Cryptography and Network Security Chapter 9

Fourth Edition

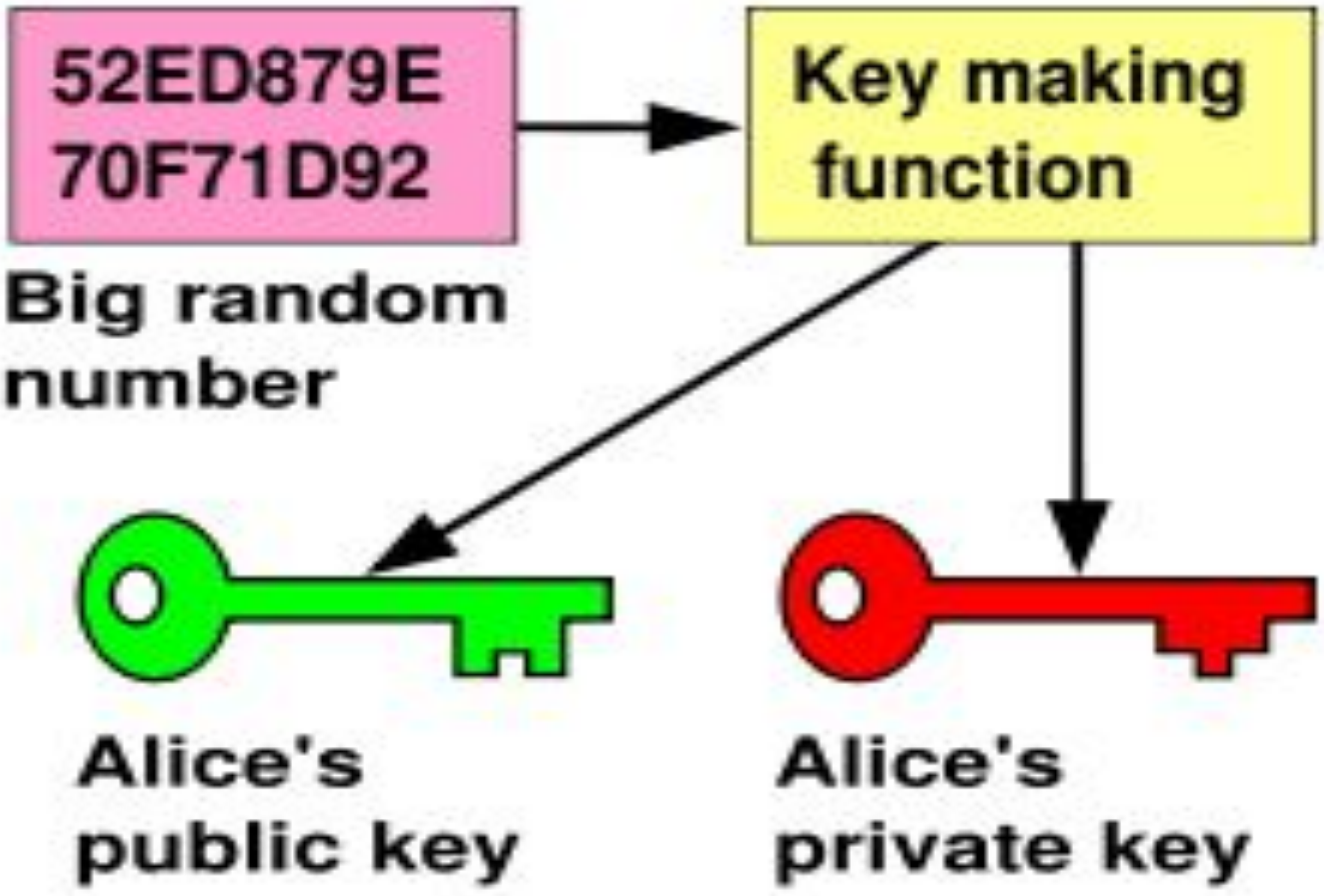
by William Stallings

Chapter 9 – Public Key Cryptography and RSA

Every Egyptian received two names, which were known respectively as the true name and the good name, or the great name and the little name; and while the good or little name was made public, the true or great name appears to have been carefully concealed.

—*The Golden Bough*, Sir James George Frazer

Alice



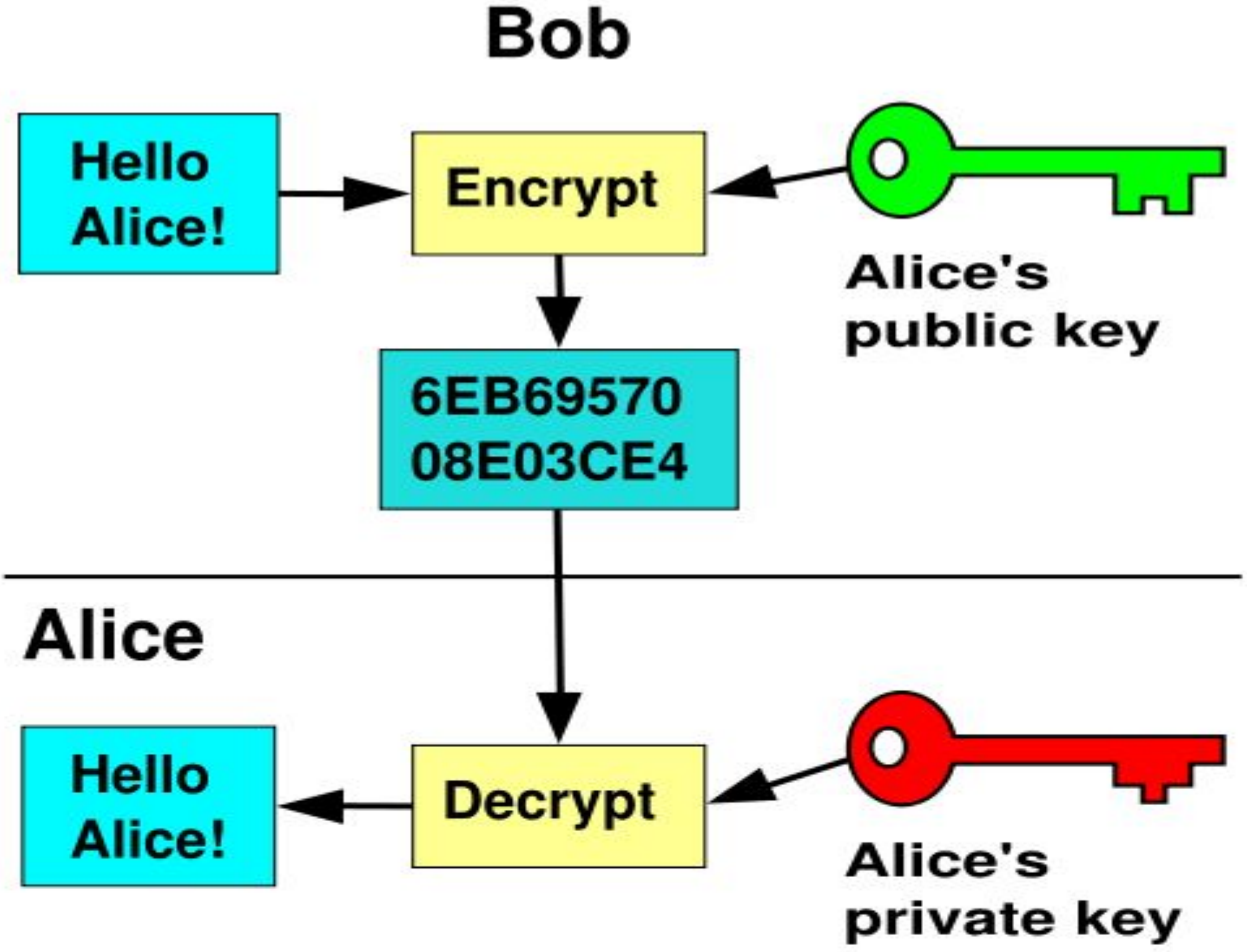
52ED879E
70F71D92

Big random number

Key making function

Alice's public key

Alice's private key



Bob

Hello Alice!

Encrypt

Alice's public key

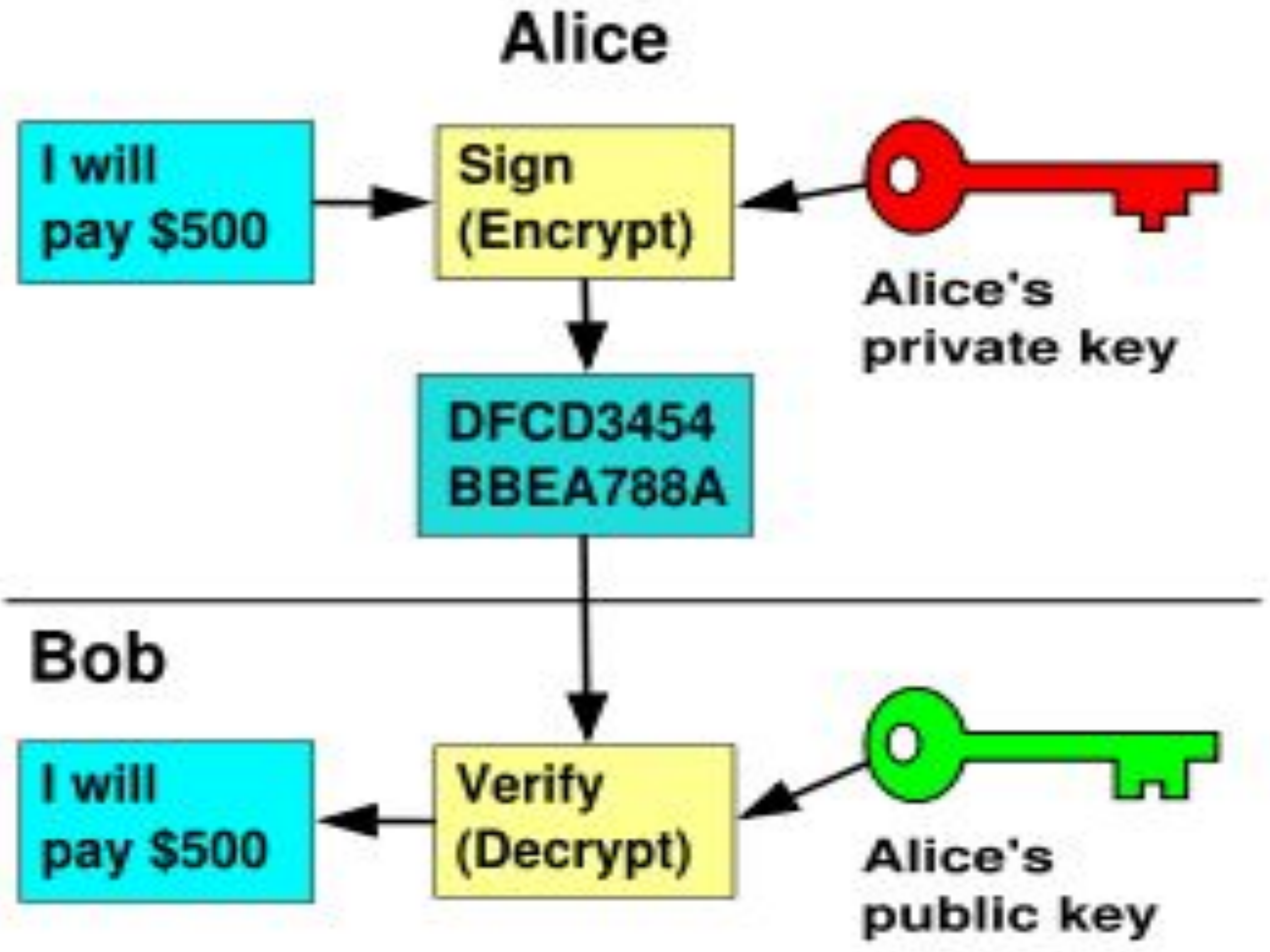
6EB69570 08E03CE4

Alice

Hello Alice!

Decrypt

Alice's private key



Alice

I will pay $500

Sign (Encrypt)

Alice's private key

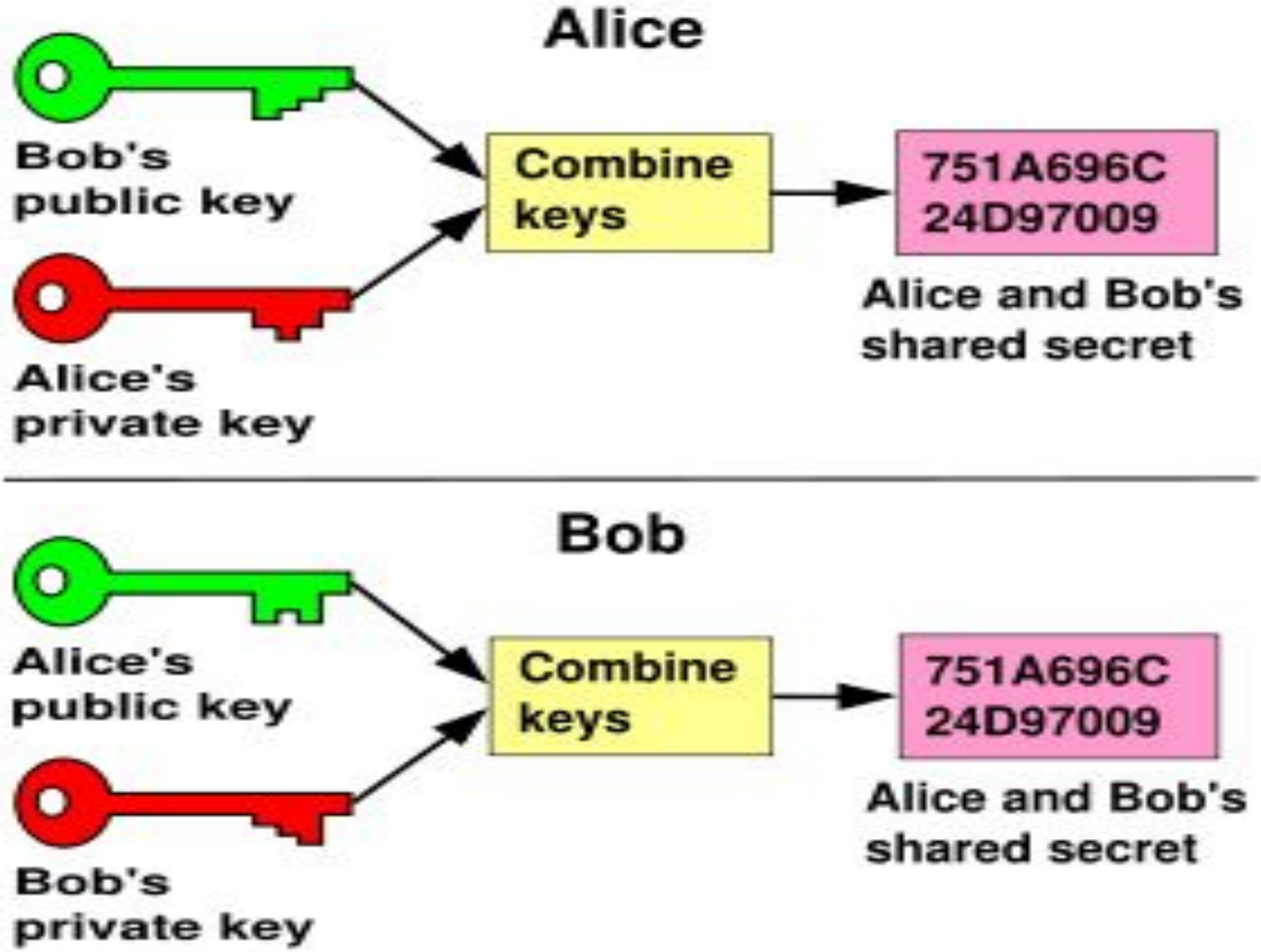
DFCD3454 BBEA788A

Bob

I will pay $500

Verify (Decrypt)

Alice's public key



Alice

Bob's public key

Alice's private key

Combine keys

751A696C 24D97009

Alice and Bob's shared secret

Bob

Alice's public key

Bob's private key

Combine keys

751A696C 24D97009

Alice and Bob's shared secret

Private-Key Cryptography

- traditional **private/secret/single key** cryptography uses **one** key
- shared by both sender and receiver
- if this key is disclosed communications are compromised
- also in **symmetric**, parties are equal
- hence does not protect sender from receiver forging a message & claiming is sent by sender

Public-Key Cryptography

- probably most significant advance in the 3000 year history of cryptography
- uses **two** keys – a public & a private key
- **asymmetric** since parties are **not** equal
- uses clever application of number theoretic concepts to function
- complements **rather than** replaces private key crypto

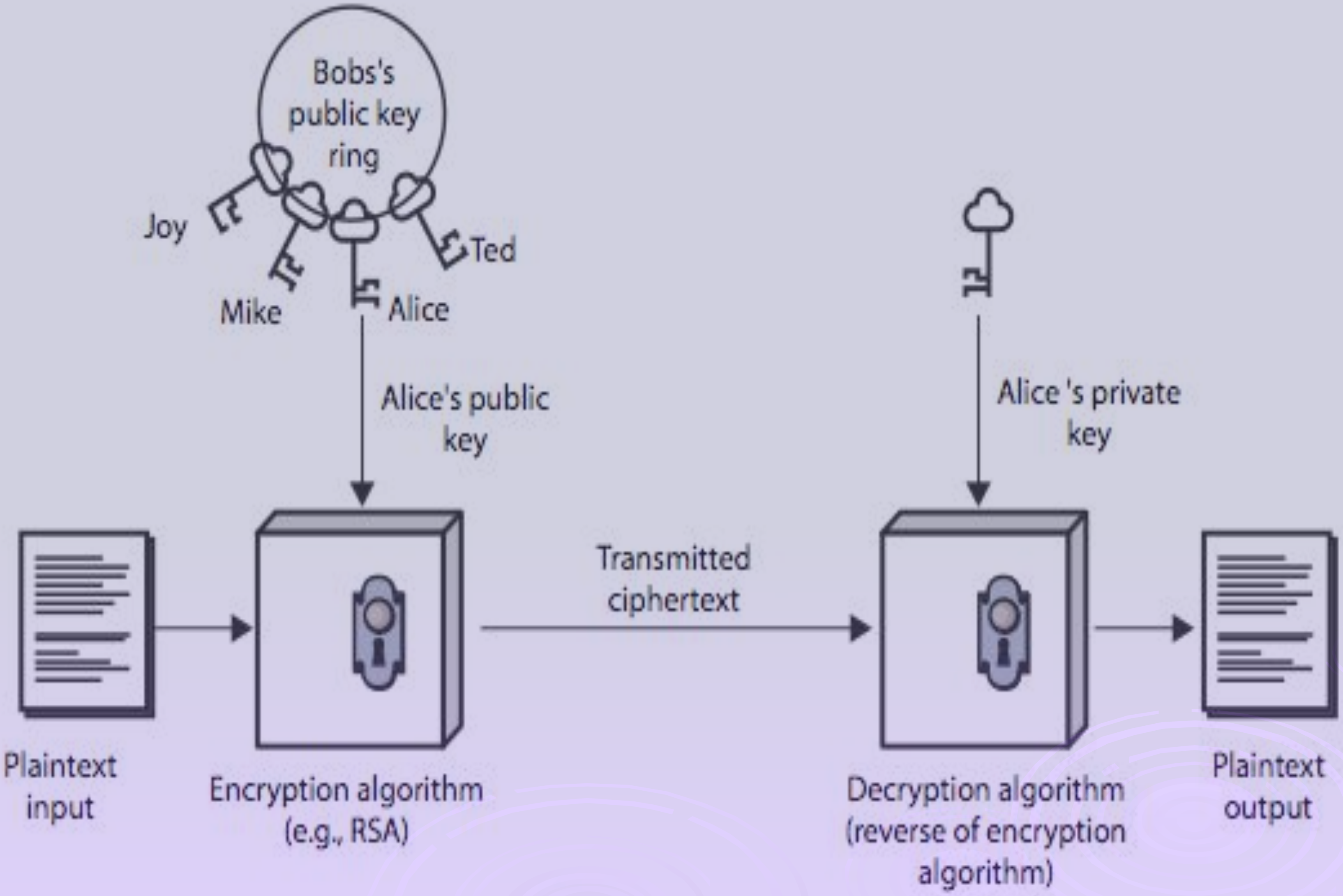
Why Public-Key Cryptography?

- developed to address two key issues:
  - **key distribution** – how to have secure communications in general without having to trust a KDC with your key
  - **digital signatures** – how to verify a message comes intact from the claimed sender
- public invention due to Whitfield Diffie & Martin Hellman at Stanford Uni in 1976
  - known earlier in classified community

Public-Key Cryptography

- **public-key/two-key/asymmetric** cryptography involves the use of **two** keys:
  - a **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
  - a **private-key**, known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures**
- is **asymmetric** because
  - those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures

Public-Key Cryptography



(a) Encryption

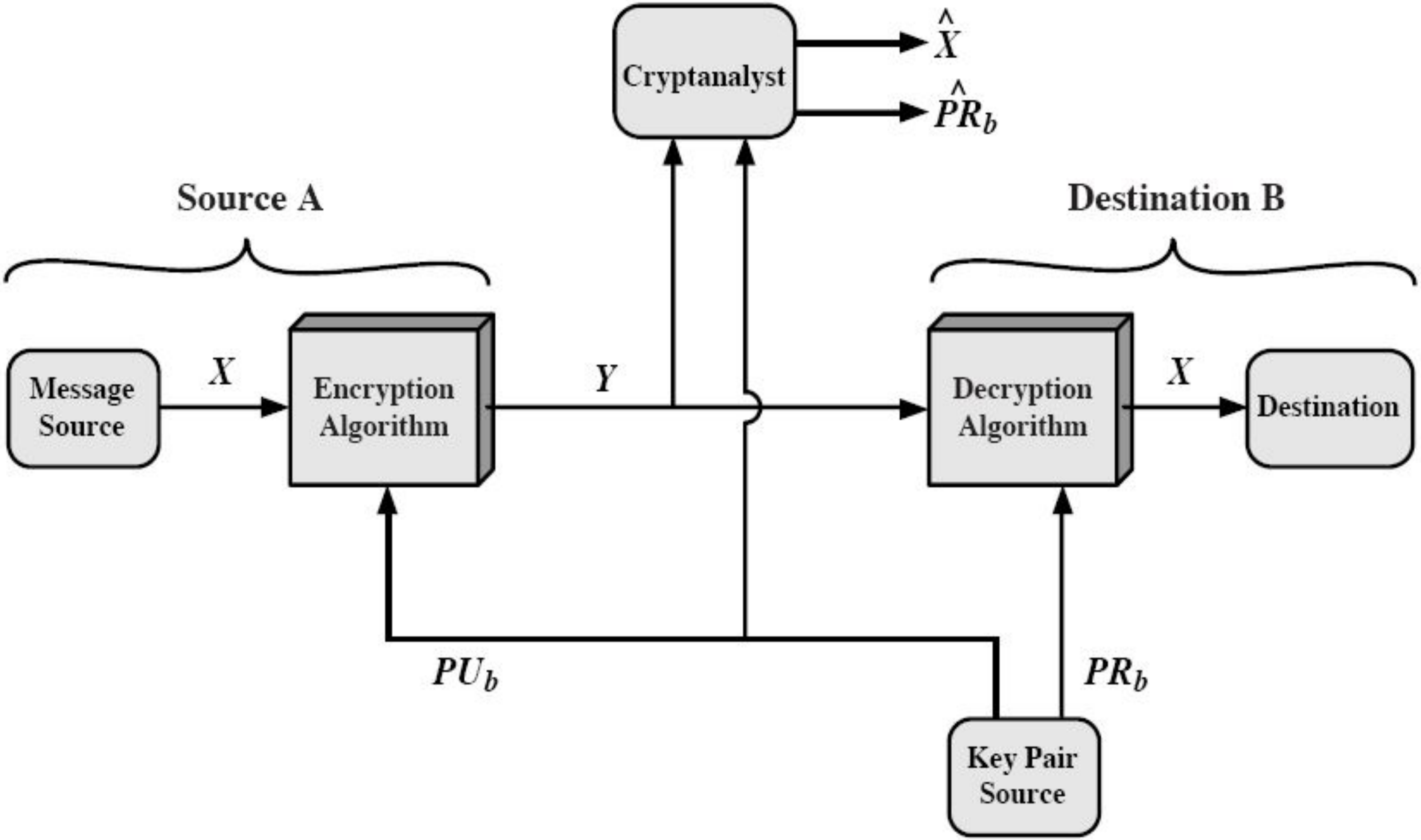
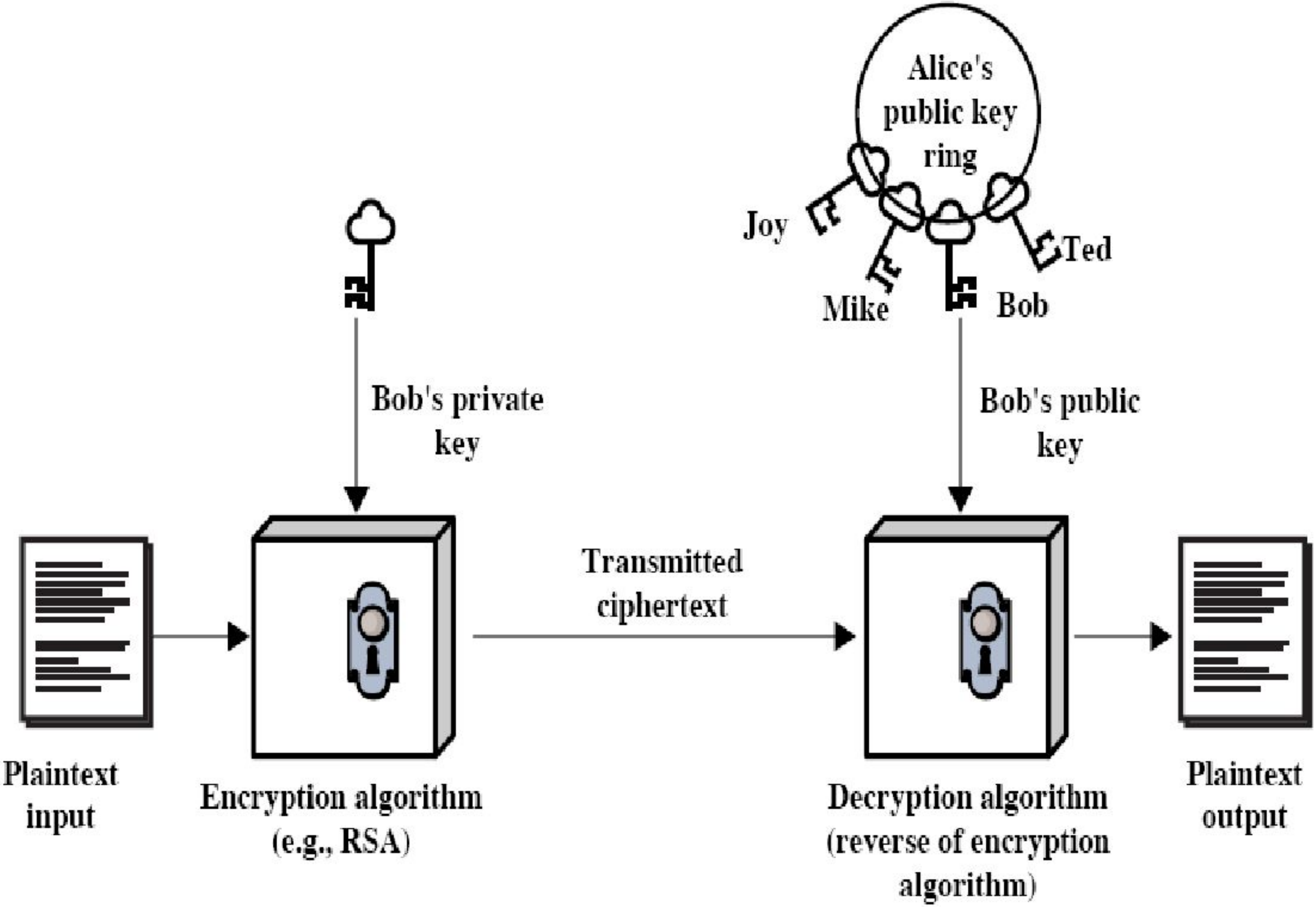


Figure 9.2 Public-Key Cryptosystem: Secrecy



(b) Authentication

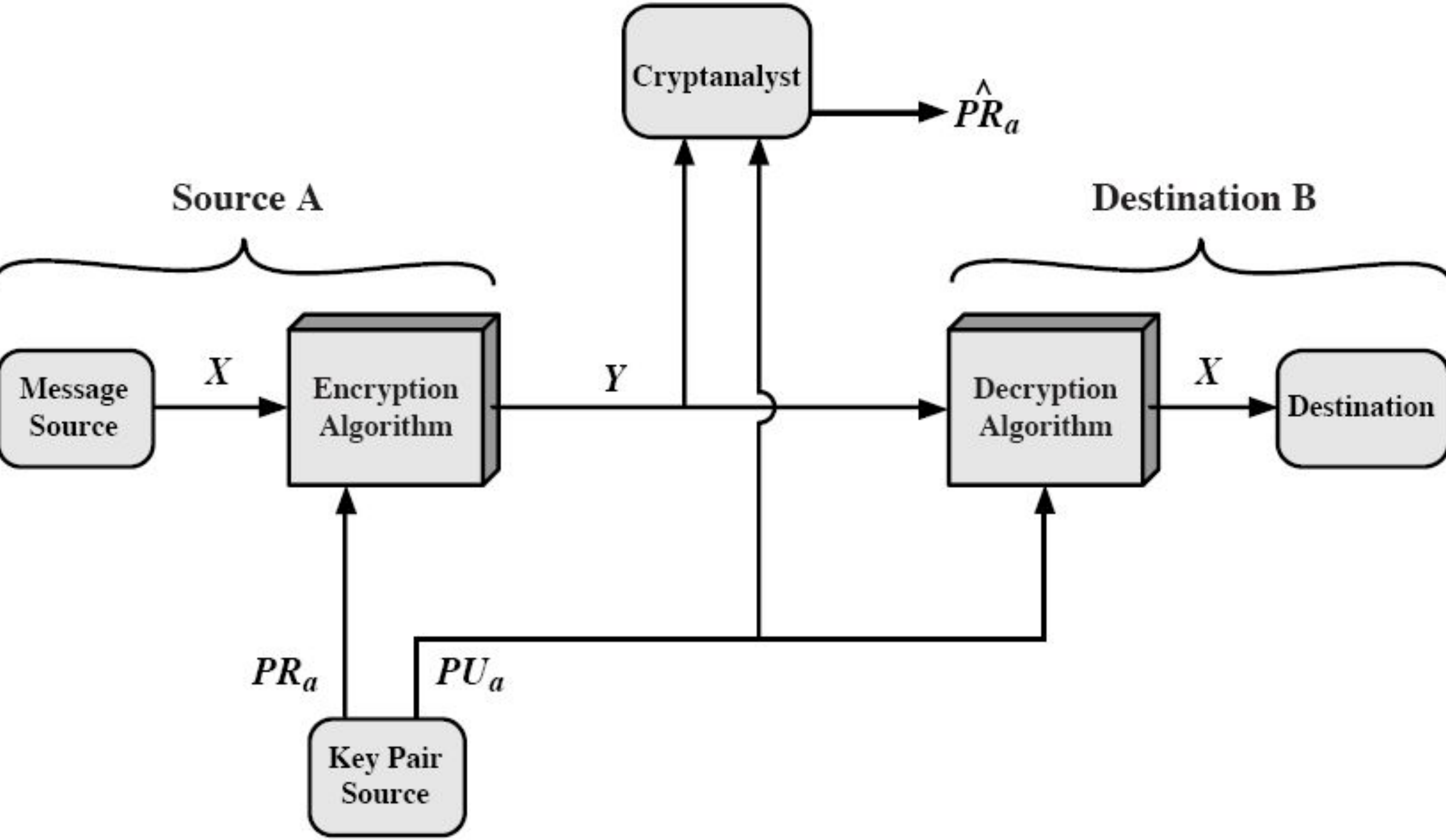
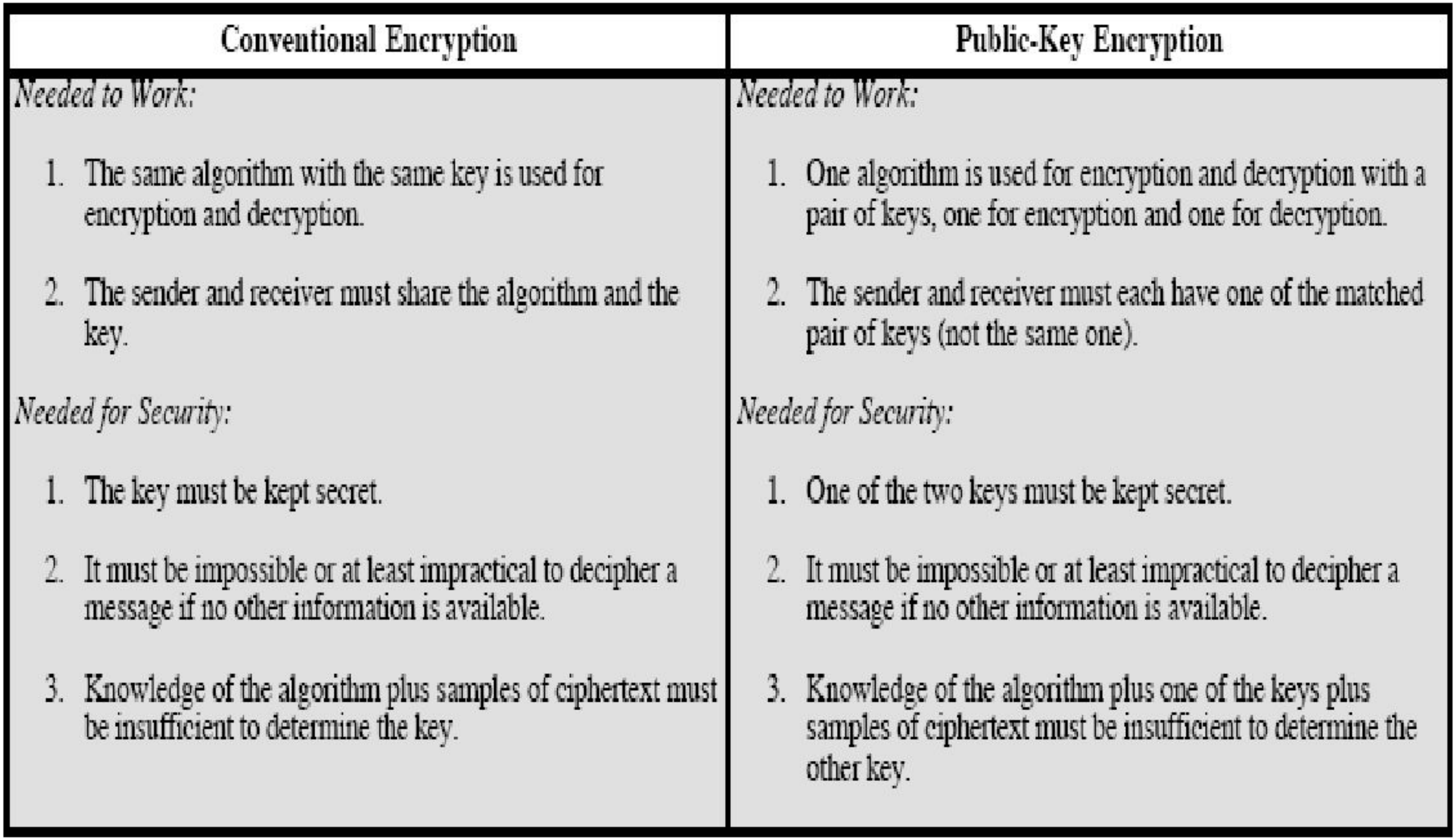


Figure 9.3 Public-Key Cryptosystem: Authentication

Table 9.1 CONVENTIONAL AND PUBLIC-KEY ENCRYPTION

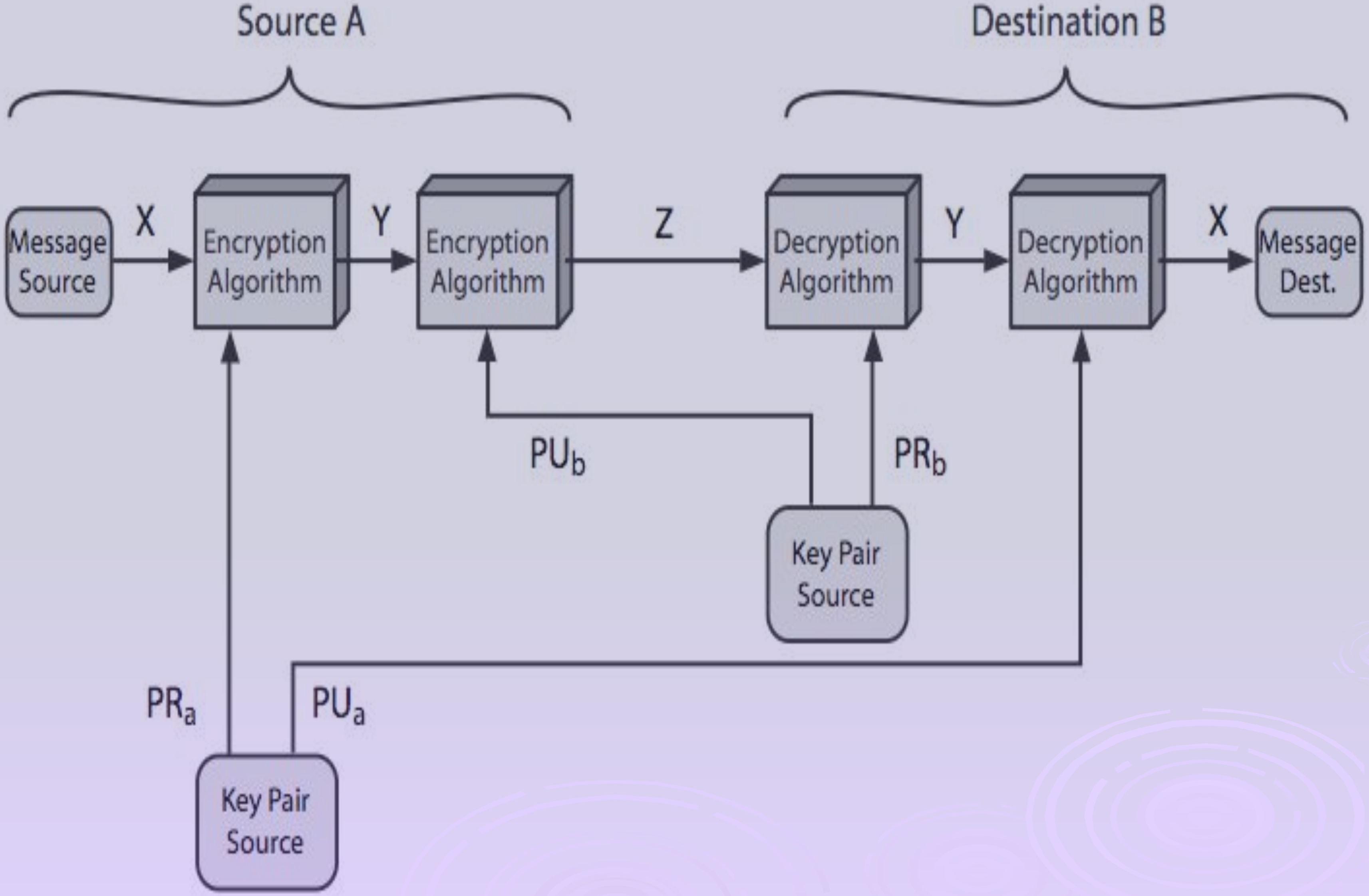


Conventional Encryption	Public-Key Encryption
Needed to Work: 1. The same algorithm with the same key is used for encryption and decryption. 2. The sender and receiver must share the algorithm and the key.	*Needed to Work:* 1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption. 2. The sender and receiver must each have one of the matched pair of keys (not the same one).
Needed for Security: 1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	*Needed for Security:* 1. One of the two keys must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available. 3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

Public-Key Characteristics

- Public-Key algorithms rely on two keys where:
  - it is computationally infeasible to find decryption key knowing only algorithm & encryption key
  - it is computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
  - either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms)

Public-Key Cryptosystems



Public-Key Applications

- can classify uses into 3 categories:
  - **encryption/decryption** (provide secrecy)
  - **digital signatures** (provide authentication)
  - **key exchange** (of session keys)
- some algorithms are suitable for all uses, others are specific to one

Security of Public Key Schemes

- like private key schemes brute force **exhaustive search** attack is always theoretically possible
- but keys used are too large (>512bits)
- security relies on a **large enough** difference in difficulty between **easy** (en/decrypt) and **hard** (cryptanalyse) problems
- more generally the **hard** problem is known, but is made hard enough to be impractical to break
- requires the use of **very large numbers**
- hence is **slow** compared to private key schemes

RSA

- by Rivest, Shamir & Adleman of MIT in 1977
- best known & widely used public-key scheme
- based on exponentiation in a finite (Galois) field over integers modulo a prime
  - nb. exponentiation takes $O((\log n)^3)$ operations (easy)
- uses large integers (eg. 1024 bits)
- security due to cost of factoring large numbers
  - nb. factorization takes $O(e^{\log n \log \log n})$ operations (hard)

RSA Key Setup

- each user generates a public/private key pair by:
- selecting two large primes at random - p, q
- computing their system modulus $n=p.q$
  - note $ø(n)=(p-1)(q-1)$
- selecting at random the encryption key e
  - where $1<e<ø(n)$, $\gcd(e,ø(n))=1$
- solve following equation to find decryption key d
  - $e.d=1 \bmod ø(n)$ and $0≤d≤n$
- publish their public encryption key: PU={e,n}
- keep secret private decryption key: PR={d,n}

RSA Use

- to encrypt a message M the sender:
  - obtains **public key** of recipient $PU=\{e,n\}$
  - computes: $C = M^e \bmod n$, where $0 \le M < n$
- to decrypt the ciphertext C the owner:
  - uses their private key $PR=\{d,n\}$
  - computes: $M = C^d \bmod n$
- note that the message M must be smaller than the modulus n (block if needed)

Why RSA Works

- because of Euler's Theorem:
  - $a^{\varnothing(n)} \bmod n = 1$ where $\gcd(a,n)=1$
- in RSA have:
  - $n=p.q$
  - $\varnothing(n)=(p-1)(q-1)$
  - carefully chose e & d to be inverses $\bmod \varnothing(n)$
  - hence $e.d=1+k.\varnothing(n)$ for some k
- hence :

$$C^{d} = M^{e.d} = M^{1+k.\varnothing(n)} = M^{1}.(M^{\varnothing(n)})^{k}$$

$$= M^{1}.(1)^{k} = M^{1} = M \bmod n$$

Key Generation

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	$d = e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^{e} \bmod n$

Decryption

Ciphertext:	C
Plaintext:	$M = C^{d} \bmod n$

Figure 9.5 The RSA Algorithm

RSA Example - Key Setup

1. Select primes: $p=17$ & $q=11$
2. Compute $n = pq = 17 \text{ x } 11=187$
3. Compute ø$(n)=(p-1)(q-1)=16 \text{ x } 10=160$
4. Select e: gcd(e,160)=1; choose $e=7$
5. Determine d: $de=1 \bmod 160$ and $d < 160$
   Value is d=23 since 23x7=161= 10x16+1
6. Publish public key PU={7,187}
7. Keep secret private key PR={23,187}

RSA Example - En/Decryption

- sample RSA encryption/decryption is:
- given message `M = 88` (nb. `88<187`)
- encryption:

  $C = 88^{7} \bmod 187 = 11$
- decryption:

  $M = 11^{23} \bmod 187 = 88$

Exponentiation

- can use the Square and Multiply Algorithm
- a fast, efficient algorithm for exponentiation
- concept is based on repeatedly squaring base
- and multiplying in the ones that are needed to compute the result
- look at binary representation of exponent
- only takes $O(\log_2 n)$ multiples for number n
  - eg. $7^5 = 7^4.7^1 = 3.7 = 10 \bmod 11$
  - eg. $3^{129} = 3^{128}.3^1 = 5.3 = 4 \bmod 11$

Exponentiation

```
c = 0; f = 1
for i = k downto 0
    do c = 2 x c
       f = (f x f) mod n
    if $b_i$ == 1 then
       c = c + 1
      f = (f x a) mod n
return f
```

i	9	8	7	6	5	4	3	2	1	0
b_i	1	0	0	0	1	1	0	0	0	0
c	1	2	4	8	17	35	70	140	280	560
f	7	49	157	526	160	241	298	166	67	1

Table 9.3 Result of the Fast Modular Exponentiation Algorithm for a^b mod n, where $a = 7$, $b = 560 = 1000110000$, $n = 561$

Efficient Encryption

- encryption uses exponentiation to power e
- hence if e small, this will be faster
  - often choose e=65537 (2^{16}-1)
  - also see choices of e=3 or e=17
- but if e too small (eg e=3) can attack
  - using Chinese remainder theorem & 3 messages with different modulii
- if e fixed must ensure gcd(e,ø(n))=1
  - ie reject any p or q not relatively prime to e

Efficient Decryption

- decryption uses exponentiation to power d
  - this is likely large, insecure if not
- can use the Chinese Remainder Theorem (CRT) to compute mod p & q separately. then combine to get desired answer
  - approx 4 times faster than doing directly
- only owner of private key who knows values of p & q can use this technique

RSA Key Generation

- users of RSA must:
  - determine two primes at random - `p, q`
  - select either `e` or `d` and compute the other
- primes `p,q` must not be easily derived from modulus `n=p.q`
  - means must be sufficiently large
  - typically guess and use probabilistic test
- exponents `e, d` are inverses, so use Inverse algorithm to compute the other

RSA Security

- possible approaches to attacking RSA are:
  - brute force key search (infeasible given size of numbers)
  - mathematical attacks (based on difficulty of computing ø(n), by factoring modulus n)
  - timing attacks (on running of decryption)
  - chosen ciphertext attacks (given properties of RSA)

Factoring Problem

- mathematical approach takes 3 forms:
  - factor `n=p.q`, hence compute `ø(n)` and then d
  - determine `ø(n)` directly and compute d
  - find d directly
- currently believe all equivalent to factoring
  - have seen slow improvements over the years
    - as of May-05 best is 200 decimal digits (663) bit with LS
  - biggest improvement comes from improved algorithm
    - cf QS to GHFS to LS
  - currently assume 1024-2048 bit RSA is secure
    - ensure p, q of similar size and matching other constraints

RSA-640 is factored!

The factoring research team of F. Bahr, M. Boehm, J. Franke, T. Kleinjung continued its productivity with a successful factorization of the challenge number RSA-640, reported on November 2, 2005. The factors [verified by RSA Laboratories] are:

16347336458092538484431338838650908598417836700330
92312181110852389333100104508151212118167511579

and

1900871281664822113126851573935413975471896789968
515493666638539088027103802104498957191261465571

The effort took approximately 30 2.2GHz-Opteron-CPU years according to the submitters, over five months of calendar time. (This is about half the effort for RSA-200, the 663-bit number that the team factored in 2004.)

Table 9.4 Progress in Factorization

Number of Decimal Digits	Approximate Number of Bits	Date Achieved	MIPS-years	Algorithm
100	332	April 1991	7	quadratic sieve
110	365	April 1992	75	quadratic sieve
120	398	June 1993	830	quadratic sieve
129	428	April 1994	5000	quadratic sieve
130	431	April 1996	1000	generalized number field sieve
140	465	February 1999	2000	generalized number field sieve
155	512	August 1999	8000	generalized number field sieve
160	530	April 2003	–	Lattice sieve
174	576	December 2003	–	Lattice sieve
200	663	May 2005	–	Lattice sieve

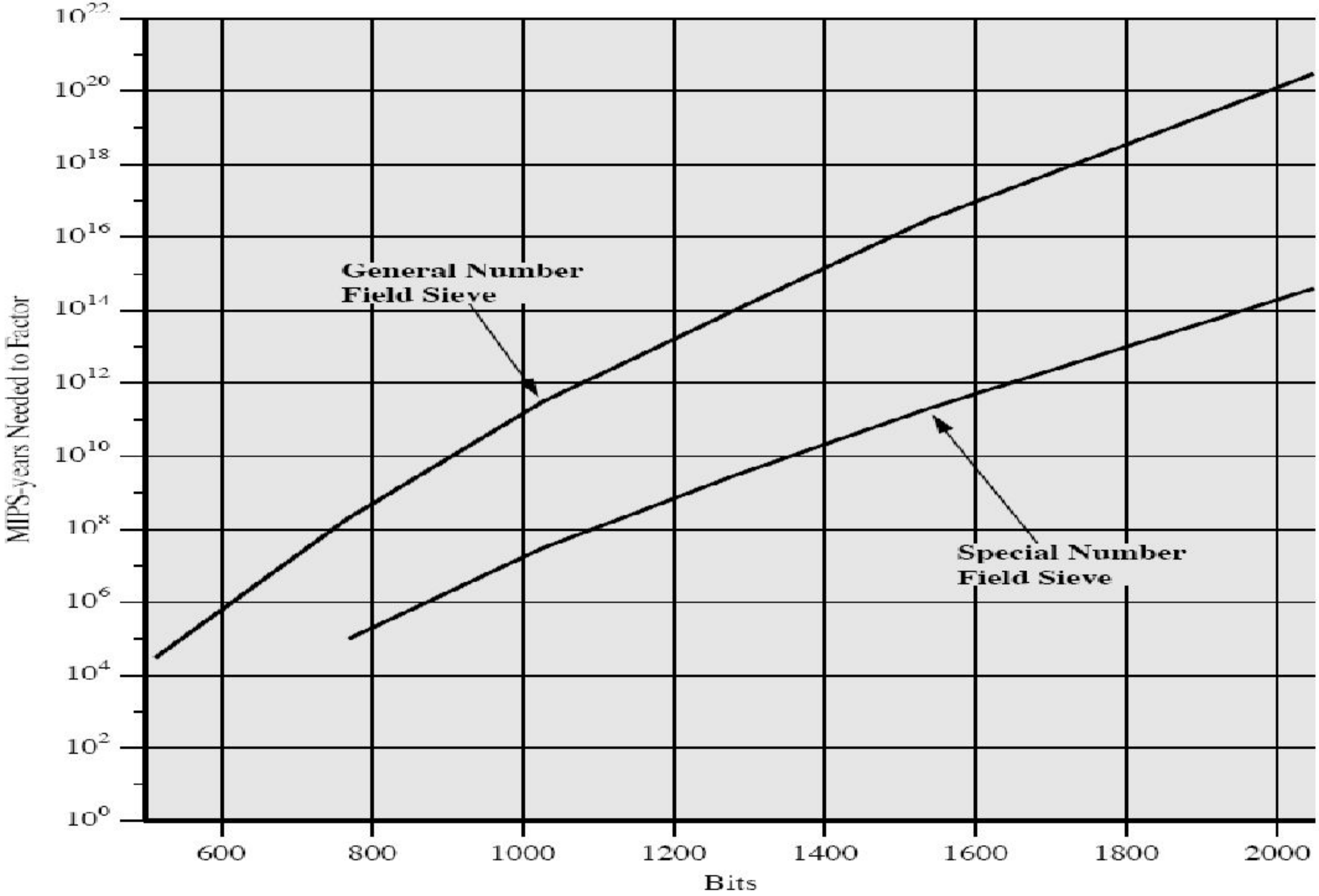


Figure 9.8 MIPS-years Needed to Factor

Timing Attacks

- developed by Paul Kocher in mid-1990’s
- exploit timing variations in operations
  - eg. multiplying by small vs large number
  - or IF's varying which instructions executed
- infer operand size based on time taken
- RSA exploits time taken in exponentiation
- countermeasures
  - use constant exponentiation time
  - add random delays
  - blind values used in calculations

Chosen Ciphertext Attacks

RSA is vulnerable to a Chosen Ciphertext Attack (CCA)

attackers chooses ciphertexts & gets decrypted plaintext back

choose ciphertext to exploit properties of RSA to provide info to help cryptanalysis

- can counter with random pad of plaintext
- or use Optimal Asymmetric Encryption Padding (OASP)

An Example of the RSA Algorithm

1. P = 61 <- first prime number (destroy this after computing E and D)
2. Q = 53 <- second prime number (destroy this after computing E and D)
3. PQ = 3233 <- modulus (give this to others)
4. E = 17 <- public exponent (give this to others)
5. D = 2753 <- private exponent (keep this secret!)
6. Your public key is (E,PQ).
7. Your private key is D.

An Example of the RSA Algorithm

8. The encryption function is:

encrypt(T) = (T^E) mod PQ
= (T^{17}) mod 3233

9. The decryption function is:

decrypt(C) = (C^D) mod PQ
= (C^{2753}) mod 3233

10. To encrypt the plaintext value 123, do this:

encrypt(123) = (123^{17}) mod 3233
= 337587917446653715596592958817679803 mod 3233
= 855

11. To decrypt the ciphertext value 855, do this:

decrypt(855) = (855^{2753}) mod 3233
= 123

An Example of the RSA Algorithm

One way to compute the value of 855^2753 mod 3233 is like this:
2753 = 101011000001 base 2, therefore
2753 = 1 + 2^6 + 2^7 + 2^9 + 2^11
= 1 + 64 + 128 + 512 + 2048
Consider this table of powers of 855:

$855^{1} = 855 \pmod{3233}$
$855^{2} = 367 \pmod{3233}$
$855^{4} = 367^{2} \pmod{3233} = 2136 \pmod{3233}$
$855^{8} = 2136^{2} \pmod{3233} = 733 \pmod{3233}$
$855^{16} = 733^{2} \pmod{3233} = 611 \pmod{3233}$
$855^{32} = 611^{2} \pmod{3233} = 1526 \pmod{3233}$
$855^{64} = 1526^{2} \pmod{3233} = 916 \pmod{3233}$
$855^{128} = 916^{2} \pmod{3233} = 1709 \pmod{3233}$
$855^{256} = 1709^{2} \pmod{3233} = 1282 \pmod{3233}$
$855^{512} = 1282^{2} \pmod{3233} = 1160 \pmod{3233}$
$855^{1024} = 1160^{2} \pmod{3233} = 672 \pmod{3233}$
$855^{2048} = 672^{2} \pmod{3233} = 2197 \pmod{3233}$

An Example of the RSA Algorithm

Given the above, we know this:

$855^{2753} \pmod{3233}$

$= 855^{(1 + 64 + 128 + 512 + 2048)} \pmod{3233}$

$= 855^{1} * 855^{64} * 855^{128} * 855^{512} * 855^{2048} \pmod{3233}$

$= 855 * 916 * 1709 * 1160 * 2197 \pmod{3233}$

$= 794 * 1709 * 1160 * 2197 \pmod{3233}$

$= 2319 * 1160 * 2197 \pmod{3233}$

$= 184 * 2197 \pmod{3233}$

$= 123 \pmod{3233}$

$= 123$

$855^{2753} \bmod 3233 =$

50432888958416068734422899127394466631453878360035509315554967564501055628612082559978744245428110054383498654289336384930246451441507851720917966547826353070
99638035387326500896686074771829745822950342950407903581845940956377938586598936883808360284013250976862076697739667533250542826093475735137988063256482639334
45309259438556242923301751977190016924916912809150596019178760171349725439279215696701789902134307146468971279610277181378394586967728986934236524031169321708
92696176437265213156658331587124597598030425031440068378832461017848307175854745472520696889259958925443667014322054695431740022855009238636942444855973333063
05160738530286321930291350374547194675777671357954965202919790505781532871558392070303159585937493663283548602090830635507044556588963193180119341220178269233
44101330116480696334024075046952588669876586690062240241020884665075302639538705266319335847348109487615622712603732759736037523738836414808894843809615775704
53800810794698006673487779588375828998513279307035335512750904399481789790548993381217329458535447413268056981087263348285463816885048824346588978393334662544
54006619645218766694795528023088412465948239275105770491133290256843065052292561427303898320890070515110552506189941712317779515797942971179547529630183784386
29139778776612982073890727967672023501139927158196427307640741898919048686074812454931579537437712441601438765069145868196402276027766869530903951314968319097
32450545234594477256587887692693353918692354818518542420923064996406822184490119135710885424428521120773712238311054554312653073940759278908226060431711333957
52266034451645259763161842774590432019134528932993216130744053222747057289481214358683197841559727649635709090121513130415756920979851832104115596935784883366
53159513273446752439408757697778908490126915322842080949630792972471304422194243906590308142893930291584830873687450789770869218452967411463211556678655283381
64806795455941891006950919658990854567980723923708463025535456869192355462995715735879062274586195721721110788286575638597094190776320509783239571346411902500
47020848560408217509491077165531176529747380317676582058767314028891032883431850884472116442719390374041315564986995913736516210845113740224335185995766577539
69362812542539006855262454561419258809437402128886669744109721845342218171980899119537075455420339119645393664617929681653426522346399367423309701835339046236
77693670380534264482173582384219251590438148524738896864244370318665419961537791396964900303958760654915244945043600135939277133952101251928572092597887511601
95962961569027116431894637342650023631004555718003693586055264910000907245183786689564417164907278356281009708545241354696608448116133878065485451517616730860
51080657829365241087232636672280540038794108643482267500907782651210137281958316531396983090887317417474535988684298559807185192215970046508106068445595364808
92249440542766329674592308898484868435865479850511542844016462352696931799377844302178570191970987516296546651302780099665800521782081393172323790132324946826
09200819981037684847167874989193694997914824716345060937125654122501953795166897601855087599313367797793952782227323337529580263122665358948205566515289466369
03208328768043239061154935095459093406676402258670848337605369986794102620470905715674470565311124286290735488849298998356099963609214112849774586146960402870
29670701478179490248282907484160083680458666855076046192252094349804715745268818131850859150194852763596503458153641656549316013061330407434457965108380304062
24027889804282518909471629226689801668448096364519809051090579651307570379245958074479752371266761011473878742144149154813591743927994969564156538668838917154
46305611805369728343470219206348999531917640161103924904391798033989754917653959236085118076531847064733180157820741276478759273908749295571685366518591266637
38312359458912678709583800022451509424457564874484086877530845395521730636693891702394037184780362774643171470855830491959895146776294392143100245613061114299
37000557751339717282549110056008940898419671319709118165542908761090083249978313382407869615784923419862991680086774959340775930660220781494380785499679894539
93640636857226974223618584114250483724512446558027085917979559108652309975651983827795294575699657424557868838354442368572236813990212613637440821314784832035
63615611346287019851423901842909741638620232051039712184983355286308685184282634615027441873586395040422815123995059959836537922272858474220716778366794513436
38070865797742198535953931662799887897216959634553463364979492211301766131620747726611310701232140371388227022172323308547267953301507998062253835458948024820
04314472619159619052603406906193093929072410284948700167172969517703467909979440975063764929635675558007116218277276031829217903502904860909762662853966270243
92536890256337101471683274045045830602286763142158159900791642627700054612322919219299716990769016902594646810414121420447240266165827568052416686147339332265
95912700645630447416085291672187007045144649793226668732146346749041185886760836840306190695786990096521390675205019744076777651043885151941619318479919134924 3
88152822038464729269446084915299958818598855195149066307311777238132267516945882593638786107243025659809149010327838482140113655678493410243151248286452917031
41004001201636482998532516634905605379458508942440385525245547779224010461489075274516342513992163738356814149047932037426337301987825405699619163520193896982
54478631309773749154478427634532593998741700138163198116645377208944002854850002696859826445621837941167021518477219093392321850877757909593326763114131296193
98495926138987901669710881027663862316769405729593253807864344410051213802508179762272379721035219677326844194648616402961059899027710532570457016332613431076
41770004323715247462639399011899727845362949303636914900881060531231630009010150839331880116682151638931046666595137827498923745560511004016477716822716267270
78370122424655126487845492350418521674263831897333324346744490397800178468972640546214802412412583384350170488532060147568786231809409001263241969092252022679
88011340807301221626440413388739260052309607238615855496515800103474611979213076722454380367188325370860671331132581992279755227718486484753261243028041779430
90938992370938053652046462551472678849615277732741192657091166135800841454214876873103944410547963930853089688036560850477214459217250012650071706896942815462
75637045883890421917739819064873190801482873905815946222786727741861011102763247972904122211994117388204526335701759090678628159281519982214576527968538925172
18720090070389138562840007332258507590485348046564543498370732876259358914278543182665872946080723896522915990217388879577364773872657461040082255112418272009
61681888284938946788104688473126554172620978905678458109651797530087306315464903021121335281808476122990409576427857316364124880930949770739567588422963171158
46456984202455109029882398517953684125891446352791897307683834073696131409745229856386682726910433575176771288945278813686239650666540898943949516191200216077
78988768647364818378253248466991683072812203107919356466684015914858269999337442767725227540385332219685229859085154811040229657916338257385513314823459591633
28144581984361459630602499361753097925561238039014690665163673718859582772525683119989984646027216462797640770570748164064507697798699551061800464719378082232
50148934078511378332510737538234034662695532926088138438957840998041704104177760846306286261061405961520706669524301843857503176293954302631267377406936404705
89608346260188591118436753252984588804084971092299919565539701911191919188327308603766775339607722455632113506572191067587511868127863441975723921952633338565
38388240057190102564949233944519659592039923922174002472341471909709645621082995477461932289811812860555658809385189881181290561427408580916876571191122476328
86587127553892843812661199193792462411263299073986785455875665245305619750989114578114735771283607554001774268660965093305172102723066635739462334136380459142
37759965220309418558880039496755829711258361621890140359542349304247490536939927761142617964071001276432804287060835315945823059463268278612702033569803461432
45697021484375 mod 3233 = 123

Summary

- have considered:
  - principles of public-key cryptography
  - RSA algorithm, implementation, security