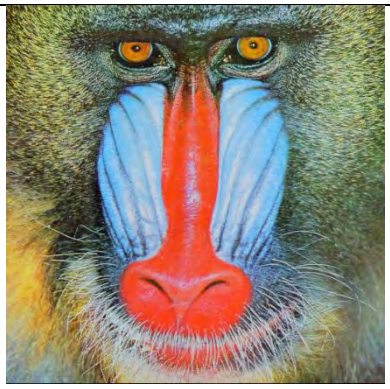
Εικόνα 67 : Στεγανογραφημένη

pic02_steg_img_67.bmp

3.2.6 Διαδικασία Εξαγωγής μηνύματος από Εικόνα.

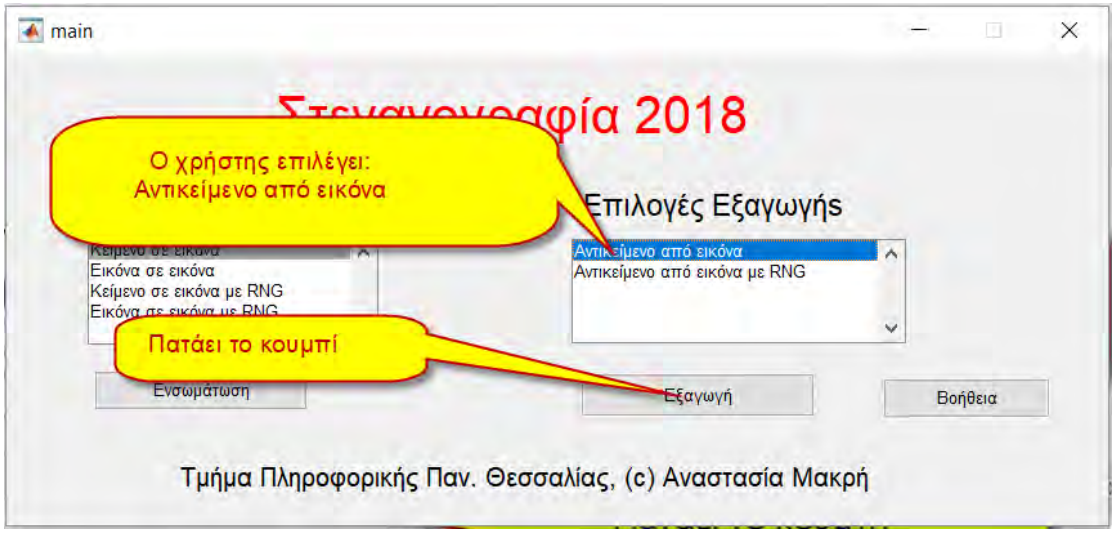
Ο χρήστης διαθέτει το στεγανογράφημα δηλαδή την εικόνα που περιέχει το μήνυμα μέσα (η εικόνα είναι σε μορφή .bmp). Πρέπει να γνωρίζει επίσης και τον κωδικό κρυπτογράφησης του μηνύματος μέσα στην εικόνα

Έστω ότι έχει την εικόνες:

 *Εικόνα 68 : Κρυμμένο κείμενο*	 *Εικόνα 69 : Κρυμμένη Εικόνα*
pic01_steg_mes_67.bmp	pic02_steg_img_67.bmp

Η πρώτη εικόνα έχει κρυμμένο κείμενο ενώ η δεύτερη έχει κρυμμένη εικόνα. Ο χρήστης δεν γνωρίζει τι υπάρχει κρυμμένο μέσα στις εικόνες απλώς γνωρίζει τον κωδικό κρυπτογράφησης ότι είναι ο «67» και με το πρόγραμμα θέλει να πάρει/διαβάσει το μήνυμα.

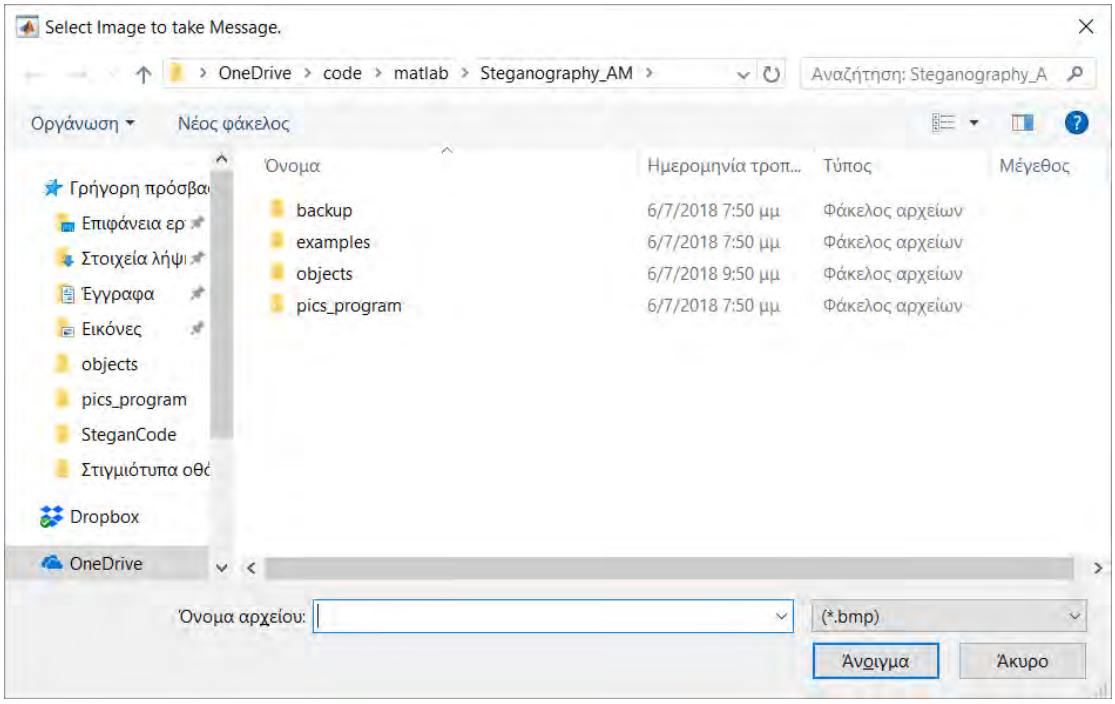
Ο χρήστης αφού επιλέξει από τις Επιλογές Εξαγωγής «Αντικείμενο από εικόνα» πατάει το κουμπί «Ενσωμάτωση»



Εικόνα 70 : Επιλογή Εξαγωγής

Η εφαρμογή θα ζητήσει από τον χρήστη να επιλέξει την εικόνα που θέλει να χρησιμοποιήσει μέσα στην οποία έχει κρυφτεί το μήνυμά.

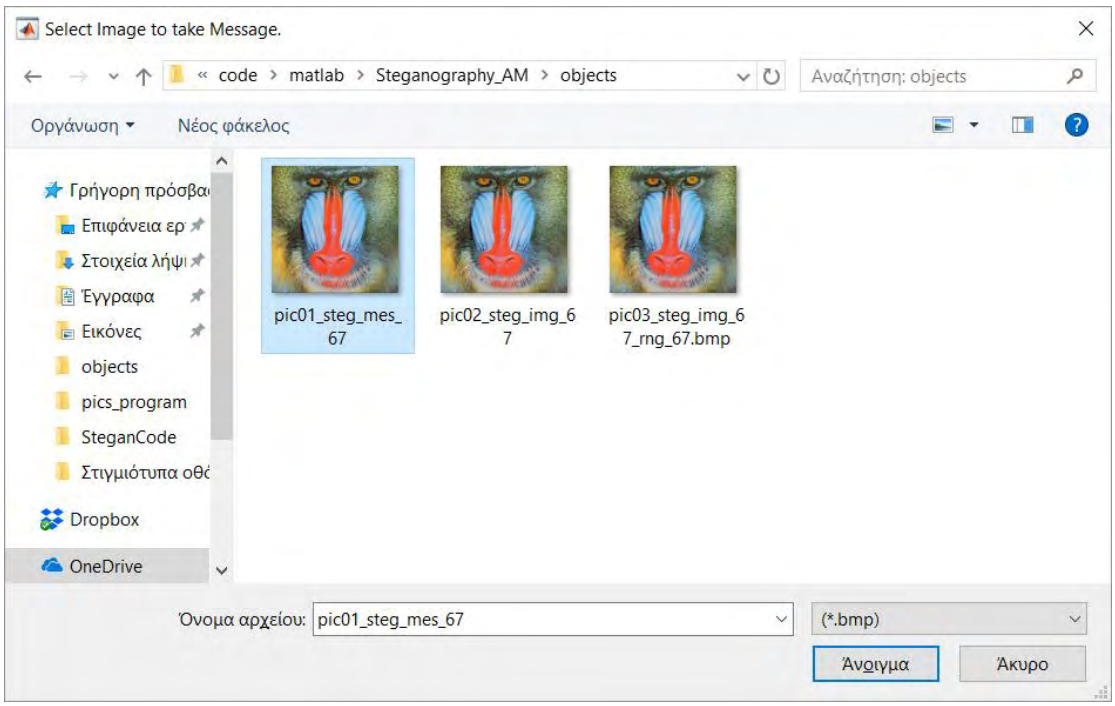
Ο χρήστης έχει την δυνατότητα από οποιοδήποτε φάκελο του υπολογιστή να βρει την εικόνα που επιθυμεί.



Εικόνα 71 : Επιλογή από φάκελο

Ως προεπιλεγμένο τύπο αρχείο η εφαρμογή έχει .bmp

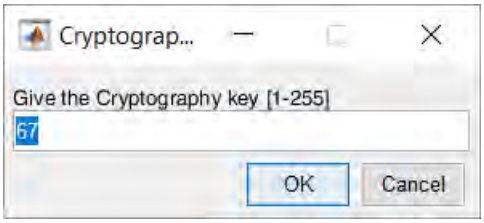
Και να επιλέξει την εικόνα που επιθυμεί από οποιοδήποτε φάκελο



Εικόνα 72 : Επιλογή εικόνας

Εάν ο χρήστης πατήσει το κουμπί «**Άκυρο**» τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Πατώντας το κουμπί άνοιγμα τότε το πρόγραμμα ζητάει από τον χρήστη να εισάγει το κωδικό κρυπτογράφησης. Ο κωδικός κρυπτογράφησης είναι ένας αριθμός από το 1-255.



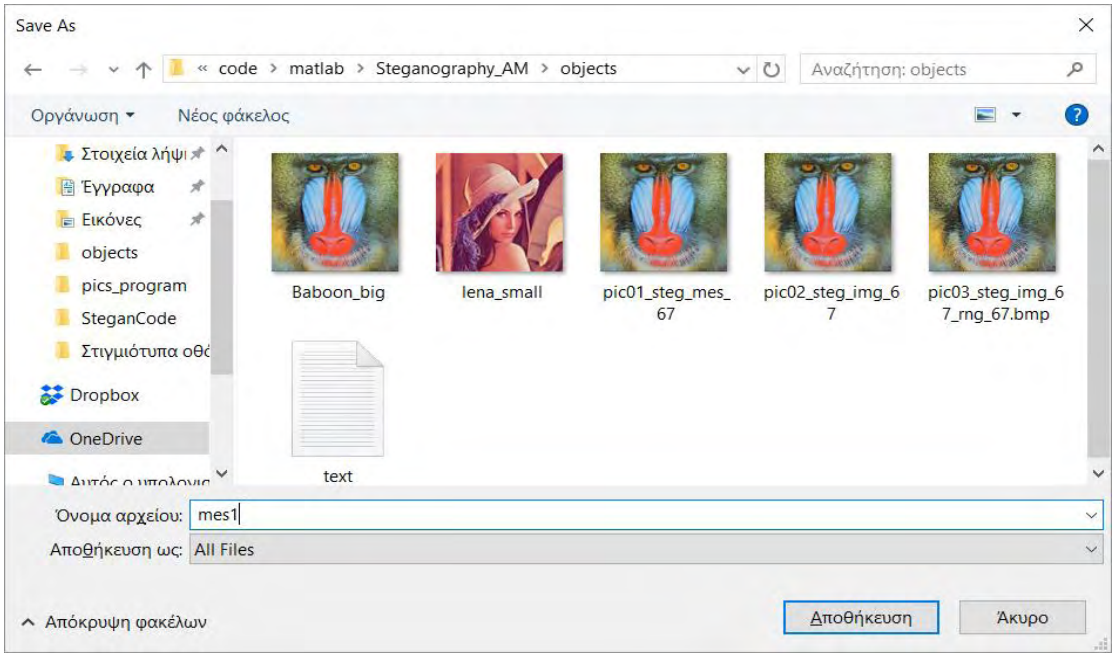
Εικόνα 73 : Κωδικός

Εάν ο χρήστης πατήσει το κουμπί «**Cancel**» ή δεν δώσει κωδικό μεταξύ του 1-255 τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Το πρόγραμμα προτείνει ως κωδικό τον αριθμό 67 (από τον ΑΜ της προγραμματίστριας στο τμήμα Πληροφορικής του Πανεπιστημίου Θεσσαλίας).

Η εξαγωγή του μηνύματος γίνεται αυτόματα εφόσον ο κωδικός κρυπτογράφησης είναι ο ίδιος με αυτόν που έγινε η κρυπτογράφηση. Σε περίπτωση που δεν είναι ο ίδιος το πρόγραμμα επιστρέφει στην κεντρική οθόνη χωρίς να κάνει τίποτα.

Σε περίπτωση που ο κωδικός κρυπτογράφησης είναι σωστός τότε το πρόγραμμα ζητάει από τον χρήστη το όνομα του αρχείου που επιθυμεί να αποθηκεύσει το μήνυμα.

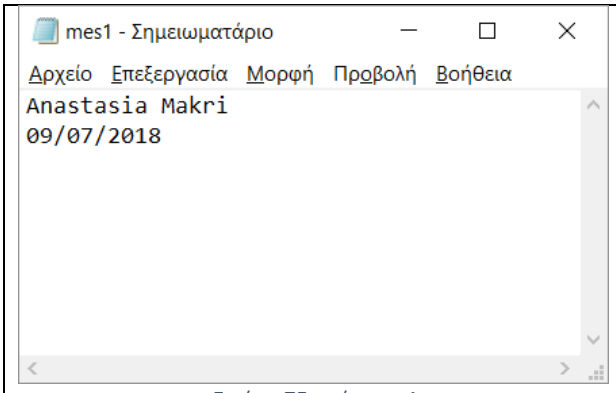


Εικόνα 74 : Αποθήκευση

Ο χρήστης πληκτρολογεί ένα όνομα που επιθυμεί (προφανώς μπορεί να επιλέξει και τον φάκελο που θέλει) και μετά πατά αποθήκευση.

Εφόσον ολοκληρωθεί η διαδικασία τότε το πρόγραμμα αποθηκεύει στο όνομα του αρχείου το μήνυμα. Το πρόγραμμα αναγνωρίζει αυτόματα εάν το μήνυμα είναι κείμενο ή εικόνα και προσθέτει μόνο του την κατάληξη .txt (σε περίπτωση κειμένου) και .bmp σε περίπτωση εικόνας (να θυμίσουμε ότι οι εικόνες για λόγους οικονομίας και επεξεργασίας αποθηκεύονται σε αποχρώσεις του γκρι και όχι έγχρωμες)

Επαναλαμβάνοντας την ίδια διαδικασία για την δεύτερη εικόνα που είχαμε στην διάθεσή μας και δίνονται ως όνομα το mes2 τότε αποθηκεύεται η εικόνα που ήταν αποθηκευμένη μέσα



Εικόνα 75 : μήνυμα 1

mes1.txt

Εικόνα 76 : μήνυμα 2

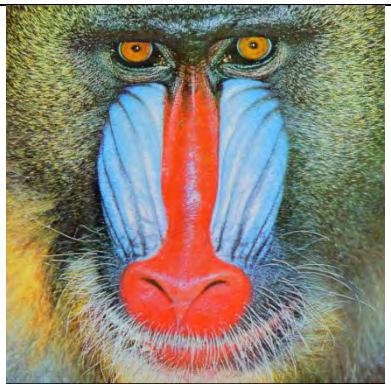
mes2.bmp

Τα οποία αρχεία τα είχαμε στεγανοποιήσει (ενσωματώσει) στην αρχική εικόνα.

3.2.7 Διαδικασία Εξαγωγής μηνύματος από Εικόνα με χρήση RNG.

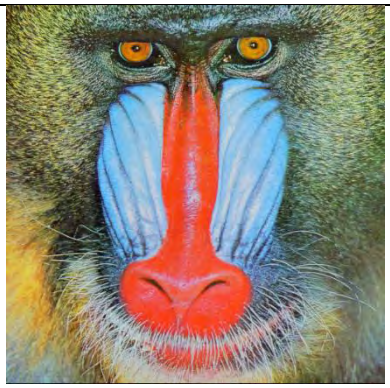
Ο χρήστης διαθέτει το στεγανογράφημα δηλαδή την εικόνα που περιέχει το μήνυμα μέσα (η εικόνα είναι σε μορφή .bmp). Πρέπει να γνωρίζει επίσης και τον κωδικό κρυπτογράφησης του μηνύματος μέσα στην εικόνα καθώς και τον σπόρο (seed) για την γεννήτρια τυχαίων αριθμών.

Έστω ότι έχει την εικόνες:



Εικόνα 77 : κείμενο σε εικόνα

pic01_steg_mes_67_rng_67.bmp

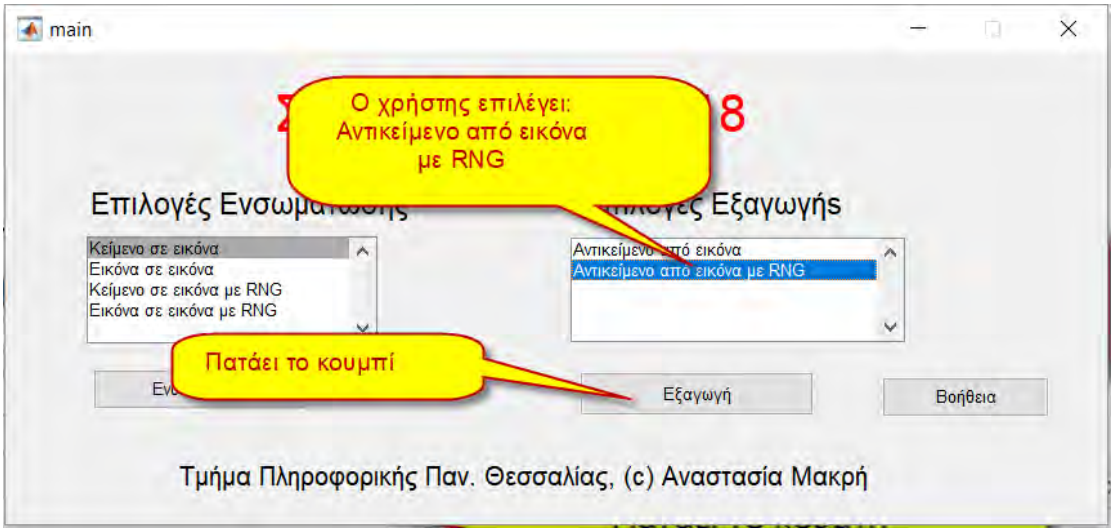


Εικόνα 78 : εικόνα σε εικόνα

pic02_steg_img_67_rng_67.bmp

Η πρώτη εικόνα έχει κρυμμένο κείμενο ενώ η δεύτερη έχει κρυμμένη εικόνα. Ο χρήστης δεν γνωρίζει τι υπάρχει κρυμμένο μέσα στις εικόνες απλώς γνωρίζει τον κωδικό κρυπτογράφησης ότι είναι ο «67» και με το πρόγραμμα θέλει να πάρει/διαβάσει το μήνυμα.

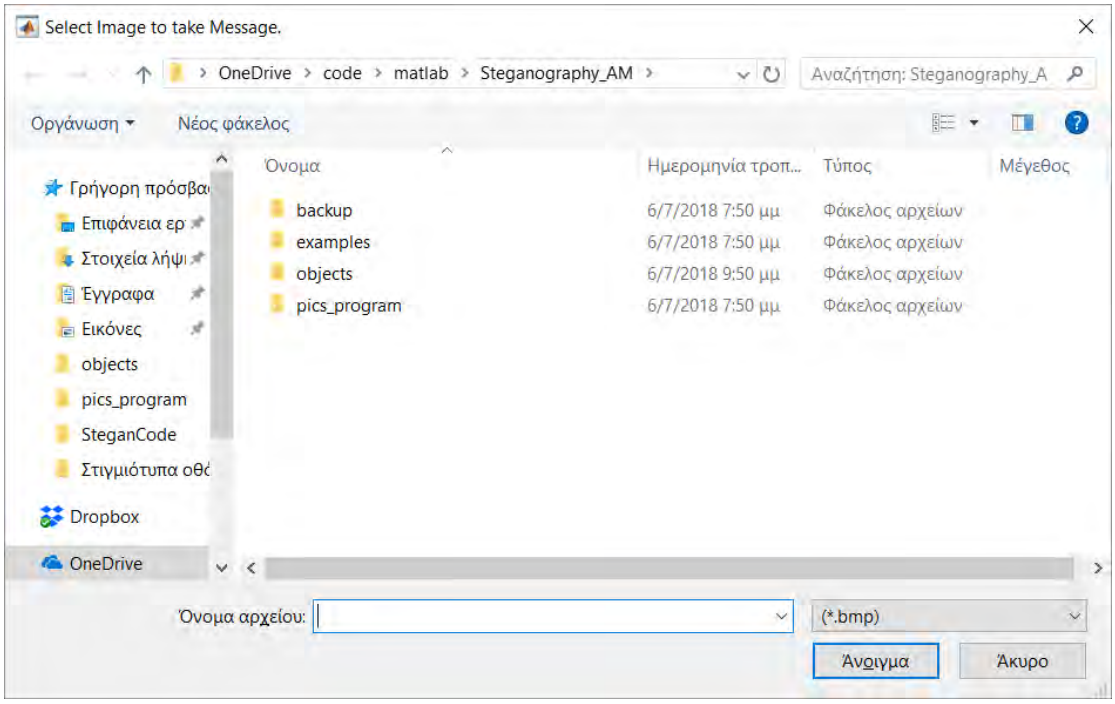
Ο χρήστης αφού επιλέξει από τις Επιλογές Εξαγωγής «Αντικείμενο από εικόνα με RNG» πατάει το κουμπί «Ενσωμάτωση»



Εικόνα 79 : Επιλογή εξαγωγής

Η εφαρμογή θα ζητήσει από τον χρήστη να επιλέξει την εικόνα που θέλει να χρησιμοποιήσει μέσα στην οποία έχει κρυφτεί το μήνυμά.

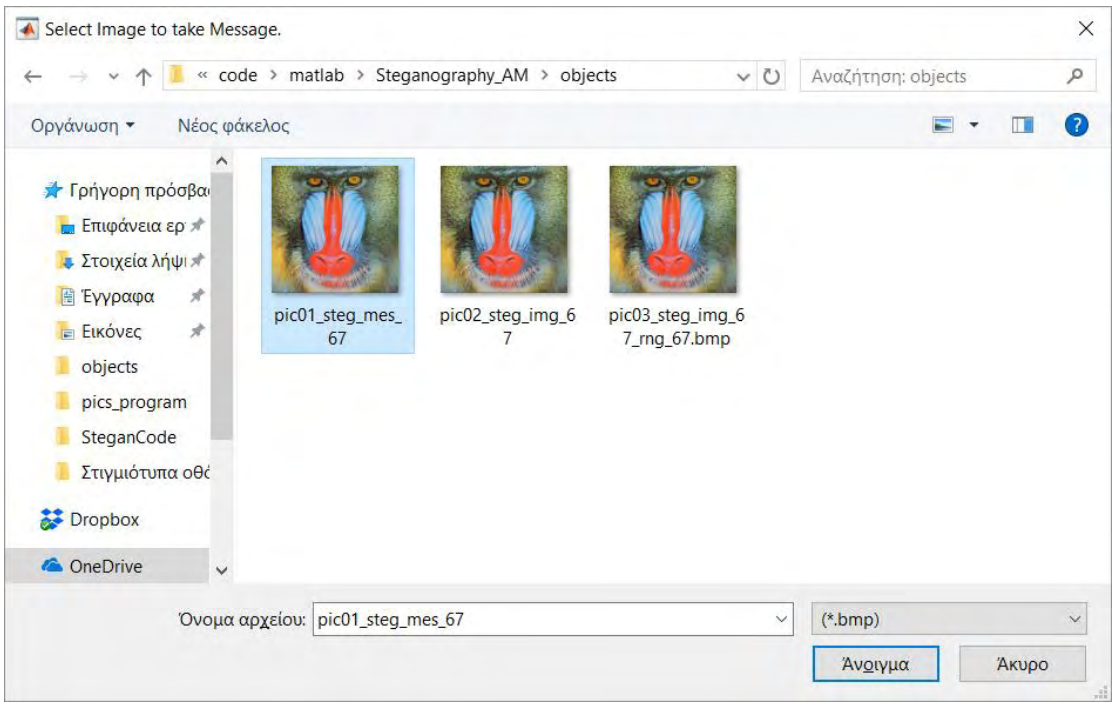
Ο χρήστης έχει την δυνατότητα από οποιοδήποτε φάκελο του υπολογιστή να βρει την εικόνα που επιθυμεί.



Εικόνα 80 : Αναζήτηση εικόνας

Ως προεπιλεγμένο τύπο αρχείο η εφαρμογή έχει .bmp

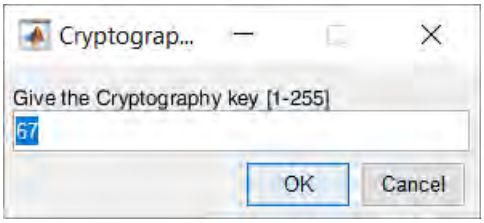
Και να επιλέξει την εικόνα που επιθυμεί από οποιοδήποτε φάκελο



Εικόνα 81 : Επιλογή εικόνας

Εάν ο χρήστης πατήσει το κουμπί «**Άκυρο**» τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Πατώντας το κουμπί άνοιγμα τότε το πρόγραμμα ζητάει από τον χρήστη να εισάγει το κωδικό κρυπτογράφησης. Ο κωδικός κρυπτογράφησης είναι ένας αριθμός από το 1-255.

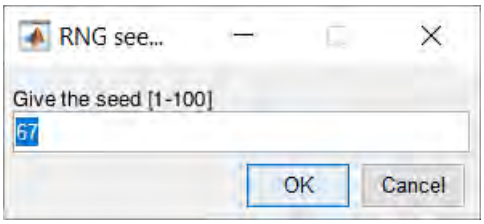


Εικόνα 82 : Κωδικός

Εάν ο χρήστης πατήσει το κουμπί «**Cancel**» ή δεν δώσει κωδικό μεταξύ του 1-255 τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Το πρόγραμμα προτείνει ως κωδικό τον αριθμό 67 (από τον ΑΜ της προγραμματίστριας στο τμήμα Πληροφορικής του Πανεπιστημίου Θεσσαλίας).

Το πρόγραμμα ζητάει από τον χρήστη να πληκτρολογήσει τον σπόρο (seed) της ακολουθίας τυχαίων αριθμών (που έχει αποθηκευτεί η πληροφορία σε τυχαίες και όχι συνεχόμενες θέσεις). Ο χρήστης πρέπει να πληκτρολογήσει έναν αριθμό μεταξύ του 1-100.

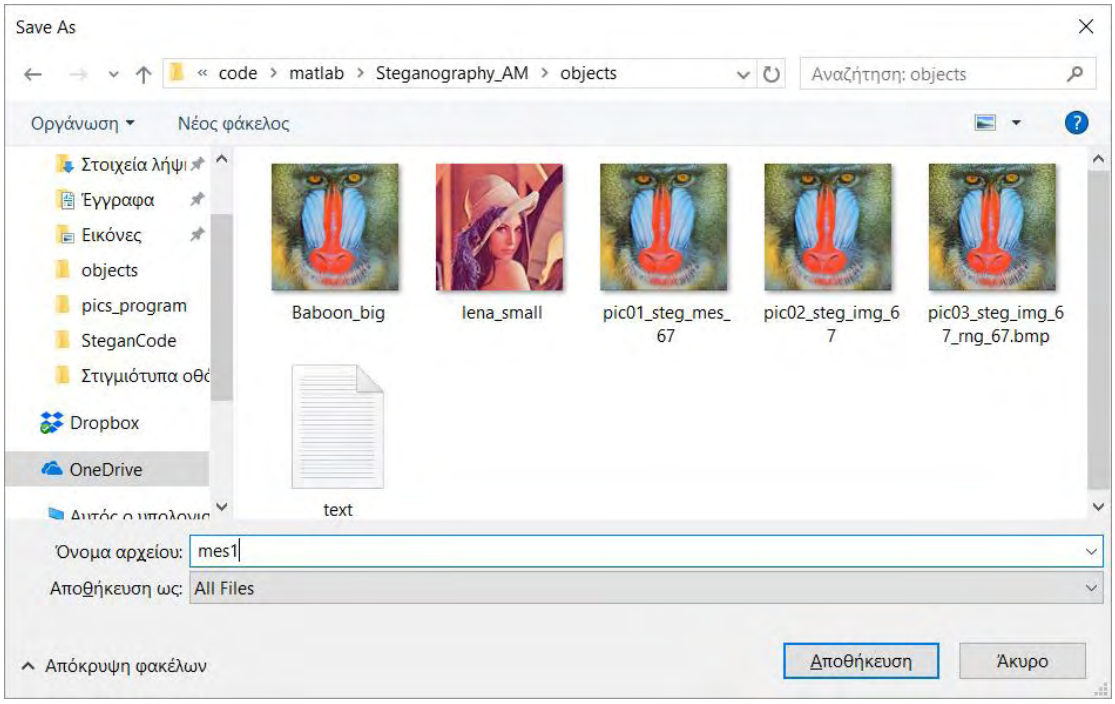


Εικόνα 83 : Εισαγωγή κωδικού

Εάν ο χρήστης πατήσει το κουμπί «**Cancel**» ή δεν δώσει κωδικό μεταξύ του 1-100 τότε ακυρώνεται η διαδικασία και το πρόγραμμα επιστρέφει στην κεντρική οθόνη.

Η εξαγωγή του μηνύματος γίνεται αυτόματα εφόσον ο κωδικός κρυπτογράφησης είναι ο ίδιος με αυτόν που έγινε η κρυπτογράφηση και το seed είναι το σωστό. Σε περίπτωση που δεν είναι τα ίδια το πρόγραμμα επιστρέφει στην κεντρική οθόνη χωρίς να κάνει τίποτα.

Σε περίπτωση που ο κωδικός κρυπτογράφησης είναι σωστός και το seed είναι σωστό τότε το πρόγραμμα ζητάει από τον χρήστη το όνομα του αρχείου που επιθυμεί να αποθηκεύσει το μήνυμα.

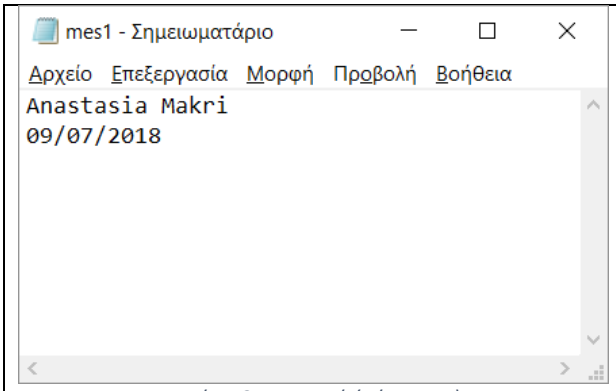


Εικόνα 84 : Αποθήκευση

Ο χρήστης πληκτρολογεί ένα όνομα που επιθυμεί (προφανώς μπορεί να επιλέξει και τον φάκελο που θέλει) και μετά πατά αποθήκευση.

Εφόσον ολοκληρωθεί η διαδικασία τότε το πρόγραμμα αποθηκεύει στο όνομα του αρχείου το μήνυμα. Το πρόγραμμα αναγνωρίζει αυτόματα εάν το μήνυμα είναι κείμενο ή εικόνα και προσθέτει μόνο του την κατάληξη .txt (σε περίπτωση κειμένου) και .bmp σε περίπτωση εικόνας (να θυμίσουμε ότι οι εικόνες για λόγους οικονομίας και επεξεργασίας αποθηκεύονται σε αποχρώσεις του γκρι και όχι έγχρωμες)

Επαναλαμβάνοντας την ίδια διαδικασία για την δεύτερη εικόνα που είχαμε στην διάθεσή μας και δίνονται ως όνομα το mes2 τότε αποθηκεύεται η εικόνα που ήταν αποθηκευμένη μέσα



Εικόνα 85 : Αρχική (μήνυμα 1)

mes1.txt

Εικόνα 86 : τελική(μήνυμα 2)

mes2.bmp

Τα οποία αρχεία τα είχαμε στεγανοποιήσει (ενσωματώσει) στην αρχική εικόνα.

Συμπεράσματα

Στην παρούσα πτυχιακή παρουσιάστηκε η τεχνική της στεγανογραφίας καθώς και εφαρμογές που έχουν αναπτυχθεί για την υλοποίησή της.

Αναπτύχθηκε εφαρμογή στο προγραμματιστικό περιβάλλον Matlab η οποία δίνει την δυνατότητα στον χρήστη να ενσωματώσει/αποκρύψει κείμενο ή εικόνα μέσα σε εικόνα, έτσι ώστε να είναι δυνατή η μεταφορά πληροφορίας χωρίς να γίνεται αντιληπτή από τρίτο πρόσωπο. Για να αυξηθεί η ασφάλεια της κρυμμένης πληροφορίας η εφαρμογή κρυπτογραφεί τα δεδομένα. Επίσης προστέθηκε επιπλέον επίπεδο ασφαλείας εισάγοντας γεννήτρια τυχαίων αριθμών για την επιλογή και τοποθέτηση σε τυχαίες θέσεις της πληροφορίας.

Η εφαρμογή είναι πολύ απλή στην χρήση και δίνει την δυνατότητα στον χρήστη μέσα από απλές επιλογές και παράθυρα διαλόγου να αποκρύψει και να εξάγει μηνύματα (κείμενα και εικόνες) από εικόνες.

Ο περιορισμός της απόκρυψης ενός μηνύματος N byte σε μία εικόνα είναι ότι απαιτούνται: $3 \times N$ pixels χρησιμοποιώντας αλλαγή 3 χρωμάτων ανά pixel, ενώ απαιτούνται $8 \times N$ pixels χρησιμοποιώντας αλλαγή 1 επιπέδου χρώματος ανά pixel (Makris G., Antoniou I., 2015).

Μελλοντικές επεκτάσεις της εφαρμογής: αποκρύπτει κείμενο, εικόνα, ήχο, video σε εικόνες, ήχους, video.

Αναφορές

Βιβλιογραφία

Ξενόγλωσση

1. Kaushal Solanki-Information Hiding, 10 conf., IH 2008-Springer (2008)
2. David Aucsmith, D. Aucsmith-Information Hiding. Second International Workshop, IH'98, Portland, Oregon, USA, April 14-17, 1998, Proceedings - Springer-Verlag
3. Hsiang-Cheh Huang-Information Hiding and Applications -Springer (2009)
4. Ira S. Moskowitz-Information Hiding_ 4th International Workshop, IH 2001, Pittsburgh, PA, USA, April 25-27, 2001. Proceedings -Springer (2001)
5. Makris G., Antoniou I., 2015, Steganography Based on Chaotic Torus Automorphisms, International Journal of Scientific and Innovative Mathematical Research (IJSIM), Volume 3, Issue 9, September 2015, pp 49-59, ISSN 2347-307X (Print) & ISSN 2347-3142 (Online).
6. Peter Wayner-Disappearing Cryptography, Third Edition_ Information Hiding_ Steganography & Watermarking-Morgan Kaufmann (2008)
7. Teddy Furon, François Cayre, Gwenaël DoërrG, Patrick Bas-Information Hiding_ 9th International Workshop, IH 2007, Saint Malo, France, Jun

Ελληνική

1. Επιτήδειος Γ. (2001). Απλά μαθήματα κρυπτογραφικής και στεγανογραφικής πολιτικής για τον μέσο πολίτη
2. Μακρής Γ., Δαλαμήτρα Ε., Αντωνίου Ι., 2013, «Στεγανογραφία με Χάος», 30ο Πανελλήνιο Συνέδριο Μαθηματικής Παιδείας. σελ 617-626

Υπερσύνδεσμοι (Link), τελευταία προσπέλαση 02/07/2018

1. https://el.wikipedia.org/wiki/%CE%A3%CF%84%CE%B5%CE%B3%CE%B1%CE%BD%CE%BF%CE%B3%CF%81%CE%B1%CF%86%CE%AF%CE%B1
2. https://el.wiktionary.org/wiki/%CE%B3%CF%81%CE%B1%CF%86%CE%AE
3. https://eclass.teicrete.gr/modules/document/file.php/TP122/01.%CE%94%CE%B9%CE%B1%CF%86%CE%AC%CE%BD%CE%B5%CE%B9%CE%B5%CF%82%20%CE%94%CE%B9%CE%B1%CE%BB%CE%AD%CE%BE%CE%B5%CF%89%CE%BD/IS-03-Steganography.pdf
4. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.551.4130&rep=rep1&type=pdf
5. http://isis.poly.edu/~steganography/pubs/spie03.pdf
6. http://www.mediateam.oulu.fi/publications/pdf/618.pdf
7. https://el.wikipedia.org/wiki/%CE%A8%CE%B7%CF%86%CE%B9%CE%B1%CE%BA%CF%8C_%CF%85%CE%B4%CE%B1%CF%84%CF%8C%CF%83%CE%B7%CE%BC%CE%BF#cite_note-A.Z.Tirkel,_G.A._Rankin_p.666-673-3
8. http://cosynet.auth.gr/sites/default/files/Thesis/MAKRIS%20Cryptography%20with%20Chaos%20%CE%95%CE%9B.pdf

9. https://el.wikipedia.org/wiki/Enigma_(%CF%83%CF%85%CF%83%CE%BA%CE%B5%CF%85%CE%AE)
10. https://el.wikipedia.org/wiki/%CE%91%CE%BA%CE%BF%CF%85%CF%83%CF%84%CF%8C_%CF%86%CE%AC%CF%83%CE%BC%CE%B1

Εικόνες

https://www.google.gr/search?rlz=1C1FWBB_enGR666GR667&biw=1366&bih=586&tbm=isch&sa=1&ei=BQY9W72QK8avsAHJg4ywBQ&q=%CE%BC%CE%BF%CE%BD%CE%B1+%CE%BB%CE%B9%CE%B6%CE%B1+&oq=%CE%BC%CE%BF%CE%BD%CE%B1+%CE%BB%CE%B9%CE%B6%CE%B1+&gs_l=img.3..0i67k1j0l7j0i67k1j0.27706.27706.0.28045.1.1.0.0.0.0.195.195.0j1.1.0....0...1c.1.64.img..0.1.194....0.M3rEJTOsQq8#imgrc=UM0lTNt0ab_GSM
http://kryptografies.blogspot.com/2014/01/blog-post_12.html
http://humorakitrello.blogspot.com/2012/09/blog-post_5003.html
https://www.popuppaperstories.com/index.php?act=viewProd&productId=103

Παράρτημα: Κώδικας της εφαρμογής

Κώδικας της Γραφικής Διεπαφής (Gui)

```
function varargout = main(varargin)
% ***************************************
% Πρόγραμμα: Steganography_AM
% Φοιτήτρια: Αναστασία Μακρή
% A.M.:      2113067
% Πανεπιστήμιο Θεσσαλίας
% Τμήμα Πληροφορικής
% ---------------------------------------
% GUI Εφαρμογής
% ***************************************
% MAIN MATLAB code for main.fig
%      MAIN, by itself, creates a new MAIN or raises the existing
%      singleton*.
%
%      H = MAIN returns the handle to a new MAIN or the handle to
%      the existing singleton*.
%
%      MAIN('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in MAIN.M with the given input
arguments.
%
%      MAIN('Property','Value',...) creates a new MAIN or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before main_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to main_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help main

% Last Modified by GUIDE v2.5 06-Jul-2018 19:21:38

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @main_OpeningFcn, ...
                   'gui_OutputFcn',  @main_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before main is made visible.
function main_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to main (see VARARGIN)

% Choose default command line output for main
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% Εκκίνηση του gui στο κέντρο της οθόνης
movegui(gcf,'center')
% UIWAIT makes main wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = main_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listbox1
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
listbox1


% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on selection change in listbox2.
```

```
function listbox2_Callback(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listbox2
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
listbox2


% --- Executes during object creation, after setting all properties.
function listbox2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Το listbox1 περιέχει την επιλογή του χρήστη για ενσωμάτωση
% στην μεταβλητή choice παίρνουμε την επιλογή του χρήστη
% Οι τιμές που μπορεί να έχει η choice είναι 1,2,3,4
choice=get(handles.listbox1,'Value');
switch choice
    % Περίπτωση Ενσωμάτωνσης Κειμένου σε Εικόνα
    % Το κείμενο πρέπει να είναι σε .txt αρχείο
    % Η εικόνα οποιασδήποτε μορφής
    % το στεγανόγραμμα που παράγεται είναι
    % εικόνα .bmp
    case 1
        % Ανοιγμα του αρχείου εικόνας (cover)
        % Σε περίπτωση που ο χρήστης πατήσει cancel επιστρέφει
        [FileName,PathName] =
uigetfile({'*.jpg';'*.png';'*.gif';'*.bmp'},'Select "Canvas Image" to
Hide Message.');
        if PathName==0, return; end
        img = imread( strcat(PathName,FileName) );

        % Ανοιγμα του αρχείου κειμένου (message)
        % Σε περίπτωση που ο χρήστης πατήσει cancel επιστρέφει
        [FileName,PathName] = uigetfile('*.txt','Select TEXT
MESSAGE.');
        if PathName==0, return; end
        testmsg = fopen( strcat(PathName,FileName) );
        [msg] = fscanf(testmsg,'%c');

        % Διάβασμα από τον χρήστη του κωδικού κρυπτογράφησης
        % μέσω inputdlg. Σε περίπτωση που δεν δώσει σωστό κωδικό
```

```
        % επιστρέφουμε στο κ. πρόγραμμα
        prompt = {'Give the Cryptography key [1-255]'};
        title = 'Cryptography Key';
        definput = {'67'};
        answer = inputdlg(prompt,title,[1 40],definput);
        if size(answer)==0
             errordlg('Πατήσατε ακύρωση της διαδικασίας !!!');
             return;
        end
        enc_key=str2num(answer{1});
        if enc_key < 1 || enc_key > 255
            errordlg('Το κλειδί πρέπει να είναι στο διάστημα [1-255]
!!!');
            return
        end
        enc_key = uint8(enc_key);
        output = stegAMcode(img,msg,enc_key);
        if size(output)==0
            errordlg('Το μήνυμα δεν χωρά στην εικόνα !!!');
            return;
        end
        [FileName, PathName] = uiputfile('*.bmp', 'Save As');
        if PathName==0, return; end    % Επιστρφοή στο πρόγραμμα
        Name = fullfile(PathName,FileName);  % Αντιστροφή των
παραμέτρων
        if exist(Name,'file')==1
            choice = questdlg(strcat('The filename',Name,' exists
Replace it?'), ...
            'Replacing File', ...
            'Yes','No');
            % Handle response
            switch choice
            case 'Yes'
                imwrite(output,Name);
            end
            return;
        end
        imwrite(output,Name);
        guidata(hObject, handles);

    % Περίπτωση Ενσωμάτωνσης Εικόνα σε Εικόνα
    % Οι εικόνες πρέπει να είναι οποιασδήποτε μορφής
    % το στεγανόγραμμα που παράγεται είναι
    % εικόνα .bmp
    case 2
        % Ανοιγμα του αρχείου εικόνας (cover)
        % Σε περίπτωση που ο χρήστης πατήσει cancel επιστρέφει
        [FileName,PathName] =
uigetfile({'*.jpg';'*.png';'*.gif';'*.bmp'},'Select "Canvas Image" to
Hide Message.');
        if PathName==0, return; end
        img = imread( strcat(PathName,FileName) );

        % Ανοιγμα του αρχείου Εικόνας (message)
        % Σε περίπτωση που ο χρήστης πατήσει cancel επιστρέφει
        [FileName,PathName] =
uigetfile({'*.jpg';'*.png';'*.gif';'*.bmp'},'Select "Image Message"
to Hide ');
        if PathName==0, return; end
        msg = imread( strcat(PathName,FileName) );
```

```
        % Διάβασμα από τον χρήστη του κωδικού κρυπτογράφησης
        % μέσω inputdlg. Σε περίπτωση που δεν δώσει σωστό κωδικό
        % επιστρέφουμε στο κ. πρόγραμμα
        prompt = {'Give the Cryptography key [1-255]'};
        title = 'Cryptography Key';
        definput = {'67'};
        answer = inputdlg(prompt,title,[1 40],definput);
        if size(answer)==0
             errordlg('Πατήσατε ακύρωση της διαδικασίας !!!');
             return;
        end
        enc_key=str2num(answer{1});
        if enc_key < 1 || enc_key > 255
            errordlg('Το κλειδί πρέπει να είναι στο διάστημα [1-255]
!!!');
            return
        end
        enc_key = uint8(enc_key);
        output = stegAMcode(img,msg,enc_key);
        if size(output)==0
            errordlg('Το μήνυμα δεν χωρά στην εικόνα !!!');
            return;
        end
        [FileName, PathName] = uiputfile('*.bmp', 'Save As');
        if PathName==0, return; end    % Επιστρφοή στο πρόγραμμα
        Name = fullfile(PathName,FileName);  % Αντιστροφή των
παραμέτρων
        if exist(Name,'file')==1
            choice = questdlg(strcat('The filename',Name,' exists
Replace it?'), ...
            'Replacing File', ...
            'Yes','No');
            % Handle response
            switch choice
            case 'Yes'
                imwrite(output,Name);
            end
            return;
        end
        imwrite(output,Name);
        guidata(hObject, handles);

    % Περίπτωση Ενσωμάτωνσης Κειμένου σε Εικόνα με χρήση RNG
    % Το κείμενο πρέπει να είναι σε .txt αρχείο
    % Η εικόνα οποιασδήποτε μορφής
    % το στεγανόγραμμα που παράγεται είναι
    % εικόνα .bmp
    case 3
        % Ανοιγμα του αρχείου εικόνας (cover)
        % Σε περίπτωση που ο χρήστης πατήσει cancel επιστρέφει
        [FileName,PathName] =
uigetfile({'*.jpg';'*.png';'*.gif';'*.bmp'},'Select "Canvas Image" to
Hide Message.');
        if PathName==0, return; end
        img = imread( strcat(PathName,FileName) );

        % Ανοιγμα του αρχείου κειμένου (message)
        % Σε περίπτωση που ο χρήστης πατήσει cancel επιστρέφει
        [FileName,PathName] = uigetfile('*.txt','Select TEXT
MESSAGE.');
        if PathName==0, return; end
```

```
        testmsg = fopen( strcat(PathName,FileName) );
        [msg] = fscanf(testmsg,'%c');

        % Διάβασμα από τον χρήστη του κωδικού κρυπτογράφησης
        % μέσω inputdlg. Σε περίπτωση που δεν δώσει σωστό κωδικό
        % επιστρέφουμε στο κ. πρόγραμμα
        prompt = {'Give the Cryptography key [1-255]'};
        title = 'Cryptography Key';
        definput = {'67'};
        answer = inputdlg(prompt,title,[1 40],definput);
        if size(answer)==0
             errordlg('Πατήσατε ακύρωση της διαδικασίας !!!');
             return;
        end
        enc_key=str2num(answer{1});
        if enc_key < 1 || enc_key > 255
            errordlg('Το κλειδί πρέπει να είναι στο διάστημα [1-255]
!!!');
            return
        end
        enc_key = uint8(enc_key);

        % Διάβασμα από τον χρήστη του σπόρου (seed) για την γεν. τυχ.
αρι
        % μέσω inputdlg. Σε περίπτωση που δεν δώσει σωστό seed
        % επιστρέφουμε στο κ. πρόγραμμα
        prompt = {'Give the seed [1-100]'};
        title = 'RNG seed key';
        definput = {'67'};
        answer = inputdlg(prompt,title,[1 40],definput);
        if size(answer)==0
             errordlg('Πατήσατε ακύρωση της διαδικασίας !!!');
             return;
        end
        randSeed=str2num(answer{1});
        if randSeed < 1 || randSeed > 100
            errordlg('Το seed πρέπει να είναι στο διάστημα [1-100]
!!!');
            return
        end
        randSeed = uint8(randSeed);

        output = stegAMcodeRand(img,msg,enc_key,randSeed);
        if size(output)==0
            errordlg('Το μήνυμα δεν χωρά στην εικόνα !!!');
            return;
        end
        [FileName, PathName] = uiputfile('*.bmp', 'Save As');
        if PathName==0, return; end    % Επιστρφοή στο πρόγραμμα
        Name = fullfile(PathName,FileName);  % Αντιστροφή των
παραμέτρων
        if exist(Name,'file')==1
            choice = questdlg(strcat('The filename',Name,' exists
Replace it?'), ...
            'Replacing File', ...
            'Yes','No');
            % Handle response
            switch choice
            case 'Yes'
                imwrite(output,Name);
            end
```

```
            return;
        end
        imwrite(output,Name);
        guidata(hObject, handles);

    % Περίπτωση Ενσωμάτωνσης Εικόνα σε Εικόνα με χρήση RNG
    % Οι εικόνες πρέπει να είναι οποιασδήποτε μορφής
    % το στεγανόγραμμα που παράγεται είναι
    % εικόνα .bmp
    case 4
        % Ανοιγμα του αρχείου εικόνας (cover)
        % Σε περίπτωση που ο χρήστης πατήσει cancel επιστρέφει
        [FileName,PathName] =
uigetfile({'*.jpg';'*.png';'*.gif';'*.bmp'},'Select "Canvas Image" to
Hide Message.');
        if PathName==0, return; end
        img = imread( strcat(PathName,FileName) );

        % Ανοιγμα του αρχείου Εικόνας (message)
        % Σε περίπτωση που ο χρήστης πατήσει cancel επιστρέφει
        [FileName,PathName] =
uigetfile({'*.jpg';'*.png';'*.gif';'*.bmp'},'Select "Image Message"
to Hide ');
        if PathName==0, return; end
        msg = imread( strcat(PathName,FileName) );

        % Διάβασμα από τον χρήστη του κωδικού κρυπτογράφησης
        % μέσω inputdlg. Σε περίπτωση που δεν δώσει σωστό κωδικό
        % επιστρέφουμε στο κ. πρόγραμμα
        prompt = {'Give the Cryptography key [1-255]'};
        title = 'Cryptography Key';
        definput = {'67'};
        answer = inputdlg(prompt,title,[1 40],definput);
        if size(answer)==0
             errordlg('Πατήσατε ακύρωση της διαδικασίας !!!');
             return;
        end
        enc_key=str2num(answer{1});
        if enc_key < 1 || enc_key > 255
            errordlg('Το κλειδί πρέπει να είναι στο διάστημα [1-255]
!!!');
            return
        end
        enc_key = uint8(enc_key);

        % Διάβασμα από τον χρήστη του σπόρου (seed) για την γεν. τυχ.
αρι
        % μέσω inputdlg. Σε περίπτωση που δεν δώσει σωστό seed
        % επιστρέφουμε στο κ. πρόγραμμα
        prompt = {'Give the seed [1-100]'};
        title = 'RNG seed key';
        definput = {'67'};
        answer = inputdlg(prompt,title,[1 40],definput);
        if size(answer)==0
             errordlg('Πατήσατε ακύρωση της διαδικασίας !!!');
             return;
        end
        randSeed=str2num(answer{1});
        if randSeed < 1 || randSeed > 100
            errordlg('Το seed πρέπει να είναι στο διάστημα [1-100]
!!!');
```

```
            return
        end
        randSeed = uint8(randSeed);

        output = stegAMcodeRand(img,msg,enc_key,randSeed);
        if size(output)==0
            errordlg('Το μήνυμα δεν χωρά στην εικόνα !!!');
            return;
        end
        [FileName, PathName] = uiputfile('*.bmp', 'Save As');
        if PathName==0, return; end    % Επιστρφοή στο πρόγραμμα
        Name = fullfile(PathName,FileName);  % Αντιστροφή των
παραμέτρων
        if exist(Name,'file')==1
            choice = questdlg(strcat('The filename',Name,' exists
Replace it?'), ...
            'Replacing File', ...
            'Yes','No');
            % Handle response
            switch choice
            case 'Yes'
                imwrite(output,Name);
            end
            return;
        end
        imwrite(output,Name);
        guidata(hObject, handles);


end;

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
choice=get(handles.listbox2,'Value');
switch choice
    % Περίπτωση Εξαγωγής αντικειμένου από Εικόνα
    % Η εικόνα (στεγανόγραμμα) πρέπει να είναι τύπου .bmp
    case 1
        [FileName,PathName] =
uigetfile({'*.bmp';'*.jpg';'*.png';'*.gif'},'Select Image to take
Message.');
        if PathName==0, return; end
        img = imread( strcat(PathName,FileName) );

        % Διάβασμα από τον χρήστη του κωδικού κρυπτογράφησης
        % μέσω inputdlg. Σε περίπτωση που δεν δώσει σωστό κωδικό
        % επιστρέφουμε στο κ. πρόγραμμα
        prompt = {'Give the Cryptography key [1-255]'};
        title = 'Cryptography Key';
        definput = {'67'};
        answer = inputdlg(prompt,title,[1 40],definput);
        if size(answer)==0
             errordlg('Πατήσατε ακύρωση της διαδικασίας !!!');
             return;
        end
        enc_key=str2num(answer{1});
        if enc_key < 1 || enc_key > 255
```

```
            errordlg('Το κλειδί πρέπει να είναι στο διάστημα [1-255]
!!!');
            return
        end
        enc_key = uint8(enc_key);
        output = stegAMdecode(img,enc_key);
        % τι είναι το εξαγώμενο κείμενο ή εικόνα
        msgtest = ischar(output);

        [FileName, PathName] = uiputfile('*.*', 'Save As');
        if PathName==0, return; end    % Επιστρφοή στο πρόγραμμα
        Name = fullfile(PathName,FileName);  % Αντιστροφή των
παραμέτρων
        if exist(Name,'file')==1
            choice = questdlg(strcat('The filename',Name,' exists
Replace it?'), ...
            'Replacing File', ...
            'Yes','No');
            % Handle response
            switch choice
            case 'Yes'
                if msgtest == 1
                    % Περίπτωση Κειμένου για μήνυμα
                    fid = fopen(strcat(Name,'.txt'),'w');
                    fwrite(fid,output,'char');
                    fclose(fid);
                else
                    % Περίπτωση Εικονα για μήνυμα
                    imwrite(output,Name);
                end
             end
            return;
        end
        if msgtest == 1
            % Περίπτωση Κειμένου για μήνυμα
            fid = fopen(strcat(Name,'.txt'),'w');
            fwrite(fid,output,'char');
            fclose(fid);
        else
            % Περίπτωση Εικονα για μήνυμα
            imwrite(output,Name);
        end
        guidata(hObject, handles);

    % Περίπτωση Εξαγωγής αντικειμένου από Εικόνα με χρήση RNG
    % Η εικόνα (στεγανόγραμμα) πρέπει να είναι τύπου .bmp
    case 2
        [FileName,PathName] =
uigetfile({'*.bmp';'*.jpg';'*.png';'*.gif'},'Select Image to take
Message.');
        if PathName==0, return; end
        img = imread( strcat(PathName,FileName) );

        % Διάβασμα από τον χρήστη του κωδικού κρυπτογράφησης
        % μέσω inputdlg. Σε περίπτωση που δεν δώσει σωστό κωδικό
        % επιστρέφουμε στο κ. πρόγραμμα
        prompt = {'Give the Cryptography key [1-255]'};
        title = 'Cryptography Key';
        definput = {'67'};
        answer = inputdlg(prompt,title,[1 40],definput);
        if size(answer)==0
```

```
            errordlg('Πατήσατε ακύρωση της διαδικασίας !!!');
            return;
        end
        enc_key=str2num(answer{1});
        if enc_key < 1 || enc_key > 255
            errordlg('Το κλειδί πρέπει να είναι στο διάστημα [1-255]
!!!');
            return
        end
        enc_key = uint8(enc_key);

        % Διάβασμα από τον χρήστη του σπόρου (seed) για την γεν. τυχ.
αρι
        % μέσω inputdlg. Σε περίπτωση που δεν δώσει σωστό seed
        % επιστρέφουμε στο κ. πρόγραμμα
        prompt = {'Give the seed [1-100]'};
        title = 'RNG seed key';
        definput = {'67'};
        answer = inputdlg(prompt,title,[1 40],definput);
        if size(answer)==0
            errordlg('Πατήσατε ακύρωση της διαδικασίας !!!');
            return;
        end
        randSeed=str2num(answer{1});

        if randSeed < 1 || randSeed > 100
            errordlg('Το seed πρέπει να είναι στο διάστημα [1-100]
!!!');
            return
        end
        randSeed = uint8(randSeed);

        output = stegAMdecodeRand(img,enc_key,randSeed);
        % τι είναι το εξαγώμενο κείμενο ή εικόνα
        msgtest = ischar(output);

        [FileName, PathName] = uiputfile('*.*', 'Save As');
        if PathName==0, return; end    % Επιστρφοή στο πρόγραμμα
        Name = fullfile(PathName,FileName);  % Αντιστροφή των
παραμέτρων
        if exist(Name,'file')==1
            choice = questdlg(strcat('The filename',Name,' exists
Replace it?'), ...
            'Replacing File', ...
            'Yes','No');
            % Handle response
            switch choice
            case 'Yes'
                if msgtest == 1
                    % Περίπτωση Κειμένου για μήνυμα
                    fid = fopen(strcat(Name,'.txt'),'w');
                    fwrite(fid,output,'char');
                    fclose(fid);
                else
                    % Περίπτωση Εικονα για μήνυμα
                    imwrite(output,Name);
                end
             end
            return;
        end
        if msgtest == 1
```

```
            % Περίπτωση Κειμένου για μήνυμα
            fid = fopen(strcat(Name,'.txt'),'w');
            fwrite(fid,output,'char');
            fclose(fid);
        else
            % Περίπτωση Εικονα για μήνυμα
            imwrite(output,Name);
        end
        guidata(hObject, handles);

end;


% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    % ANοιγμα του αρχείου Βοήθειας
    winopen('SteganographyAM_help.pdf');
```

Κώδικας Συναρτήσεων

```
function [image_with_message] = stegAMcode(img,msg,enc_key)
% ****************************************
% Πρόγραμμα: Steganography_AM
% Φοιτήτρια: Αναστασία Μακρή
% A.M.:      2113067
% Πανεπιστήμιο Θεσσαλίας
% Τμήμα Πληροφορικής
% ----------------------------------------
% Συνάρτηση stegAMcode
% ****************************************
% stegAMcode:  Η συνάρτηση αυτή κρύβει ένα μήνυμα (msg) μέσα σε
%               μία εικόνα msg με δεδομένο κάποιο κλειδί
κρυπτογράφησης
%               enc_key
% Είσοδοι:
% - img: Αρχική εικόνα στην οποία θα κρυφτεί το μήνυμα
% - msg: Μήνυμα (είναι κείμενο ή εικόνα η οποία μετατρέπεται σε gray
% - enc_key: Κλειδί κρυπτογράφησης
%
% Έξοδοι:
% - image_with_message: Το στεγανόγραμμα (εικόνα + μήνυμα)

% Εξέταση έαν το msg είναι κείμενο ή εικόνα
% η msgtype θα πάρει τιμή 1 εάν είναι κείμενο αλλιώς την τιμή 0
msgtype = ischar(msg);

if msgtype == 1     % είναι κείμενο
    msg_temp = double(msg);     % Μετατροπή του ASCII σε ακέραιους
αριθμούς
    msg_dim = num2str(length(msg_temp)); % Μετατροπή σε strings όλων
των αριθμών
    msg_length = length(msg_dim);  % Υπολογισμός του μήκους του
μηνύματος
    % προετοιμασία της κεφαλίδας του μηνύματος
```

```
    z = 0;
    if msg_length < 7
        padtext = 7 - msg_length;
        for z = 1:padtext
            msg_dim = horzcat('0',msg_dim);
        end
        msg_head = horzcat('t',msg_dim);
        % Τοποθέτηση της κεφαλίδας μπροστά από το μήνυμα
        msg_temp_head = horzcat(msg_head,msg_temp);
    end

else
    % Πρόκειται για εικόνα που θέλουμε να κρύψουμε
    msg = im2uint8(msg);        % μετατροπή της εικόνας σε ακέραια
αναπαράσταση για επεξεργασία

    msg_temp = rgb2gray(msg);   % Μετατροπή της εικόνας σε grayscele
.

    % Καθορισμός του μήκους της ΕΙκόνας για κωδικοποίηση στην
κεφαλίδα
    [hideM1,hideN1] = size(msg_temp);
    hideM = num2str(hideM1);
    hideN = num2str(hideN1);
    dimM = length(hideM);
    dimN = length(hideN);
    padM = 0; padN = 0;
    z = 0;

    if dimM < 4
        padM = 4 - dimM;
        for z = 1:padM
            % Μηδενισμός εάν είναι μικρότερο από 4
            hideM = horzcat('0',hideM);
        end
    end
    z = 0;

    if dimN < 4
        padN = 4 - dimN;
        for z = 1:padN
            % Μηδενισμός εάν είναι μικρότερο από 4
            hideN = horzcat('0',hideN);
        end
    end
    msg_head = horzcat(hideM,hideN);
    msg_temp_head = msg_head;

    y = 0;  k = hideM1;
    for y = 1:k
        % Προσθήκη της κεφαλίδας στην αρχή του μηνύματος που
πρόκειται να κωδικοποιηθεί
        msg_temp_head = horzcat(msg_temp_head,msg_temp(y,:));
    end

end

%% Βήμα 2: Έλεγχος του χώρου αποθήκευσης/τοποθέτησης
% Σε περίπτωση που δεν είναι δυνατόν να χωρέσει το μήνυμα
% μέσα στην εικόνα επιστρέφεται η κενή εικόνα
tot_hiding_pix = max(cumprod(size(img)));
```

```
tot_data = max(cumprod(size(msg_temp_head)));
if tot_hiding_pix <= tot_data
    image_with_message=[];
    return;
end

%% Βήμα 3: Κρυπογράφηση χρησιμοποιώντας την συνάρτηση XOR

msg_enc = bitxor(uint8(msg_temp_head),uint8(enc_key));
msg_enc_set = dec2bin(msg_enc, 8);

%% Βήμα 4: Προετοιμία του Cover
img_prep = im2uint8(img);


%% Βήμα 5: Τοποθέτηση / κρύψιμο Δεδομένων
% Τα δεδομένα τοποθετούνται σειριακά κατά στήλες
% 3 pixels για κάθε byte πληροφορίας
rm = 1; gm = 1; bm = 1;     % Αρχικοποίηση μετρητών
rn = 1; gn = 1; bn = 1;

[maxM,maxN] = size(img_prep);
z = 0;

% RUN_TIME : Περιέχει το πλήθος των λέξεων που πρόκειται να κρυφτούν
run_time = length(msg_enc_set);

for z = 1:run_time;
    temp_code = msg_enc_set(z,:);
    % Bit 1: Red
    if str2double(temp_code(1)) == 0
        img_prep(rm,rn,1) = bitand(img_prep(rm,rn,1),uint8(254));
    else
        img_prep(rm,rn,1) = bitor(img_prep(rm,rn,1),uint8(1));
    end

    rm = rm + 1;
    % Έλεγχος εάν έχουμε φτάσει στο τέλος της εικόνας όποτε ξαναπάμε
από
    % την αρχή
    % Οι παρακάτω εντολές θα μπορούσαν να αντικατασταθούν και με
πράξεις
    % mod maxM
    if rm > maxM
        rn = rn + 1;
        rm = 1;
    end
    % Bit 2: Green
    if str2double(temp_code(2)) == 0
        img_prep(gm,gn,2) = bitand(img_prep(gm,gn,2),uint8(254));
    else
        img_prep(gm,gn,2) = bitor(img_prep(gm,gn,2),uint8(1));
    end

    gm = gm + 1;
    if gm > maxM
        gn = gn + 1;
        gm = 1;
    end
    % Bit 3: Blue
    if str2double(temp_code(3)) == 0
```

```
        img_prep(bm,bn,3) = bitand(img_prep(bm,bn,3),uint8(254));
    else
        img_prep(bm,bn,3) = bitor(img_prep(bm,bn,3),uint8(1));
    end

    bm = bm + 1;
    if bm > maxM
        bn = bn + 1;
        bm = 1;
    end
    % Bit 4: Blue
    if str2double(temp_code(4)) == 0
        img_prep(bm,bn,3) = bitand(img_prep(bm,bn,3),uint8(254));
    else
        img_prep(bm,bn,3) = bitor(img_prep(bm,bn,3),uint8(1));
    end

    bm = bm + 1;
    if bm > maxM
        bn = bn + 1;
        bm = 1;
    end
    % Bit 5: Green
    if str2double(temp_code(5)) == 0
        img_prep(gm,gn,2) = bitand(img_prep(gm,gn,2),uint8(254));
    else
        img_prep(gm,gn,2) = bitor(img_prep(gm,gn,2),uint8(1));
    end

    gm = gm + 1;
    if gm > maxM
        gn = gn + 1;
        gm = 1;
    end
    % Bit 6: Red
    if str2double(temp_code(6)) == 0
        img_prep(rm,rn,1) = bitand(img_prep(rm,rn,1),uint8(254));
    else
        img_prep(rm,rn,1) = bitor(img_prep(rm,rn,1),uint8(1));
    end
    rm = rm + 1;
    if rm > maxM
        rn = rn + 1;
        rm = 1;
    end
    % Bit 7: Red
    if str2double(temp_code(7)) == 0
        img_prep(rm,rn,1) = bitand(img_prep(rm,rn,1),uint8(254));
    else
        img_prep(rm,rn,1) = bitor(img_prep(rm,rn,1),uint8(1));
    end
    rm = rm + 1;
    if rm > maxM
        rn = rn + 1;
        rm = 1;
    end
    % Bit 8: Green
    if str2double(temp_code(8)) == 0
        img_prep(gm,gn,2) = bitand(img_prep(gm,gn,2),uint8(254));
    else
        img_prep(gm,gn,2) = bitor(img_prep(gm,gn,2),uint8(1));
```

```
    end

    gm = gm + 1;
    if gm > maxM
        gn = gn + 1;
        gm = 1;
    end

end

%% Τελική έξοδος της συνάρτηση το στεγανόγραμμα
%  Το οποίο είναι η αρχική εικόνα μαζί με το μήνυμα
image_with_message = img_prep;
%
end


function [image_with_message] =
stegAMcodeRand(img,msg,enc_key,randSeed)
% ****************************************
% Πρόγραμμα: Steganography_AM
% Φοιτήτρια: Αναστασία Μακρή
% Α.Μ.:      2113067
% Πανεπιστήμιο Θεσσαλίας
% Τμήμα Πληροφορικής
% ----------------------------------------
% Συνάρτηση stegAMcodeRand
% ****************************************
% stegAMcodeRand:   Η συνάρτηση αυτή κρύβει ένα μήνυμα (msg) μέσα σε
%                   μία εικόνα msg με δεδομένο κάποιο κλειδί
κρυπτογράφησης
%                   enc_key και ένα δεδομένο seed (randSeed)
% Είσοδοι:
% - img: Αρχική εικόνα στην οποία θα κρυφτεί το μήνυμα
% - msg: Μήνυμα (είναι κείμενο ή εικόνα η οποία μετατρέπεται σε gray
% - enc_key: Κλειδί κρυπτογράφησης
% - randSeed: seed της γεννήτριας τυχαίων αριθμών
%
% Έξοδοι:
% - image_with_message: Το στεγανόγραμμα (εικόνα + μήνυμα)

% Εξέταση έαν το msg είναι κείμενο ή εικόνα
%   η msgtype θα πάρει τιμή 1 εάν είναι κείμενο αλλιώς την τιμή 0

msgtype = ischar(msg);


if msgtype == 1     % Message = TEXT
    msg_temp = double(msg);     % Converts from ASCII to Integer
Values.
    msg_dim = num2str(length(msg_temp));
    msg_length = length(msg_dim);
    z = 0;
    if msg_length < 7
        padtext = 7 - msg_length;
        for z = 1:padtext
            msg_dim = horzcat('0',msg_dim);
        end
        msg_head = horzcat('t',msg_dim);
        % Προσθήκη της κεφαλίδας στην αρχή του μηνύματος που
πρόκειται να κωδικοποιηθεί
```

```
        msg_temp_head = horzcat(msg_head,msg_temp);
    end

else
    % Εικόνα ως Μήνυμα
    msg = im2uint8(msg);        % μετατροπή της εικόνας σε ακέραια
αναπαράσταση για επεξεργασία

    msg_temp = rgb2gray(msg);   % Μετατροπή της εικόνας σε grayscale
.


    % Καθορισμός του μήκους της ΕΙκόνας για κωδικοποίηση στην
κεφαλίδα
    [hideM1,hideN1] = size(msg_temp);
    hideM = num2str(hideM1);
    hideN = num2str(hideN1);
    dimM = length(hideM);
    dimN = length(hideN);
    padM = 0; padN = 0;
    z = 0;

    if dimM < 4
        padM = 4 - dimM;
        for z = 1:padM
            % Μηδενισμός εάν είναι μικρότερο από 4
            hideM = horzcat('0',hideM);
        end
    end
    z = 0;

    if dimN < 4
        padN = 4 - dimN;
        for z = 1:padN
            % Μηδενισμός εάν είναι μικρότερο από 4
            hideN = horzcat('0',hideN);
        end
    end
    msg_head = horzcat(hideM,hideN);
    msg_temp_head = msg_head;

    y = 0;  k = hideM1;
    for y = 1:k
        % Προσθήκη της κεφαλίδας στην αρχή του μηνύματος που
πρόκειται να κωδικοποιηθεί
        msg_temp_head = horzcat(msg_temp_head,msg_temp(y,:));
    end

end

%% Βήμα 2: Έλεγχος του χώρου αποθήκευσης/τοποθέτησης
% Σε περίπτωση που δεν είναι δυνατόν να χωρέσει το μήνυμα
% μέσα στην εικόνα επιστρέφεται η κενή εικόνα
tot_hiding_pix = max(cumprod(size(img)));
tot_data = max(cumprod(size(msg_temp_head)));
if tot_hiding_pix <= tot_data
    image_with_message=[];
    return;
end

%% Βήμα 3: Κρυπογράφηση χρησιμοποιώντας την συνάρτηση XOR
```

```
msg_enc = bitxor(uint8(msg_temp_head),uint8(enc_key));
msg_enc_set = dec2bin(msg_enc, 8);

%% Βήμα 4: Προετοιμασία του Cover  and αρχικοποίηση των Τυχαίων
αριθμών
% "Canvas" Image
img_prep = im2uint8(img);

% Random Permutation Set
rng(randSeed); % Αρχικοποίηση της γεννήτριας τυχαίων αριθμών
[canM,canN,chan]=size(img);
canvas_Dim = canM * canN;
% Εξασφάλιση ότι οι διαστάσεις είναι πολλαπλάσιες του 3

canTest = rem(canvas_Dim,3);
if canTest ~= 0
    canvas_Dim = canvas_Dim - canTest;
end

randSet = randperm(canvas_Dim);     % Επιλογή τυχαίου Pixel
randGroup = reshape(randSet,[],3);  % Ομαδοποίηση των Pixels ανά 3


%% Βήμα 5: Τοποθέτηση / κρύψιμο Δεδομένων

% RUN_TIME : Περιέχει το πλήθος των λέξεων που πρόκειται να κρυφτούν
run_time = length(msg_enc_set);

for z = 1:run_time;
    temp_code = msg_enc_set(z,:);
    temp_loc = randGroup(z,:);

    % Ενσωμάτωνση των 3 πρώτων bits του RGB μοντέλου

    [row1,col1] = ind2sub([canM,canN],temp_loc(1));

    % Bit 1: Red
    if str2double(temp_code(1)) == 0
        img_prep(row1,col1,1) =
bitand(img_prep(row1,col1,1),uint8(254));
    else
        img_prep(row1,col1,1) =
bitor(img_prep(row1,col1,1),uint8(1));
    end

    % Bit 2: Green
    if str2double(temp_code(2)) == 0
        img_prep(row1,col1,2) =
bitand(img_prep(row1,col1,2),uint8(254));
    else
        img_prep(row1,col1,2) =
bitor(img_prep(row1,col1,2),uint8(1));
    end

    % Bit 3: Blue
    if str2double(temp_code(3)) == 0
        img_prep(row1,col1,3) =
bitand(img_prep(row1,col1,3),uint8(254));
    else
```

```
        img_prep(row1,col1,3) =
bitor(img_prep(row1,col1,3),uint8(1));
    end

    % Τα επόμενα 3 Bits

    [row2,col2] = ind2sub([canM,canN],temp_loc(2));

    % Bit 4: Blue
    if str2double(temp_code(4)) == 0
        img_prep(row2,col2,3) =
bitand(img_prep(row2,col2,3),uint8(254));
    else
        img_prep(row2,col2,3) =
bitor(img_prep(row2,col2,3),uint8(1));
    end

    % Bit 5: Green
    if str2double(temp_code(5)) == 0
        img_prep(row2,col2,2) =
bitand(img_prep(row2,col2,2),uint8(254));
    else
        img_prep(row2,col2,2) =
bitor(img_prep(row2,col2,2),uint8(1));
    end

    % Bit 6: Red
    if str2double(temp_code(6)) == 0
        img_prep(row2,col2,1) =
bitand(img_prep(row2,col2,1),uint8(254));
    else
        img_prep(row2,col2,1) =
bitor(img_prep(row2,col2,1),uint8(1));
    end

    % Τα επόμενα 2 Bits RG_

    [row3,col3] = ind2sub([canM,canN],temp_loc(3));

    % Bit 7: Red
    if str2double(temp_code(7)) == 0
        img_prep(row3,col3,1) =
bitand(img_prep(row3,col3,1),uint8(254));
    else
        img_prep(row3,col3,1) =
bitor(img_prep(row3,col3,1),uint8(1));
    end

    % Bit 8: Green
    if str2double(temp_code(8)) == 0
        img_prep(row3,col3,2) =
bitand(img_prep(row3,col3,2),uint8(254));
    else
        img_prep(row3,col3,2) =
bitor(img_prep(row3,col3,2),uint8(1));
    end

end

%% Βήμα 6: Τελική έξοδος/ΕΠιστροφή
image_with_message = img_prep;       % Final Encoding Output
```

```
% J = msg_enc_set;  % ENCRYPTION Βήμα TEST OUTPUT
end


function [msg] = stegAMdecode(img,enc_key)
% ****************************************
% Πρόγραμμα: Steganography_AM
% Φοιτήτρια: Αναστασία Μακρή
% A.M.:      2113067
% Πανεπιστήμιο Θεσσαλίας
% Τμήμα Πληροφορικής
% ----------------------------------------
% Συνάρτηση stegAMdecode
% ****************************************
% stegAMdecode: Η συνάρτηση εξάγει ένα κρυμένο μήνυμα
%   από μία εικόνα (Το στεγανόγραμμα (εικόνα + μήνυμα))
% Είσοδοι:
% - img: Το στεγανόγραμμα (εικόνα + μήνυμα)
% - enc_key: Κλειδί κρυπτογράφησης
%
% Έξοδοι:
% - msg: Μήνυμα (είναι κείμενο ή εικόνα σε gray)

%% Βήμα 1a: Ανάκτηση της κεφαλιδα με τα στοιχεία του μηνύματος
rm = 1; gm = 1; bm = 1;     % Αρχικοποίηση μετρητών
rn = 1; gn = 1; bn = 1;

header = [];
[maxM, maxN, chan] = size(img);

for z = 1:8;
    temp = zeros(1,8);
    % Red
    temp(1,1) = mod(img(rm,rn,1),2);
    rm = rm + 1;
    % Έλεγχος εάν έχουμε φτάσει στο τέλος της εικόνας όποτε ξαναπάμε
από
    % την αρχή
    % Οι παρακάτω εντολές θα μπορούσαν να αντικατασταθούν και με
πράξεις
    % mod maxM
    if rm > maxM
        rn = rn + 1;
        rm = 1;
        if rn > maxN
            break
        end
    end
    % Green
    temp(1,2) = mod(img(gm,gn,2),2);
    gm = gm + 1;
    if gm > maxM
        gn = gn + 1;
        gm = 1;
        if gn > maxN
            break
        end
    end
    % Blue
    temp(1,3) = mod(img(bm,bn,3),2);
    bm = bm + 1;
```

```
    if bm > maxM
        bn = bn + 1;
        bm = 1;
    end
    % Blue
    temp(1,4) = mod(img(bm,bn,3),2);
    bm = bm + 1;
    if bm > maxM
        bn = bn + 1;
        bm = 1;
    end
    % Green
    temp(1,5) = mod(img(gm,gn,2),2);
    gm = gm + 1;
    if gm > maxM
        gn = gn + 1;
        gm = 1;
        if gn > maxN
            break
        end
    end
    % Red
    temp(1,6) = mod(img(rm,rn,1),2);
    rm = rm + 1;
    if rm > maxM
        rn = rn + 1;
        rm = 1;
        if rn > maxN
            break
        end
    end
    % Red
    temp(1,7) = mod(img(rm,rn,1),2);
    rm = rm + 1;
    if rm > maxM
        rn = rn + 1;
        rm = 1;
        if rn > maxN
            break
        end
    end
    % Green
    temp(1,8) = mod(img(gm,gn,2),2);
    gm = gm + 1;
    if gm > maxM
        gn = gn + 1;
        gm = 1;
        if gn > maxN
            break
        end
    end
    tempstr = num2str(temp);
    header = vertcat(header,tempstr);
end

%% Βήμα 1b: Ανάλυση της Κεφαλίδας και προσδιορισμός του μεγέθους του
μηνύματος
msg_head_set = bin2dec(header);
temp_head = bitxor(uint8(msg_head_set),uint8(enc_key));

% Case 1: Εάν η κεφαλίδα ξεκινάει με 't' είναι κείμενο το μήνυμα
```

```
% Case 2: Εάν η κεφαλίδα δεν ξεκινάει με 't' είναι εικόνα το μήνυμα
if temp_head(1) == 116
    % CASE 1: Κείμενο
    dim1 = char(temp_head(2:8));
    m = str2double(dim1);
    n = 1;
else
    % CASE 2: Εικόνα
    % Προσιορισμός των διαστάσεων
    tempm = char(temp_head(1:4));
    tempn = char(temp_head(5:8));
    m = str2double(tempm');
    n = str2double(tempn');
end


%% Βήμα 2: Προσδιορισμός των θέσεων που είναι τοποθετημένο το μήνυμα

z = 0;

enc_msg = [];
stopmax = (m * n);

for z = 1:stopmax
    temp = zeros(1,8);
    % Red
    temp(1,1) = mod(img(rm,rn,1),2);
    rm = rm + 1;
    % Έλεγχος εάν έχουμε φτάσει στο τέλος της εικόνας όποτε ξαναπάμε
από
    % την αρχή
    % Οι παρακάτω εντολές θα μπορούσαν να αντικατασταθούν και με
πράξεις
    % mod maxM
    if rm > maxM
        rn = rn + 1;
        rm = 1;
        if rn > maxN
            break
        end
    end
    % Green
    temp(1,2) = mod(img(gm,gn,2),2);
    gm = gm + 1;
    if gm > maxM
        gn = gn + 1;
        gm = 1;
        if gn > maxN
            break
        end
    end
    % Blue
    temp(1,3) = mod(img(bm,bn,3),2);
    bm = bm + 1;
    if bm > maxM
        bn = bn + 1;
        bm = 1;
    end
    % Blue
    temp(1,4) = mod(img(bm,bn,3),2);
    bm = bm + 1;
```

```
    if bm > maxM
        bn = bn + 1;
        bm = 1;
    end
    % Green
    temp(1,5) = mod(img(gm,gn,2),2);
    gm = gm + 1;
    if gm > maxM
        gn = gn + 1;
        gm = 1;
        if gn > maxN
            break
        end
    end
    % Red
    temp(1,6) = mod(img(rm,rn,1),2);
    rm = rm + 1;
    if rm > maxM
        rn = rn + 1;
        rm = 1;
        if rn > maxN
            break
        end
    end
    % Red
    temp(1,7) = mod(img(rm,rn,1),2);
    rm = rm + 1;
    if rm > maxM
        rn = rn + 1;
        rm = 1;
        if rn > maxN
            break
        end
    end
    % Green
    temp(1,8) = mod(img(gm,gn,2),2);
    gm = gm + 1;
    if gm > maxM
        gn = gn + 1;
        gm = 1;
        if gn > maxN
            break
        end
    end
    tempstr = num2str(temp);
    enc_msg = vertcat(enc_msg,tempstr);
end

%% Βήμα 3: Decryption Βήμα
msg_dec_set = bin2dec(enc_msg);
msg_dec = bitxor(uint8(msg_dec_set),uint8(enc_key));


%% Βήμα 4: Προετοιμασία για ανάκτηση μηνύματος
if temp_head(1) == 116
    % CASE 1: Κείμενο
    msg_set = msg_dec;
    msg_out = char(msg_set');
else
    % CASE 2: Εικόνα
    % Προσδιορισμός των διαστάσεων της εικόνας
```

```
    tempm = char(temp_head(1:4));
    tempn = char(temp_head(5:8));
    m = str2double(tempm');
    n = str2double(tempn');

    % Αντιγραφή της κωδικοποιημένης εικόνας
    msg_set = msg_dec;

    count = 1;
    msg_out = uint8(zeros(m,n));
    for y = 1:m
        for x = 1:n
            msg_out(y,x) = msg_set(count);
            count = count + 1;
        end
    end

    msg_out = im2uint8(msg_out);

end


%% Βήμα 5: Τελική έξοδος/ΕΠιστροφή
msg = msg_out;
end




function [msg] = stegAMdecodeRand(img,enc_key,randSeed)
% ****************************************
% Πρόγραμμα: Steganography_AM
% Φοιτήτρια: Αναστασία Μακρή
% A.M.:      2113067
% Πανεπιστήμιο Θεσσαλίας
% Τμήμα Πληροφορικής
% ----------------------------------------
% Συνάρτηση stegAMdecodeRand
% ****************************************
% stegAMdecodeRand: Η συνάρτηση εξάγει ένα κρυμένο μήνυμα
%   από μία εικόνα (Το στεγανόγραμμα (εικόνα + μήνυμα))
% Είσοδοι:
% - img: Το στεγανόγραμμα (εικόνα + μήνυμα)
% - enc_key: Κλειδί κρυπτογράφησης
% - randSeed: seed της γεννήτριας τυχαίων αριθμών
%
% Έξοδοι:
% - msg: Μήνυμα (είναι κείμενο ή εικόνα σε gray)

%% Βήμα 1a: Ανάκτηση της κεφαλιδα με τα στοιχεία του μηνύματος
% Random Permutation Set
[canM,canN,chan]=size(img);
canvas_Dim = canM * canN;
% Εξασφάλιση ότι οι διαστάσεις είναι πολλαπλάσιες του 3
canTest = rem(canvas_Dim,3);
if canTest ~= 0
    canvas_Dim = canvas_Dim - canTest;
end

rng(randSeed); % Αρχικοποίηση της γεννήτριας τυχαίων αριθμών
```

```
randSet = randperm(canvas_Dim);     % Επιλογή τυχαίου Pixel
randGroup = reshape(randSet,[],3);  % Ομαδοποίηση των Pixels ανά 3

% Αρχικοποίηση της κεφαλίδας
header = [];

for z = 1:8;
    temp = zeros(1,8);
    temp_loc = randGroup(z,:);

    % Τα τρία πρώτα bit του header
    % --------------------------------
    [row1,col1] = ind2sub([canM,canN],temp_loc(1));
    % Red
    temp(1,1) = mod(img(row1,col1,1),2);

    % Green
    temp(1,2) = mod(img(row1,col1,2),2);

    % Blue
    temp(1,3) = mod(img(row1,col1,3),2);

    % Τα επόμενα 3 bit του header
    % --------------------------------
    [row2,col2] = ind2sub([canM,canN],temp_loc(2));

    % Blue
    temp(1,4) = mod(img(row2,col2,3),2);

    % Green
    temp(1,5) = mod(img(row2,col2,2),2);

    % Red
    temp(1,6) = mod(img(row2,col2,1),2);

    % Τα τελευταία 2 Bit για το header
    % --------------------------------
    [row3,col3] = ind2sub([canM,canN],temp_loc(3));
    % Red
    temp(1,7) = mod(img(row3,col3,1),2);

    % Green
    temp(1,8) = mod(img(row3,col3,2),2);

    % Μετατροπή σε κείμενο για προσθήκη στην κεφαλίδα
    tempstr = num2str(temp);
    header = vertcat(header,tempstr);
end

%% Βήμα 1b: Ανάλυση της Κεφαλίδας και προσδιορισμός του μεγέθους του
μηνύματος
msg_head_set = bin2dec(header);
temp_head = bitxor(uint8(msg_head_set),uint8(enc_key));

% Case 1: Εάν η κεφαλίδα ξεκινάει με 't' είναι κείμενο το μήνυμα
% Case 2: Εάν η κεφαλίδα δεν ξεκινάει με 't' είναι εικόνα το μήνυμα
if temp_head(1) == 116
    % CASE 1: Κείμενο
    dim1 = char(temp_head(2:8));
    m = str2double(dim1);
    n = 1;
```

```
else
    % CASE 2: Εικόνα
    % Προσιορισμός των διαστάσεων
    tempm = char(temp_head(1:4));
    tempn = char(temp_head(5:8));
    m = str2double(tempm');
    n = str2double(tempn');
end


%% Βήμα 2: Προσδιορισμός των θέσεων που είναι τοποθετημένο το μήνυμα


z = 0;

enc_msg = [];
stopmax = (m * n);

for z = 1:stopmax
    temp = zeros(1,8);
    temp_loc = randGroup(z+8,:);

    %  Τα πρώτα 3 επίπεδα (RGB) χρώματος από το 1 pixel
    % ----------------------------------
    [row1,col1] = ind2sub([canM,canN],temp_loc(1));
    % Red
    temp(1,1) = mod(img(row1,col1,1),2);

    % Green
    temp(1,2) = mod(img(row1,col1,2),2);

    % Blue
    temp(1,3) = mod(img(row1,col1,3),2);

    % Τα επόμενα 3 επίπεδα του pixel (RGB)
    % ----------------------------------
    [row2,col2] = ind2sub([canM,canN],temp_loc(2));

    % Blue
    temp(1,4) = mod(img(row2,col2,3),2);

    % Green
    temp(1,5) = mod(img(row2,col2,2),2);

    % Red
    temp(1,6) = mod(img(row2,col2,1),2);

    % Τα επόμενα 2 από τα 3 επιπεδα του pixel (RG_)
    % ----------------------------------
    [row3,col3] = ind2sub([canM,canN],temp_loc(3));
    % Red
    temp(1,7) = mod(img(row3,col3,1),2);

    % Green
    temp(1,8) = mod(img(row3,col3,2),2);

    % Μετατροπή σε string και προσθήκη στο μήνυμα
    tempstr = num2str(temp);
    enc_msg = vertcat(enc_msg,tempstr);
end
```

```
%% Βήμα 3: Αποκρυπτογράφηση Βήμα
msg_dec_set = bin2dec(enc_msg);
msg_dec = bitxor(uint8(msg_dec_set),uint8(enc_key));


%% Βήμα 4: Προετοιμασία μηνύματος προς ανάκτηση
if temp_head(1) == 116
    % CASE 1: Κείμενο
    msg_set = msg_dec;
    msg_out = char(msg_set');
    output = msg_out;
else
    % CASE 2: Εικόαν
    % Προσδιορισμός της διάστασης της εικόνας
    tempm = char(temp_head(1:4));
    tempn = char(temp_head(5:8));
    m = str2double(tempm');
    n = str2double(tempn');

    % Αντιγραφή του κρυπτογραφημένου μηνύματος
    msg_set = msg_dec;

    count = 1;
    msg_out = uint8(zeros(m,n));
    for y = 1:m
        for x = 1:n
            msg_out(y,x) = msg_set(count);
            count = count + 1;
        end
    end
    output = im2uint8(msg_out);

end


%% Βήμα 5: Τελική έξοδος/ΕΠιστροφή
msg = output;
end
```