09 / 10 / 2024

p l a i n t e x t

#1 - RSA and Coppersmith's Method

Introductions!

Join the SIGNAL! -->

Introduce yourself:

- Name
- Where are you from
- Major and class year
- 1 thing you like

Challenge Solution!

(BEGIN CIPHERTEXT MESSAGE)

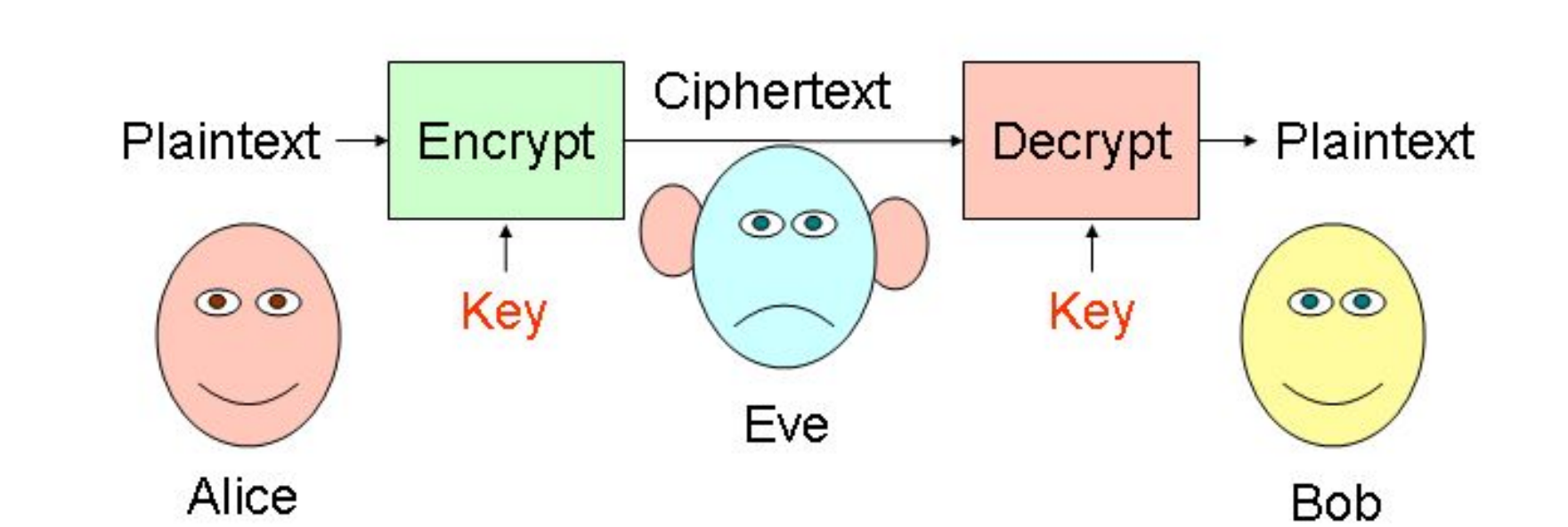
8S0)jDfm14@;[3.ART*lFD,B0/0K4VFWb.)FED)7+EM47G9D$I+Wrfp

(END CIPHERTEXT MESSAGE)

Outline

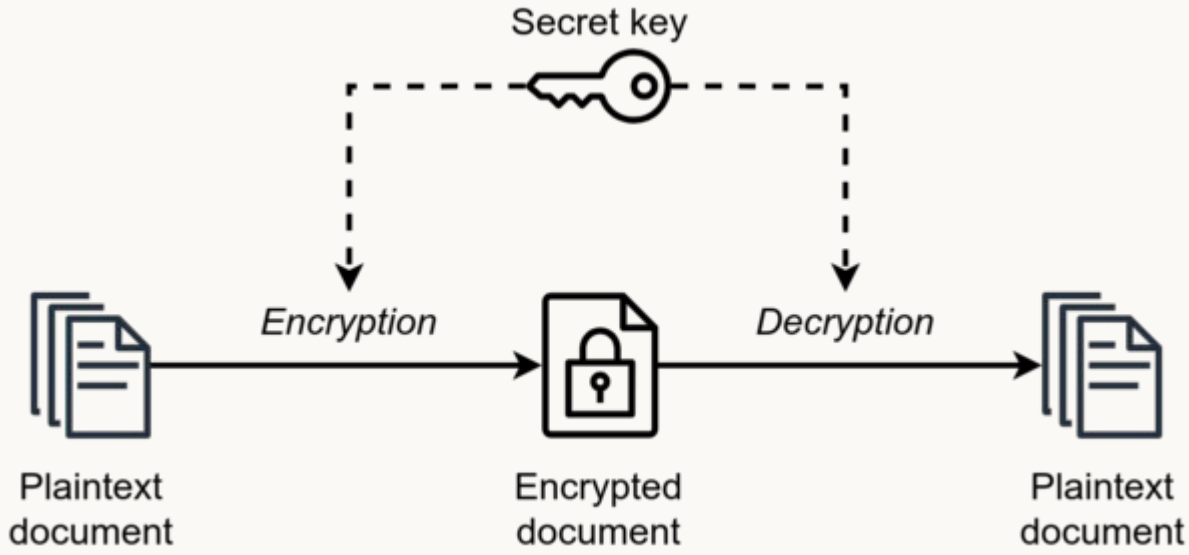
- What is encryption?
- Symmetric key vs Asymmetric key encryption
- RSA & Stereotyped message attack
- Lattices and the SVP problem
- LLL algorithm
- Coppersmith’s method

Why do we need cryptography?



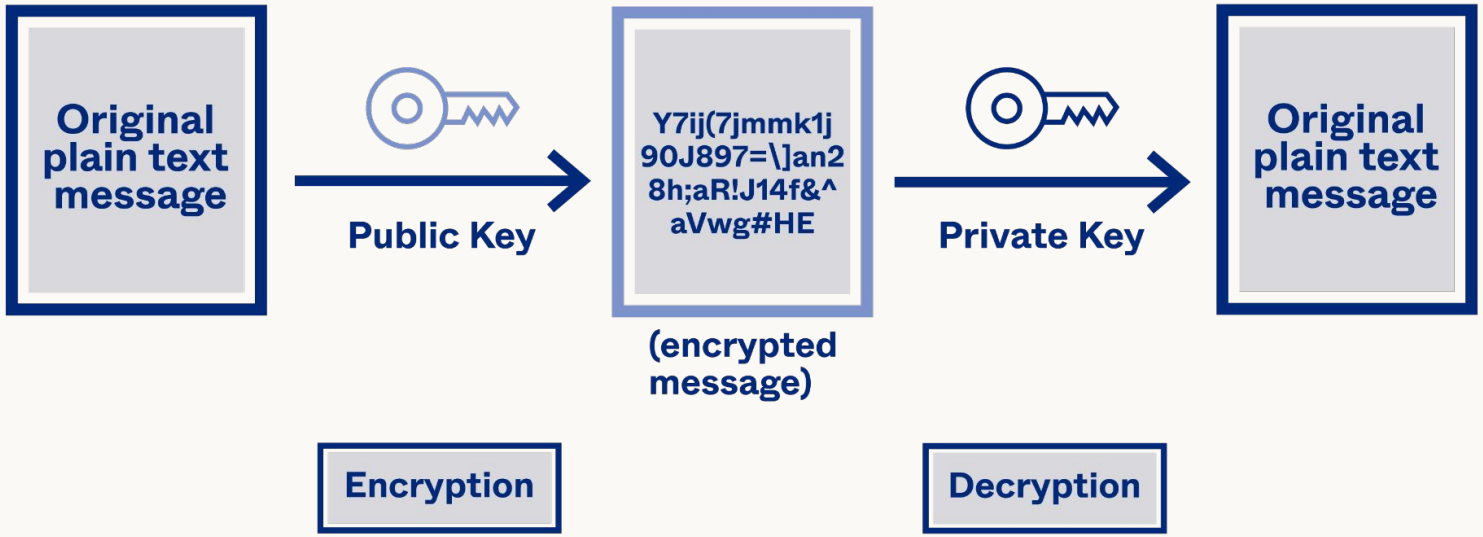
2 main types of encryption: Symmetric and Asymmetric

Symmetric-Key Encryption



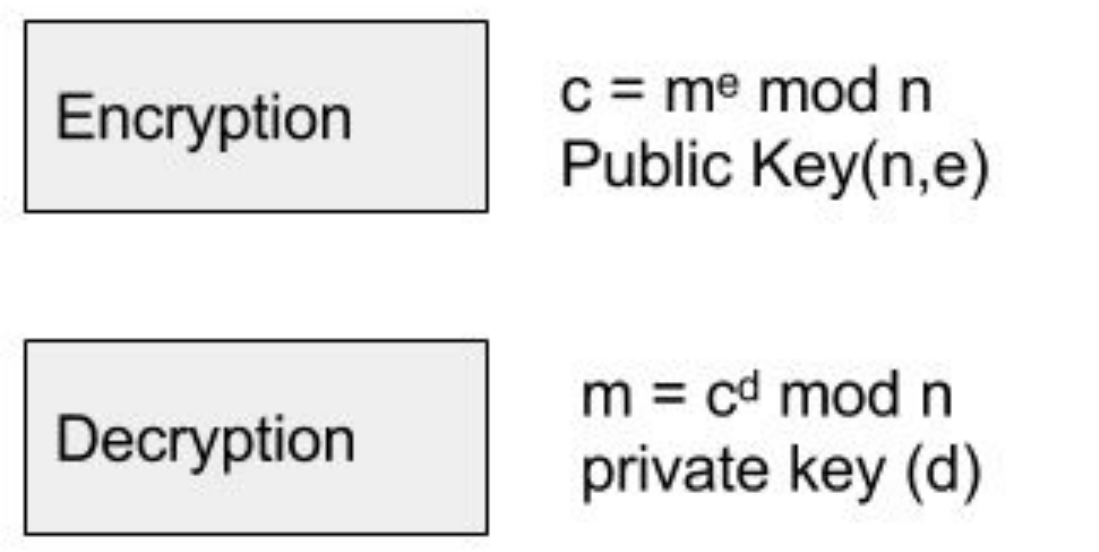
Examples: AES and ChaCha20

Public-Key (Asymmetric Key) Encryption



Rivest-Shamir-Adleman (RSA)

m is **plaintext**, c is **ciphertext**, numbers (n,e) are **public key**, d is **private key**



RSA Details

How to convert plaintext message to numbers m?

ASCII BINARY ALPHABET			
A.	01000001	S.	01010011
B.	01000010	T.	01010100
C.	01000011	U.	01010101
D.	01000100	V.	01010110
E.	01000101	W.	01010111
F.	01000110	X.	01011000
G.	01000111	Y.	01011001
H.	01001000	Z.	01011010
I.	01001001	0.	00000000
J.	01001010	1.	00000001
K.	01001011	2.	00000010
L.	01001100	3.	00000011
M.	01001101	4.	00000100
N.	01001110	5.	00000101
O.	01001111	6.	00000110
P.	01010000	7.	00000111
Q.	01010001	8.	00001000
R.	01010010	9.	00001001

What is that “mod n” thing?

$$c = m^e \bmod n$$

Public Key(n,e)

It is just the remainder when divided by n!

Onto the juicy stuff!

Stereotyped message attack

A bank sends the following using RSA to all its customers:

“Your secret 10 letter password is {secret}! Don’t share it with anyone!”

(Password is randomized for each customer).

Is this **secure**?

Modular Polynomials

Just a classic polynomial but “mod n”

Normal polynomial: x^2 - 9x + 8 (has a root at x = 1)

Modular polynomial x^2 - 9x + 8 mod 6 (has roots at x = 1 and x = 2!)

Attack plan

Stereotyped message attack is a **modular polynomial problem**!

We have ciphertext **c**, and we have public key **n** and **e**.

And template message **m'**. What we don't know is **x**.

Root of modular polynomial $f(x) = (m' + x)^e - c \pmod n$ is our solution. **Yippee! We are done**!

Not quite!

Nope… not so fast :(

Finding all roots of modular polynomial is equivalent to **factoring N**. Takes too long, even on supercomputer!

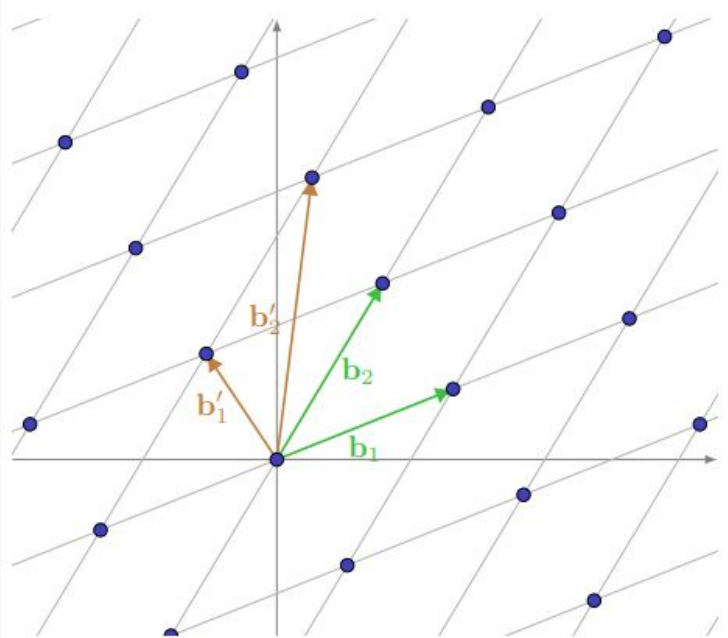
How about if we relax the problem?

It is possible to quickly find **“small”** roots of modular polynomial!

How do we do this?

Lattices to the rescue!

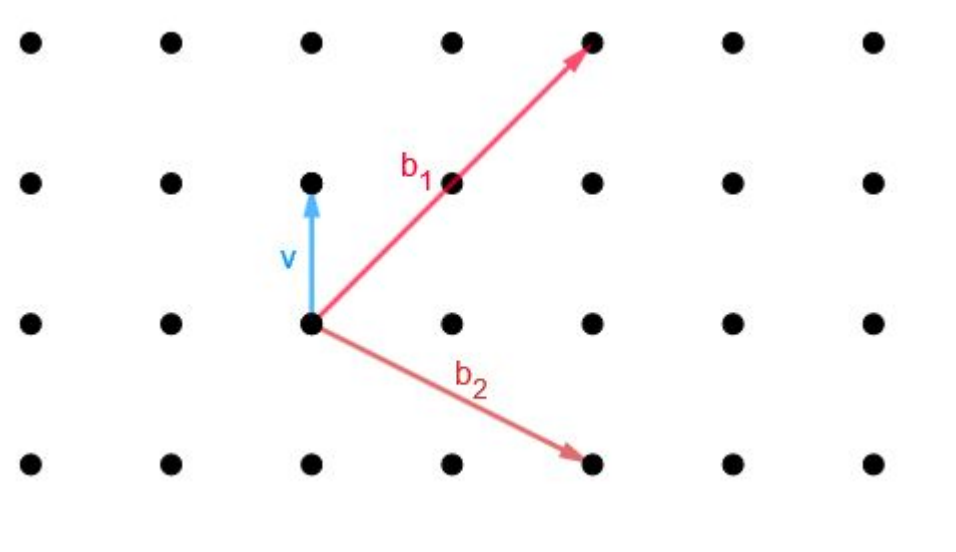
Lattice is **vector space** but only **integral** linear combinations.



$$\mathcal{P}(\mathbf{B}) = \left\{ \sum_{i=1}^{m} a_i \mathbf{b}_i \mid a_i \in [0, 1) \right\}$$

Lattice problems

SVP (shortest vector problem)



This problem is

NP-Hard

Again, let's relax it.

The approximation problem SVPγ can be solved efficiently!

Gamma is the approximation factor.

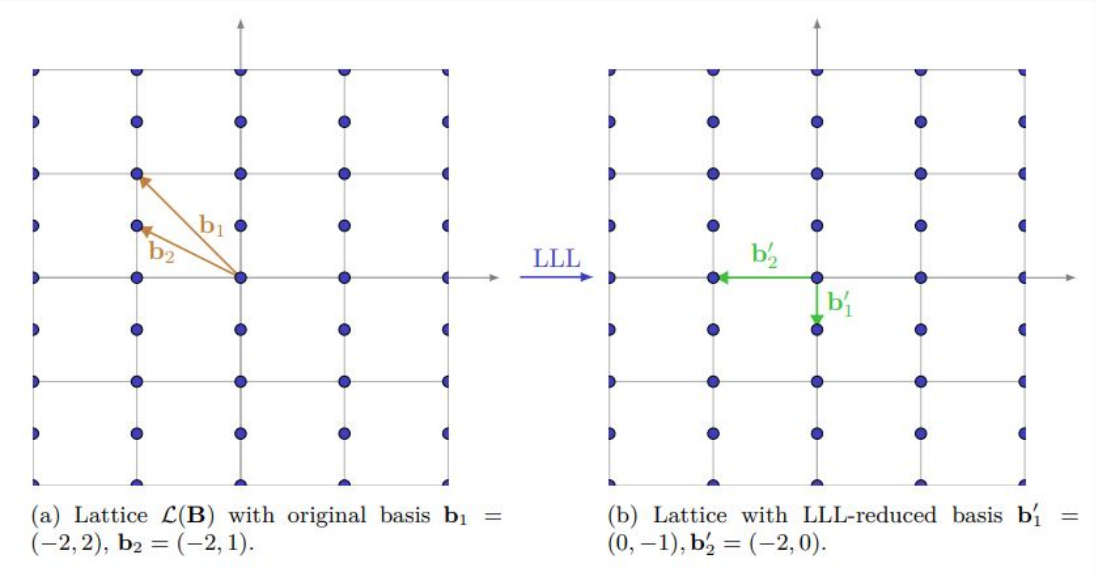
Lenstra–Lenstra–Lovász (LLL) Algorithm

"LLL"-reduces lattice basis in polynomial time.

"LLL-reduced" = shorter and more orthogonal basis vectors

(like Gram–Schmidt)

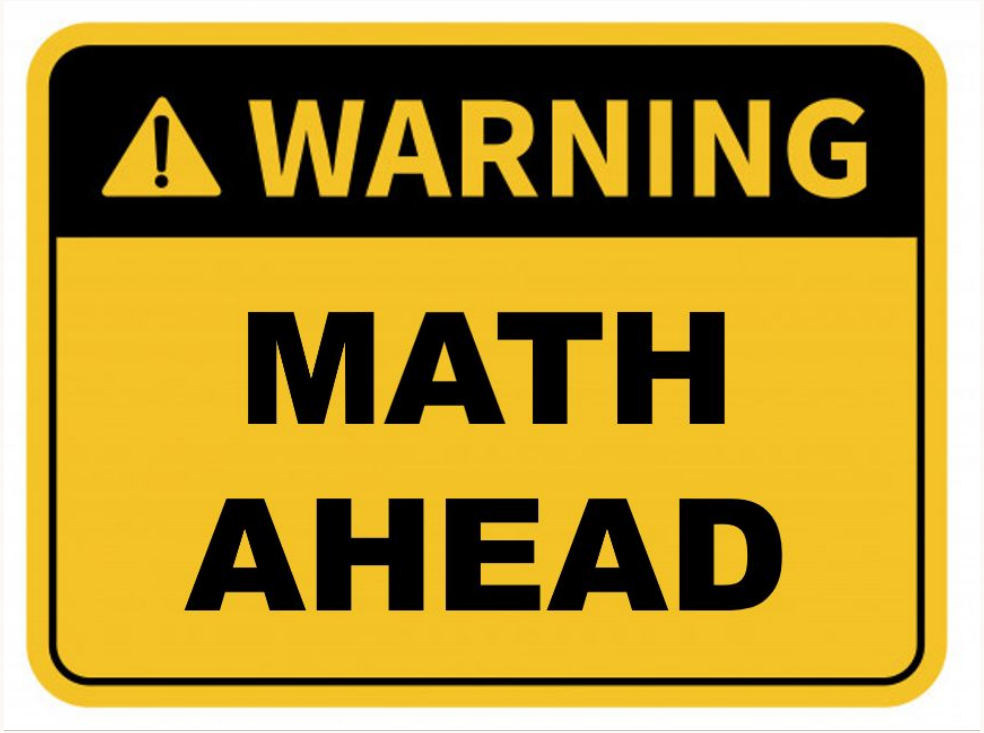
Automatically solves SVPγ!



(a) Lattice $\mathcal{L}(\mathbf{B})$ with original basis $\mathbf{b}_1 = (-2, 2)$, $\mathbf{b}_2 = (-2, 1)$.

(b) Lattice with LLL-reduced basis $\mathbf{b}'_1 = (0, -1), \mathbf{b}'_2 = (-2, 0)$.

Coppersmith's Method (overview)



Coppersmith's Method (overview)

How do we find small roots of modular polynomials with**SVPγ**?

Main idea: Transform modular polynomial to polynomial over the reals, where **small roots of modular polynomial** are **still roots** of our new polynomial.

and then use traditional root finding algorithms on new polynomial.

transform f(x) (modular polynomial) to h(x) normal polynomial

Coppersmith's Method (details)

1) Construct a **lattice of polynomials** that shares roots of f mod N.

2) Find a relatively small vector/polynomial in this lattice

3) If this vector is small enough, it will share the roots over the reals.

5) So, a short vector in lattice can be the polynomial h(x) that we want!

6) We are done! **(for real this time)**

Demo time!

Extensions and examples

Finding small root mod some divisor d of N without knowing d

Finding small roots for multivariate modular polynomial

All kinds of cool attacks, called **lattice-based methods**

Still relevant: ROCA = Return of Coppersmith's Attack **(2017)**

- Recover RSA private keys from public keys in vulnerable systems
- National ID cards of 50% of Estonia's population had to be recalled

Summary

- What is encryption?
- Symmetric key vs Asymmetric key encryption
- RSA & Stereotyped message attack
- Lattices and the SVP problem
- LLL algorithm
- Coppersmith's method + extensions

References and Resources

- Understanding the RSA Algorithm
- A Gentle Tutorial for Lattice-Based Cryptanalysis
- Analysis of the ROCA vulnerability
- General resources:
- https://cryptohack.org/ & https://cryptopals.com/

Thanks for coming :)