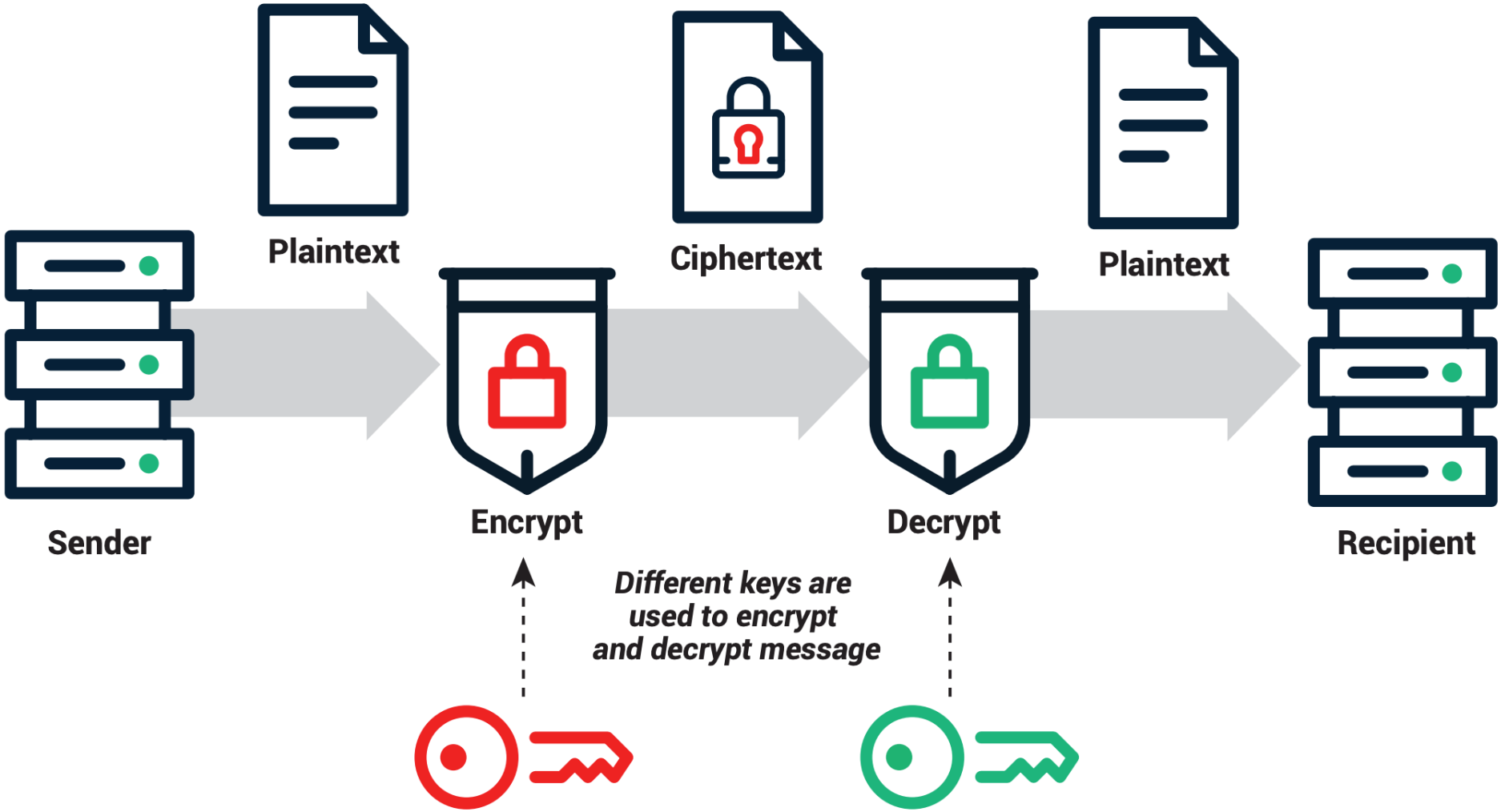
Public Key Cryptography

Public Key Cryptography

- A.k.a. asymmetric cryptography
- Two keys – public and private
- Public key is shared
- Private key is kept secret
- Well suited for organizations

How does it work



https://sectigo.com/resource-library/public-key-vs-private-key

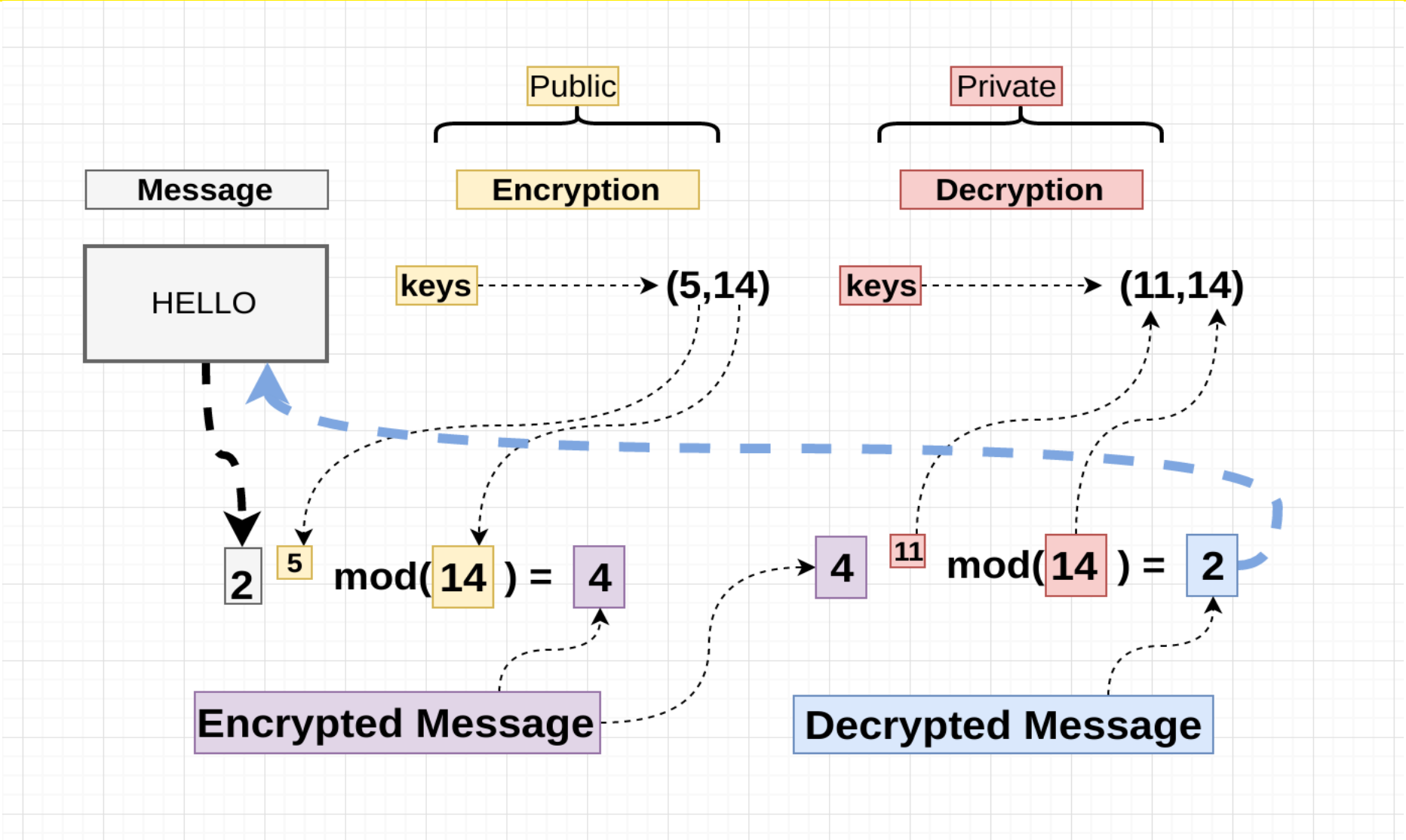
Public VS Private Key Cryptography

Public	Private
Slower	Faster
Two key	One keys
One is is public	Key is kept secret
Asymmetrical	Symmetrical
Sender and Receiver don’t share key	Share the same key

RSA

- Rivest–Shamir–Adleman (RSA)
- Invented in 1977
- Asymmetric
- A bit slow
- Not commonly to directly encrypt user data
- OpenPGP, S/MIME, SSL/TLS

How does RSA work



https://hackernoon.com/how-does-rsa-work-f44918df914b

RSA: Choosing keys

1. Choose two large prime numbers p, q. (e.g., 1024 bits each)
2. Compute $n = p.q$, $z = (p-1)(q-1)$
3. Choose e (with $e<n$) that has no common factors with z. (e, z are "relatively prime").
4. Choose d such that $ed-1$ is exactly divisible by z. (in other words: $e.d \bmod z = 1$).
5. *Public* key is $\underbrace{(n,e)}_{K_B^+}$. *Private* key is $\underbrace{(n,d)}_{K_B^-}$.

RSA: Encryption, decryption

0. Given (n,e) and (n,d) as computed above

1. To encrypt bit pattern, m, compute

   $c = m^e \bmod n$ (i.e., remainder when m^e is divided by n)

2. To decrypt received bit pattern, c, compute

   $m = c^d \bmod n$ (i.e., remainder when c^d is divided by n)

Magic happens!

$$m = (m^e \bmod n)^d \bmod n$$

RSA example:

Bob chooses $p=5$, $q=7$. Then $n=35$, $z=24$.

$e=5$ (so e, z relatively prime).

$d=29$ (so $ed-1$ exactly divisible by z.

encrypt:

letter	m	m^e	$c = m^e \bmod n$
l	12	1524832	17

decrypt:

c	c^d	$m = c^d \bmod n$	letter
17	481968572106750915091411825223072000	12	l