

# **Public Key Infrastructure**

**An Overview of Asymmetric Key Encryption**

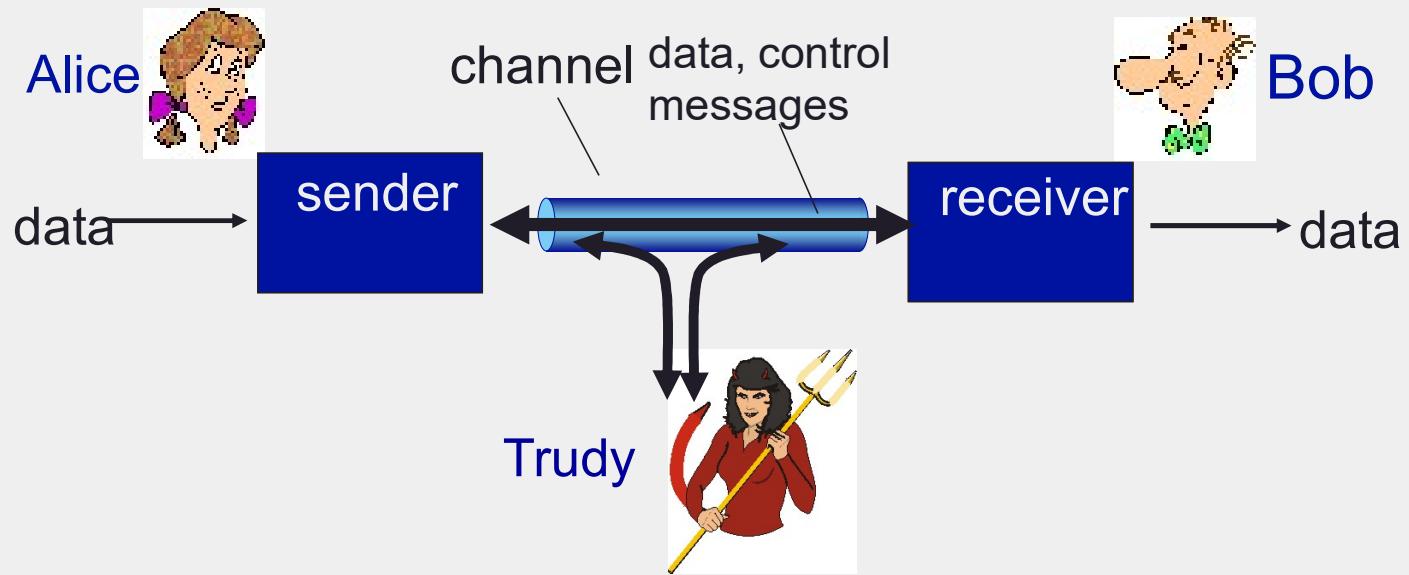
**R. Venkateswaran**

## There are bad guys (and girls) out there!

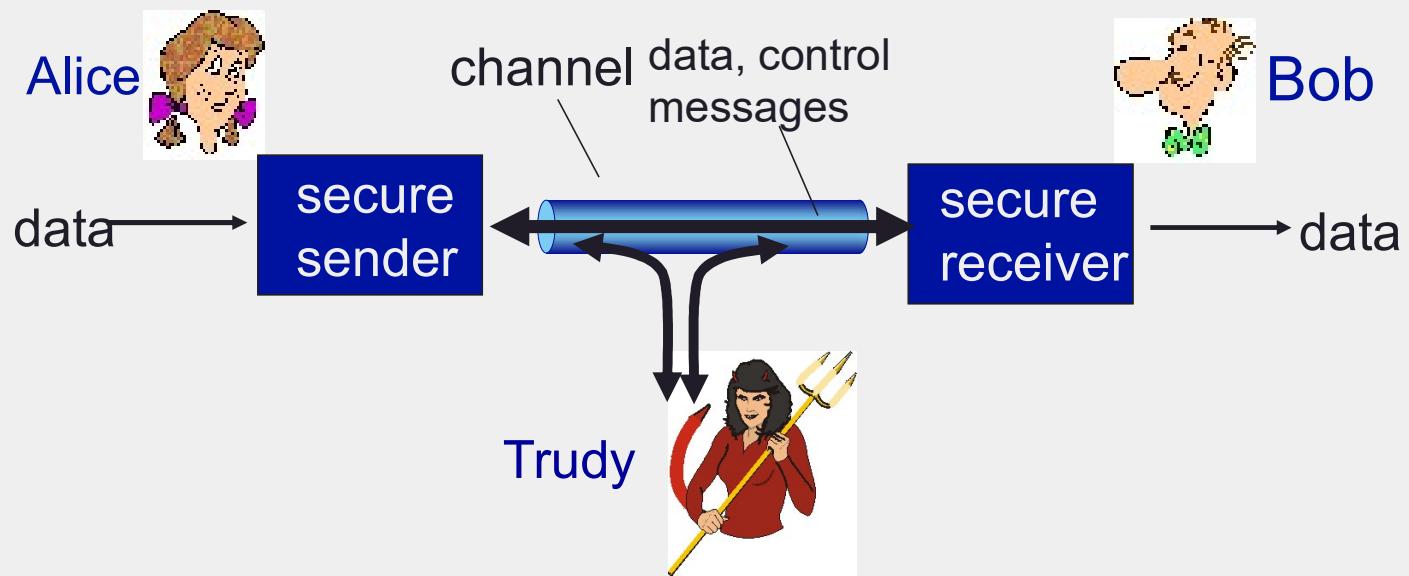
- Q: What can a “bad guy” do?
- A: A lot!
  - eavesdrop: intercept messages
  - actively insert messages into connection
  - impersonation: can fake (spoof) source address in packet (or any field in packet)
  - hijacking: “take over” ongoing connection by removing sender or receiver, inserting himself in place
  - denial of service: prevent service from being used by others (e.g., by overloading resources)

## Friends and enemies: Alice, Bob, Trudy

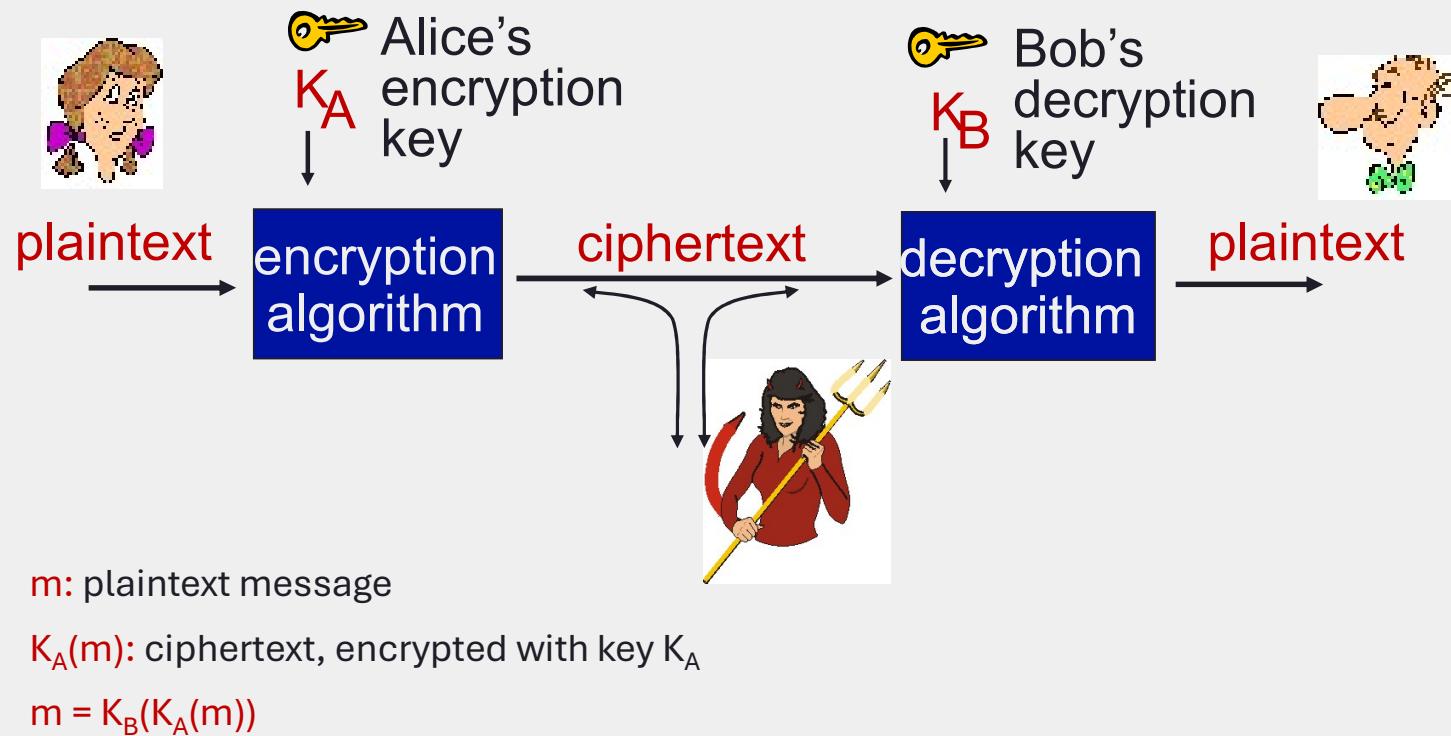
- well-known in network security world
- Alice & Bob want to communicate with each other
- Trudy (intruder) may intercept, delete, add messages



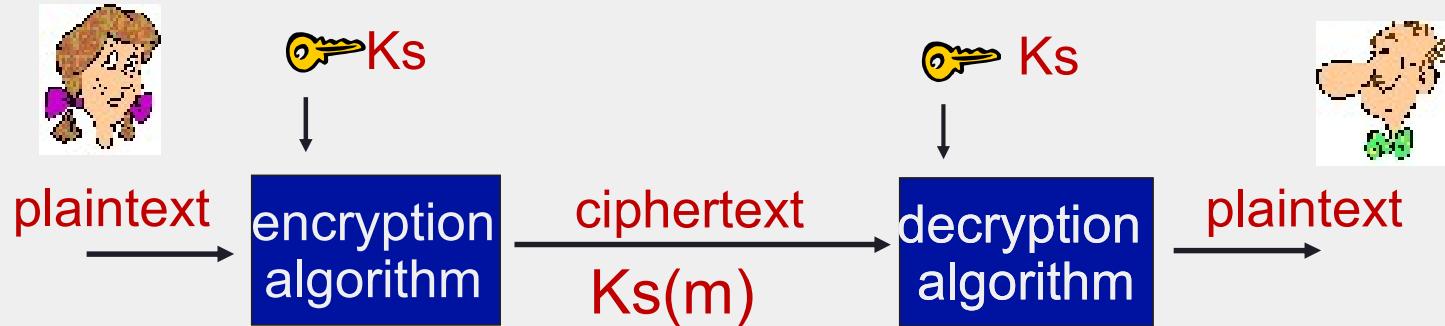
Alice and Bob want to communicate **securely**



## The language of cryptography



## Symmetric key cryptography

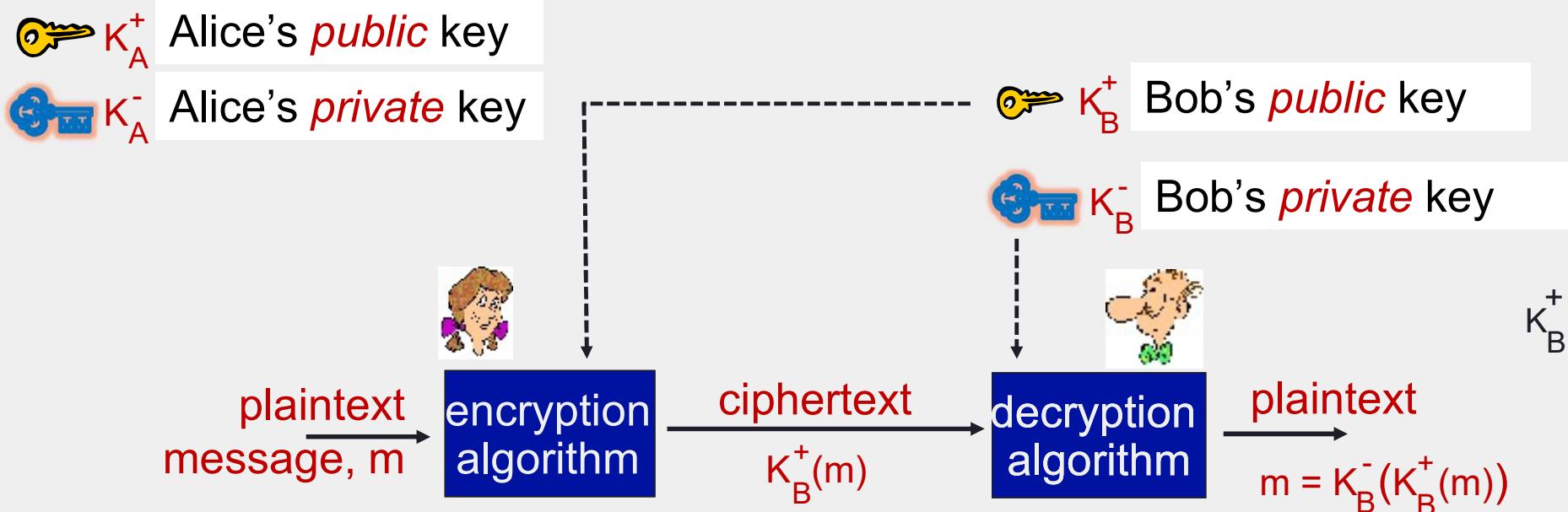


**symmetric key crypto:** Bob and Alice share same (symmetric) key Ks

Ks satisfies the following property :  $Ks(Ks(m)) = m$

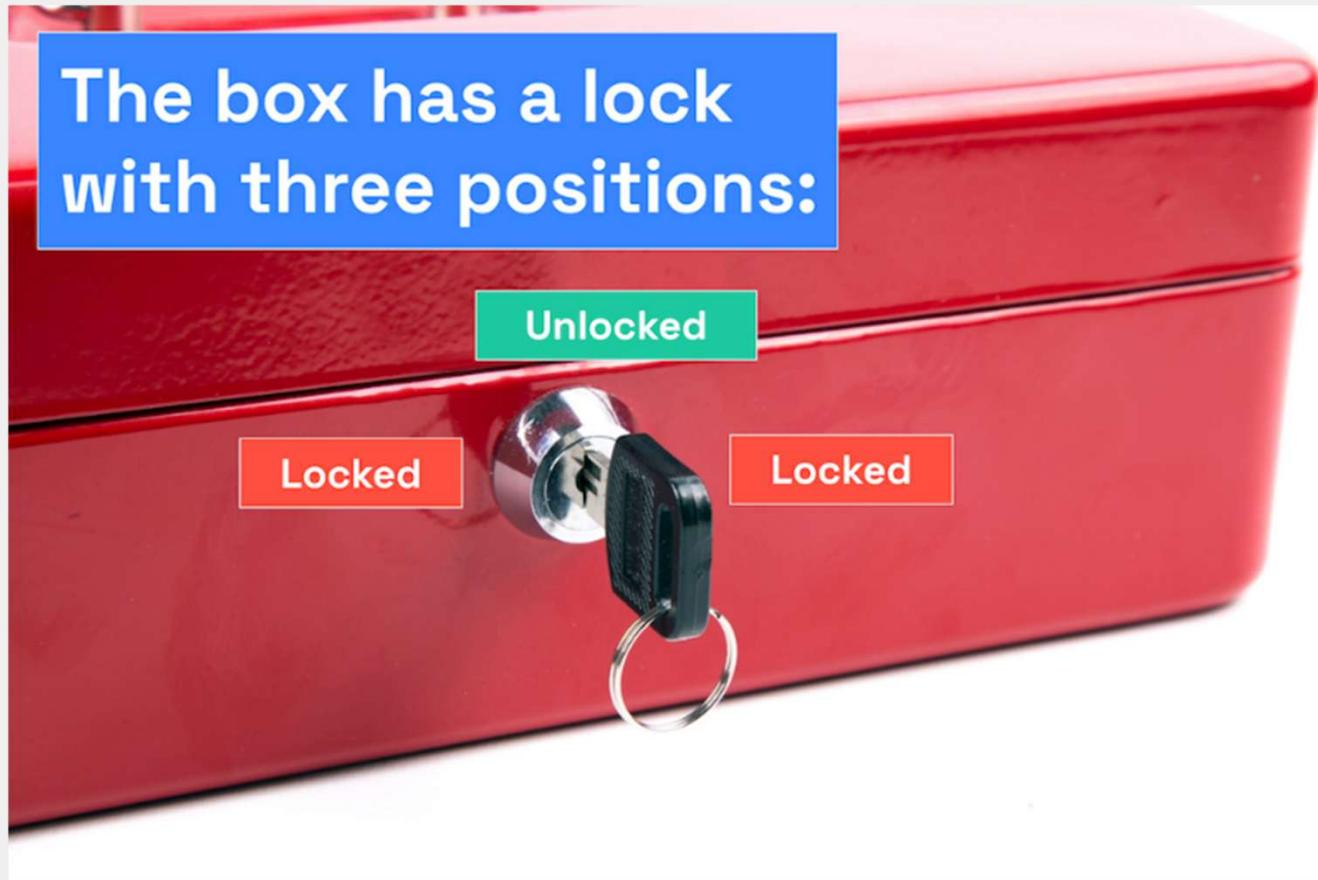
**Challenge :** How to ensure that both Alice and Bob have the same key

## Public Key Cryptography

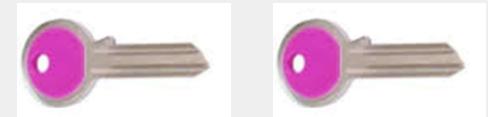


Public key cryptography revolutionized 2000-year-old (previously only symmetric key) cryptography!

## Public Key Infrastructure - Simple Analogy



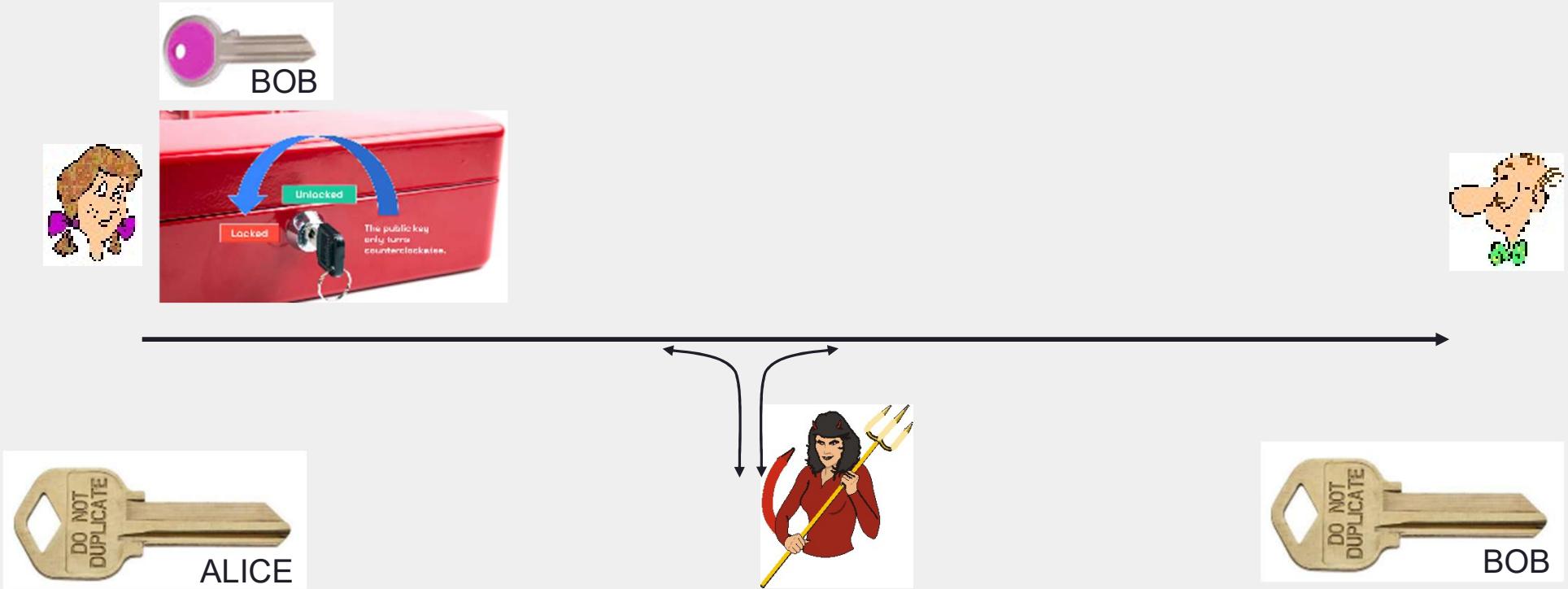
Golden Key – ONLY ONE COPY.  
Turns **CLOCKWISE**



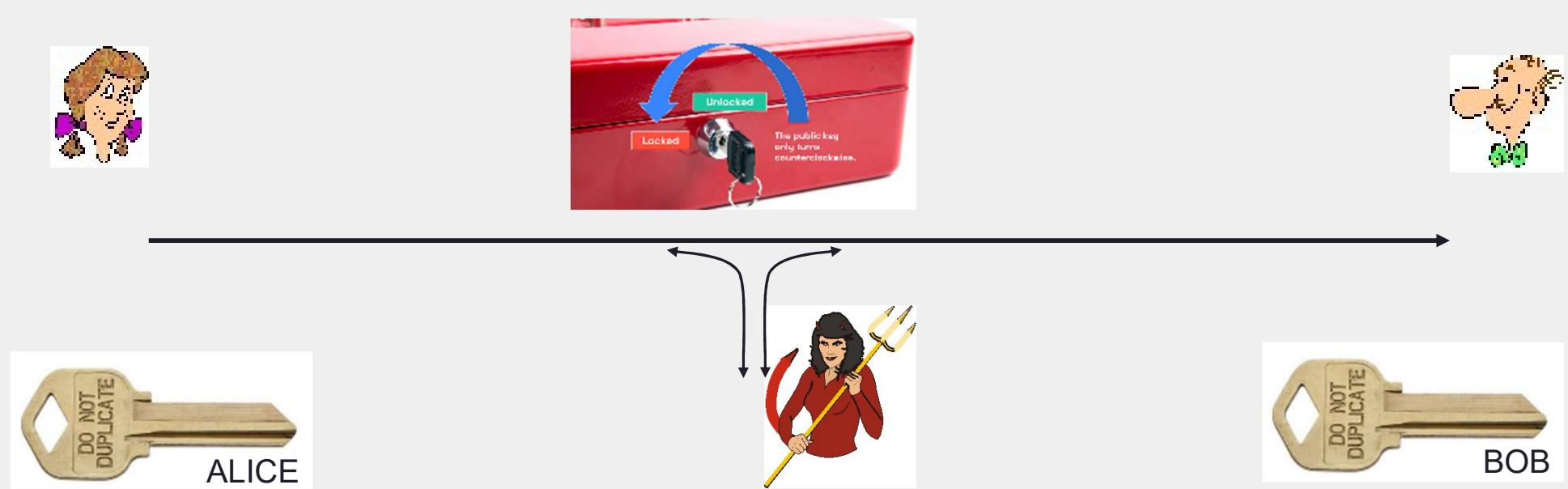
Regular Key – Many Copies.  
Turns **ANTI-CLOCKWISE**



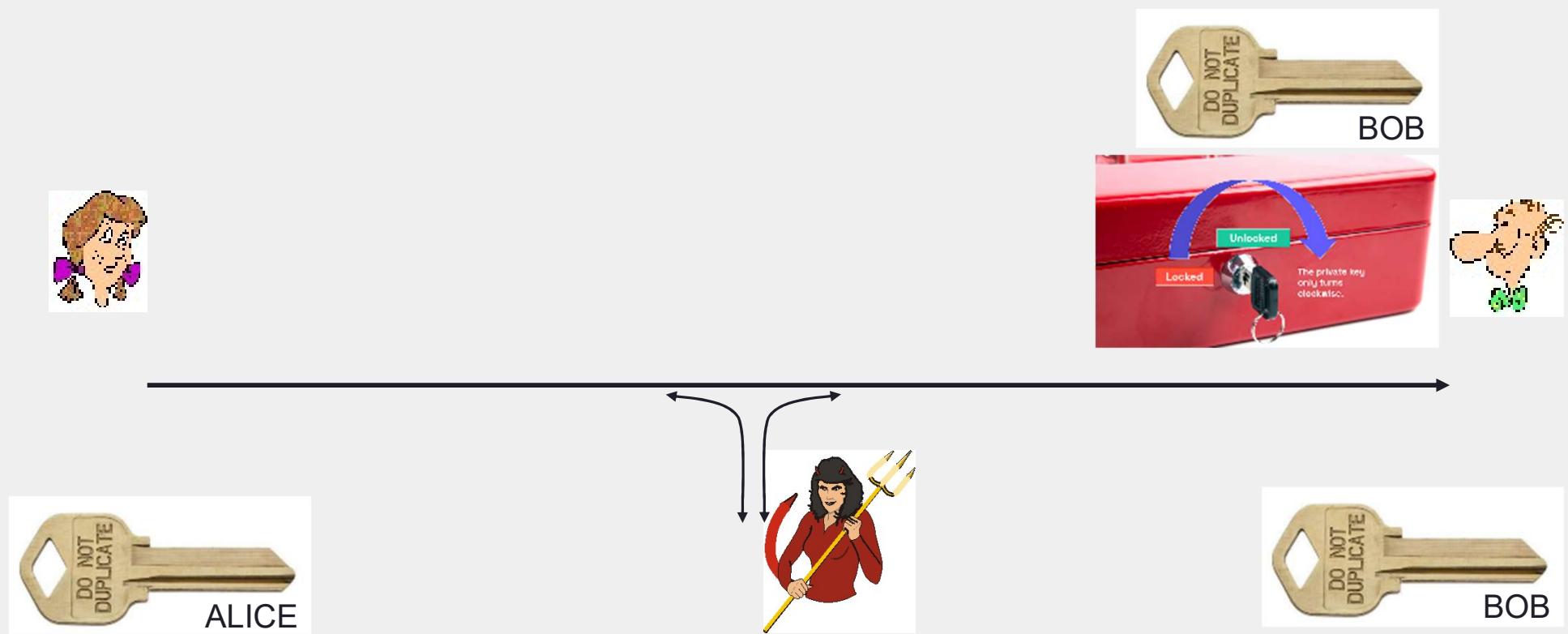
## Alice sends a message to Bob securely



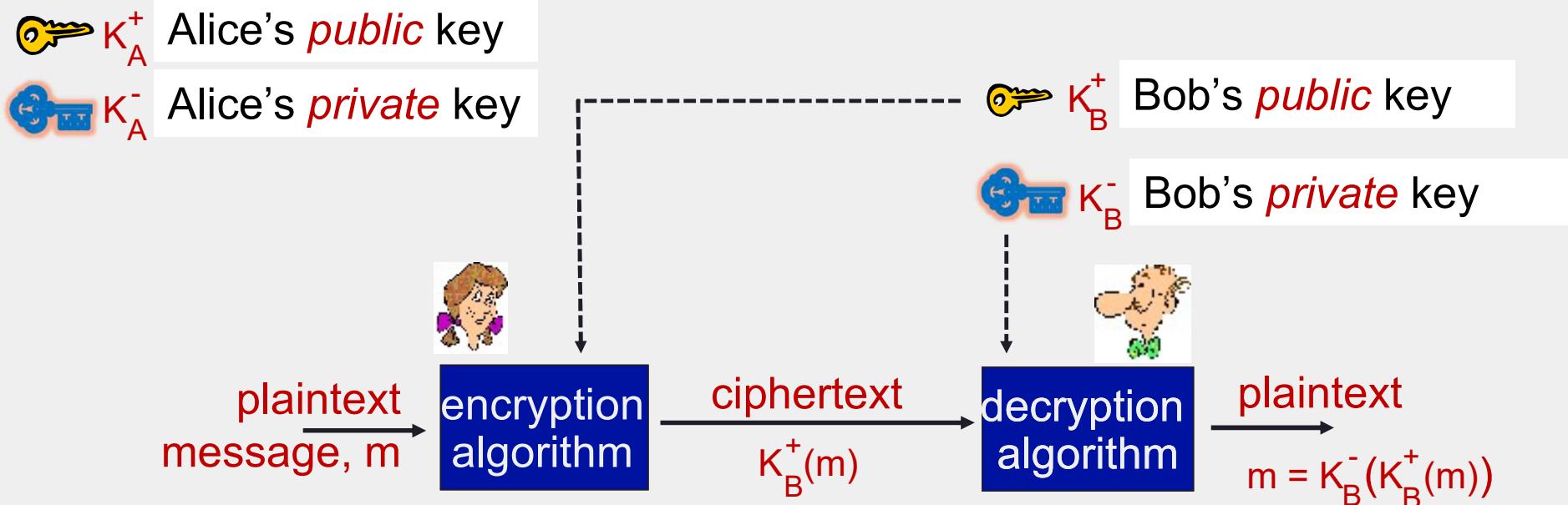
## Alice sends a message to Bob securely



Alice sends a message to Bob securely



## Public Key Cryptography



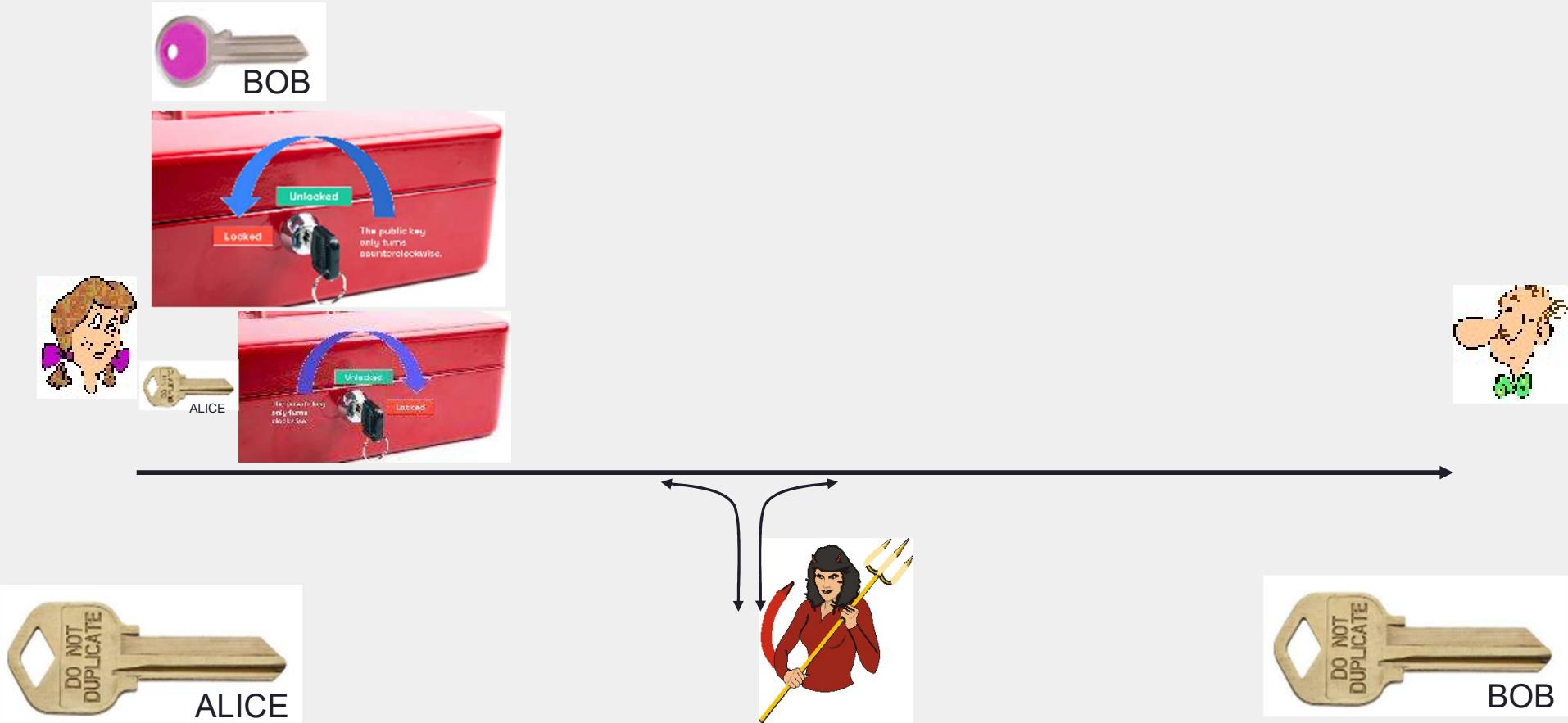
$$\text{Property 1 } K_B^-(K_B^+(m)) = K_B^+(K_B^-(m)) = m \quad \& \quad K_A^-(K_A^+(m)) = K_A^+(K_A^-(m)) = m$$

Property 2 – Given  $K_B^+$ , it must be computationally hard to compute  $K_B^-$

Trudy can intercept Alice's box, create her own message and pretend to be Alice



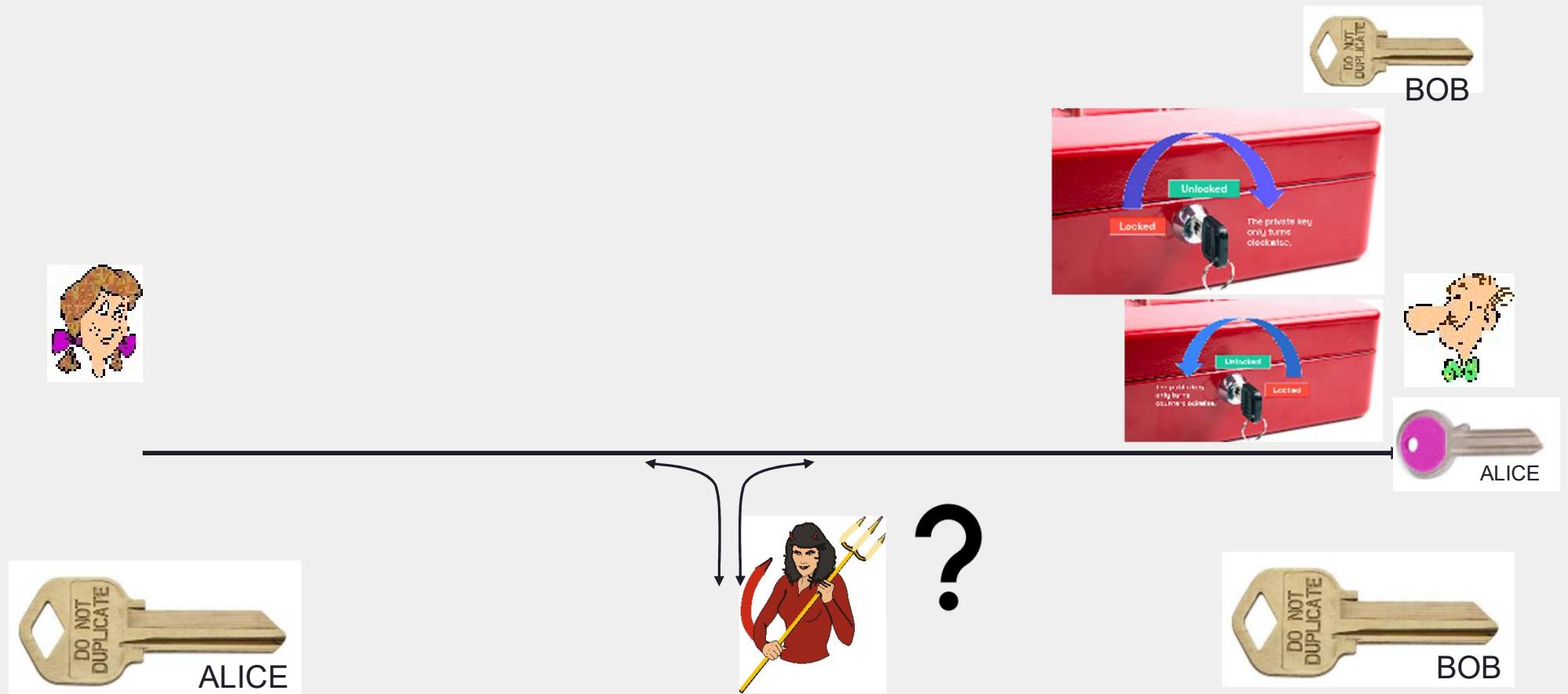
## Alice sends a message to Bob securely with her signature



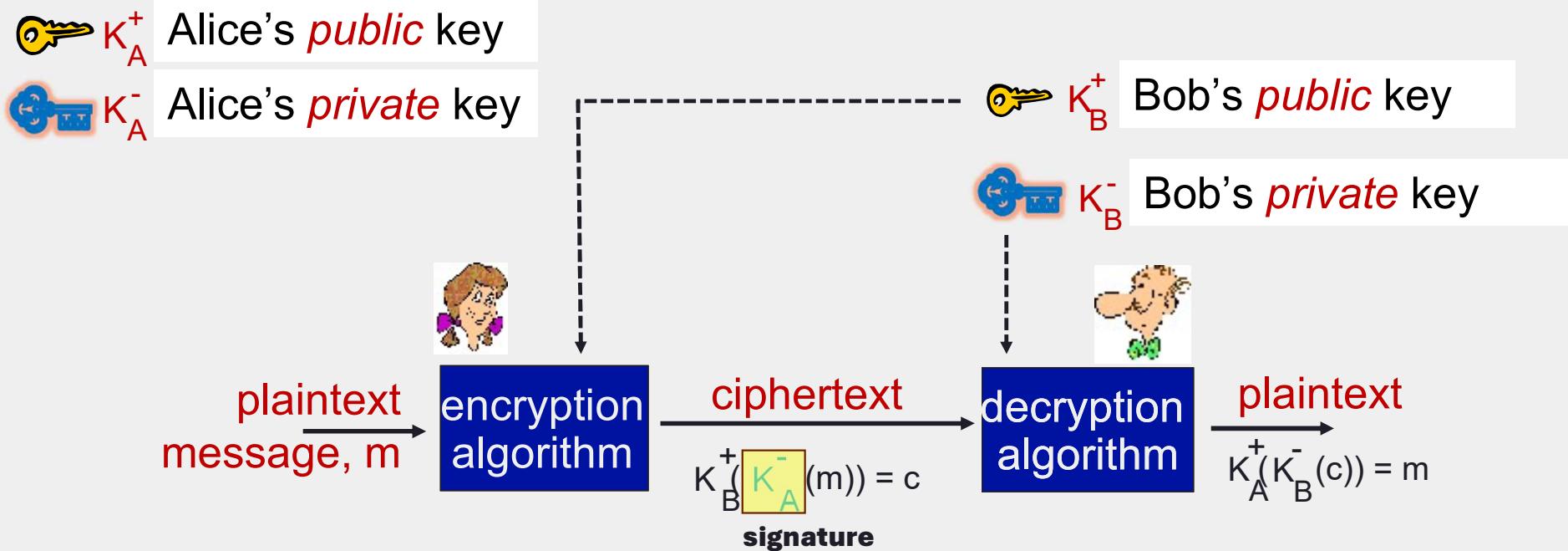
Alice sends a message to Bob securely with her signature



Alice sends a message to Bob securely with her signature



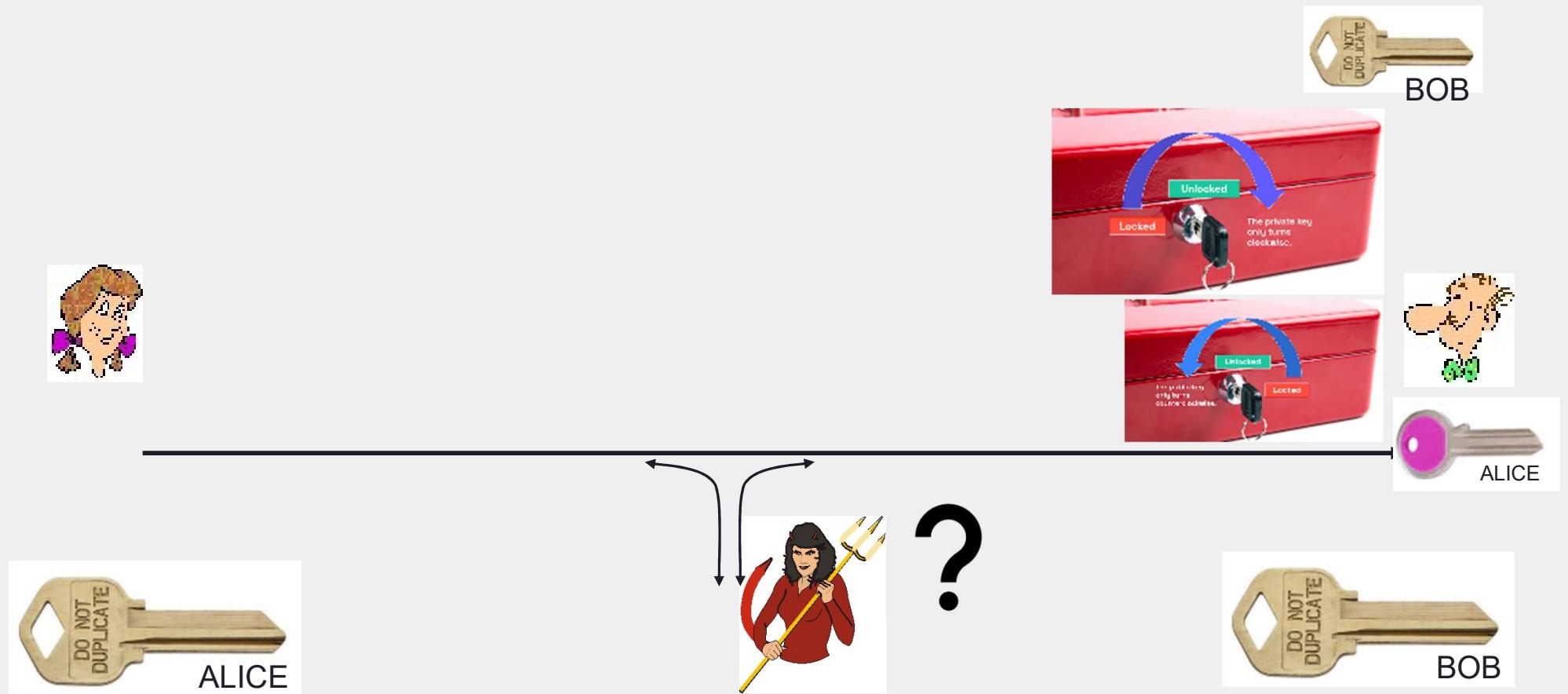
## Public Key Cryptography



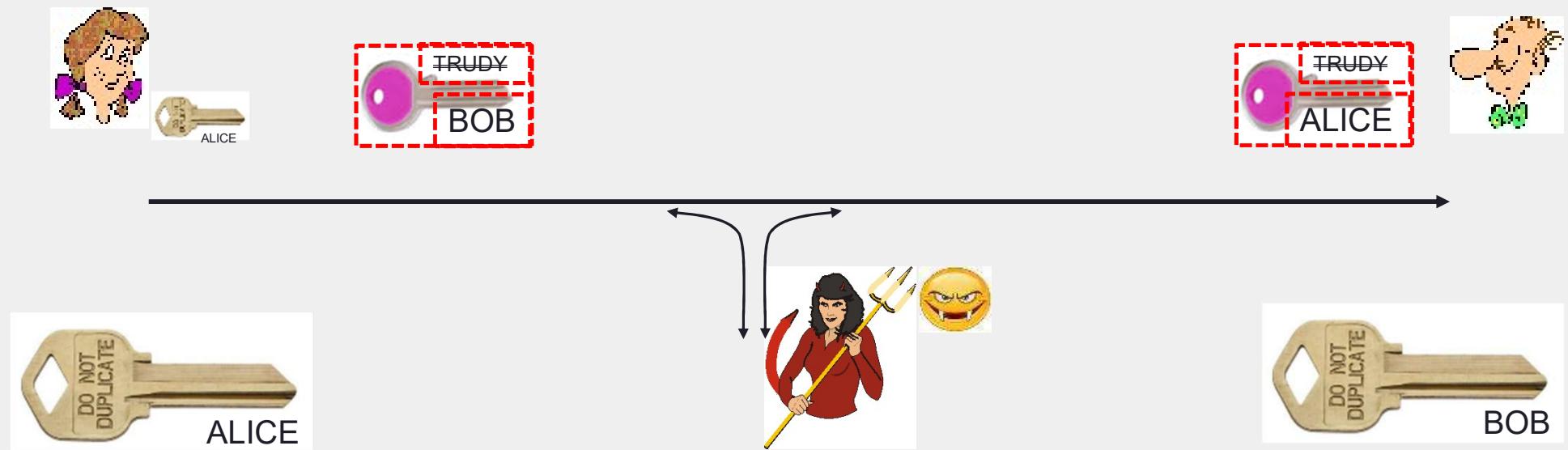
Property 1  $K_B^-(K_B^+(m)) = K_B^+(K_B^-(m)) = m \quad \& \quad K_A^-(K_A^+(m)) = K_A^+(K_A^-(m)) = m$

Property 2 – Given  $K_B^+$ , it must be computationally hard to compute  $K_B^-$

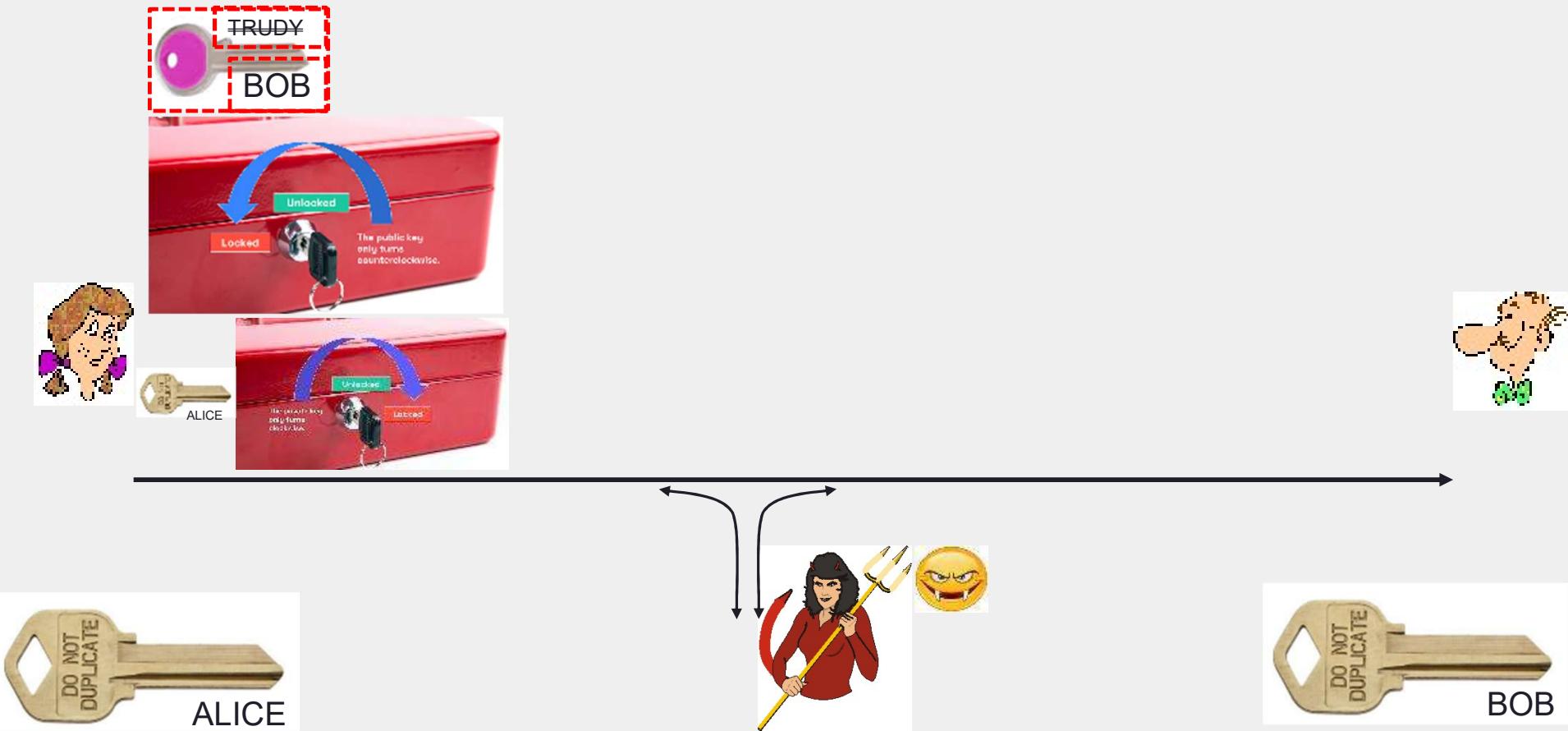
Alice sends a message to Bob securely with her signature



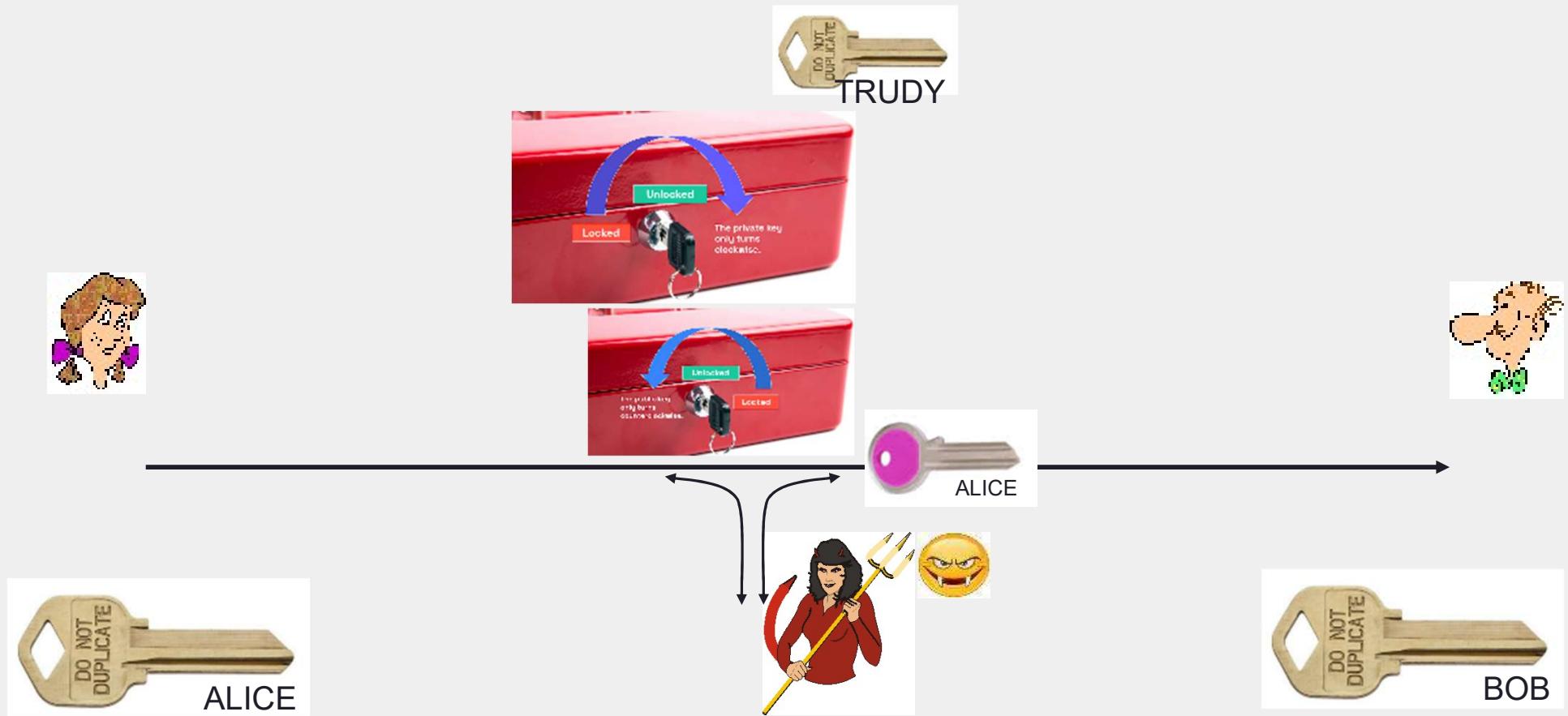
Trudy maliciously shares her Public key as Bob's and Alice's respectively



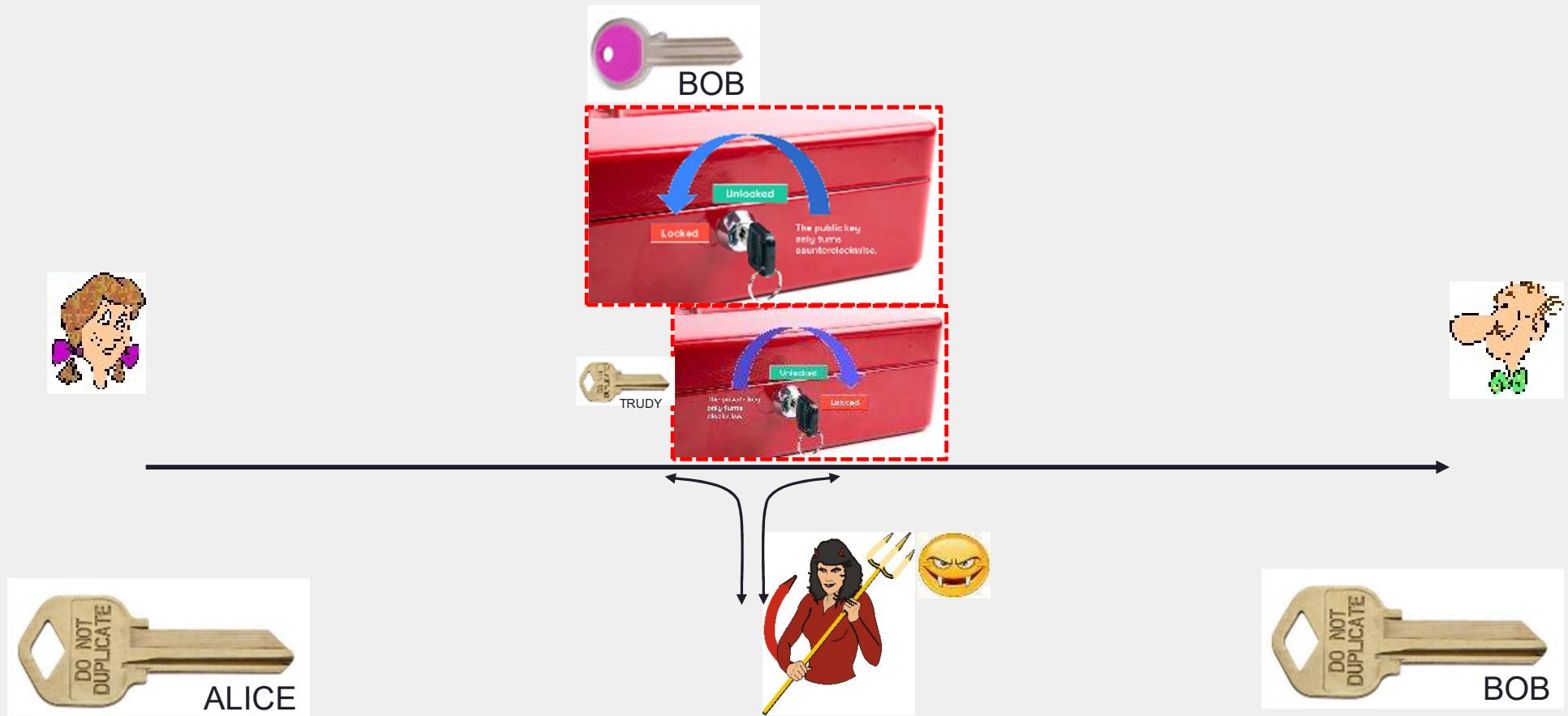
Alice sends a message to Bob securely with Trudy's signature (thinking it is Bob's)



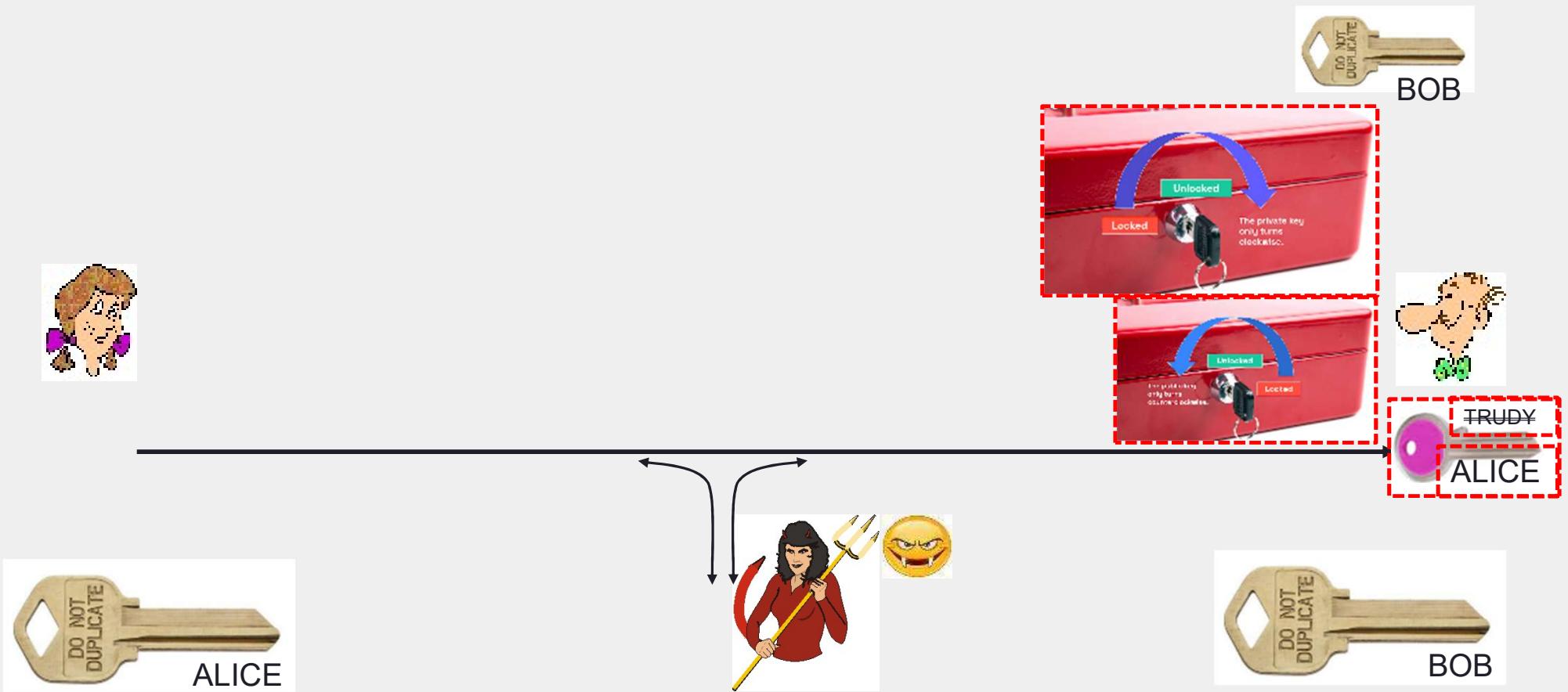
Trudy intercepts the message – and can access it using her Private key and Alice's public key



Trudy sends malicious message to Bob securely with **Trudy's** signature, pretending to be **Alice**

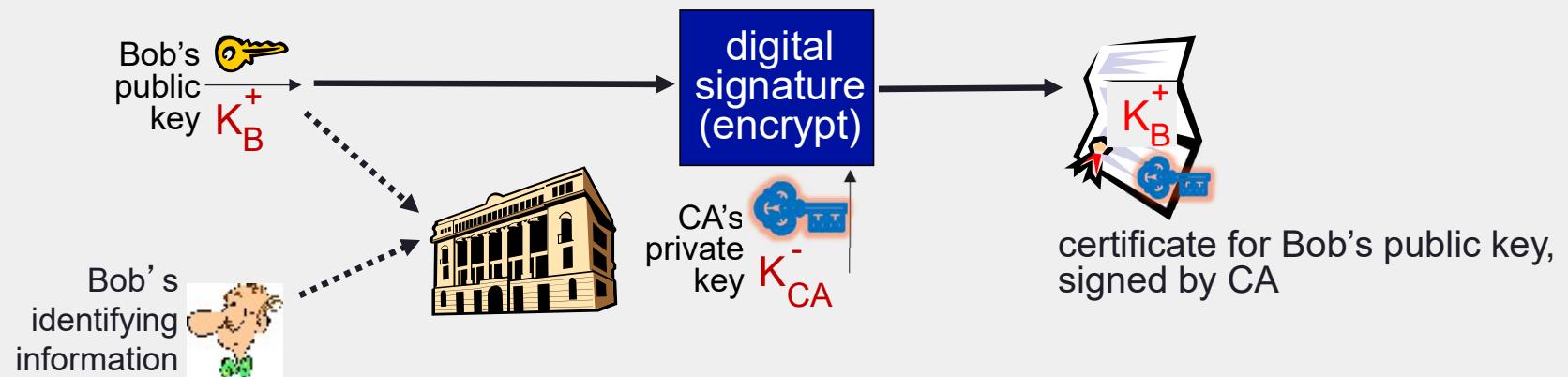


Bob uses his private key and **Trudy's** public key (thinking it is **Alice's**)



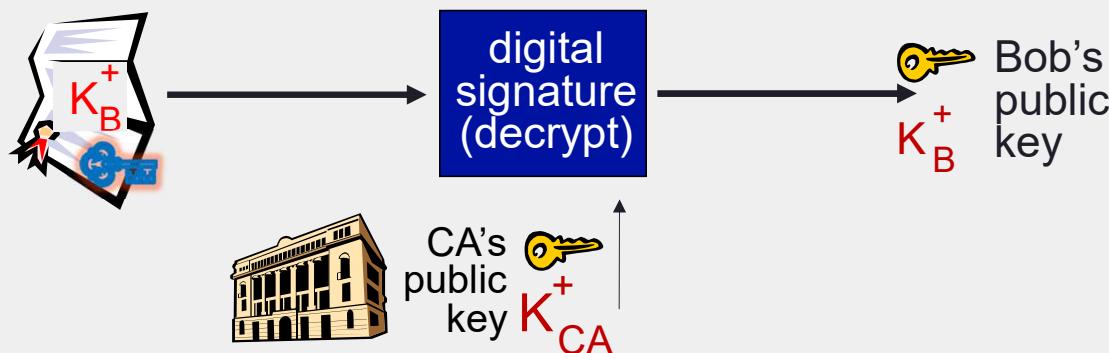
## Public key Certification Authorities (CA)

- Certification authority (CA): binds public key to particular entity, E
- Entity (person, website, router) registers its public key with CA provides “proof of identity” to CA
  - CA creates certificate binding identity E to E's public key
  - certificate containing E's public key digitally signed by CA: CA says “this is E's public key”



## Public key Certification Authorities (CA)

- when Alice wants Bob's public key:
  - gets Bob's certificate (Bob or elsewhere)
  - apply CA's public key to Bob's certificate, get Bob's public key



## Certificate Manager on Windows – Public Keys of several Certification Authorities

The screenshot shows the Windows Certificate Manager window titled "certmgr - [Certificates - Current User\Trusted Root Certification Authorities\Certificates]". The left pane displays a tree view of certificate categories under "Certificates - Current User", including Personal, Trusted Root Certification Authorities (which is expanded to show Certificates, Baltimore CyberTrust Root, Certum Trusted Network CA, Class 3 Public Primary Certificate, COMODO RSA Certification Authority, Copyright (c) 1997 Microsoft Corp., DigiCert Assured ID Root CA, DigiCert Global Root CA, DigiCert Global Root G2, DigiCert Global Root G3, DigiCert High Assurance EV Root, DigiCert Trusted Root G4, DST Root CA X3, Entrust Root Certification Authority, Entrust Root Certification Authority, Entrust.net Certification Authority, GlobalSign, GlobalSign, GlobalSign Code Signing Root, GlobalSign Root CA, Go Daddy Class 2 Certification Authority, Go Daddy Root Certificate Authority, ISRG Root X1, Microsoft Authenticode(tm) Root, Microsoft ECC Product Root Certificate, Microsoft ECC TS Root Certificate, Microsoft Identity Verification Root, Microsoft Root Authority, Microsoft Root Certificate Authority, and Trustwave SSL CA); Enterprise Trust, Intermediate Certification Authority, Active Directory User Object, Trusted Publishers, Untrusted Certificates, Third-Party Root Certification, Trusted People, Client Authentication Issuers, Other People, Certificate Enrollment Requests, and Smart Card Trusted Roots. The right pane is a table listing 44 certificates, with columns for Issued To, Issued By, Expiration Date, and Intended Purposes. The table includes rows for various well-known root certificates from companies like Baltimore, Certum, COMODO, DigiCert, Entrust, GlobalSign, Go Daddy, ISRG, Microsoft, and Trustwave.

Issued To	Issued By	Expiration Date	Intended Purposes
127.0.0.1	127.0.0.1	27-08-2031	Server Authentication
AAA Certificate Services	AAA Certificate Services	01-01-2029	Client Authentication, Code Signing
Baltimore CyberTrust Root	Baltimore CyberTrust Root	13-05-2025	Client Authentication
Certum Trusted Network CA	Certum Trusted Network CA	31-12-2029	Client Authentication
Certum Trusted Network CA 2	Certum Trusted Network CA 2	06-10-2046	Client Authentication
Class 3 Public Primary Certificate	Class 3 Public Primary Certificate	02-08-2028	Client Authentication
COMODO RSA Certification Authority	COMODO RSA Certification Authority	19-01-2038	Client Authentication
Copyright (c) 1997 Microsoft Corp.	Copyright (c) 1997 Microsoft Corp.	31-12-1999	Time Stamping
DigiCert Assured ID Root CA	DigiCert Assured ID Root CA	10-11-2031	Client Authentication
DigiCert Global Root CA	DigiCert Global Root CA	10-11-2031	Client Authentication
DigiCert Global Root G2	DigiCert Global Root G2	15-01-2038	Client Authentication
DigiCert Global Root G3	DigiCert Global Root G3	15-01-2038	Client Authentication
DigiCert High Assurance EV Root	DigiCert High Assurance EV Root	10-11-2031	Client Authentication
DigiCert Trusted Root G4	DigiCert Trusted Root G4	15-01-2038	Client Authentication
DST Root CA X3	DST Root CA X3	30-09-2021	Client Authentication
Entrust Root Certification Authority	Entrust Root Certification Authority	28-11-2026	Client Authentication
Entrust Root Certification Authority	Entrust Root Certification Authority	07-12-2030	Client Authentication
Entrust.net Certification Authority	Entrust.net Certification Authority	24-07-2029	Client Authentication
GlobalSign	GlobalSign	18-03-2029	Client Authentication
GlobalSign	GlobalSign	19-01-2038	Client Authentication
GlobalSign Code Signing Root	GlobalSign Code Signing Root R45	18-03-2045	Code Signing
GlobalSign Root CA	GlobalSign Root CA	28-01-2028	Client Authentication
Go Daddy Class 2 Certification Authority	Go Daddy Class 2 Certification Authority	29-06-2034	Client Authentication
Go Daddy Root Certificate Authority	Go Daddy Root Certificate Authority	01-01-2038	Client Authentication
ISRG Root X1	ISRG Root X1	04-06-2035	Client Authentication
Microsoft Authenticode(tm) Root	Microsoft Authenticode(tm) Root	01-01-2000	Secure Email, Code ..
Microsoft ECC Product Root Certificate	Microsoft ECC Product Root Certificate	28-02-2043	<All>
Microsoft ECC TS Root Certificate	Microsoft ECC TS Root Certificate	28-02-2043	<All>
Microsoft Identity Verification Root	Microsoft Identity Verification Root	17-04-2045	Code Signing, Time..
Microsoft Root Authority	Microsoft Root Authority	31-12-2020	<All>
Microsoft Root Certificate Authority	Microsoft Root Certificate Authority	10-05-2021	<All>

Trusted Root Certification Authorities store contains 44 certificates.

# RSA Details

## Recap: Modulo Arithmetic

- $x \bmod n = \text{remainder of } x \text{ when divide by } n$
- Modulo Arithmetic facts:
  - $[(a \bmod n) + (b \bmod n)] \bmod n = (a+b) \bmod n$
  - $[(a \bmod n) - (b \bmod n)] \bmod n = (a-b) \bmod n$
  - $[(a \bmod n) * (b \bmod n)] \bmod n = (a*b) \bmod n$
- thus
  - $(a \bmod n)^d \bmod n = a^d \bmod n$
- example:  $a = 14$ ,  $n = 10$ ,  $d = 2$ :  
LHS:  $(a \bmod n)^d \bmod n = 4^2 \bmod 10 = 6$   
RHS:  $a^d \bmod n = 14^2 \bmod 10 = 196 \bmod 10 = 6$

## RSA: getting ready

- Message: sequence of bits
- Each bit pattern can be uniquely represented by an integer number
- Encrypting a message is equivalent to encrypting a number
- example:
  - $m = \textbf{10010001}$ . This message is uniquely represented by the decimal number **145**.
- To encrypt  $m$ ,
  - We encrypt the corresponding number (eg 145), which gives a new number (the ciphertext)

## RSA: Creating public/private key pair

1. Choose two large prime numbers  $p, q$ . (e.g., 1024 bits each)
2. Compute  $n = pq, z = (p-1)(q-1)$
3. Choose a **small  $e$**  (with  $e < n$ ) that has no common factors with  $z$  ( $e, z$  are “relatively prime”).
4. choose  $d$  ( $\neq e$ ) such that  $ed-1$  is exactly divisible by  $z$ . (in other words:  $ed \bmod z = 1$  ).
5. **public** key is  $(n, e)$ . **private** key is  $(n, d)$ .

$\overbrace{K_B^+}$

$\overbrace{K_B^-}$

## RSA: encryption, decryption

1. Given  $(n,e)$  and  $(n,d)$  as computed above
2. To encrypt message  $m (< n)$ , compute ciphertext  $c$

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

3. To decrypt received ciphertext,  $c$ , compute

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

Magic happens!  $m = (m^e \bmod n)^d \bmod n$

## RSA Example: Bob creates public/private key pair

1. Choose two large prime numbers  $p, q$ . ( $\textcolor{blue}{p = 5, q = 7}$ )
2. Compute  $n = pq, z = (p-1)(q-1)$  ---  $\textcolor{blue}{n = 35, z = 24}$
3. Choose  $e$  (with  $e < n$ ) that has no common factors with  $z$  ( $e, z$  are “relatively prime”).  $\textcolor{blue}{e = 5}$
4. choose  $d$  such that  $ed - 1$  is exactly divisible by  $z$ . (in other words:  $ed \bmod z = 1$ ).  $\textcolor{blue}{d = 29}$
5. *public* key is  $(\textcolor{red}{n}, e)$ . *private* key is  $(\textcolor{red}{n}, d)$ .  
 $(\textcolor{blue}{35}, 5)$                      $(\textcolor{blue}{35}, 29)$

To encrypt a message  $\textcolor{blue}{m = 12}$ , Alice uses Bob's Public Key  $(\textcolor{blue}{35}, 5)$  and computes

$$c = m^e \bmod n = 12^5 \bmod 35 = 248832 \bmod 35 = 17$$

To decrypt the ciphertext  $c = 17$ , Bob uses his Private Key  $(35, 29)$  and computes

$$\begin{aligned} c &= c^d \bmod n = 17^{29} \bmod 35 = \\ &481968572106750915091411825223071697 \bmod 35 = 12 \end{aligned}$$

## Why does RSA work?

- We must show that for any cipher  $c$ ,  $c^d \bmod n = m$ , where cipher  $c = m^e \bmod n$
- fact: for any  $x$  and  $y$ :  $x^y \bmod n = x^{(y \bmod z)} \bmod n$ 
  - where  $n = pq$  and  $z = (p-1)(q-1)$
- thus,
$$\begin{aligned} c^d \bmod n &= (m^e \bmod n)^d \bmod n \\ &= m^{ed} \bmod n \\ &= m^{(ed \bmod z)} \bmod n \\ &= m^1 \bmod n \\ &= m \end{aligned}$$

## RSA: another important property

The following property will be *very* useful later:

$$K_B^-(K_B^+(m)) = K_B^+(K_B^-(m)) = m$$

   
use public key      use private key  
first, followed by      first, followed by  
private key      public key

*result is the same!*

$$\begin{aligned} (m^e \bmod n)^d \bmod n &= m^{ed} \bmod n \\ &= m^{de} \bmod n \\ &= (m^d \bmod n)^e \bmod n \end{aligned}$$

## Why is RSA secure?

- Suppose you know Bob's public key  $(n, e)$ . How hard is it to determine  $d$ ?
- Essentially need to find factors of  $n$  without knowing the two factors  $p$  and  $q$
- Finding Prime Factors of a large number is **computationally HARD**

## **Questions and Discussions**

*Email:* venkateswaranr.comp@coeptech.ac.in