

Caesar Cipher

a b c d e f g h i j k l m n o p q r s t u v w x y z

b c d e f g h i j k l m n o p q r s t u v w x y z a shift of 1

c d e f g h i j k l m n o p q r s t u v w x y z a b shift of 2

...

z a b c d e f g h i j k l m n o p q r s t u v w x y shift of 25

“attack at dawn” => “buubdl bu ebxo” with shift of 1

How can we crack this?

1. Letter frequency analysis
2. Word constraints
3. Brute force search

Vignere (vee gen air) – circa 1568

1. a b c d e f g h i j k l m n o p q r s t u v w x y z

2. b c d e f g h i j k l m n o p q r s t u v w x y z a

3. c d e f g h i j k l m n o p q r s t u v w x y z a b

...

26. z a b c d e f g h i j k l m n o p q r s t u v w x y

Now choose catch phrase (key), e.g., “candy”.

Turn that into sequence of numbers: 3,1,14,4,25.

3 1 14 25 3 1 14 4 25 3 1 14

a t t a c k a t d a w n “attackatdawn”

y t g z e k n w b d w a “ytgzeknwbdfa”

How can it be cracked?

1. Look for recurring patterns – in large text, common words often repeat because they align with the key.
2. Try to crack key. First guess length. Then use freq analysis.

One Time Pad Cipher

Use Vigenere table as base. But instead of choosing a key and sticking with it, generate k random keys.

Both parties have a copy of these random keys on a “pad”, e.g., piece of paper.

When Alice sends Bob a message, she uses the first random key to encrypt, and Bob uses the first random key to decrypt.

So far the same as before.

But now, both scratch off the first key, and use the 2nd key for next message.

They can send k messages before needing a new pad (new list).

How can it be cracked?

1. Steal/see/copy the pad.
2. Otherwise, unbreakable if never repeat keys.

NSA Steps in

National Security Agency is now involved.

The problem is that you don't want to keep generating pads – risk of getting in wrong hands. So instead of “scratching out” a random key, reuse it later.

If k is way large, then little danger of a cracker seeing the repeat.

NSA wants two things:

1. Commercial sector to use One Pad Ciphers successfully to protect themselves and their customers, e.g., VISA numbers, etc.
2. NSA wants ability to crack any message it wants using its super computers.

Sooooo, NSA decided that it could crack a message if the pad length $k < 10^{*}18$, i.e., less than $10^{*}18$ random keys on pad.

This is called DES56. The 56 comes from 56 bits, which can hold roughly $10^{*}18$ as a number.

No one in US is supposed to use a pad greater than $10^{*}18$.

What problems does shared-key crypto have?

There still remains the problem of distributing shared keys.

If we try to distribute them on the internet, others can sniff packets and see what we are doing.

Is there a way to stop the sniffers?

Let's use physical example first: box where we can put locks and chains. We will put shared key inside the box.

1. Alice puts key K inside the box, then locks it with her special key, and sends to Bob.
2. Eve intercepts the box, but can't open it – she does not have Alice's key.
3. Bob gets it, but is now stuck: he does not have a copy of Alice's key.

Hmmmm. That did not work ☹

Very Clever Idea

Let's try that again with a slight twist.

1. Alice puts key K inside the box, then locks it with her special key, and sends to Bob.
2. Eve intercepts the box, but can't open it – she does not have Alice's key.
3. Bob gets it, and now puts his own lock on it and sends it back to Alice.
4. Eve intercepts the box, but can't open it – she has neither key.
5. Alice gets it and removes her lock and sends it back to Bob.
6. Eve intercepts but still can't do anything.
7. Bob gets it, and removes his lock. He now can open and get key K.

The old double-lock trick!

Big, Big Step

We need to replace locks with some electronic mechanism. We want to use the similar idea on the internet.

Diffie-Helman algorithm does the trick. It will *generate* a shared key for both parties, all in front of Eve's eyes.

Alice	Bob
Step 0.	Both agree on two prime numbers, Y and P. This is done in public! Y=7 and P=11.
Step 1. Alice chooses a secret number A (say, 3).	Bob chooses a secret number B (say 6)
Step 2. Alice computes a new number $\alpha = Y^{**}A\%P = 7^{**}3\%11 = 2$	Bob computes a new number $\beta = Y^{**}B\%P = 7^{**}6\%11 = 4$
Step 3. Alice sends Bob alpha (public)	Bob sends Alice beta (public)
Step 4. Alice computes $\beta^{**}A\%P = 9$	Bob computes $\alpha^{**}B\%P = 9$
Step 5. Alice knows the shared key = 9	Bob knows the shared key = 9.

How does this work?

Theorem:

$$(Y^{**B\%P})^{**A\%P} = (Y^{**A\%P})^{**B\%P}$$

That means that A and B are good private keys. If Alice holds the key A, she does not have to pass it to Bob. She only has to pass him $(Y^{**A\%P})$, which we are calling `alpha`.

She then knows that Bob can use that to plug into `alpha^{**B\%P}`.

The theorem says that when they both do this, they will get the same result. This is the shared key 😊

What does Eve know?

$Y=7$ and $P=11$

$\alpha=2$ and $\beta=4$

$7^{**}A\%11=2$ and $7^{**}B\%11=4$

$4^{**}A\%11 = 2^{**}B\%11$

Alice	Bob
Step 0.	Both agree on two prime numbers, Y and P. This is done in public! $Y=7$ and $P=11$.
Step 1. Alice chooses a secret number A (say, 3).	Bob chooses a secret number B (say 6)
Step 2. Alice computes a new number $\alpha=Y^{**}A\%P$ $= 7^{**}3\%11 = 2$	Bob computes a new number $\beta=Y^{**}B\%P$ $= 7^{**}6\%11 = 4$
Step 3. Alice sends Bob α (public)	Bob sends Alice β (public)
Step 4. Alice computes $\beta^{**}A\%P=9$	Bob computes $\alpha^{**}B\%P=9$
Step 5. Alice knows the shared key = 9	Bob knows the shared key = 9.

How can Eve crack it?

$Y=7$ and $P=11$

$\alpha=2$ and $\beta=4$

$7^{**}A\%11=2$ and $7^{**}B\%11=4$

$4^{**}A\%11 = 2^{**}B\%11$

Rough algorithm:

1. loop through all A's such that $7^{**}A\%11=2$

1. loop through all B's such that $7^{**}B\%11=4$

1. if $4^{**}A\%11=2^{**}B\%11$ then winner!