Διαδικαστικός Προγραμματισμός

(ECE_Y215)

Επιπρόσθετες Βιβλιοθήκες στη C

Πρόγραμμα Διαλέξεων & Δραστηριοτήτων

Ημερομηνία	Περιεχόμενο
23 Απριλίου 2021 09:00 – 10:00	**Εισαγωγή & GSL - GNU Scientific Library (1/2)** • *εξειδικευμένος επιστημονικός υπολογισμός*
14 Μαΐου 2021 09:00 – 10:00	**GSL - GNU Scientific Library (2/2)** • *εξειδικευμένος επιστημονικός υπολογισμός*
21 Μαΐου 2021 09:00 – 10:00	GMP - GNU MultiPrecision Library (1/2) • ***αυθαίρετη αριθμητική ακρίβεια σε αριθμούς***
28 Μαΐου 2021 09:00 – 10:00	GMP - GNU MultiPrecision Library (2/2) • *αυθαίρετη αριθμητική ακρίβεια σε αριθμούς*

Δραστηριότητες	Παράδοση
1η Δραστηριότητα: 23 Απριλίου	10 Μαΐου 2021 (+ 0.25 μονάδες στον τελικό βαθμό)
2η Δραστηριότητα: 14 Μαΐου	28 Μαΐου 2021 (+ 0.25 μονάδες στον τελικό βαθμό)

Στόχοι

- Ο σκοπός της σημερινής διάλεξης είναι η περαιτέρω ενασχόληση με τη **βιβλιοθήκης GMP**: *GNU MultiPrecision Library*
- Να αναφερθούμε σε **παραδείγματα** ώστε να εμπεδώσουμε καλύτερα τις θεωρητικές γνώσεις
- Να **ασχοληθούμε με προβλήματα** που απαιτούν αυθαίρετη αριθμητική ακρίβεια στους υπολογιστικούς υπολογισμούς:
  - **κρυπτογραφία**
  - **αλγόριθμος ασύμμετρης κρυπτογραφίας – RSA**
  - **υλοποίηση του RSA μέσω της GMP**

Από προηγούμενες διαλέξεις

Βιβλιοθήκη είναι μια συλλογή **μεταγλωττισμένων μονάδων** που μπορούν να συνδεθούν (**linked**) στα προγράμματά μας μέσω των διεπαφών (**header files - interfaces**) που παρέχουν.

Με άλλα λόγια κάθε βιβλιοθήκη αποτελείτε από **δυαδικά** αρχεία σε object code (*.o) που περιέχουν **υλοποιήσεις** όλων των συναρτήσεων που έχουν **δηλωθεί** στα αρχεία επικεφαλίδων .h


```
• # include <stdio.h>
• # include "gmp.h"
            …
```

Η Βιβλιοθήκη της GMP: Υπολογισμός Παραγοντικού

Θέλουμε να γράψουμε ένα πρόγραμμα το οποίο να βρίσκει το παραγοντικό οποιαδήποτε αριθμού n. Παραγοντικό 5!=1*2*3*4*5

```
void compute_fact_basic_c(int n){
  int c;
  int p = 1;
  for (c=1; c <= n ; c++){
    p = p * c;
  }
  printf("%d! = %d",n,p);
}
```

```
10! = 3628800
```

```
15! = 2004310016
```

```
100! = 0
```

```
void compute_fact_gmp(int n){
  int c;
  mpz_t p;
  mpz_init_set_ui(p,1);
  for (c=1; c <= n ; c++) mpz_mul_ui(p,p,c);
  printf ("%d! = ", n);
  gmp_printf("%Zd\n",p);
  mpz_clear(p);
}
```

```
100! = 93326215443944152681699238856266700490715968264
381621468592963895217599993229915608941463976156518286253697920827223758251185210916864000000000000000000000000
```

Ο πηγαίος κώδικας των σημερινών ασκήσεων υπάρχει στο eclass [lecture 24]

Η Βιβλιοθήκη της GMP: Υπολογισμός Παραγοντικού

Ο πηγαίος κώδικας του παραδείγματος βρίσκεται στο eclass: [lecture24] **GMP-FACTOR**

- Τρίτη έκδοση (πλήρως GMP συμβατή):

```
void compute_fact_gmp2(mpz_t *n){
  mpz_t fact;
  mpz_init_set_ui(fact,1);
  mpz_t p;
  mpz_init_set_ui(p,1);
  while(mpz_cmp (p, *n)<=0){
        mpz_mul(fact,fact,p);
        mpz_add_ui(p, p, 1);
  }
  gmp_printf("%Zd\n",fact);
  mpz_clear(p);
  mpz_clear(fact);
}
```

Η συνάρτηση compute_fact_gmp2:
- **Δέχεται ως όρισμα έναν δείκτη σε δομή ακεραίου αριθμού αυθαίρετης ακρίβειας**

Αντικαθιστούμε τη *for με μια while*

Αποδεσμεύουμε τη μνήμη

Ο πηγαίος κώδικας των σημερινών ασκήσεων υπάρχει στο eclass [lecture 24]

Η Βιβλιοθήκη της GMP: Ακολουθία Fibonacci

Θέλουμε να γράψουμε ένα πρόγραμμα το οποίο να βρίσκει τον αριθμό **Fibonacci** για ακολουθία **n αριθμών**. π.χ. 0,1,1,2,3,5,6,13,21 etc.

```c
void compute_fibonacci_basic_c(int n){
   int i,fibo,prev1,prev2;
   prev1=0;
   prev2=1;
   printf("Fibonacci Series: %d, %d, ", prev1, prev2);
   for(i=2;i<=n;i++){
        fibo=prev1+prev2;
        printf(" %d,",fibo);
        prev1=prev2;
        prev2=fibo;
    }
}
```

```c
void compute_fibonacci_gmp(int n){
   int i;
   mpz_t fibo,prev1,prev2;
   mpz_init(fibo);
   mpz_init_set_ui(prev1,0);
   mpz_init_set_ui(prev2,1);
   gmp_printf("Fibonacci Series (n=%d):%Zd,%Zd", n, prev1, prev2);
   for(i=2;i<=n;i++){
      mpz_add( fibo,prev1,prev2 ) ;
      gmp_printf(",%Zd", fibo);
      mpz_set(prev1,prev2);
      mpz_set(prev2,fibo);
   }
   mpz_clear(fibo);
   mpz_clear(prev1);
   mpz_clear(prev2);
}
```

```
Fibonacci Series (n=30):0,1, 1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,2
584,4181,6765,10946,17711,28657,46368,75025,121393,196418,317811,514229,832040,


Fibonacci Series (n=30):0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,25
84,4181,6765,10946,17711,28657,46368,75025,121393,196418,317811,514229,832040,
```

Ο πηγαίος κώδικας των σημερινών ασκήσεων υπάρχει στο eclass [lecture 24]

Η Βιβλιοθήκη της GMP: Ακολουθία Fibonacci

Ο πηγαίος κώδικας του παραδείγματος βρίσκεται στο eclass: [lecture24] **GMP-FIBONACCI**

```
void compute_fibonacci_basic_c(int n);
void compute_fibonacci_gmp(int n);
int main()
{
    compute_fibonacci_basic_c(100);
    printf("\n\n\n");
    compute_fibonacci_gmp(100);
    return 0;
}
```

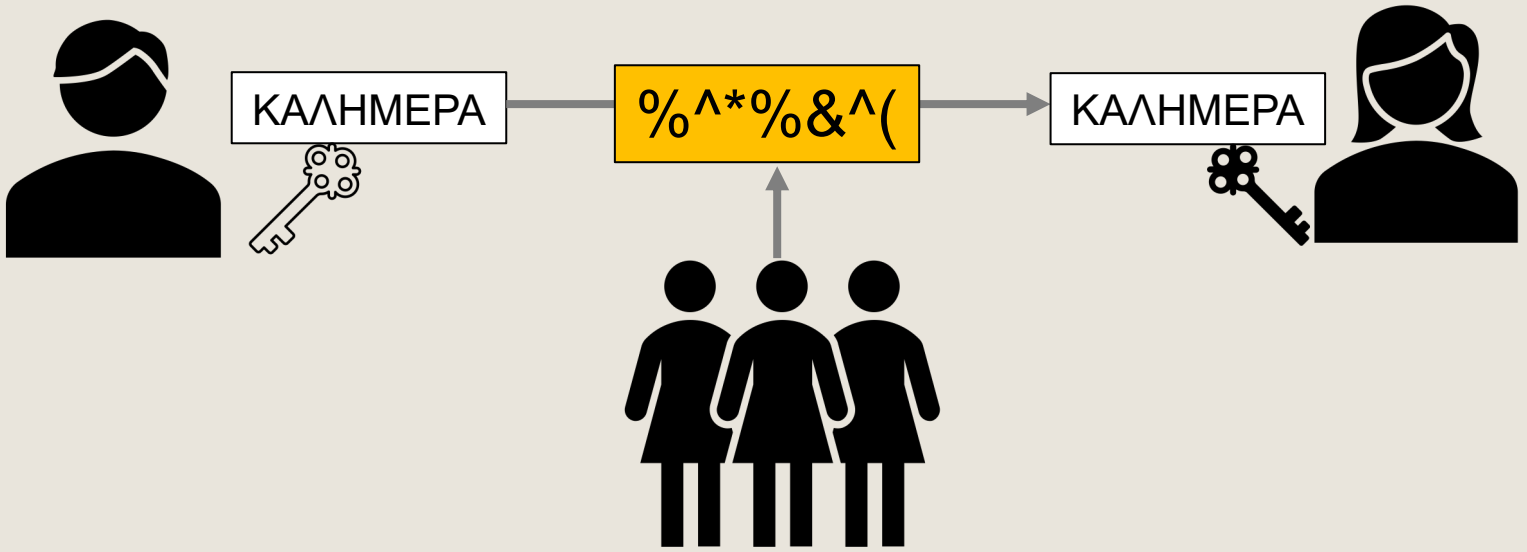
32-bit: **2.147.483.647**

```
Fibonacci Series (n=100):0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,25
84,4181,6765,10946,17711,28657,46368,75025,121393,196418,317811,514229,832040,134
6269,2178309,3524578,5702887,9227465,14930352,24157817,39088169,63245986,10233415
5,165580141,267914296,433494437,701408733,1134903170,1836311903,-1323752223,5
9680,-811192543,-298632863,-1109825406,-1408458269,1776683621,3
3,-1781832971,363076002,-1418756969,-1055680967,1820529360,764848393,-1709589543,
-944741150,1640636603,695895453,-1958435240,-1262539787,1073992269,-188547518,885
444751,696897233,1582341984,-2015728079,-433386095,1845853122,1412467027,-1036647
147,375819880,-660827267,-285007387,-945834654,-1230842041,2118290601,887448560,-
1289228135,-401779575,-1691007710,-2092787285,511172301,-1581614984,-1070442683,1
642909629,572466946,-2079590721,-1507123775,708252800,-798870975,-90618175,-88948
9150,-980107325


Fibonacci Series (n=100):0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,25
84,4181,6765,10946,17711,28657,46368,75025,121393,196418,317811,514229,832040,134
6269,2178309,3524578,5702887,9227465,14930352,24157817,39088169,63245986,10233415
5,165580141,267914296,433494437,701408733,1134903170,1836311903,2971215073,48
6976,7778742049,12586269025,20365011074,32951280099,53316291173,
3862445,225851433717,365435296162,591286729879,956722026041,1548008755920,2504730
781961,4052739537881,6557470319842,10610209857723,17167680177565,27777890035288,4
4945570212853,72723460248141,117669030460994,190392490709135,308061521170129,4984
54011879264,806515533049393,1304969544928657,2111485077978050,3416454622906707,55
27939700884757,8944394323791464,14472334024676221,23416728348467685,3788906237314
3906,61305790721611591,99194853094755497,160500643816367088,259695496911122585,42
0196140727489673,679891637638612258,1100087778366101931,1779979416004714189,28800
67194370816120,4660046610375530309,7540113804746346429,12200160415121876738,19740
274219868223167,31940434634990099905,51680708854858323072,83621143489848422977,13
```

Ο πηγαίος κώδικας των σημερινών ασκήσεων υπάρχει στο eclass [lecture 24]

Από την προηγούμενη διάλεξη: Κρυπτογραφία

Το πρόβλημα της κρυπτογραφίας είναι ένα διαχρονικό πρόβλημα της ασφαλούς επικοινωνίας. Στόχος είναι αν το μήνυμα της επικοινωνίας υποκλαπεί να μην μπορεί να κατανοηθεί με εύκολο τρόπο.

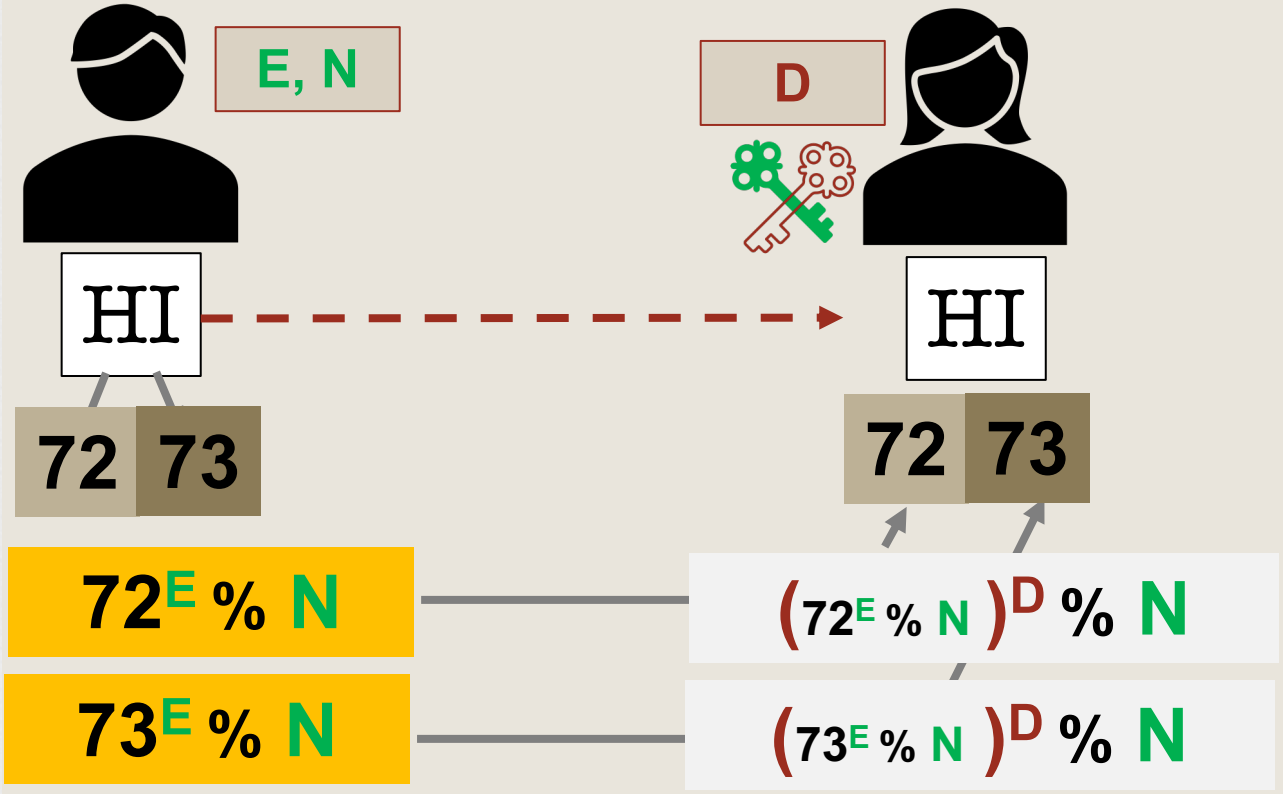


https://en.wikipedia.org/wiki/Cryptography

Από το προηγούμενο μάθημα: Ασύμμετρη Κρυπτογραφία

ASCII printable characters

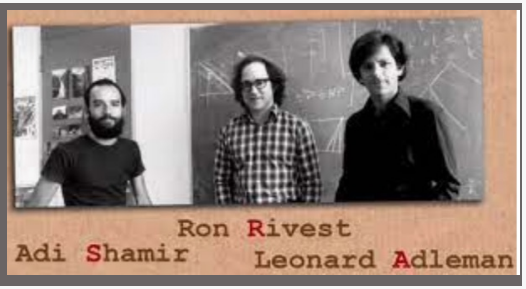
32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		



https://en.wikipedia.org/wiki/RSA_(cryptosystem)

https://en.wikipedia.org/wiki/Cryptography

RSA - Ασύμμετρη Κρυπτογραφία Αλγόριθμος (1977)



- Βρίσκουμε δυο μεγάλους πρώτους αριθμούς **p, q**.
- Υπολογίζουμε τον αριθμό **N=p*q**
- Υπολογίζουμε τον αριθμό **Φ (N) =(p-1)*(q-1)** (αναπαριστά το σύνολο των αριθμών οι οποίοι δεν έχουν κανένα κοινό παράγοντα με τον N)
- Υπολογίζουμε έναν ακέραιο **E** στο διάστημα **(1,Φ)** με κριτήριο τέτοιο ώστε οι αριθμοί **E** και **Φ** να μη έχουν κοινό παράγοντα παρά μόνο τον αριθμό **1** (Μ.Κ.Δ.=1)
- Υπολογίζουμε έναν ακέραιο **D** στο διάστημα **(1,Φ)** με κριτήριο ο αριθμός **D*E-1** να είναι πολλαπλάσιος του **A**

Δημόσιο Κλειδί: Οι αριθμοί **N** και **E** απαρτίζουν το δημόσιο κλειδί. Οι δυο αριθμοί χρησιμοποιούνται στην κρυπτογράφηση

Ιδιωτικό Κλειδί: Οι αριθμός **D** είναι το ιδιωτικό κλειδί και πρέπει να παραμείνει κρυφό

https://en.wikipedia.org/wiki/RSA_(cryptosystem)

Ασύμμετρη Κρυπτογραφία RSA: Βήμα 1ο - Υπολογισμός του N

Ενδεικτικό πηγαίος κώδικας βρίσκεται στην εκφώνηση της 3ης Δραστηριότητας στο eclass: [lecture24] **GMP – Δραστηριότητα 3**

```c
#include "gmp.h"
#include <time.h>
#include<unistd.h>
void init_gmp_rand(gmp_randstate_t  *stat);
void generate_prime(mpz_t* p,gmp_randstate_t
int main()
{
    int prime_bits;
    printf("Enter RSA Encryption bits: ");
    scanf("%d",&prime_bits);

    gmp_randstate_t stat;
    init_gmp_rand(&stat);

    mpz_t p,q;
    generate_prime(&p,&stat, &prime_bits);
    generate_prime(&q,&stat, &prime_bits);
    gmp_printf("p = %Zd\n",p);
    gmp_printf("q = %Zd\n",q);

   mpz_mul(N,p,q);
```

```
RSA - ENCRYPTION
RSA - Calculate p,q, N
Enter RSA Encryption bits: 1024
p = 561576059790923652897206600000603730229600048207394681173581044563934915972761324100318915865432579823084491517789662521211872105495187144474661044446180316632139553550021473286399824923531231736662345864260258107735260561649380073324378033628016549207558213460989016197866294660597128482049593075331918150631

q = 681506758444778544084406877359958925526221910805826900936414376312235286216060143634955668904399907853023564555679351139307406538619083188474354778344958344522420371466955746794164175620818078346414356666822528287104307421698621207378654441649947140490281209129649928183986989867962602081012926748166885671631

N = 382717880128303518777957425580456633017015691972733479352566662218178900316555750049214871389120656590128524861771830136500918398365332954145564854723050274437330586177747983304074024069795303757203698793473451395101425277394262544766607493352228497594209797784616545178559245931028074839525483177460048781930177319914317339797679408097598111907614121670913041726264551490736529643956129898562619362868987407872929607996117008326198982932968983544971303242526524172424568453672187485644422506954657138444950228420085704285230065214860726017933670352625234791125627624090106687937196140491718922664784396984075057626 9
```

https://en.wikipedia.org/wiki/RSA_(cryptosystem)

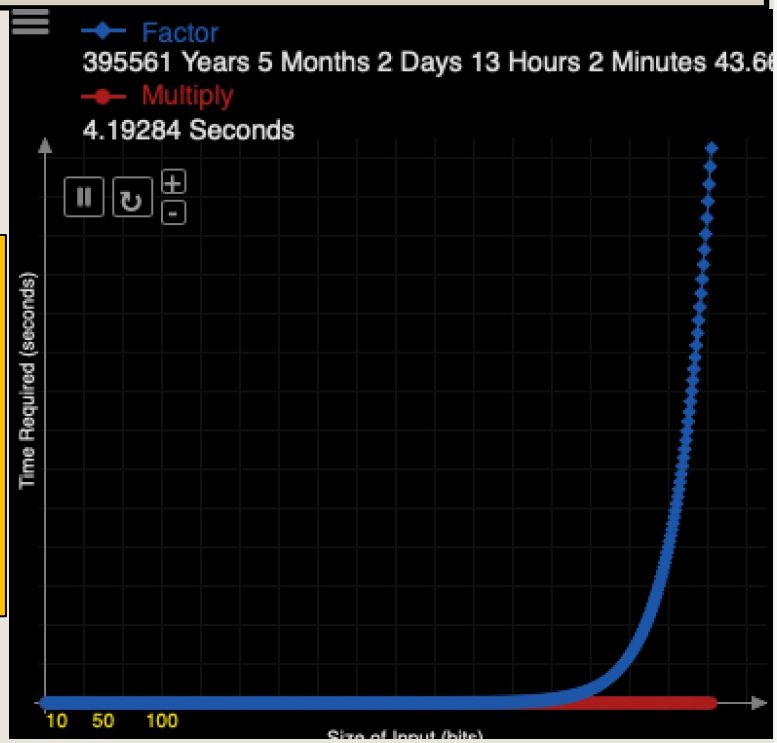
Ασύμμετρη Κρυπτογραφία RSA
Βήμα 1ο - Υπολογισμός του N

- Ο N είναι **ο modulo** για το δημόσιο και ιδιωτικό κλειδί
- Το μήκος του N σε **bits** ονομάζεται **μήκος κλειδιού** (key length)

$$N = p * q \qquad p \equiv q \pmod{N}$$

Οι αριθμοί p και q είναι πρώτοι αριθμοί μεγάλου μήκους σε bit.
Ισοδύναμοι modulo (δίνουν το ίδιο υπόλοιπο διαίρεσης από το συντελεστή αναδίπλωσης N)

Είναι αρκετά δύσκολο (*εκθετικός χρόνος ως προς το μήκος του N*) να βρεθούν οι πρώτοι παράγοντες (p,q) του αριθμού N [The factoring problem]



Δημόσιο Κλειδί: Οι αριθμός N είναι μέρος του δημοσίου κλειδιού

https://en.wikipedia.org/wiki/Shor%27s_algorithm

Ωστόσο: Σύντομα θα χρειαστούν νέοι αλγόριθμοι κρυπτογραφίας

How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits

Craig Gidney[1] and Martin Ekerå[2]

[1]Google Inc., Santa Barbara, California 93117, USA
[2]KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
Swedish NCSA, Swedish Armed Forces, SE-107 85 Stockholm, Sweden

13 Apr 2021

We significantly reduce the cost of factoring integers and computing discrete logarithms in finite fields on a quantum computer by combining techniques from Shor 1994, Griffiths-Niu 1996, Zalka 2006, Fowler 2012, Ekerå-Håstad 2017, Ekerå 2017, Ekerå 2018, Gidney-Fowler 2019, Gidney 2019. We estimate the approximate cost of our construction using plausible physical assumptions for large-scale superconducting qubit platforms: a planar grid of qubits with nearest-neighbor connectivity, a characteristic physical gate error rate of 10^{-3}, a surface code cycle time of 1 microsecond, and a reaction time of 10 microseconds. We account for factors that are normally ignored such as noise, the need to make repeated attempts, and the spacetime layout of the

https://arxiv.org/pdf/1905.09749.pdf

Ασύμμετρη Κρυπτογραφία RSA: Βήμα 2ο – Υπολογισμός του E

- *Υπολόγισε* θετικό ακέραιο **E** στο διάστημα (1, A) όπου A =(p -1) * (q - 1).
- Οι αριθμοί **E** και **A** **θα πρέπει να είναι σχετικά πρώτοι,** δηλαδή να έχουμε μέγιστο κοινό διαιρέτη (Μ.Κ.Δ.) τον αριθμό 1.

Ο αλγόριθμος του Ευκλείδη: Αλγόριθμος εύρεσης ΜΚΔ δυο ακέραιων αριθμών.

```
int euclid (int a , int b) {
    if (b == 0) return a;
    else
    return euclid(b, a % b);
}
```

Παράδειγμα εκτέλεσης:
(30,21) -> (21,9)->
(9,3) ->(3,0) -> 3

Αλεξάνδρεια 323–283 π.Χ.

```
void mpz_gcd (…);
```

Greatest Common Divisor

Δημόσιο Κλειδί: Οι αριθμοί **N** και **E** είναι το δημόσιο κλειδί

https://el.wikipedia.org/wiki/Αλγόριθμος_του_Ευκλείδη

Ασύμμετρη Κρυπτογραφία RSA: Βήμα 2ο - Υπολογισμός του E, D

Ενδεικτικό πηγαίος κώδικας βρίσκεται στην εκφώνηση της 3ης Δραστηριότητας στο eclass: [lecture24] **GMP - Δραστηριότητα 3**

```
mpz_t temp,e;
mpz_init(temp);
mpz_init(e);
do {
   mpz_urandomm(e, stat, phi + 1);
   mpz_gcd(temp, phi, e);
} while (mpz_cmp_ui(temp, 1) != 0);

gmp_printf("e = %Zd\n",e);
```

```
Enter RSA Encryption bits: 512
p = 151734340627195151767715774422086910192878504699170367713175496221163136736368176315773464827821304651960921207417228384368995112120805570208195811878341 3
q = 126830240768659756204204962923876442697040195779707294637505966196085581581766344997245517997090448829464774917986754909274668626073819978365466838645243397
N = 192445029546209928998414336459497449521982647362438026199052162523084644016643700148424390090344668904363693832480589205669263452637427883094096557909716171327120772436105245166583553923454593061577383160146961585030222802742391471612933053844768413683066880064338226737863570776400941465309070359926079890079
phi = 192445029546209928998414336459497449521982647362438026199052162523084644016643700148424390090344668904363693832480589205669263452637427883094096557909716029323445941056833864190043187838320876733531133535815552761514404600847136068450304230980288541103772219197299498260115859208263655564773684073506246812560
e = 250526199125815536008107252635165487528565306442638294537001153240346779621982446024638743686419856426395618257604947621712988891905174360445871952557322903449229113581654076483335541360817690299441508508353715850941143563028996774764460290581417382246283635370509110838493998876279711327874433824212851332 5
```

Οι αριθμοί N και E είναι το δημόσιο κλειδί και ο D είναι το ιδιωτικό κλειδί

https://en.wikipedia.org/wiki/RSA_(cryptosystem)

Ασύμμετρη Κρυπτογραφία RSA: Βήμα 2ο - Υπολογισμός του Φ, D

Ενδεικτικό πηγαίος κώδικας βρίσκεται στην εκφώνηση της 3ης Δραστηριότητας στο eclass: [lecture24] **GMP – Δραστηριότητα 3**

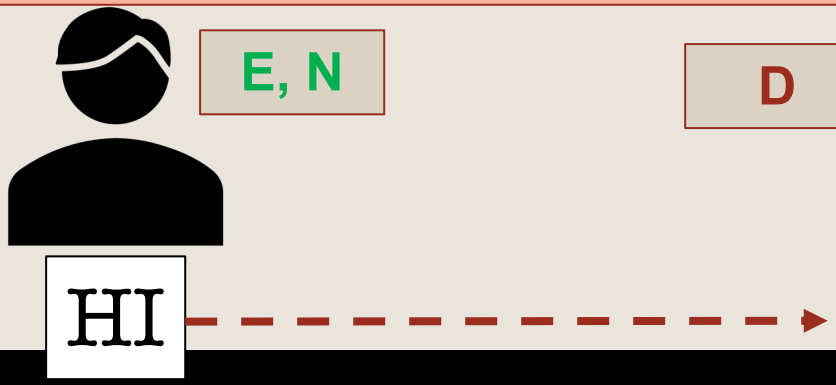
```
mpz_t phi,temp1, temp2;
mpz_init(phi);
mpz_init(temp1);
mpz_init(temp2);

mpz_sub_ui(temp1, q, 1);
mpz_sub_ui(temp2, p, 1);
mpz_mul(phi, temp1, temp2);
gmp_printf("\nphi = %Zd\n",phi);
```

```
mpz_t d;
mpz_init(d);
mpz_invert(d, e, phi);
 gmp_printf("\nd = %Zd\n",d);
```

```
Enter RSA Encryption bits: 512
p = 19798978527421175382905821667342466536693943510907890668431607226270659158
4
10077001705370680328159019625788781709386435652373520637250670068942702932829
69
q = 64322474127408331767901173385122106542621539522099047144153847845606687559
6
62969996430857854591833974860594494845438885431295872732683615487334471602003
29
N = 12735192840791616655036282912741466811667857231567691119314074396215682824
5
00507105813446181091297302239872094187359358040895620439057849313756271376216
26
97441103332032250493794576369097229056881967558689545086460252886401699083709
35
73762705391559920625402598258938513496921313687965047818089496300112389680
1

phi = 12735192840791616655036282912741466811667857231567691119314074396215682
82
45005071058134461810912973022398720941873593580408956204390578493137562713762
16
26133196506772027433429875813166326559775664845256826072738747978145243523656
36
31037813431062107206954577344261729652437104769940313484752339335283670413504

e = 10929522261421456837192603892228657934008003078780348837852777373649775559
0
19147464393608883310574892929870245584569379751095226525416355785212350966630
69
01384285189823025976626346764883611578207435807457267828843512848669870236416
76
61592606023802285954205145382057607884698664675617695495928333206587164826
5

d = 95615778686039101348711871308716325457303997335170692589353640623903492489
4
88012349436093435055690005828579445595747688902242244048601179792494348077303
58
87662508618940700702484558234952397412585907954362975785505521775837815853967
91
78306974996443164849639568508453673759516380978132190537961643838229143737
```

Ασύμμετρη Κρυπτογραφία RSA: Βήμα 3ο - Κωδικοποίηση

Ενδεικτικό πηγαίος κώδικας βρίσκεται στην εκφώνηση της 3ης Δραστηριότητας στο eclass: [lecture24] **GMP - Δραστηριότητα 3**




```
72
73
message as int before encryption

-------------------------------------------------------------------------------
encrypt message  = 781163576037670450628691425083231135210478919051309135383858912326522073424099494282772021000249568621836172831451023987165808881885668451339413230571384813228907851127553058452275238376361742373288708167460176103369049219879484103806902766666224967060704801448037071540042117140849699696768102516030756149 2

-------------------------------------------------------------------------------
message as int after decr
message as string after decr  = HI
```

Encryption **c = (msg ^ E) % N**

Decryption **msg = (c ^ D) % N**

https://en.wikipedia.org/wiki/RSA_(cryptosystem)

Ανασκόπηση Διαλέξεων & Δραστηριοτήτων

Ημερομηνία	Περιεχόμενο
23 Απριλίου 2021 09:00 – 10:00	**Εισαγωγή & GSL - GNU Scientific Library (1/2)** • *εξειδικευμένος επιστημονικός υπολογισμός*
14 Μαΐου 2021 09:00 – 10:00	**GSL - GNU Scientific Library (2/2)** • *εξειδικευμένος επιστημονικός υπολογισμός*
21 Μαΐου 2021 09:00 – 10:00	GMP - GNU MultiPrecision Library (1/2) • ***αυθαίρετη αριθμητική ακρίβεια σε αριθμούς***
28 Μαΐου 2021 09:00 – 10:00	GMP - GNU MultiPrecision Library (2/2) • *αυθαίρετη αριθμητική ακρίβεια σε αριθμούς*

Δραστηριότητες	Παράδοση
1η Δραστηριότητα: 23 Απριλίου	10 Μαΐου 2021 (+ 0.25 μονάδες στον τελικό βαθμό)
2η Δραστηριότητα: 14 Μαΐου	28 Μαΐου 2021 (+ 0.25 μονάδες στον τελικό βαθμό)

Ευχαριστώ για την προσοχή σας

- **Επικοινωνία**
  - Skype: fidas.christos
  - Email: fidas@upatras.gr
  - Phone: 2610 – 996491
  - Web: http://cfidas.info

Το υλικό της διάλεξης είναι διαθέσιμο στο eclass