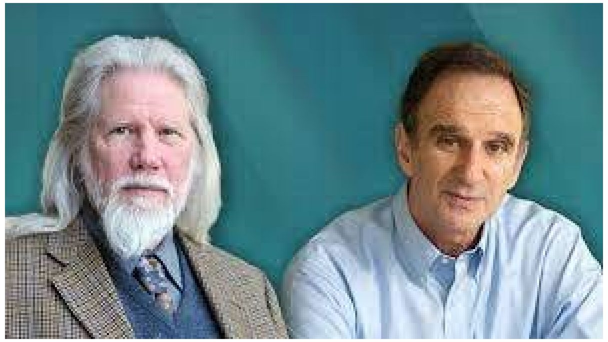


Public Key Cryptography and RSA

By

Prof. Muhammad Iqbal Bhat
GDC Beerwah

Some unanswered questions on symmetric cryptosystems

- **Key management: changing the secret key or establishing one is nontrivial**
  - Change the keys two users share (should be done reasonably often)
  - Establish a secret key with somebody you do not know and cannot meet in person: (e.g., visiting secure websites such as e-shops)
  - This could be done via a trusted Key Distribution Center (details in a future lecture)
  - Can (or should) we really trust the KDC?
  - “What good would it do after all to develop impenetrable cryptosystems, if their users were forced to share their keys with a KDC that could be compromised by either burglary or subpoena?” –Diffie, 1988
- **Digital signatures: one should make sure that a message came *infact from the claimed sender***

A breakthrough idea

- Rather than having a secret key that the two users must share, each users has **two keys**
  - **One key is secret and he is the only one who knows it**
  - **The other key is public and anyone who wishes to send him a message uses that key to encrypt the message**
  - Diffie and Hellman first (publicly) introduced the idea in 1976 –this was radically different than all previous efforts
  - NSA claims to have known it sine mid-1960s!
  - Communications-Electronic Security Group (the British counterpart of NSA) documented the idea in a classified report in 1970

A word of warning

- Public-key cryptography complements rather than replaces symmetric cryptography
- There is nothing in principle to make public-key crypto more secure than symmetric crypto
- Public-key crypto does not make symmetric crypto obsolete: it has its advantages but also its (major) drawbacks such as speed
- Due to its low speed, it is mostly confined to key management and digital signatures

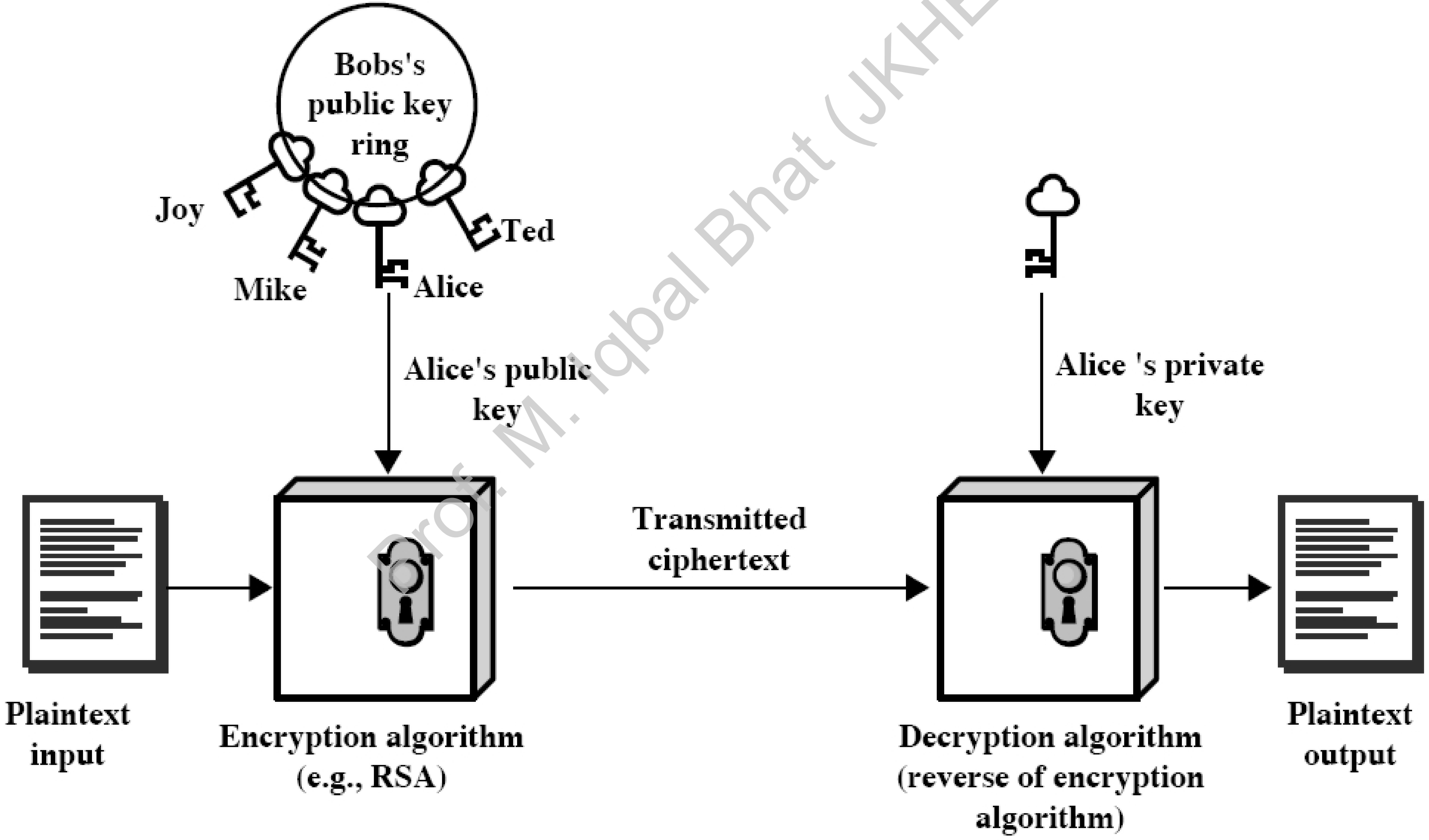
The idea of public-key cryptography

- The concept was proposed in 1976 by Diffieand Hellman although no practical way to design such a system was suggested
- Each user has two keys: one encryption key that he makes public and one decryption key that he keeps secret
  - Clearly, it should be computationally infeasible to determine the decryption key given only the encryption key and the cryptographic algorithm
- Some algorithms (such as RSA) satisfy also the following useful characteristic:
  - Either one of the two keys can be used for encryption –the other one should then be used to decrypt the message
- *First we will investigate the concept with no reference yet to practical design of a public-key system*

Essential steps in public-key encryption

- Each user generates a pair of keys to be used for encryption and decryption
- Each user places one of the two keys in a public register and the other key is kept private
- If B wants to send a confidential message to A, B encrypts the message using A's public key
- When A receives the message, she decrypts it using her private key
  - Nobody else can decrypt the message because that can only be done using A's private key
  - Deducing a private key should be infeasible
- If a user wishes to change his keys –generate another pair of keys and publish the public one: no interaction with other users is needed

Bob sends an encrypted message to Alice



(a) Encryption

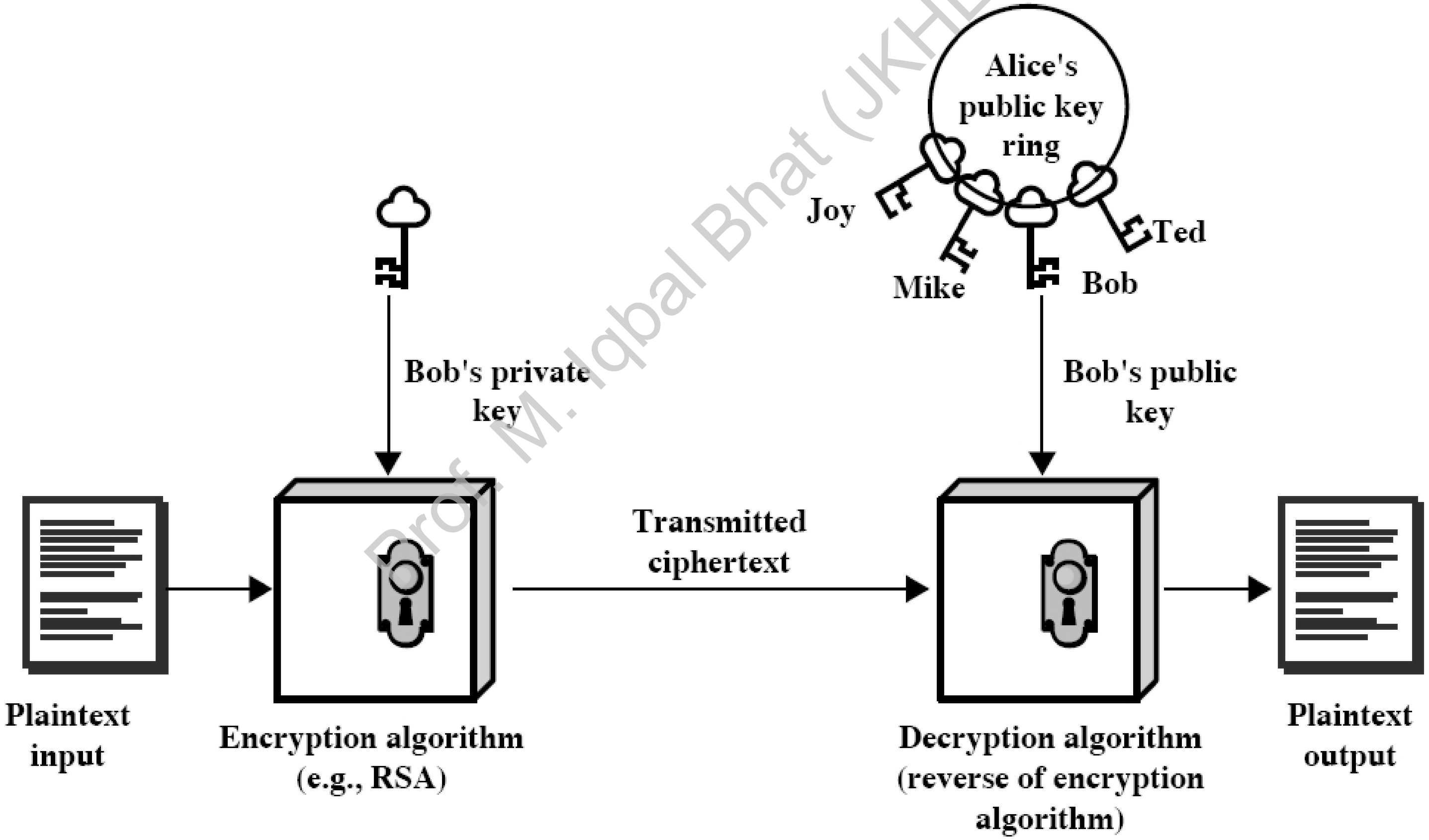
Some notation

- The public key of user A will be denoted KU_A
- The private key of user A will be denoted KR_A
- Encryption method will be a function E
- Decryption method will be a function D
- If B wishes to send a plain message X to A, then he sends the cryptotext $Y=E(KU_A,X)$
- The intended receiver A will decrypt the message: D(KRA,Y)=X

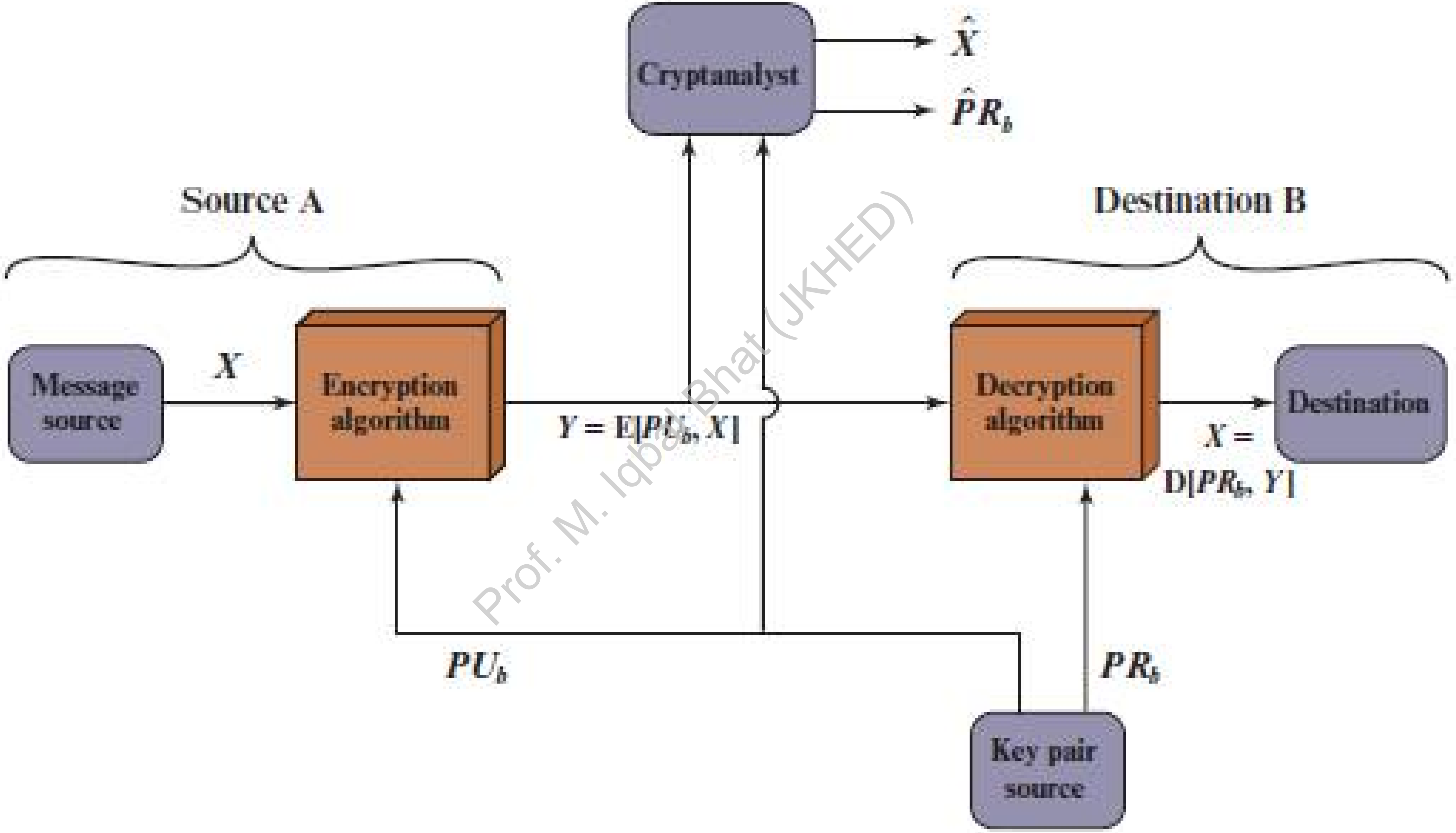
A first attack on the public-key scheme –authenticity

- **Immediate attack on this scheme:**
- **An attacker may impersonate user B: he sends a message E(KU_A,X) and claims in the message to be B –A has no guarantee this is so**
  - *This was guaranteed in classical cryptosystems simply through knowing the key (only A and B are supposed to know the symmetric key)*
- The **authenticity of user B can be established as follows:**
- B will encrypt the message using his private key: Y=E(KR_B,X)
- This shows the authenticity of the sender because (supposedly) he is the only one who knows the private key
- The entire encrypted message serves as a digital signature
  - *Note: this may not be the best possible solution: ideally, digital signatures should be rather small so that one can preserve many of them over a long period of time*
- Better schemes will be presented a couple of lectures on

A scheme to authenticate the sender of the message



(b) Authentication


Cryptanalyst
X̂
P̂R_b
Source A
Destination B
Message source
X
Encryption algorithm
Y = E[PU_b, X]
Decryption algorithm
Destination
X = D[PR_b, Y]
PU_b
PR_b
Key pair source

Encryption and authenticity

- Still a drawback: the scheme on the previous slide authenticate but does not ensure security: anybody can decrypt the message using B’s public key
- One can provide both authentication and confidentiality using the public-key scheme twice:
  - B encrypts X with his private key: $Y=E(KR_B,X)$
  - B encrypts Y with A’s public key: $Z=E(KU_A,Y)$
  - A will decrypt Z (and she is the only one capable of doing it): $Y=D(KR_A,Z)$
  - A can now get the plaintext and ensure that it comes from B (he is the only one who knows his private key): decrypt Y using B’s public key: X=E(KUB,Y)

Applications for public-key cryptosystems

1. **Encryption/decryption: sender encrypts the message with the receiver's public key**
2. **Digital signature: sender "signs" the message (or a representative part of the message) using his private key**
3. **Key exchange: two sides cooperate to exchange a secret key for later use in a secret-key cryptosystem**

Requirements for public-key cryptosystems

- *Generating a key pair (public key, private key) is computationally easy*
- *Encrypting a message using a known key (his own private or somebody else's public) is computationally easy*
- *Decrypting a message using a known key (his own private or somebody else's public) is computationally easy*
- Knowing the public key, it is *computationally infeasible for an opponent to deduce the private key*
- Knowing the public key and a ciphertext, it is *computationally infeasible for an opponent to deduce the private key*
- *Useful extra feature: encryption and decryption can be applied in any order:*
- E(KUA, D(KRA,X)) =D(KRA, E(KUA, X)

RSA

- One of the first proposals on implementing the concept of public-key cryptography was that of Rivest, Shamir, Adleman–1977: RSA
- The RSA scheme is a block cipher in which the plaintext and the ciphertext are integers between 0 and n-1 for some fixed n
  - Typical size for n is 1024 bits (or 309 decimal digits)
  - To be secure with today's technology size should between 1024 and 2048 bits
- Idea of RSA: it is a difficult math problem to factorize (large)integers
  - **Choose p and q odd primes, n=pq**
  - **Choose integers d,e such that M^ed=M mod n, for all M<n**
  - **Plaintext**: block of k bits, where 2k<n≤2k+1–can be considered a number M with M<n
  - **Encryption**: C=M^e mod n
  - **Decryption**: C^d mod n = M^de mod n = M
  - **Public key**: KU={e,n}
  - **Private key:**KR={d,n}
- **Question: How do we find d,e?**
  - **Answer: Number Theory!**

RSA Background Mathematics

Modulo Congruence

$$b \equiv c \pmod{m}.$$

It refers to the relationship between two integers that have the same remainder when divided by a given positive integer, which is known as the modulus.

In other words, two integers a and b are said to be congruent modulo n, denoted as a ≡ b (mod n), if they have the same remainder when divided by n.

$$14 \equiv 2 \pmod{6}$$

$$25 \equiv 19 \pmod{3}$$

Leonhard Euler

Euler Totient Function

$$\phi\,(24) = 8$$

The Euler totient function, denoted as φ(n), is an important function in number theory.

It plays a critical role in the RSA algorithm, which is a widely-used public-key cryptosystem.

The Euler totient function is defined as the number of positive integers less than or equal to n that are relatively prime to n.

It is denoted as φ(n), where n is a positive integer.

For example, if n = 10, then φ(10) = 4, because the only positive integers less than or equal to 10 that are relatively prime to 10 are 1, 3, 7, and 9.

Euler Totient Function Properties

The Euler totient function has several important properties that make it useful in number theory and cryptography.

The first property is that $\varphi(n)$ is always even for $n > 2$.

The second property is that if p is a prime number, then $\varphi(p) = p-1$.

The third property is that if p and q are distinct prime numbers, then $\varphi(pq) = (p-1)(q-1)$.

The fourth property is that if n is a positive integer and a is a positive integer relatively prime to n, then $\varphi(an) = \varphi(n) \times a^{(k-1)}$, where k is the highest power of a that divides n

Euler Totient Function

$$\phi(24) = 8$$

- To calculate the Euler totient function, we need to find prime factors, this is the hardest part, for a big number it is very hard to find all of its prime factors, this is the security in RSA which is hard to factor a large number.
- Easy to see that for any two primes **p,q**, **φ(pq)=(p-1)(q-1)**
- **Euler's theorem**: for any relatively prime integers a, n we have $\mathbf{a^{\varphi(n)} \equiv 1 \bmod n}$
- This theorem is a key component of the RSA algorithm, which uses modular exponentiation to encrypt and decrypt messages.

Prime Factorization

399760339672075088299557194017994567848983719760280566510461694198205436599518040051609502918604597414419179645650
998925708942577701920324033371575390574186156167631547516696710042520586452798497447384811672279083833231162063897
143659010062936034531839664028520772113299916962527889285003765714831019968123230958171768533632424964448875962377
078042837494063139104883994762820590497060974377110730411808196584312443757015404013600816108244776781822715222280
487948885602891430551650343684833697508477347034068075581190428464004788029400785827187277658551697056720485143717
011331224562985166784945961020525526670307756664656683691096233395890399071292575368183955406298948459615017178976
589275589240697212677388363838696819778600947582732866303216309460317451170184747470967187230774940537917802928905
780901527837799407146052986645573549120932710534109085795989517987504933214154915043064847597748633594255240081708
044616224751379558415455307418881346824082708674251261376084842828103422282525194432486586473790101955453982199098
969325481320752971192060087421984226757415176637661276628865634427536174713218080266452709631427460305013127571386
648966579247634232317435989105989088161949040735570263443109044973232705707311002576576678066683072917328539925663
359737373307306613006695604852804908372900687534899618296547910551795990408196228661988812135959671312027358656432
208648655705813883934230357238160797349324322287397100734318061354951319796504264551931998473813724469978244667900
6857094084846747764429611563524533411145800040687260272920166093684984120202858303471568982546311184047115152560655
979581537132837581220281821041744241572142375094457837092090654952385531509535755916961007563743885420187608455761
720979654622842939408042375675576164796164270466163041533964224602964232818613284083747007563616277140254890531672
427918355509701575111655665528460924504606631314533835936660304462796162754132587480832546563640204763549008836692
113332281875801556192253737734859069239263227304774684564963747914203485939664199754026900386570369177890683431111
878984895776014203495411474180932505480291873347967036941169598535751087933696095266461130499714914324990135426188
107314102245762596998117708829043939108430744757884808328588684893719008668535005610995904449954416009020623666341
212979150966420910783836965379477622460062900392959714845992745236856069681738596300132408922794225624048358400945
23853941592466542312875122105855473259392197682657054698911591461443581

8192 bit Number

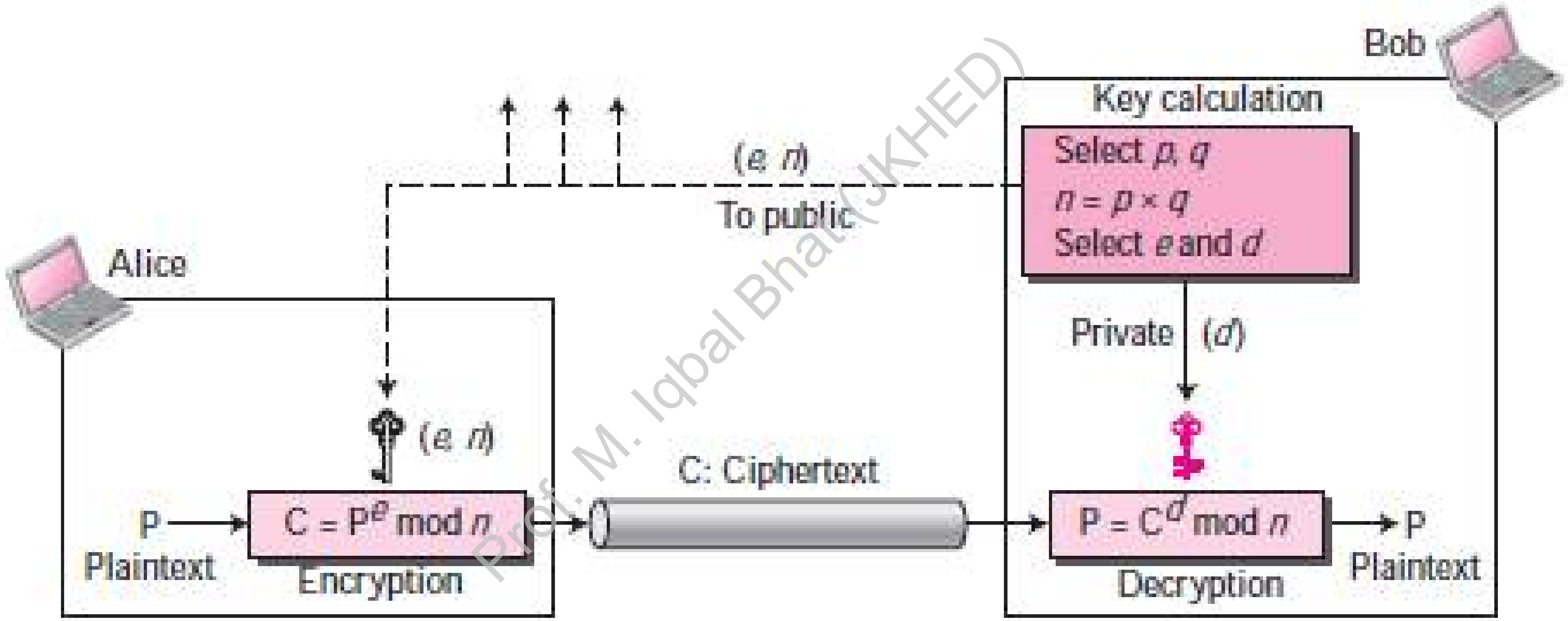
Back to RSA

Key Generation by Alice	
Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n)=(p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption by Bob with Alice's Public Key	
Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption by Alice with Alice's Private Key	
Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

RSA algorithm



RSA Algorithm

Encrypt $m^e \bmod n = c$

$\varphi = (p - 1)(q - 1)$

$x^{\varphi} \bmod n = 1$

$e * d \bmod \varphi = 1$

Decrypt $c^d \bmod n = m$

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

Encrypt $m^e \bmod n = c$

Decrypt $c^d \bmod n = m$

$m = 42$

$p = 61, q = 53, e = 17,$
$n = 3233, d = 2753$

Encrypt $42^{17} \bmod 3233 = c$

Decrypt $2557^{2753} \bmod 3233 = m$

Example

- Key generation
- Select primes p=17, q=11
- Compute n=p*q=187
- Compute φ(n)=(p-1)(q-1)=160
- Select e=7
- Compute d: d=23 (use the *extended Euclid's algorithm*)
- Pr_k={7,187}
- Pu_k={23,187}

- **Encrypt** M=88: 88^7mod 187
- 88^7mod 187 = 11
- **Decrypt** C=11: 11^{23}mod 187
- M=11^{23} mod 187=88

Attacking RSA

- Brute force attacks: try all possible private keys
- As in the other cases defend using large keys: nowadays integers between 1024 and 2048 bits
- **Mathematical attacks**
- Factor n into its two primes p,q: this is a hard problem for large n
  - Challenges by RSA Labs to factorize large integers
  - Smallest unsolved challenge: 704 bits
- Determine φ**(n)** directly without first determining p,q: this math problem is equivalent to factoring
- Determine d directly, without first determining φ**(n)**: this is believed to be at least as difficult as factoring
- Suggestions for design
- The larger the keys, the better but also the slower the algorithm
- Choosing p,q badly may weaken the algorithm
  - p,q should differ in length by only a few bits: for a 1024-bit key, p,q should be on the order of magnitude 1075to 10100
  - p-1 and q-1 should both contain a large prime factor
  - gcd(p-1,q-1) should be small
  - hd should be larger than n1/4**RSA**