Open Schools Journal for Open Science

Vol 3, No 6 (2020)

Γενετικός κώδικας - κρυπτογράφια

Ιωάννης Δροσσάς, Αλέξανδρος Νικολάου, Κυριακή Γρηγοριάδου, Ιωάννη Ζήκος

doi: 10.12681/osj.24297

To cite this article:

Δροσσάς Ι., Νικολάου Α., Γρηγοριάδου Κ., & Ζήκος Ι. (2020). Γενετικός κώδικας - κρυπτογράφια. *Open Schools Journal for Open Science*, *3*(6). https://doi.org/10.12681/osj.24297

Open Schools Journal for Open Science

Γενετικός κώδικας - κρυπτογράφια

Δροσσάς Ιωάννης[1], Νικολάου Αλέξανδρος[1], Γρηγοριάδου Κυριακή[2], Ζήκου Ιωάννης[3]

[1]1ο Πειραματικό ΓΕΛ Θεσσαλονίκης «Μανόλης Ανδρόνικος», Θεσσαλονίκη, Ελλάδα,

[2]Βιολόγος, 1ο Πειραματικό ΓΕΛ Θεσσαλονίκης «Μανόλης Ανδρόνικος», Θεσσαλονίκη, Ελλάδα,

[3]Πληροφορικός, 1ο Πειραματικό ΓΕΛ Θεσσαλονίκης «Μανόλης Ανδρόνικος», Θεσσαλονίκη, Ελλάδα

ΠΕΡΙΛΗΨΗ

Η ομάδα μας δημιούργησε μια εφαρμογή που κωδικοποιεί και αποκωδικοποιεί μηνύματα συνδυάζοντας δεδομένα της Βιολογίας και της Πληροφορικής. Δημιουργήσαμε ένα σύστημα κρυπτογράφησης, βασισμένο στον γενετικό κώδικα. Αποφασίσαμε να πρωτοτυπήσουμε και να δημιουργήσουμε ένα σύστημα κρυπτογράφησης με βάση τον γενετικό κώδικα. Πώς θα καταφέρναμε, λοιπόν, να υλοποιήσουμε ένα τέτοιο πρόγραμμα; Ποια εργαλεία θα χρησιμοποιούσαμε; Α) Γενετικός κώδικας: Ο γενετικός κώδικας είναι παγκόσμιος, ισχύει για όλους τους ζωντανούς οργανισμούς. Αυτό το χαρακτηριστικό είναι και ένας από τους λόγους για τους οποίους επιλέχθηκε για το σύστημα κρυπτογράφησης. Επίσης, είναι τριαδικός, που σημαίνει πως μια τριάδα νουκλεοτιδίων κωδικοποιεί ένα αμινοξύ ή στην περίπτωσή μας έναν χαρακτήρα. Οι δυνατοί συνδυασμοί των 4 νουκλεοτιδίων ανέρχονται σε 64, γεγονός που μας δίνει την δυνατότητα κωδικοποίησης πολλών χαρακτήρων. Β) Γλώσσα προγραμματισμού C: Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής ήταν η γλώσσα προγραμματισμού C, η οποία τρέχει σε όλους τους ηλεκτρονικούς υπολογιστές ανεξαρτήτως του λειτουργικού συστήματός τους. Γ) Κρυπτογραφία: Το πρόγραμμα μας μετατρέπει τους χαρακτήρες που εισάγονται ως τριπλέτες νουκλεοτιδίων. Η μέθοδος που χρησιμοποιούμε είναι ένα substitution cipher, δηλαδή σε κάθε χαρακτήρα που εισάγεται αντιστοιχεί πάντα η ίδια τριπλέτα. Το αποτέλεσμα που προέκυψε ήταν μια εκτελέσιμη εφαρμογή για λειτουργικά συστήματα Windows και UNIX, η οποία ανάλογα με τις παραμέτρους

που της θέτουν οι χρήστες κωδικοποιεί ή αποκωδικοποιεί μηνύματα με την σχετική κρυπτογράφηση.

ΛΕΞΕΙΣ-ΚΛΕΙΔΙΑ

προγραμματισμός, γενετικός κώδικας, κρυπτογραφία, τριαδικός

ΕΙΣΑΓΩΓΗ

Η ομάδα μας προσπάθησε να γεφυρώσει τις επιστήμες της Βιολογίας και της Πληροφορικής, θέλοντας να δείξει, με αυτόν τον τρόπο, τα θετικά αποτελέσματα που αποφέρει η «ανάμειξή» τους. Για να το πετύχουμε αυτό θεωρήσαμε πως θα έπρεπε να δημιουργήσουμε μια εφαρμογή, χρηστική, εύκολα προσβάσιμη και πρωτότυπη, που να δείχνει με τον καλύτερο δυνατό τρόπο τη σχέση που ενώνει τα προαναφερθέντα επιστημονικά πεδία. Με αφορμή μαθήματα σχετικά με τον γενετικό κώδικα, παρατηρήσαμε τις ομοιότητές του με άλλα συστήματα κωδικοποίησης και συνειδητοποιήσαμε ότι μοιάζουν πολύ. Αποφασίσαμε, λοιπόν, να υλοποιήσουμε ένα σύστημα κρυπτογράφησης με βάση τον γενετικό κώδικα, χρησιμοποιώντας για την επίτευξη του - την όχι και τόσο γνωστή επιστήμη της Κρυπτογραφίας. Συνδυάζοντας τις ήδη αρκετές γνώσεις μας στον τομέα της Πληροφορικής και τις ικανότητές μας με την γλώσσα προγραμματισμού C, τις διάφορες μεθόδους κρυπτογράφησης, που γνωρίζαμε και τις πληροφορίες του βιβλίου της Βιολογίας της Β Λυκείου, καταφέραμε, τελικά, να υλοποιήσουμε το σύστημα κρυπτογράφησης.

ΓΕΝΕΤΙΚΟΣ ΚΩΔΙΚΑΣ

Ο γενετικός κώδικας είναι παγκόσμιος, ισχύει για όλους τους ζωντανούς οργανισμούς. Αυτό το χαρακτηριστικό είναι και ένας από τους λόγους για τους οποίους επιλέχθηκε για το σύστημα κρυπτογράφησης. Επίσης, είναι τριαδικός, καθώς, μια τριάδα νουκλεοτιδίων (γνωστή ως «κωδικόνιο») κωδικοποιεί ένα αμινοξύ.

Γενετικός κώδικας

Πρώτο γράμμα	Δεύτερο γράμμα U	Δεύτερο γράμμα C	Δεύτερο γράμμα A	Δεύτερο γράμμα G	Τρίτο γράμμα
U	UUU, UUC: φαινυλαλανίνη; UUA, UUG: λευκίνη	UCU, UCC, UCA, UCG: σερίνη	UAU, UAC: τυροσίνη; UAA, UAG: λήξη	UGU, UGC: κυστεΐνη; UGA: λήξη; UGG: τρυπτοφάνη	U C A G
C	CUU, CUC, CUA, CUG: λευκίνη	CCU, CCC, CCA, CCG: προλίνη	CAU, CAC: ιστιδίνη; CAA, CAG: γλουταμίνη	CGU, CGC, CGA, CGG: αργινίνη	U C A G
A	AUU, AUC, AUA: ισολευκίνη; AUG: μεθειονίνη έναρξη	ACU, ACC, ACA, ACG: θρεονίνη	AAU, AAC: ασπαραγγίνη; AAA, AAG: λυσίνη	AGU, AGC: σερίνη; AGA, AGG: αργινίνη	U C A G
G	GUU, GUC, GUA, GUG: βαλίνη	GCU, GCC, GCA, GCG: αλανίνη	GAU, GAC: ασπαρτικό οξύ; GAA, GAG: γλουταμινικό οξύ	GGU, GGC, GGA, GGG: γλυκίνη	U C A G

Σχήμα 1: Όλες οι τριάδες νουκλεοτιδίων, οι οποίες ανέρχονται σε 64.

Από τα εξήντα τέσσερα διαφορετικά αμινοξέα, τέσσερα έχουν διαφορετικό ρόλο από τα υπόλοιπα, στο στάδιο της έκφρασης της γενετικής πληροφορίας (μεταφραστική διαδικασία). Τα τρία από αυτά δεν κωδικοποιούν κανένα αμινοξύ και λειτουργούν ως σήματα λήξης της μετάφρασης (UAA,UAG, UGA), ενώ το τέταρτο (AUG) λειτουργεί ως σήμα έναρξης της μετάφρασης, αλλά παράλληλα κωδικοποιεί το αμινοξύ μεθειονίνη (Καψάλης κ.α., 2014). Η ύπαρξη αυτών των κωδικονίων ήταν ένα ακόμα χαρακτηριστικό του γενετικού κώδικα για το οποίο τον επιλέξαμε, καθώς διευκολύνεται ο διαχωρισμός των κωδικοποιημένων λέξεων μέσα στο σώμα ενός κωδικοποιημένου κειμένου. Τέλος, αξίζει να αναφερθεί ότι οι συνδυασμοί των νουκλεοτιδίων σε τριάδες είναι αρκετοί, για να συμπεριληφθούν στο σύστημα κρυπτογράφησης: όλο το λατινικό αλφάβητο (κεφαλαία και μικρά γράμματα), όλοι οι αριθμοί από το μηδέν έως το 9 και άλλα σύμβολα, όπως η τελεία, το θαυμαστικό και η κόμμα.

ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C

Η γλώσσα προγραμματισμού C είναι μια διαδικαστική γλώσσα προγραμματισμού, γενικής χρήσης, η οποία παρέχει οικονομία στην έκφραση, σύγχρονη διαχείριση ροής και δομής δεδομένων, καθώς και μια πλούσια «γκάμα» συντελεστών. Η C δεν είναι γλώσσα «υψηλού επιπέδου», ούτε απευθύνεται σε κάποιο συγκεκριμένο πεδίο εφαρμογής. Αυτό που την κατατάσσει ανάμεσα στις ισχυρότερες γλώσσες προγραμματισμού είναι η απουσία περιορισμών και η γενικότητά της, που την καθιστούν ικανή να χρησιμοποιηθεί για την υλοποίηση ενός ευρέως φάσματος εφαρμογών.

Αρχικά, είχε σχεδιαστεί και δημιουργηθεί στο λειτουργικό σύστημα UNIX, από τον Dennis Ritchie. Το λειτουργικό σύστημα, ο μεταγλωττιστής και ουσιαστικά όλες οι εφαρμογές του UNIX είναι γραμμένα σε C (Brian at all, 1988).

ΚΡΥΠΤΟΓΡΑΦΙΑ

Ως κρυπτογραφία ορίζεται η επιστήμη της κατάλληλης κωδικοποίησης ενός κειμένου, ή γενικά κάποιου μηνύματος έτσι ώστε κανένας “τρίτος” να μην μπορεί να αξιοποιήσει το μήνυμα σε περίπτωση υποκλοπής του. Μπορεί η κρυπτογραφία να φαντάζει κάτι απόμακρο, ωστόσο όλοι την χρησιμοποιούμε έστω και έμμεσα σε καθημερινή βάση. Χαρακτηριστικό παράδειγμα είναι η κωδικοποίηση των μηνυμάτων ηλεκτρονικού ταχυδρομείου και όλες τις διαδικασίες που περιλαμβάνουν μετακίνηση χρημάτων μέσω διαδικτύου. Η κρυπτογραφία επιτελείται με την χρήση ειδικών μεθόδων που λέγονται “κλειδιά”. Όσο πιο περίπλοκο είναι το “κλειδί”, τόσο πιο δύσκολο είναι για αυτόν που θα υποκλέψει το σήμα να αποκωδικοποιήσει τα δεδομένα. Για να επιτευχθεί λοιπόν η μέγιστη δυνατή ασφάλεια των χρηστών, πριν ένας αλγόριθμος κωδικοποίησης τεθεί σε χρήση περνάει χρόνια δοκιμών από ειδικούς κρυπτογράφους με την χρήση τόσο brute-force όσο και analytic επιθέσεων (Christof Paar & Jan Pelzl, 2009).Έμπνευση μας για την χρήση της Κρυπτογραφίας στην εφαρμογή, στάθηκε η ταινία «The Imitation Game» του 2014, που είναι βασισμένη σε αληθινά γεγονότα που αφορούσαν το σπάσιμο του κώδικα Enigma. Στην ταινία ο Alan Turing αποδεικνύει την σημαντικότητα της κρυπτογραφίας στην εξέλιξη του 2ου παγκοσμίου πολέμου.

ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΚΡΥΠΤΟΓΡΑΦΗΣΗΣ

Η υλοποίηση του κώδικα του συστήματος κρυπτογράφησης πραγματοποιήθηκε σε τρία βασικά βήματα, την δημιουργία ενός προσχέδιου του κώδικα, την υλοποίηση του κώδικα και την διαδικασία των δοκιμών και της εξσφαλμάτωσης.

Κατά τη δημιουργία του προσχέδιου του κώδικα σχεδιάσαμε σε ένα φύλλο χαρτί το γραφικό περιβάλλον της εφαρμογής, όπως θα θέλαμε να προβάλλεται στην οθόνη, καθώς και τον τρόπο με τον οποίο το πρόγραμμα θα κωδικοποιούσε και αποκωδικοποιούσε τα μηνύματα που θα εισήγαγε ο χρήστης. Έπειτα, πρόχειρα γράψαμε ποιες ιδιότητες της γλώσσας προγραμματισμού C θα χρησιμοποιούσαμε για την υλοποίησή τους. Στο βήμα αυτό κατασκευάσαμε, επίσης, ένα ενδεικτικό αλφάβητο για τις δοκιμές του συστήματος κρυπτογράφησης.

UUU	A
UUC	B
UUA	C
UUG	D
UCU	E
UCC	F
UCA	G
UCG	H
UAU	I
UAC	J
UAA	K
UAG	L
UGU	M
UGC	N
UGA	O
UGG	P
CUU	Q

CUC	R
CUA	S
CUG	T
CCU	U
CCC	V
CCA	W
CCG	X
CAU	Y
CAC	Z
CAA	a
CAG	b
CGU	c
CGC	d
CGA	e
CGG	f
AUU	g
AUC	h
AUA	i
AUG	j
ACU	k
ACC	l
ACA	m
ACG	n
AAU	o
AAC	p
AAA	q
AAG	r

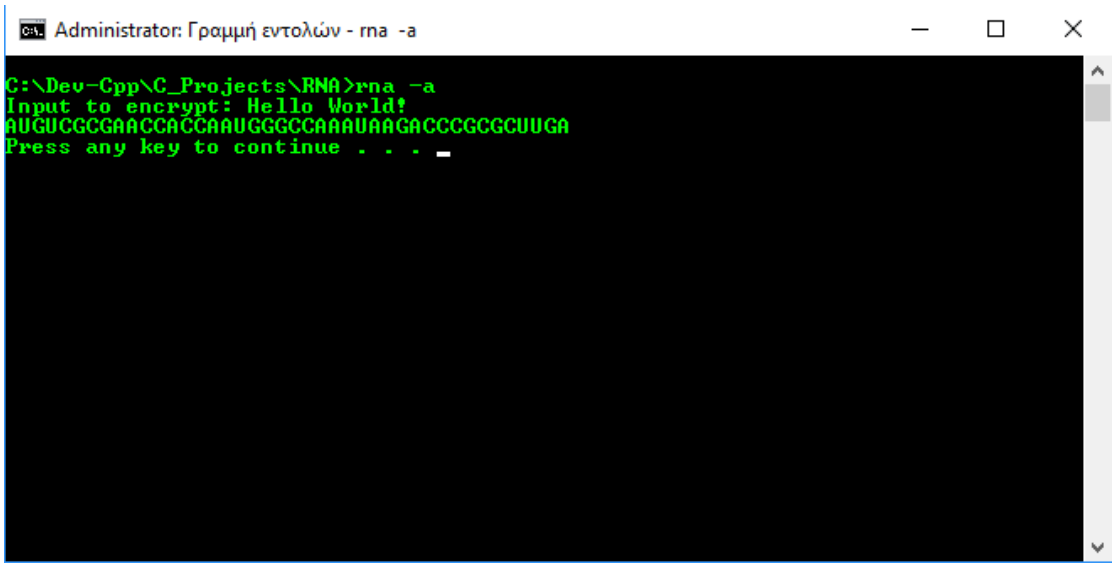
AGU	s
AGC	t
AGA	u
AGG	v
GUU	w
GUC	x
GUA	y
GUG	z
GCU	!
GCC	0
GCA	1
GCG	2
GAU	3
GAC	4
GAA	5
GAG	6
GGU	7
GGC	8
GGA	9
GGG	(blank)

Πίνακας 1: Ενδεικτικό αλφάβητο του συστήματος κρυπτογράφησης.

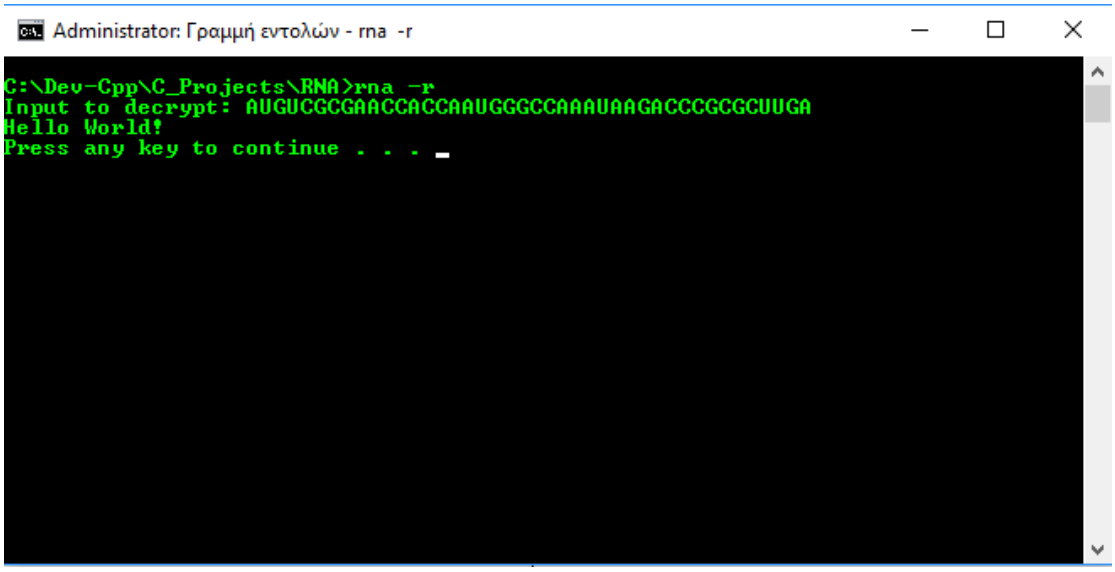
Αφού το προσχέδιο ήταν έτοιμο, προχωρήσαμε στην υλοποίηση του κώδικα. Το «γράψιμο» του κώδικα και η μεταγλώττισή του έλαβαν χώρα σε λειτουργικό σύστημα UNIX με την βοήθεια του μεταγλωττιστή GCC. Στην πορεία μεταγλωττίσαμε κατάλληλα την εφαρμογή, έτσι ώστε να λειτουργεί και σε Windows, σε περιβάλλον DOS. Η κύρια λειτουργία της γλώσσας προγραμματισμού C που χρησιμοποιήσαμε ήταν οι δομές δεδομένων (structs), την οποία εκμεταλλευτήκαμε δημιουργώντας μια δικιά μας δομή δεδομένων, παρόμοια με αυτή του

γενετικού κώδικα, με την διαφορά ότι υποστήριζε την κωδικοποίηση μηνυμάτων και όχι πρωτεϊνικών μορίων.

Στο τρίτο και τελευταίο στάδιο, αφού η εφαρμογή είχε υλοποιηθεί, προσπαθήσαμε να εξαλείψουμε οποιοδήποτε πιθανό σφάλμα, δοκιμάζοντας την εφαρμογή σε λογικές, αλλά και σε ακραίες συνθήκες εισαγωγής δεδομένων. Το κυριότερο πρόβλημα που αντιμετωπίσαμε σε αυτό το βήμα ήταν να καταφέρουμε να καταστήσουμε το πρόγραμμα δυνατό να ξεχωρίζει την εκάστοτε χρήση των τεσσάρων κωδικονίων (AUG, UAA, UAG, UAA), που έχουν διπλή χρήση στο σύστημα, είτε για να σημάνουν έναρξη – λήξη, είτε για την κωδικοποίηση ορισμένων χαρακτήρων. Όταν το πρόγραμμα πέρασε όλες τις δοκιμασίες επιτυχώς, τότε πλέον μπορούσαμε να το παρουσιάσουμε και να αρχίσουμε να σκεφτόμαστε μελλοντικές επεκτάσεις. Το τελικό προϊόν είναι μια εφαρμογή σε δύο εκδόσεις (μια για λειτουργικά τύπου UNIX και μια για Windows), η οποία τρέχει σε Terminal, στο UNIX και στο Command Prompt στα Windows. Για να επικοινωνήσουν δύο άτομα μέσω αυτής της εφαρμογής αρκεί ο χρήστης που θέλει να στείλει ένα μήνυμα, να επιλέξει πρώτα την λειτουργία κωδικοποίησης και να το εισάγει στην εφαρμογή και αφού αυτό κωδικοποιηθεί να στείλει το μήνυμα στον εκάστοτε παραλήπτη, μέσω οποιουδήποτε μέσου επικοινωνίας επιθυμεί αυτός (γράμμα σε φάκελο, μήνυμα στο Messenger κτλ) (Σχήμα 2). Αφού ο παραλήπτης λάβει το κωδικοποιημένο μήνυμα, για να το αποκωδικοποιήσει πρέπει να επιλέξει την λειτουργία αποκωδικοποίησης και να το εισάγει στο πρόγραμμα. Αυτό με την σειρά του θα αποκωδικοποιήσει το μήνυμα και θα εμφανίσει το αποτέλεσμα στο περιβάλλον της εφαρμογής (Σχήμα 3).



Σχήμα 2: Κωδικοποίηση του μηνύματος “Hello World!” σε τριάδες αμινοξέων.



Σχήμα 3: Αποκωδικοποίηση των τριάδων αμινοξέων στο μήνυμα του αποστολέα (“Hello World!”).

ΣΥΜΠΕΡΑΣΜΑΤΑ

Ανακεφαλαιώνοντας, καταφέραμε να γεφυρώσουμε δύο κατά πολλούς άσχετα επιστημονικά πεδία μεταξύ τους, της Βιολογίας και της Πληροφορικής, χρησιμοποιώντας ως γέφυρα την επιστήμη της Κρυπτογραφίας. Αυτό το επιτύχαμε με την υλοποίηση ενός συστήματος

κρυπτογράφησης, το οποίο θα κωδικοποιεί μηνύματα που θα εισάγει ο χρήστης σε κωδικόνια RNA και το αντίστροφο. Η εφαρμογή αυτή μπορεί να αξιοποιηθεί από πολλούς για την ασφαλή μεταφορά «ευαίσθητων» μηνυμάτων, χωρίς να απαιτούνται ιδιαίτερες γνώσεις πάνω στην λειτουργία της. Αποτελεί, επίσης, μια πρόκληση για όσους ενδιαφέρονται και τους αρέσει το «σπάσιμο» συστημάτων κωδικοποίησης.

Ως μελλοντική επέκταση, η εφαρμογή θα μπορούσε κάλλιστα να τροποποιηθεί κατάλληλα, για να λειτουργεί και σε συστήματα Android και IOS. Η επέκταση αυτή θα μπορούσε να περιλαμβάνει την κρυπτογράφηση αρχείων και φωτογραφιών, έτσι ώστε να παραμείνουν ασφαλή ακόμα και αν η συσκευή κλαπεί ή αν τα αρχεία υποκλαπούν.

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

[1] Καψάλης Α., Μπουρμπουχάκης Ι., Περάκη Β., Σαλαμαστράκης Σ. (2014). Βιολογία ΄Β Γενικού Λυκείου, Αθήνα: Εκδόσεις Διόφαντος, 122 – 130.

[2] Brian W. Kernighan & Dennis M. Ritchie. (1988). *“The C Programming Language”* – Second Edition, Prentice Hall Software Series, 1.

[3] Christof Paar & Jan Pelzl. (2009). *“Understanding Cryptography: A Textbook for Students and Practitioners”*, Springer.