Defensive Info Operations - Part I
Data Security & Cryptology

Hüseyin Hışıl

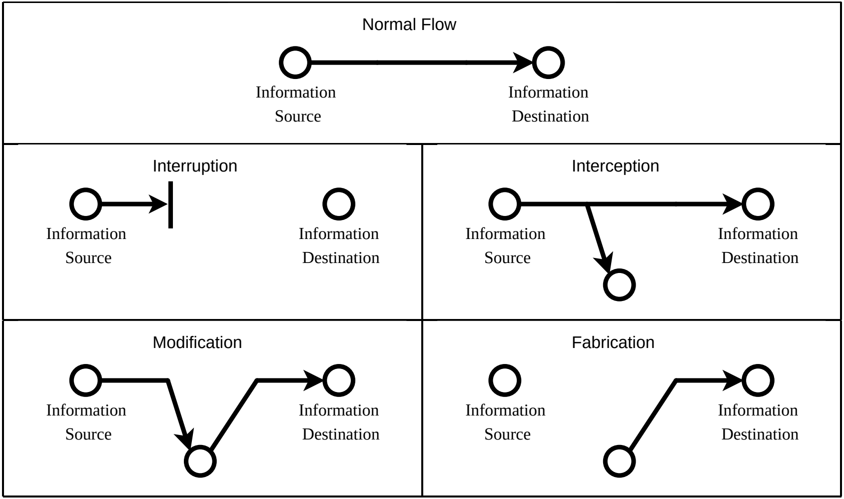
Computer Engineering Department
Engineering Faculty
Yaşar University

22 April 2012 / İzmir

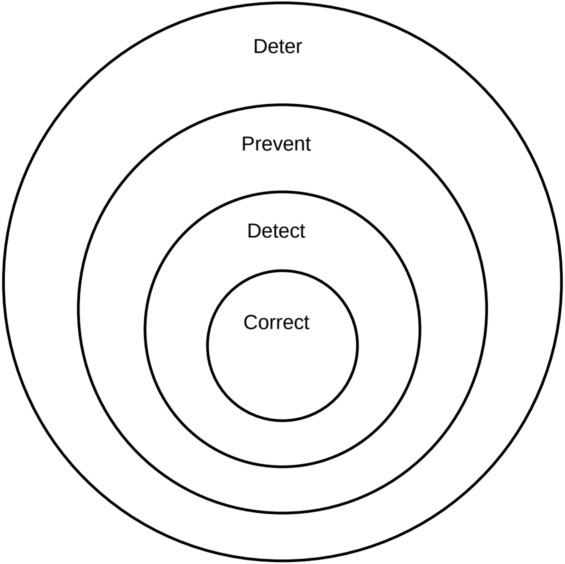
Acknowledgement: Some of the slides are compiled from Dr. Koltuksuz's lecture notes.

Possible Data Security Threats

Threat is a potential violation of security.



Layers of Defensive Data Security



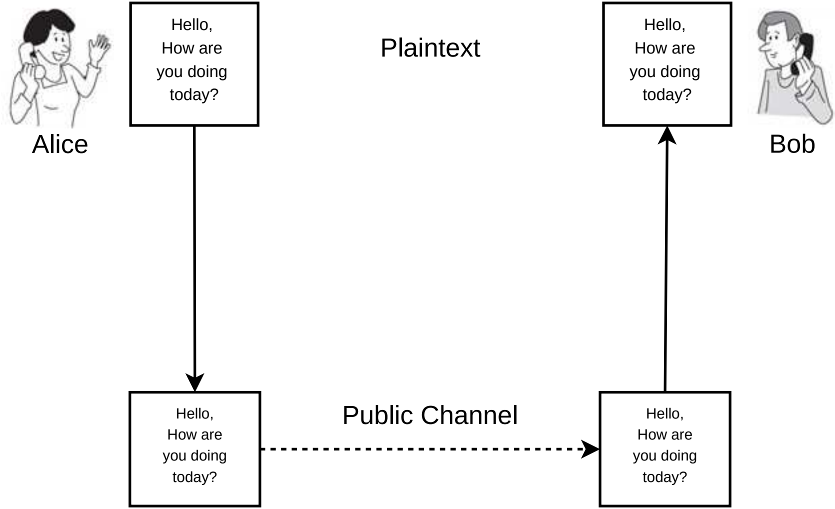
Essential Services for Data Security & Network Security

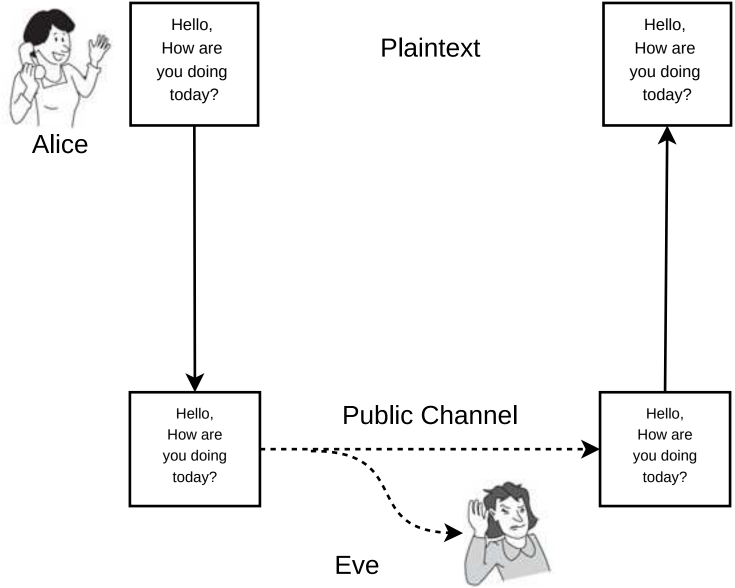
- Availability:
  - Ensures that data remain to be sucessfully accessible. (Networking)
  - Interruption targets availability.
- Authentication:
  - Ensures that data really were sent by the claimed sender. (Cryptology)
  - Fabrication targets authentication.
- Confidentiality:
  - Ensures that data are accessed only by authorized parties. (Cryptology)
  - Interception targets confidentiality.
- Integrity:
  - Ensures that the original data is intact. (Coding Theory)
  - Modification targets integrity.

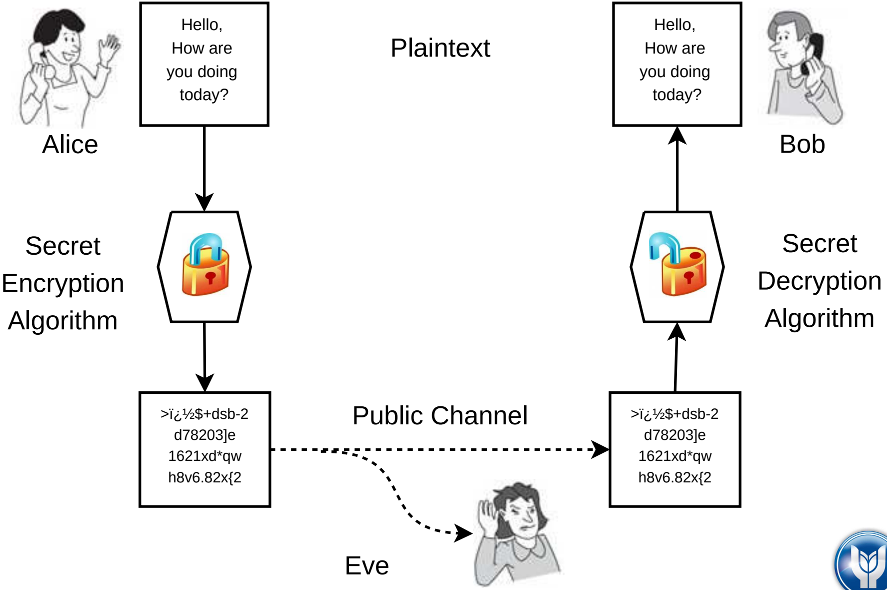
Note: Higher level services such as non-repudiation, access control, utility, possession, can be defined as needed.

The World of Crypto

- Cryptography: The science of securing data.
- Cryptanalysis: The science of defeating cryptographic security.
- Coding theory: The science of converting the representation of data.
- Cryptology $=$ Cryptography $+$ Cryptanalysis $\pm$ Coding theory.
- $$\Big(\text{Cryptology}\Big) \sim \Big(\begin{array}{l} (\text{Logic}) \wedge \\ (\text{Mathematics}) \wedge \\ (\text{Computer science}) \wedge \\ (\text{Computer engineering}) \wedge \\ (\text{Electrical \& Electronics engineering}) \end{array} \Big).$$


Alice
Hello,
How are
you doing
today?
Plaintext
Hello,
How are
you doing
today?
Bob
Hello,
How are
you doing
today?
Public Channel
Hello,
How are
you doing
today?


Hello,
How are
you doing
today?
Plaintext
Hello,
How are
you doing
today?
Alice
Bob
Hello,
How are
you doing
today?
Public Channel
Hello,
How are
you doing
today?
Eve


Hello,
How are
you doing
today?
Plaintext
Hello,
How are
you doing
today?
Alice
Bob
Secret
Encryption
Algorithm
Secret
Decryption
Algorithm
>ï¿½$+dsb-2
d78203]e
1621xd*qw
h8v6.82x{2
Public Channel
>ï¿½$+dsb-2
d78203]e
1621xd*qw
h8v6.82x{2
Eve

The Archaic Ciphers - Selected Examples

- Ancient Greeks and Romans
  - 475 B.C. Spartans - Scytale Cipher.
  - 60 B.C. Julius Caesar - Substitution Cipher.

- Middle Ages
  - 1378 - 1417 Gabriele de Lavinde of Parma
- Renaissance
  - 1518 Johannes Trithemius: “Polygraphiae” (Steganographia), first printed work!
- 20th Century
  - 1917 Zimmermann Telegramme (codebooks)
  - 1926 Vernam, “one-time-pad”
  - 1939 - 1945 2nd World War: “Enigma” - “Purple”

Zimmermann Telegramme

WESTERN UNION TELEGRAM

NEWCOMB CARLTON, PRESIDENT

Send the following telegram, subject to the terms on back hereof, which are hereby agreed to

via Galveston

JAN 19 1917

GERMAN LEGATION

MEXICO CITY

130	13042	13401	8501	115	3528	416	17214	6491	11310
18147	18222	21560	10247	11518	23677	13605	3494	14936	
98092	5905	11311	10392	10371	0302	21290	5161	39695	
23571	17504	11269	18276	18101	0317	0228	17694	4473	
23284	22200	19452	21589	67893	5569	13918	8958	12137	
1333	4725	4458	5905	17166	13851	4458	17149	14471	6706
13850	12224	6929	14991	7382	15857	67893	14218	36477	
5870	17553	67893	5870	5454	16102	15217	22801	17138	
21001	17388	7446	23638	18222	6719	14331	15021	23845	
3156	23552	22096	21604	4797	9497	22464	20855	4377	
23610	18140	22260	5905	13347	20420	39689	13732	20667	
6929	5275	18507	52262	1340	22049	13339	11265	22295	
10439	14814	4178	6992	8784	7632	7357	6926	52262	11267
21100	21272	9346	9559	22464	15874	18502	18500	15857	
2188	5376	7381	98092	16127	13486	9350	9220	76036	14219
5144	2831	17920	11347	17142	11264	7667	7762	15099	9110
10482	97556	3569	3670						

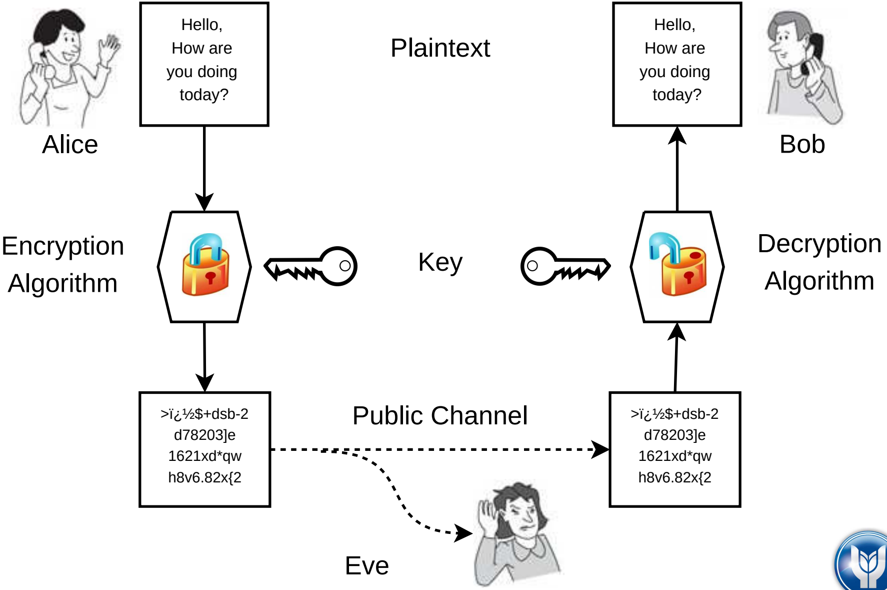
BEPNSTOPFF.

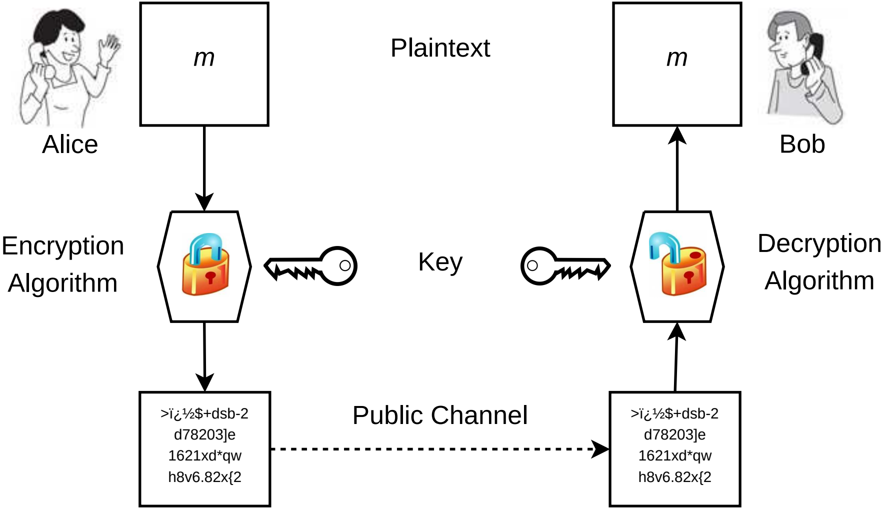
Charge German Embassy.

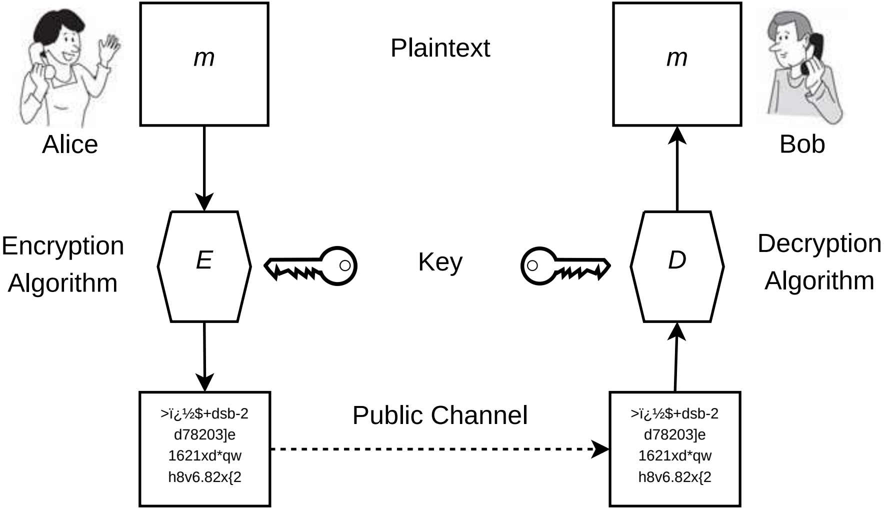
TELEGRAM RECEIVED.

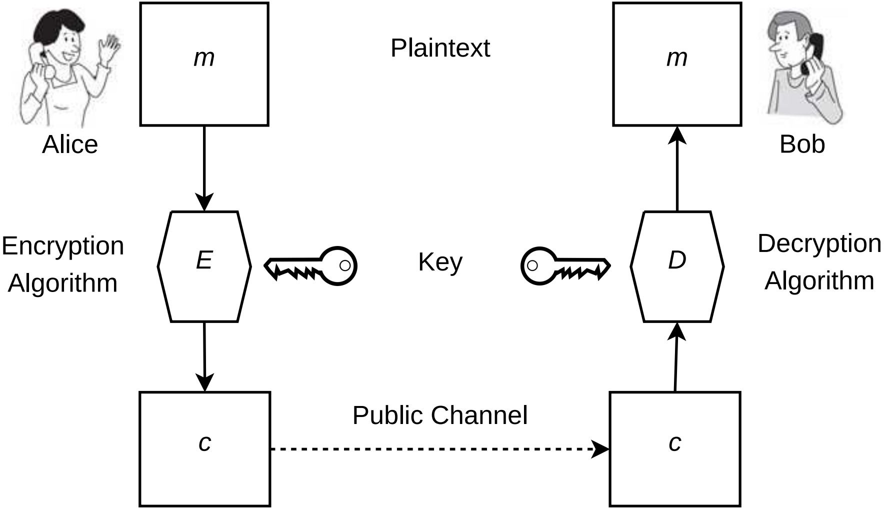
FROM 2nd from London # 5747.

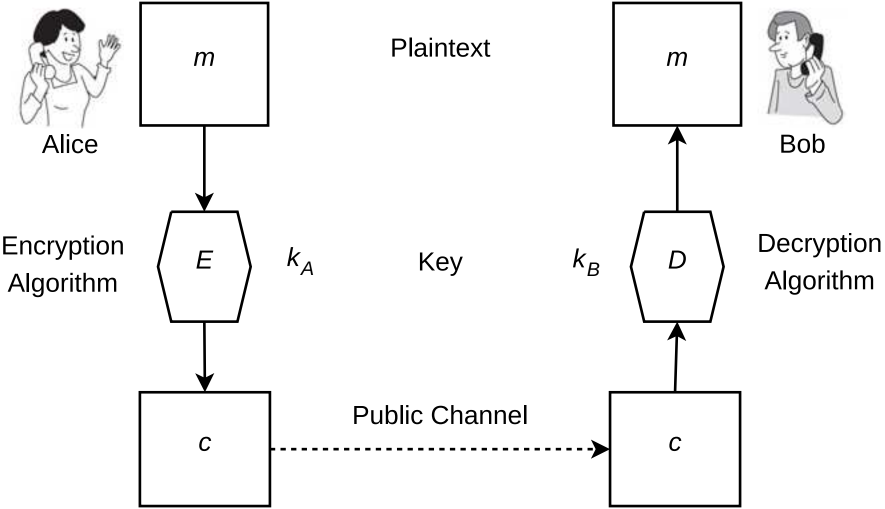
"We intend to begin on the first of February unrestricted submarine warfare. We shall endeavor in spite of this to keep the United States of America neutral. In the event of this not succeeding, we make Mexico a proposal of alliance on the following basis: make war together, make peace together, generous financial support and an understanding on our part that Mexico is to reconquer the lost territory in Texas, New Mexico, and Arizona. The settlement in detail is left to you. You will inform the President of the above most secretly as soon as the outbreak of war with the United States of America is certain and add the suggestion that he should, on his own initiative, invite Japan to immediate adherence and at the same time mediate between Japan and ourselves. Please call the President's attention to the fact that the ruthless employment of our submarines now offers the prospect of compelling England in a few months to make peace." Signed, ZIMMERMANN.


Hello,
How are
you doing
today?
Plaintext
Hello,
How are
you doing
today?
Alice
Bob
Encryption
Algorithm
Key
Decryption
Algorithm
>ï¿½$+dsb-2
d78203]e
1621xd*qw
h8v6.82x{2
Public Channel
>ï¿½$+dsb-2
d78203]e
1621xd*qw
h8v6.82x{2
Eve

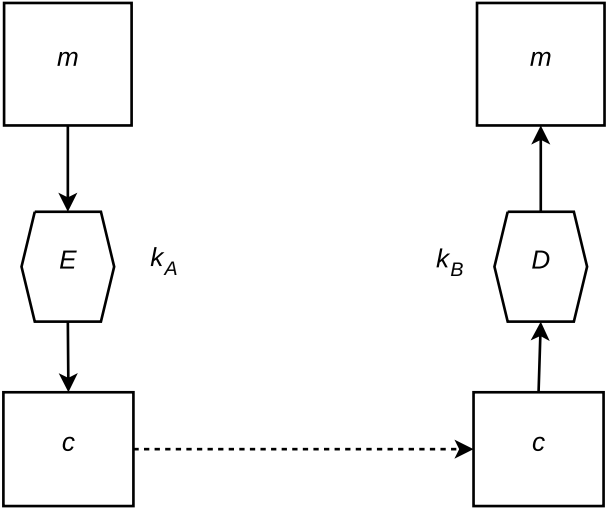

Alice
m
Plaintext
m
Bob
Encryption Algorithm
Key
Decryption Algorithm
>ï¿½$+dsb-2 d78203]e 1621xd*qw h8v6.82x{2
Public Channel
>ï¿½$+dsb-2 d78203]e 1621xd*qw h8v6.82x{2

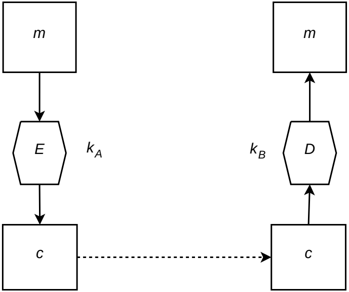

m
Plaintext
m
Alice
Bob
Encryption
Algorithm
E
Key
D
Decryption
Algorithm
>ï¿½$+dsb-2
d78203]e
1621xd*qw
h8v6.82x{2
Public Channel
>ï¿½$+dsb-2
d78203]e
1621xd*qw
h8v6.82x{2


Plaintext
m
m
Alice
Bob
Encryption Algorithm
E
Key
D
Decryption Algorithm
c
Public Channel
c

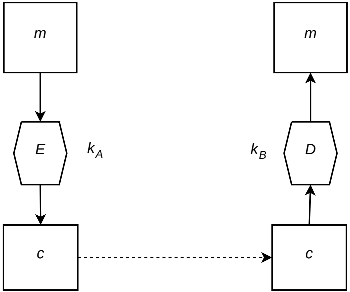


Alice
m
Plaintext
m
Bob
Encryption Algorithm
E
k_A
Key
k_B
D
Decryption Algorithm
c
Public Channel
c

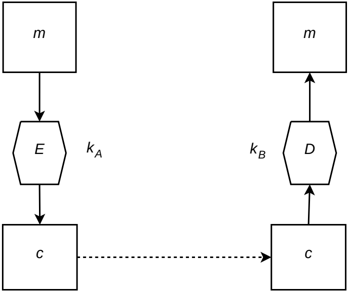

m
m
E
k_A
k_B
D
c
c



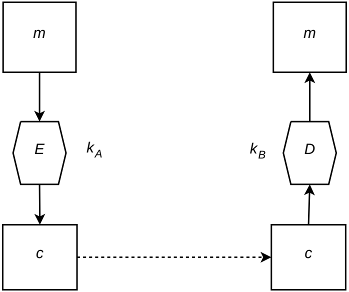
- Obtaing the ciphertext: $E(m, k_A) = c$.
- Recovering the plaintext: $D(c, k_B) = m$.



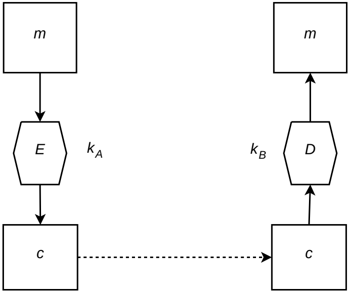
- Obtaing the ciphertext: $E(m, k_A) = c$.
- Recovering the plaintext: $D(c, k_B) = m$.
- Symmetric Key Cryptography: $k_A = k_B$.



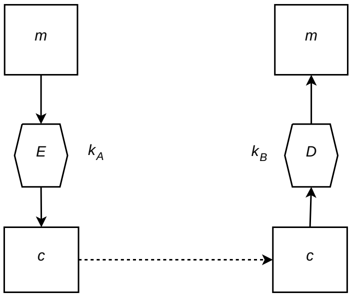
- Obtaing the ciphertext: $E(m, k_A) = c$.
- Recovering the plaintext: $D(c, k_B) = m$.
- Symmetric Key Cryptography: $k_A = k_B$. (Shared / Secret Key Cry.)



- Obtaing the ciphertext: $E(m, k_A) = c$.
- Recovering the plaintext: $D(c, k_B) = m$.
- Symmetric Key Cryptography: $k_A = k_B$. (Shared / Secret Key Cry.)
- Asymmetric Key Cryptography: $k_A \neq k_B$.



- Obtaing the ciphertext: $E(m, k_A) = c$.
- Recovering the plaintext: $D(c, k_B) = m$.
- Symmetric Key Cryptography: $k_A = k_B$. (Shared / Secret Key Cry.)
- Asymmetric Key Cryptography: $k_A \neq k_B$. (Public Key Cry.)



- Obtaing the ciphertext: $E(m, k_A) = c$.
- Recovering the plaintext: $D(c, k_B) = m$.
- Symmetric Key Cryptography: $k_A = k_B$. (Shared / Secret Key Cry.)
- Asymmetric Key Cryptography: $k_A \neq k_B$. (Public Key Cry.)
- Cryptanalysis: Given c, E, D, find m.

Brute force attacks

Try all possible keys!

- 1-bit key: You have $2^1 = 2$ keys; one or the other
- 2-bit key: You have $2^2 = 4$ keys: one of the four
- 3-bit key: You have $2^3 = 8$ keys: one of the eight
- 4-bit key: You have $2^4 = 16$ keys: one of the sixteen
- ...
- 256-bit key: You have $2^{256} =$

  $$115792089237316195423570985008687907853269984665640564039457584007913129639936$$

  $$\approx 3671743063080802746815416825491118336290905145409708398004109 \cdot 365 \cdot 24 \cdot 60 \cdot 60 \cdot 10^9$$

  keys!!!

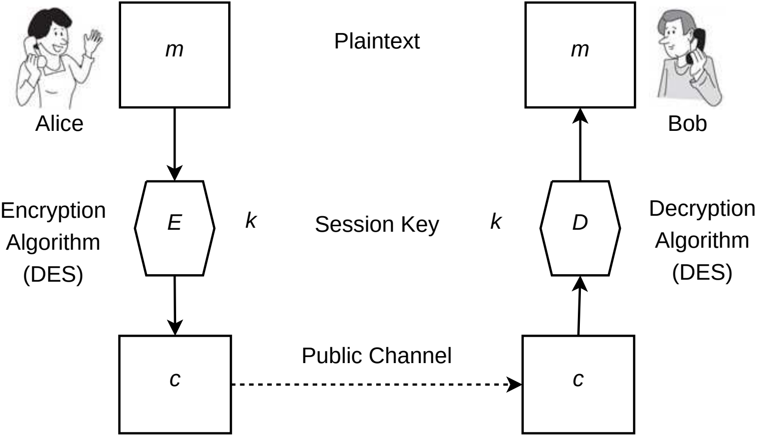
Contemporary Ciphers: Early Years

- 1971 IBM announces Lucifer, A Block cipher
- 1975 IBM offers Lucifer as a standard
- 1976 Diffie & Hellman, Public Key concept
- 1977 Lucifer gets approved by NIST as Data Encryption Standard (DES), a Block Cipher
- 1978 Rivest-Shamir-Adleman (RSA), Public Key Cryptosystem
- 1984 Shamir, Identity Based Cryptography
- 1985 Elliptic Curve Cryptograpgy (ECC)
- 1987 Stream cipher RC4
- 2001 Advanced Encryption Standard (AES)
- 2001 Boneh & Franklin, Identity Based Cryptography is feasible!

A Basic Taxonomy

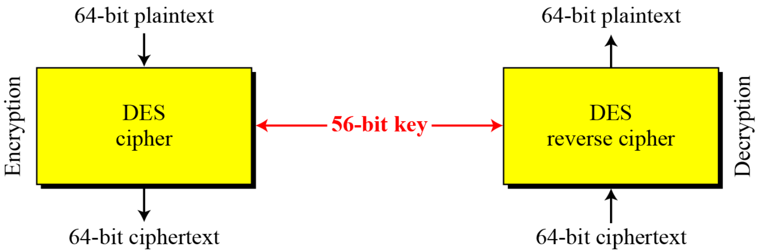
- Symmetric systems
  - Block ciphers: DES, 3DES, IDEA, BLOWFISH, TWOFISH, AES, ...
  - Stream ciphers: RC4, Dragon, HC-256, MICKEY, MOUSTIQUE, ...
- Asymmetric systems:
  - Key exchange: DH
  - Encryption/Decryption: RSA, ELGAMAL, ECC, NTRU
  - Digital Signature: DSA, ECDSA

Data Encryption Standard (DES)

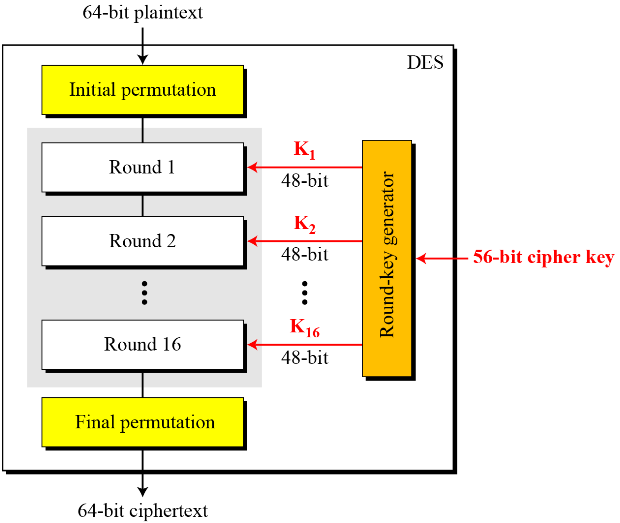


- Obtaing the ciphertext: $E(m,k) = c$.
- Recovering the plaintext: $D(c,k) = m$.

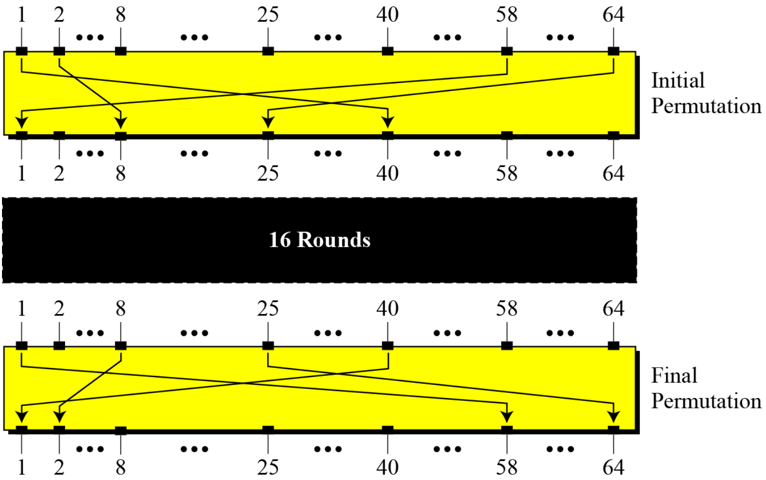
Data Encryption Standard (DES)



Data Encryption Standard (DES) / Rounds Overview

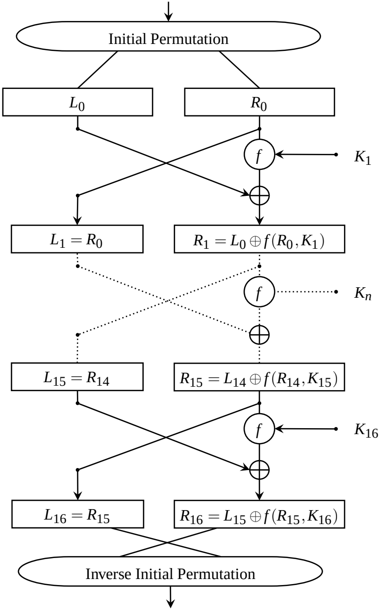


Data Encryption Standard (DES) / Initial & Final Permutation



Data Encryption Standard (DES) / Initial & Final Permutation

Initial Permutation								*Final Permutation*							
58	50	42	34	26	18	10	02	40	08	48	16	56	24	64	32
60	52	44	36	28	20	12	04	39	07	47	15	55	23	63	31
62	54	46	38	30	22	14	06	38	06	46	14	54	22	62	30
64	56	48	40	32	24	16	08	37	05	45	13	53	21	61	29
57	49	41	33	25	17	09	01	36	04	44	12	52	20	60	28
59	51	43	35	27	19	11	03	35	03	43	11	51	19	59	27
61	53	45	37	29	21	13	05	34	02	42	10	50	18	58	26
63	55	47	39	31	23	15	07	33	01	41	09	49	17	57	25



We have

- $L_j = R_{j-1}$,
- $R_j = L_{j-1} \oplus f(R_{j-1}, k_j)$.

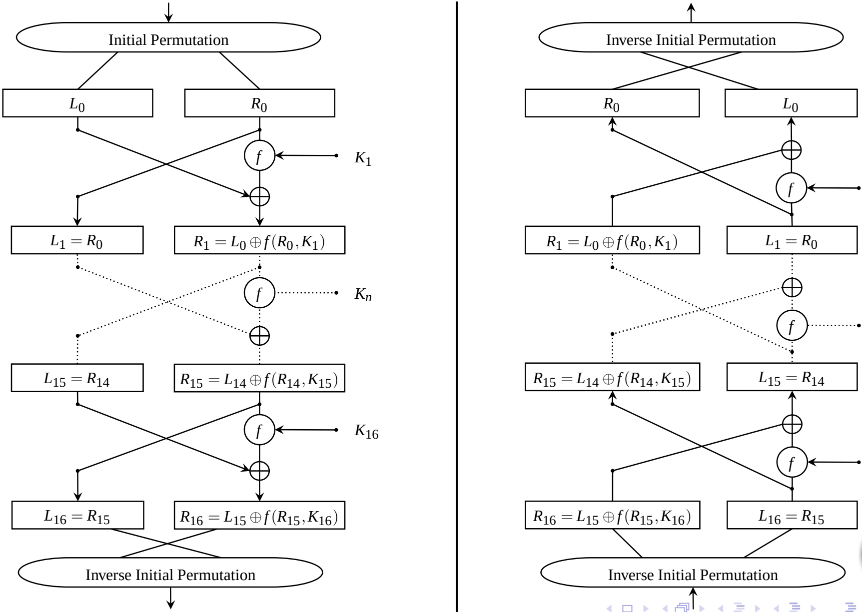
We can rewrite in the form

- $R_{j-1} = L_j$,
- $L_{j-1} = R_j \oplus f(R_{j-1}, k_j)$.

By substitution

- $L_{j-1} = R_j \oplus f(L_j, k_j)$.

Data Encryption Standard (DES) / Encryption & Decryption



Data Encryption Standard (DES) / The function $f(R_{i-1}, K_i)$

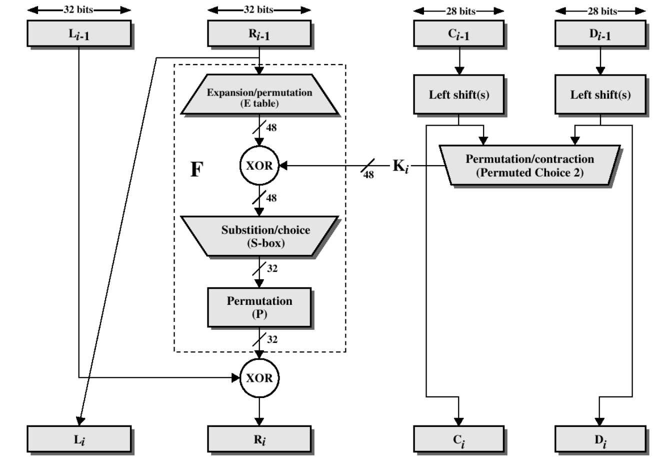


Figure 2.4 Single Round of DES Algorithm

EXPANSION PERMUTATION (32 -> 48): 32, 1, 2, 3, 4, 5, 4, 5, 6, 7, 8, 9, 8, 9, 10, 11, 12, 13, 12, 13, 14, 15, 16, 17, 16, 17, 18, 19, 20, 21, 20, 21, 22, 23, 24 25, 24, 25 26 27, 28, 29, 28 29, 30, 31, 32, 1.

P-BOX PERMUTATION (56 -> 48): 16, 7, 20, 21, 29, 12, 28, 17, 1, 15, 23, 26, 5, 18, 31, 10, 2, 8, 24, 14, 32, 27, 3, 9, 19, 13, 30, 6, 22, 11, 4, 25.

S-BOX-1 to S-BOX-8 (6 -> 4): The fist two bits determine the row; the next four bits determines the column.

14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12, 5, 9, 0, 7,
0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11, 9, 5, 3, 8,
4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3, 10, 5, 0,
15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10, 0, 6, 13.

15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12, 0, 5, 10,
3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6, 9, 11, 5,
0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9, 3, 12, 15,
13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0, 5, 14, 9.

10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7, 11, 4, 2, 8,
13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12, 11, 15, 1,
13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5, 10, 14, 7,
1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11, 5, 2, 12.

7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15,
13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9,
10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4,
3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14.

2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13, 0, 14, 9,
14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10, 3, 9, 8, 6,
4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6, 3, 0, 14,
11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9, 10, 4, 5, 3.

12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14, 7, 5, 11,
10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0, 11, 3, 8,
9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1, 13, 11, 6,
4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6, 0, 8, 13.

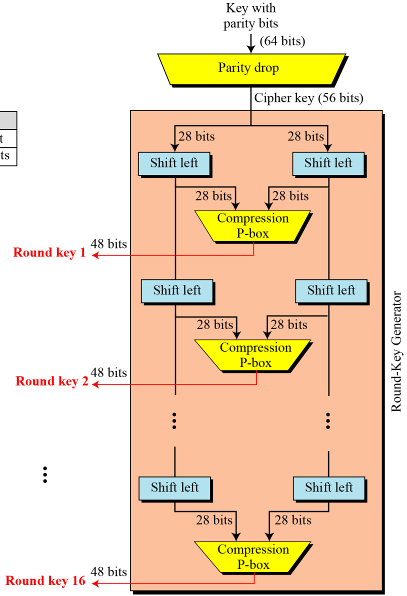
4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5, 10, 6, 1,
13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2, 15, 8, 6,
1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8, 0, 5, 9, 2,
6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14, 2, 3, 12.

13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5, 0, 12, 7,
1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0, 14, 9, 2,
7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13, 15, 3, 5, 8,
2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3, 5, 6, 11.

Data Encryption Standard (DES) / Key Scheduling

Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits



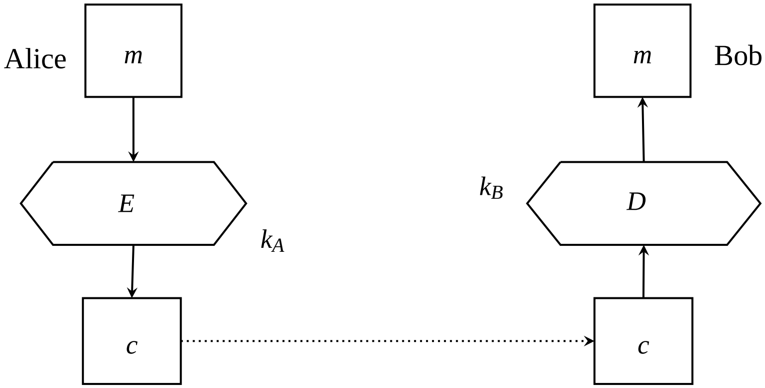
KEY PERMUTATION (64 -> 56): 57, 49, 41, 33, 25, 17, 9, 1, 58, 50, 42, 34, 26, 18, 10, 2, 59, 51, 43 35 27 19, 11, 3, 60, 52, 44, 36, 63, 55, 47 39, 31, 23, 15, 7, 62, 54, 46, 38, 30, 22, 1, 6, 61, 53, 45, 37, 29, 21, 13, 5, 28, 20, 12, 4.

KEY SHIFTS PER ROUND

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
# of shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

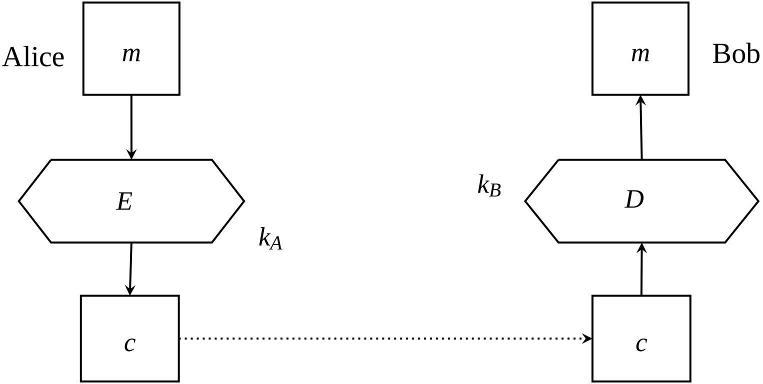
COMPRESSION PERMUTATION (56 -> 48): 14, 17, 11, 24, 1, 5, 3, 28, 15, 6, 21, 10, 23, 19, 12, 4, 26, 8, 16, 7, 27, 20, 13, 2, 41, 52, 31 37, 47, 55, 30, 40, 51, 45, 33, 48, 44, 49, 39 56, 34, 53, 46, 42, 50, 36, 29, 32.

The RSA algorithm



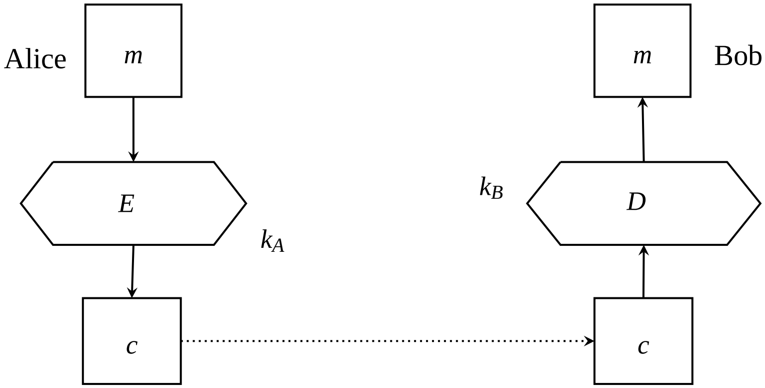
The RSA algorithm

1. Bob chooses primes p and q. Computes $n = p \cdot q$.



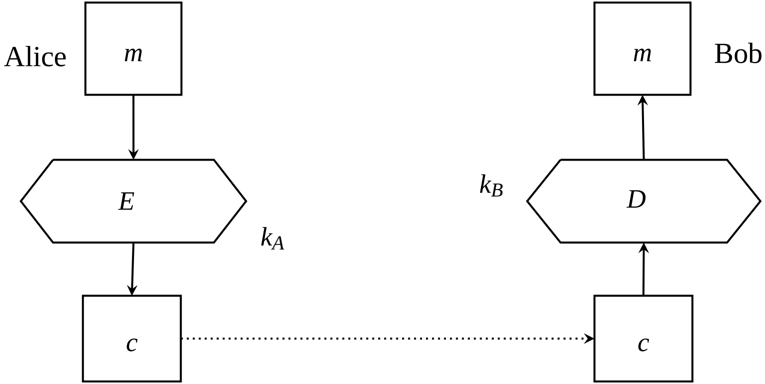
The RSA algorithm

1. Bob chooses primes p and q. Computes $n = p \cdot q$.
2. Bob chooses e with $\mathrm{GCD}\Big(e, (p-1)\cdot(q-1)\Big) = 1$.



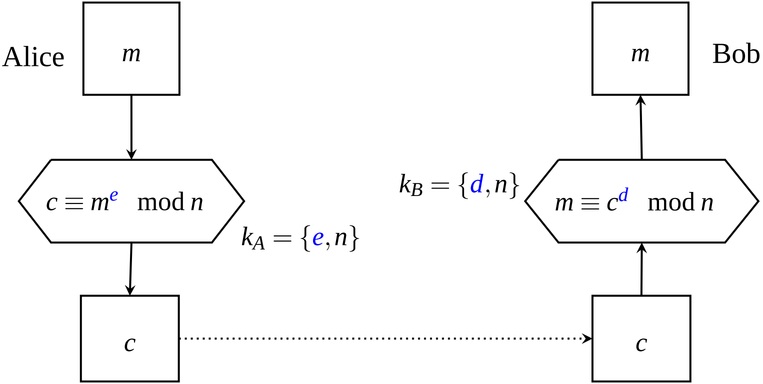
The RSA algorithm

1. Bob chooses primes p and q. Computes $n = p \cdot q$.
2. Bob chooses e with $\mathrm{GCD}\Big(e, (p-1)\cdot(q-1)\Big) = 1$.
3. Bob computes d with $d \cdot e \equiv 1 \mod (p-1)\cdot(q-1)$.



The RSA algorithm

1. Bob chooses primes p and q. Computes $n = p \cdot q$.
2. Bob chooses e with $\mathrm{GCD}\Big(e, (p-1)\cdot(q-1)\Big) = 1$.
3. Bob computes d with $d \cdot e \equiv 1 \mod (p-1)\cdot(q-1)$.



RSA in action

Alice wants to send a message m to Bob:

Let m = "**`Hello`**" = **`0x48656C6C6F`** = 310939249775.

RSA in action

Alice wants to send a message m to Bob:

$$\text{Let } m = \text{“}\mathtt{Hello}\text{”} = \mathtt{0x48656C6C6F} = 310939249775.$$

1. Bob chooses primes $p = 1048583$, $q = 2097211$ and computes $n = p \cdot q = 2199099802013$.

RSA in action

Alice wants to send a message m to Bob:

$$\text{Let } m = \text{“}\texttt{Hello}\text{”} = \texttt{0x48656C6C6F} = 310939249775.$$

1. Bob chooses primes $p = 1048583$, $q = 2097211$ and computes $n = p \cdot q = 2199099802013$.
2. Bob chooses $e = 1644903229909$ with $\mathrm{GCD}\big(e, (p-1)\cdot(q-1)\big) = 1$.

RSA in action

Alice wants to send a message m to Bob:

$$\text{Let } m = \text{“}\texttt{Hello}\text{”} = \texttt{0x48656C6C6F} = 310939249775.$$

1. Bob chooses primes $p = 1048583$, $q = 2097211$ and computes $n = p \cdot q = 2199099802013$.
2. Bob chooses $e = 1644903229909$ with $\mathrm{GCD}\Big(e, (p-1)\cdot(q-1)\Big) = 1$.
3. Bob computes $d = 2055797390629$ with $d \cdot e \equiv 1 \mod (p-1)\cdot(q-1)$.

RSA in action

Alice wants to send a message m to Bob:

$$\text{Let } m = \text{“}\mathtt{\textbf{Hello}}\text{”} = \mathtt{\textbf{0x48656C6C6F}} = 310939249775.$$

1. Bob chooses primes $p = 1048583$, $q = 2097211$ and computes $n = p \cdot q = 2199099802013$.
2. Bob chooses $e = 1644903229909$ with $\mathrm{GCD}\Big(e, (p-1)\cdot(q-1)\Big) = 1$.
3. Bob computes $d = 2055797390629$ with $d \cdot e \equiv 1 \mod (p-1)\cdot(q-1)$.

Alice gets $\{e, n\}$ from Bob, computes & sends the ciphertext c:

$$\begin{aligned} c \equiv m^e &\equiv 310939249775^{1644903229909} \qquad \Big(\mathrm{mod}\ 2199099802013\Big). \\ &\equiv 858640968629 \quad (= \text{“}\mathtt{\textbf{Çéık\&}}\text{”}) \end{aligned}$$

RSA in action

Alice wants to send a message m to Bob:

$$\text{Let } m = \text{“}\texttt{Hello}\text{”} = \texttt{0x48656C6C6F} = 310939249775.$$

1. Bob chooses primes $p = 1048583$, $q = 2097211$ and computes $n = p \cdot q = 2199099802013$.
2. Bob chooses $e = 1644903229909$ with $\mathrm{GCD}\big(e, (p-1)\cdot(q-1)\big) = 1$.
3. Bob computes $d = 2055797390629$ with $d \cdot e \equiv 1 \mod (p-1)\cdot(q-1)$.

Alice gets $\{e, n\}$ from Bob, computes & sends the ciphertext c:

$$\begin{aligned} c \equiv m^e &\equiv 310939249775^{1644903229909} \\ &\equiv 858640968629 \quad (= \text{“}\texttt{Çéık\&}\text{”}) \end{aligned} \quad \big(\mathrm{mod}\ 2199099802013\big).$$

Bob receives the ciphertext c and decrypt it using {d, n}:

$$\begin{aligned} m \equiv c^d &\equiv 858640968629^{2055797390629} \\ &\equiv 310939249775 \quad (= \text{“}\texttt{Hello}\text{”}) \end{aligned} \quad \big(\mathrm{mod}\ 2199099802013\big).$$

How does the RSA decryption works?

Definition (Euler's totient function)

Let n be an integer.

$$\phi(n) := \text{"The number of integers } 1 \leq a \leq n \text{ such that } \mathrm{GCD}(a,n) = 1\text{"}.$$

Lemma

$\phi(n) = \phi(p \cdot q) = (p-1) \cdot (q-1)$.

Theorem

If $GCD(a,n) = 1$ then $a^{\phi(n)} \equiv 1 \mod n$.

Now,

$$c^d \equiv (m^e)^d \equiv m^{1+k\cdot\phi(n)} \equiv m \cdot (m^{\phi(n)})^k \equiv m \cdot 1^k \equiv m \mod n.$$

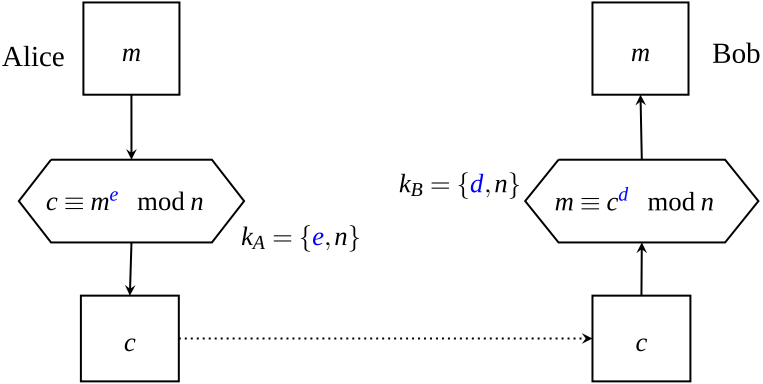
A 1024-bit RSA Key Pair

- $\{e, n\}$ is Bob's public key.
- $\{d, n\}$ is Bob's private key.
- A 1024-bit real life example for $\{e, n\}$ and $\{d, n\}$:

```
-----BEGIN RSA PUBLIC KEY-----
98903585440747599384191320255954189656318148122089021285657780860309099864821 41\
61076066778910536737051038839999777729455374045174487243350037733416639711850 53\
39293845960769712410984156894969469738678553933337641721313425918180516603240 62\
4522290105265565886483476797092062038811264788746288467833203265965221 9,
10438979421523677490638252295322065521654535721060522931311413897271600366022 68\
08705772017491398949753817944988638218003392833393273918097591973229650906151 49\
20362836849521069991467875040592817931791644012871146435291241331010484648733 53\
3791438145557822003985410337672074315914945733262496182265372296273437 77
-----END RSA PUBLIC KEY-----

-----BEGIN RSA PRIVATE KEY-----
60896885011141551638324626835139206081818916976938394877942378438363254397688 48\
96604426412160960361021198228627940644422432475043851974209073046926271643191 54\
62555051230485641077819927134910694147560629919427454813253574609201185666958 87\
6224525091700085797266395012286691829822826276550454575385878946349844 4,
10438979421523677490638252295322065521654535721060522931311413897271600366022 68\
08705772017491398949753817944988638218003392833393273918097591973229650906151 49\
20362836849521069991467875040592817931791644012871146435291241331010484648733 53\
3791438145557822003985410337672074315914945733262496182265372296273437 77
-----END RSA PRIVATE KEY-----
```

Threats against RSA

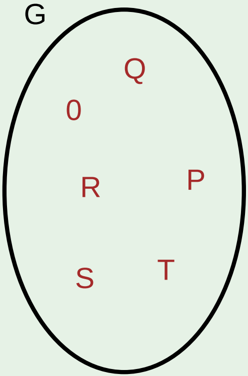


What can Eve do?

- Eve can intercept n, e, c.
- Eve does not know p, q, d.
- Eve cannot factor n. (assumption)

Group

A set G



A binary operation on G

- Q + R = T
- S + P = R
- 0 + S = S (identity)
- 0 + 0 = 0
- P + T = 0 (inverse)

Group

Definition

A group is a pair $(G,+)$ consisting of a nonempty set G and a binary operation $+$, (closed) on G, such that $(\forall\, P,Q,R \in G)$

- Binary operation is associative; $(P+Q)+R = P+(Q+R)$,
- A unique identity exists; $0+P = P+0 = P$,
- Every element has a unique inverse; $P+Q = Q+P = 0$.

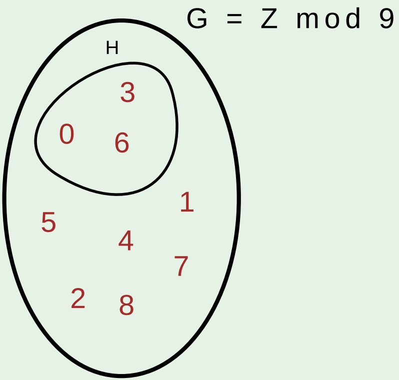
Furthermore, $(G,+)$ is abelian if $P+Q = Q+P \quad \forall\, P,Q \in G$.

Examples

- $\mathbb{Z}/p\mathbb{Z}$ is an abelian group. (Simply "mod p" arithmetic)
- An elliptic curve is a group. (We will define this later)

Subgroup

H is a subset of G



$H \subset G$

+	**0**	**3**	**6**
0	0	3	6
3	3	6	0
6	6	0	3

Check

- Closed
- Identity
- Inverses
- Associativity

Subgroup

Definition

A subset H of a group G which is

- closed under the binary operation of G,
- a group itself,

is called a subgroup of G. $(H \subseteq G)$

Cyclic (Sub)group, Generator

Definition

Let $P \in G$, then

$$H = \left\{ nP = \underbrace{P + P + \ldots + P}_{n \text{ times}} \mid n \in \mathbb{Z} \right\}$$

is the cyclic subgroup of G generated by P. $\quad \left(H = \langle P \rangle \right)$

Cyclic (Sub)group, Generator

Definition

Let $P \in G$, then

$$H = \left\{ nP = \underbrace{P + P + \ldots + P}_{n \text{ times}} \mid n \in \mathbb{Z} \right\}$$

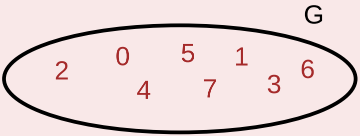
is the cyclic subgroup of G generated by P. $\quad \big(H = \langle P \rangle\big)$

Remark

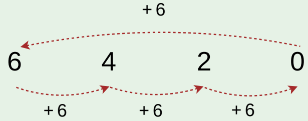
- If an element $P \in G$ generates G, then P is a generator for G. $\big(G = \langle P \rangle\big)$
- G is a cyclic group if there is some element $P \in G$ that generates G.
- The number of elements in $\langle P \rangle$ is called the order of P and is denoted by $|\langle P \rangle|$.

Cyclic (Sub)group, Generator

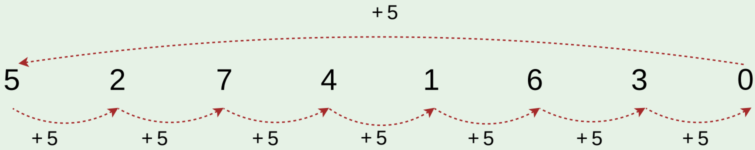
Consider integers modulo 8.



6 is not a generator for G.



5 is a generator for G. $G = \langle 5 \rangle$



Discrete Logarithm Problem

- Let $(G,+)$ be a cyclic group of order n and let P be a generator of G.
- Given $Q \in G$ find the unique k such that $0 \leq k \leq n-1$ and $Q = kP$.
- Finding k is called Discrete Logarithm Problem (DLP).
- The complexity of DLP depends on the selection of the group G.

Note: If the group is written multiplicatively, the notation is changed to $Q = P^k$.

Diffie Hellman Key Exchange (DH)

1. Either Alice or Bob picks a prime p and a generator α.
2. Alice makes makes p and α public.

Diffie Hellman Key Exchange (DH)

1. Either Alice or Bob picks a prime p and a generator α.
2. Alice makes makes p and α public.
3. Alice chooses a secret random $1 \leq x \leq p-2$.
4. Bob chooses a secret random $1 \leq y \leq p-2$.

Diffie Hellman Key Exchange (DH)

1. Either Alice or Bob picks a prime p and a generator α.
2. Alice makes makes p and α public.
3. Alice chooses a secret random $1 \leq x \leq p-2$.
4. Bob chooses a secret random $1 \leq y \leq p-2$.
5. Alice sends $\alpha^x \mod p$ to Bob.
6. Bob sends $\alpha^y \mod p$ to Alice.

Diffie Hellman Key Exchange (DH)

1. Either Alice or Bob picks a prime p and a generator α.
2. Alice makes makes p and α public.
3. Alice chooses a secret random $1 \leq x \leq p-2$.
4. Bob chooses a secret random $1 \leq y \leq p-2$.
5. Alice sends $\alpha^x \mod p$ to Bob.
6. Bob sends $\alpha^y \mod p$ to Alice.
7. Alice calculates the shared secret as $K \equiv (\alpha^y)^x \mod p$.
8. Bob calculates the same shared secret as $K \equiv (\alpha^x)^y \mod p$.

Diffie Hellman Key Exchange (DH)

1. Either Alice or Bob picks a prime p and a generator α.
2. Alice makes makes p and α public.
3. Alice chooses a secret random $1 \leq x \leq p-2$.
4. Bob chooses a secret random $1 \leq y \leq p-2$.
5. Alice sends $\alpha^x \mod p$ to Bob.
6. Bob sends $\alpha^y \mod p$ to Alice.
7. Alice calculates the shared secret as $K \equiv (\alpha^y)^x \mod p$.
8. Bob calculates the same shared secret as $K \equiv (\alpha^x)^y \mod p$.

- Though Eve may know p, α, $\alpha^x \mod p$ and $\alpha^y \mod p$,
- She cannot recover K
- Unless she solves the DLP and finds out either x or y.

DH in action

1. Alice picks a prime $p = 558494556463$ and a generator $\alpha = 197214177966$.

DH in action

1. Alice picks a prime $p = 558494556463$ and a generator $\alpha = 197214177966$.
2. Alice makes makes p and α public.

DH in action

1. Alice picks a prime $p = 558494556463$ and a generator $\alpha = 197214177966$.
2. Alice makes makes p and α public.
3. Alice chooses a secret random $x = 282910484039$ with $1 \leq x \leq p - 2$.

DH in action

1. Alice picks a prime $p = 558494556463$ and a generator $\alpha = 197214177966$.
2. Alice makes makes p and α public.
3. Alice chooses a secret random $x = 282910484039$ with $1 \leq x \leq p-2$.
4. Bob chooses a secret random $y = 306801011233$ with $1 \leq y \leq p-2$.

DH in action

1. Alice picks a prime $p = 558494556463$ and a generator $\alpha = 197214177966$.
2. Alice makes makes p and α public.
3. Alice chooses a secret random $x = 282910484039$ with $1 \leq x \leq p-2$.
4. Bob chooses a secret random $y = 306801011233$ with $1 \leq y \leq p-2$.
5. Alice sends $\alpha^x \equiv 197214177966^{282910484039} \equiv 542167786127 \mod p$.

DH in action

1. Alice picks a prime $p = 558494556463$ and a generator $\alpha = 197214177966$.
2. Alice makes makes p and α public.
3. Alice chooses a secret random $x = 282910484039$ with $1 \leq x \leq p-2$.
4. Bob chooses a secret random $y = 306801011233$ with $1 \leq y \leq p-2$.
5. Alice sends $\alpha^x \equiv 197214177966^{282910484039} \equiv 542167786127 \mod p$.
6. Bob sends $\alpha^y \equiv 197214177966^{306801011233} \equiv 416704295064 \mod p$.

DH in action

1. Alice picks a prime $p = 558494556463$ and a generator $\alpha = 197214177966$.
2. Alice makes makes p and α public.
3. Alice chooses a secret random $x = 282910484039$ with $1 \leq x \leq p-2$.
4. Bob chooses a secret random $y = 306801011233$ with $1 \leq y \leq p-2$.
5. Alice sends $\alpha^x \equiv 197214177966^{282910484039} \equiv 542167786127 \mod p$.
6. Bob sends $\alpha^y \equiv 197214177966^{306801011233} \equiv 416704295064 \mod p$.
7. Alice: $K \equiv (\alpha^y)^x \equiv 416704295064^{282910484039} \equiv 306801011233 \mod p$.

DH in action

1. Alice picks a prime $p = 558494556463$ and a generator $\alpha = 197214177966$.
2. Alice makes makes p and α public.
3. Alice chooses a secret random $x = 282910484039$ with $1 \leq x \leq p - 2$.
4. Bob chooses a secret random $y = 306801011233$ with $1 \leq y \leq p - 2$.
5. Alice sends $\alpha^x \equiv 197214177966^{282910484039} \equiv 542167786127 \mod p$.
6. Bob sends $\alpha^y \equiv 197214177966^{306801011233} \equiv 416704295064 \mod p$.
7. Alice: $K \equiv (\alpha^y)^x \equiv 416704295064^{282910484039} \equiv 306801011233 \mod p$.
8. Bob: $K \equiv (\alpha^x)^y \equiv 542167786127^{306801011233} \equiv 306801011233 \mod p$.

ElGamal

As usual Alice wants to send a message to Bob.

- Let $G = \langle P \rangle$ be a cyclic group.
- Bob's public key is $Q = kP$.
- Bob's private key is k.
- Plaintext is $M \in G$.

Alice performs:

ElGamal Encryption

input : Q, M.
output : $\{C_0, C_1\}$.

Select a random r, $0 < r < |\langle P \rangle|$.

Compute $C_0 = rP$.

Compute $C_1 = M + rQ$.

return $\{C_0, C_1\}$. (The ciphertext)

Bob performs:

ElGamal Decryption

input : k, $\{C_0, C_1\}$.
output : M.

Compute $M = C_1 - kC_0$.

return M.

ElGamal

How does the encryption works?

- We have the relation $Q = kP$.
- Encryption is $C_1 = (M + rQ), \quad C_0 = (rP)$.
- Decryption is $M = (C_1 - kC_0)$.
- So, decryption corresponds to

$$\begin{cases} C_1 - kC_0 = \\ (M + rQ) - k(rP) = \\ M + rQ - r(kP) = \\ M + rQ - rQ = \\ M \end{cases}$$

Elliptic Curves

Definition *(A simplified non-technical version)*

Let $p > 2$ be a prime. Let A, B be integers satisfying

$$0 \leq A < p, \qquad 0 \leq B < p, \qquad 4A^3 + 27B^3 \not\equiv 0 \mod p.$$

An elliptic curve is the set of points

$$E := \left\{ (x,y) \mid (0 \leq x < p) \text{ and } (0 \leq y < p) \text{ and } (y^2 \equiv x^3 + Ax + B \mod p) \right\}$$

together with a distinguished point $\mathcal{O}$ (the point at infinity).

- We have a set of points.
- Our goal is to form a group.
- All we need is a binary operation!

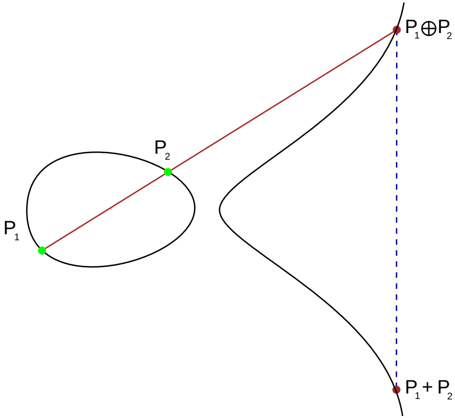
Bezout's Theorem *(A simplified non-technical version)*

Two curves of degree m and n intersect in mn points.

Remark

An elliptic curve and a line intersect at 3 points.

- We have a set of points.
- Our goal is to form a group.
- And the binary operation is:



With this binary operation;

- We select $\mathcal{O}$ as the identity element.
- The inverse of a point (x,y) is $(x,-y)$.

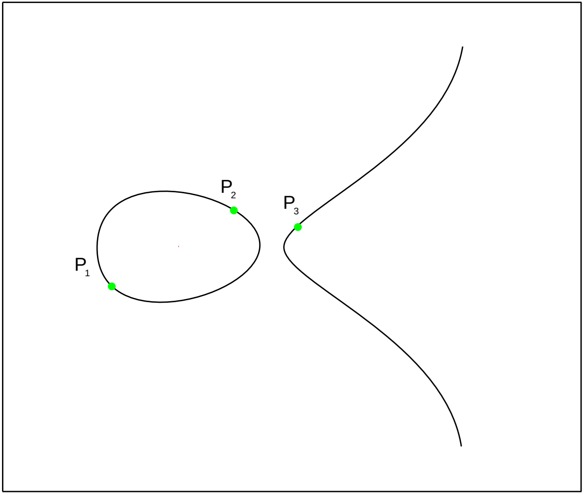
$$y^2 = x^3 + Ax + B$$

$$y = \pm\sqrt{x^3 + Ax + B}$$

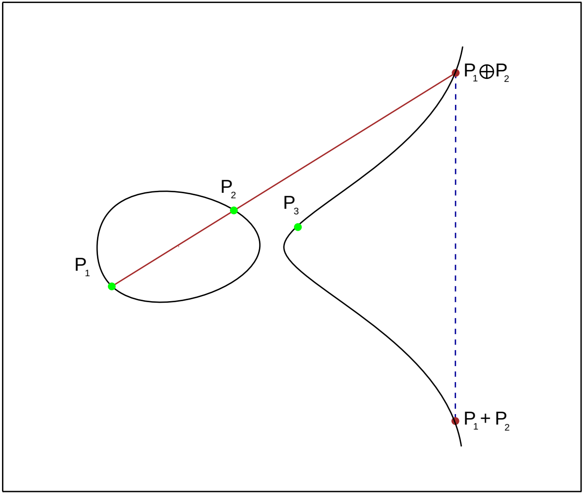
- The only axiom to check is the associativity, i.e.

$$(P_1 + P_2) + P_3 = P_1 + (P_2 + P_3).$$

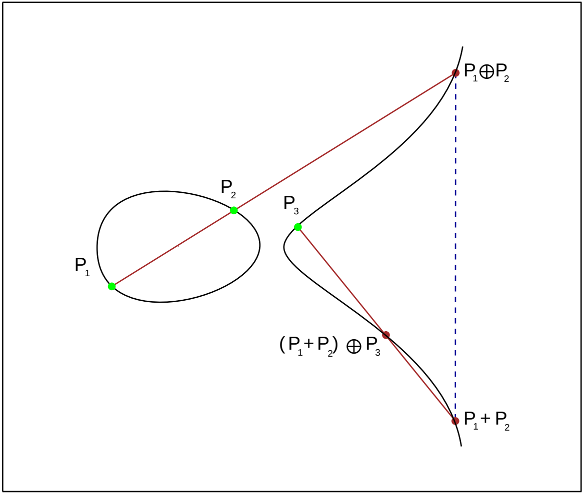
Elliptic Curves / The Group Law



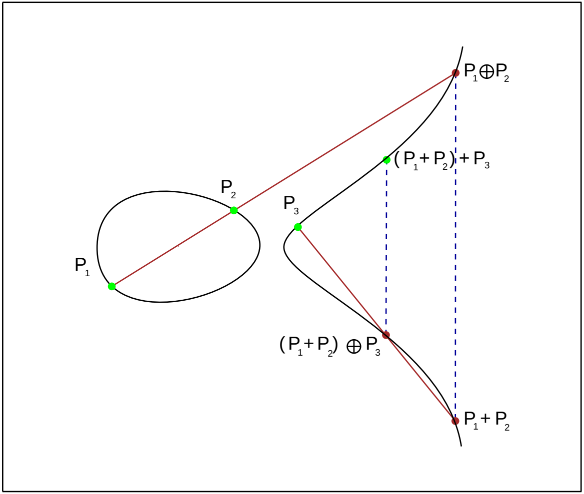
Elliptic Curves / The Group Law



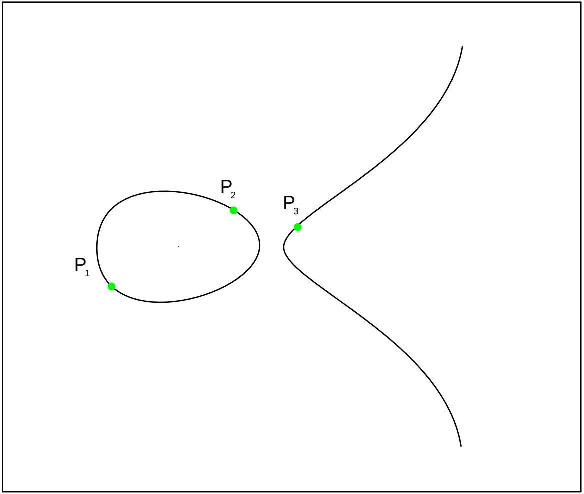
Elliptic Curves / The Group Law



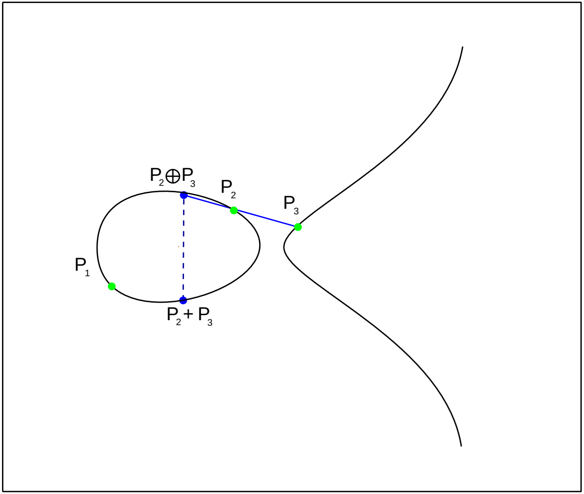
Elliptic Curves / The Group Law



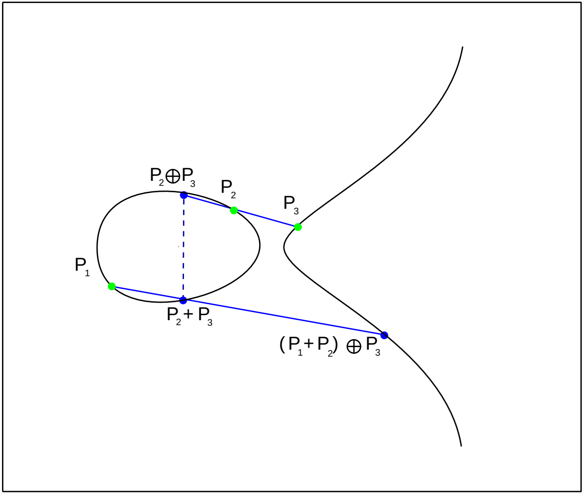
Elliptic Curves / The Group Law



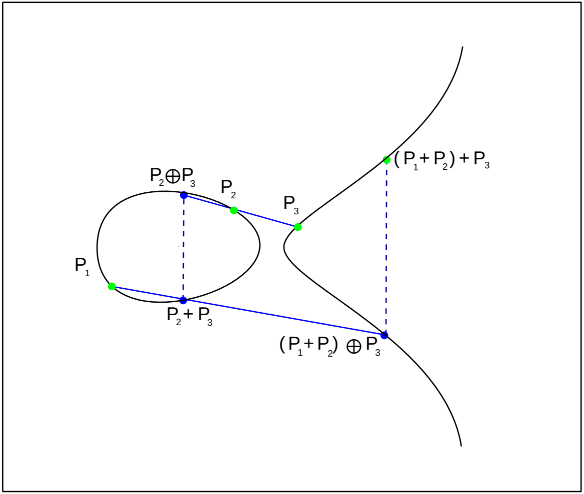
Elliptic Curves / The Group Law

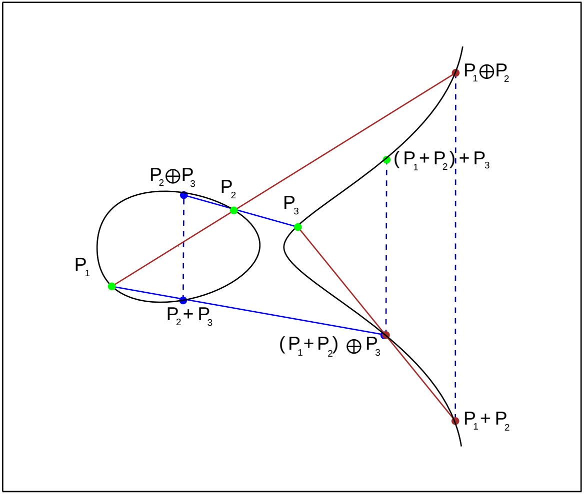


Elliptic Curves / The Group Law

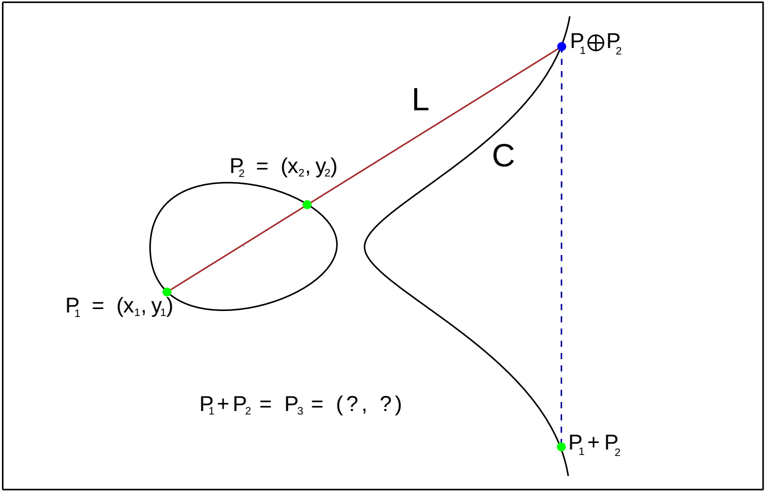


Elliptic Curves / The Group Law

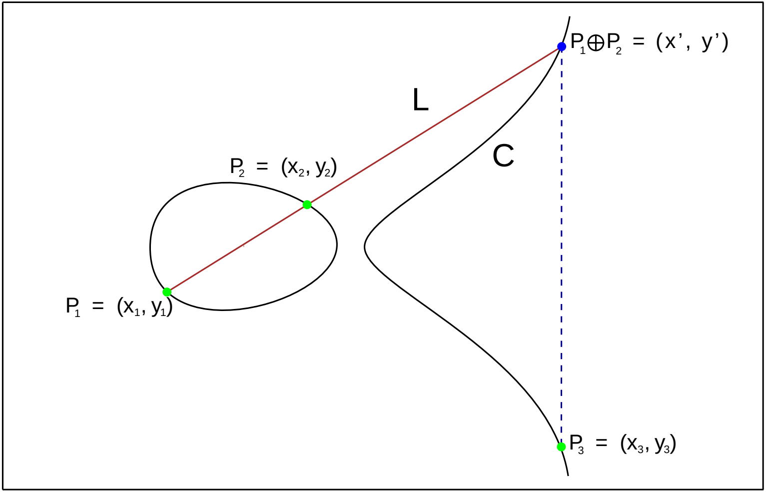



$P_1 \oplus P_2$
$(P_1+P_2)+P_3$
$P_2 \oplus P_3$
P_2
P_3
P_1
P_2+P_3
$(P_1+P_2) \oplus P_3$
P_1+P_2

Elliptic Curves / Explicit Point Addition Formulae ($P_1 \neq P_2$)



Elliptic Curves / Explicit Point Addition Formulae ($P_1 \neq P_2$)

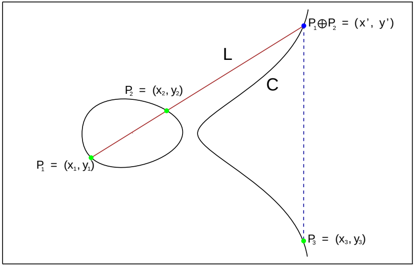


Elliptic Curves / Explicit Point Addition Formulae $(P_1 \neq P_2)$

$$\mathbf{L} : y = \lambda x + \beta$$

where

$$\lambda = (y_2 - y_1)/(x_2 - x_1)$$

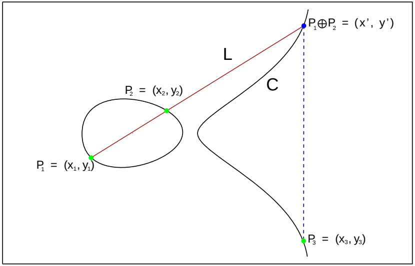


$$\mathbf{C} : y^2 = x^3 + Ax + B \quad \longrightarrow \quad (x^3 + Ax + B - y^2) = (x - x_1)(x - x_2)(x - x')$$

$$x^3 + Ax + B - (\lambda x + \beta)^2 = x^3 - (x_1 + x_2 + x')x^2 + (x_1x_2 + x_2x' + x'x_1)x - (x_1x_2x')$$

$$x^3 - \lambda^2 x^2 + (A - 2\lambda\beta)x + (B - \beta^2) = x^3 - (x_1 + x_2 + x')x^2 + (x_1x_2 + x_2x' + x'x_1)x - (x_1x_2x')$$

Elliptic Curves / Explicit Point Addition Formulae $(P_1 \neq P_2)$



$$\mathbf{L} : y = \lambda x + \beta$$

where

$$\lambda = (y_2 - y_1)/(x_2 - x_1)$$

$$\mathbf{C} : y^2 = x^3 + Ax + B \quad \longrightarrow \quad (x^3 + Ax + B - y^2) = (x - x_1)(x - x_2)(x - x')$$

$$x^3 + Ax + B - (\lambda x + \beta)^2 = x^3 - (x_1 + x_2 + x')x^2 + (x_1x_2 + x_2x' + x'x_1)x - (x_1x_2x')$$

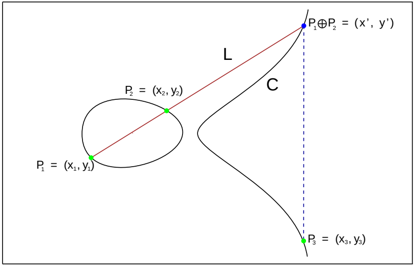
$$x^3 - \lambda^2 x^2 + (A - 2\lambda\beta)x + (B - \beta^2) = x^3 - (x_1 + x_2 + x')x^2 + (x_1x_2 + x_2x' + x'x_1)x - (x_1x_2x')$$

Elliptic Curves / Explicit Point Addition Formulae ($P_1 \neq P_2$)

$$\mathbf{L} : y = \lambda x + \beta$$

where

$$\lambda = (y_2 - y_1)/(x_2 - x_1)$$



$$\mathbf{C} : y^2 = x^3 + Ax + B \quad \longrightarrow \quad (x^3 + Ax + B - y^2) = (x - x_1)(x - x_2)(x - x')$$

$$x^3 + Ax + B - (\lambda x + \beta)^2 = x^3 - (x_1 + x_2 + x')x^2 + (x_1x_2 + x_2x' + x'x_1)x - (x_1x_2x')$$

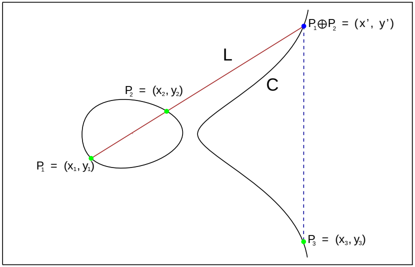
$$x^3 - \lambda^2 x^2 + (A - 2\lambda\beta)x + (B - \beta^2) = x^3 - (x_1 + x_2 + x')x^2 + (x_1x_2 + x_2x' + x'x_1)x - (x_1x_2x')$$

Elliptic Curves / Explicit Point Addition Formulae $(P_1 \neq P_2)$

$$\lambda^2 = x_1 + x_2 + x'$$

$$x' = \lambda^2 - x_1 - x_2$$

$$x_3 = x' = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2$$



$$\mathbf{L} : y = \lambda x + \beta$$

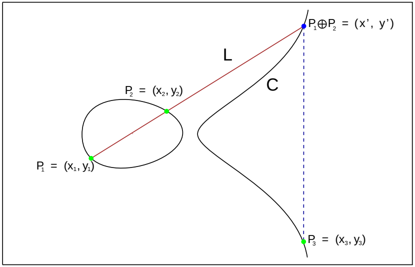
$$y_3 = -y' = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1$$

Elliptic Curves / Explicit Point Addition Formulae $(P_1 \neq P_2)$

$$\lambda^2 = x_1 + x_2 + x'$$

$$x' = \lambda^2 - x_1 - x_2$$

$$x_3 = x' = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2$$



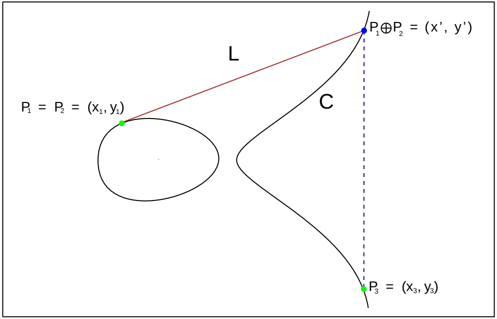
$\mathbf{L} : y = \lambda x + \beta$

$$y_3 = -y' = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1$$

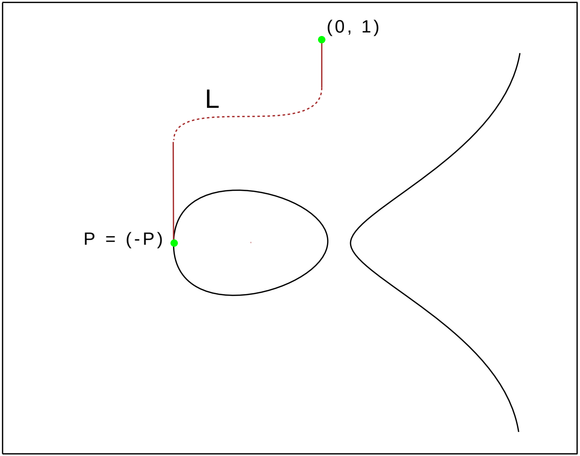
Elliptic Curves / Explicit Point Addition Formulae $(P_1 = P_2)$

$$x_3 = \left(\frac{3{x_1}^2 + A}{2y_1}\right)^2 - 2x_1$$

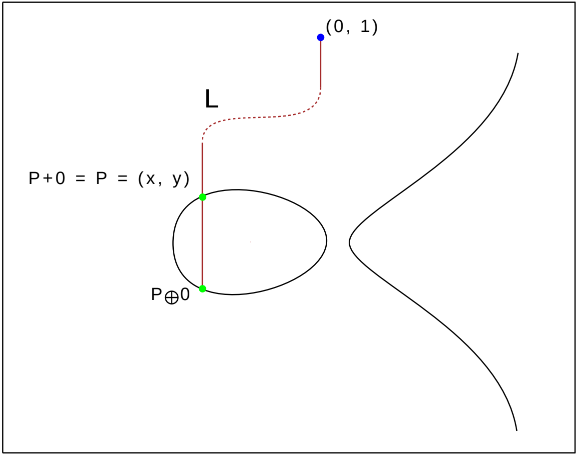
$$y_3 = \left(\frac{3{x_1}^2 + A}{2y_1}\right)(x_1 - x_3) - y_1$$



Elliptic Curves / Explicit Point Addition ($P = -P$)



Elliptic Curves / Explicit Point Addition $(P + \mathcal{O} = P)$



input : $P_1 = (x_1, y_1), P_2 = (x_2, y_2) \in E \mod p$.
output: $P_1 + P_2 = (x_3, y_3) \in E \mod p$.
if $P_1 = \mathcal{O}$ **then** **return** P_2.
else if $P_2 = \mathcal{O}$ **then** **return** P_1.
else if $x_1 = x_2$ **then**
  if $y_1 \neq y_2$ **then** **return** $\mathcal{O}$.
  else if $y_1 = 0$ **then** **return** $\mathcal{O}$.
  else
    $x_3 := ((3x_1^2 + a)/(2y_1))^2 - 2x_1 \mod p$.
    $y_3 := ((3x_1^2 + a)/(2y_1))(x_1 - x_3) - y_1 \mod p$.
    return (x_3, y_3).
  end
else
  $x_3 := ((y_1 - y_2)/(x_1 - x_2))^2 - x_1 - x_2 \mod p$.
  $y_3 := ((y_1 - y_2)/(x_1 - x_2))(x_1 - x_3) - y_1 \mod p$.
  return (x_3, y_3).
end

Elliptic Curves / A toy example

$E : y^2 = x^3 + 77x + 92 \mod 137.$

$4A^3 + 27B^3 \equiv 67 \not\equiv 0 \mod p$. So, E is an elliptic curve.

- $(x_1, y_1) = (95, 77) = P$ satisfies E.
- $(95, 77) + (95, 77) = (56, 31) = 2P$
- $(56, 31) + (95, 77) = (98, 67) = 3P$
- $(98, 67) + (95, 77) = (16, 25) = 4P$
- $\ldots$

ElGamal (Revisited)

As usual Alice wants to send a message to Bob.

- Let $G = \langle P \rangle$ be a cyclic group.
- Bob's public key is $Q = kP$.
- Bob's private key is k.
- Plaintext is $M \in G$.

Alice performs:

ElGamal Encryption

input : Q, M.
output : $\{C_0, C_1\}$.

Select a random r, $0 < r < |\langle P \rangle|$.

Compute $C_0 = rP$.

Compute $C_1 = M + rQ$.

return $\{C_0, C_1\}$. (The ciphertext)

Bob performs:

ElGamal Decryption

input : k, $\{C_0, C_1\}$.
output : M.

Compute $M = C_1 - kC_0$.

return M.

Thanks.