Introduction to Cyber Security

Fall 2017 | Sherman Chow | CUHK IERG 4130

Chapter 8
Public Key Cryptography

Secret Key Distributions

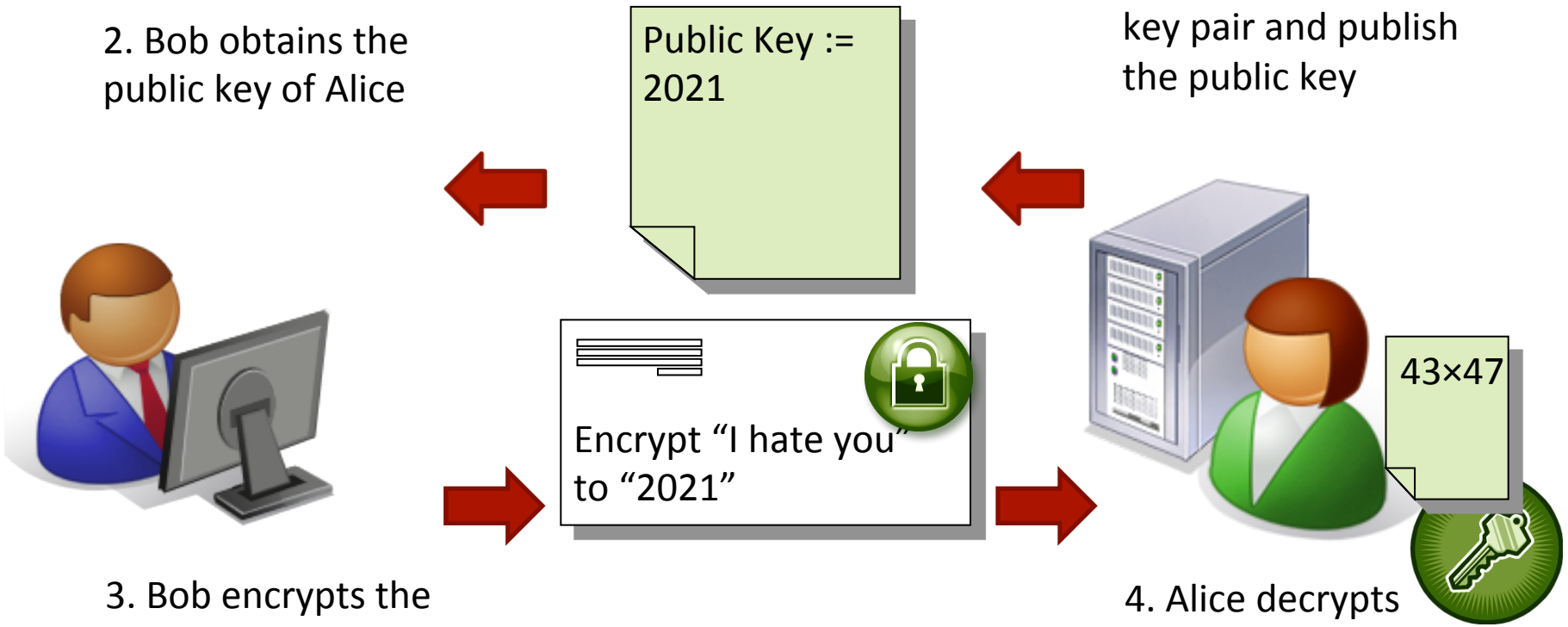
- How many secret keys need to be established if *n* people wants to have pairwise confidential communication?
- Can we achieve "non-repudiation" with just shared private key?
  - No, the message sender can repudiate since the recipient also can create the MAC

- How about we assume a trusted key distribution center (KDC)
  - Security and performance problems, more details later

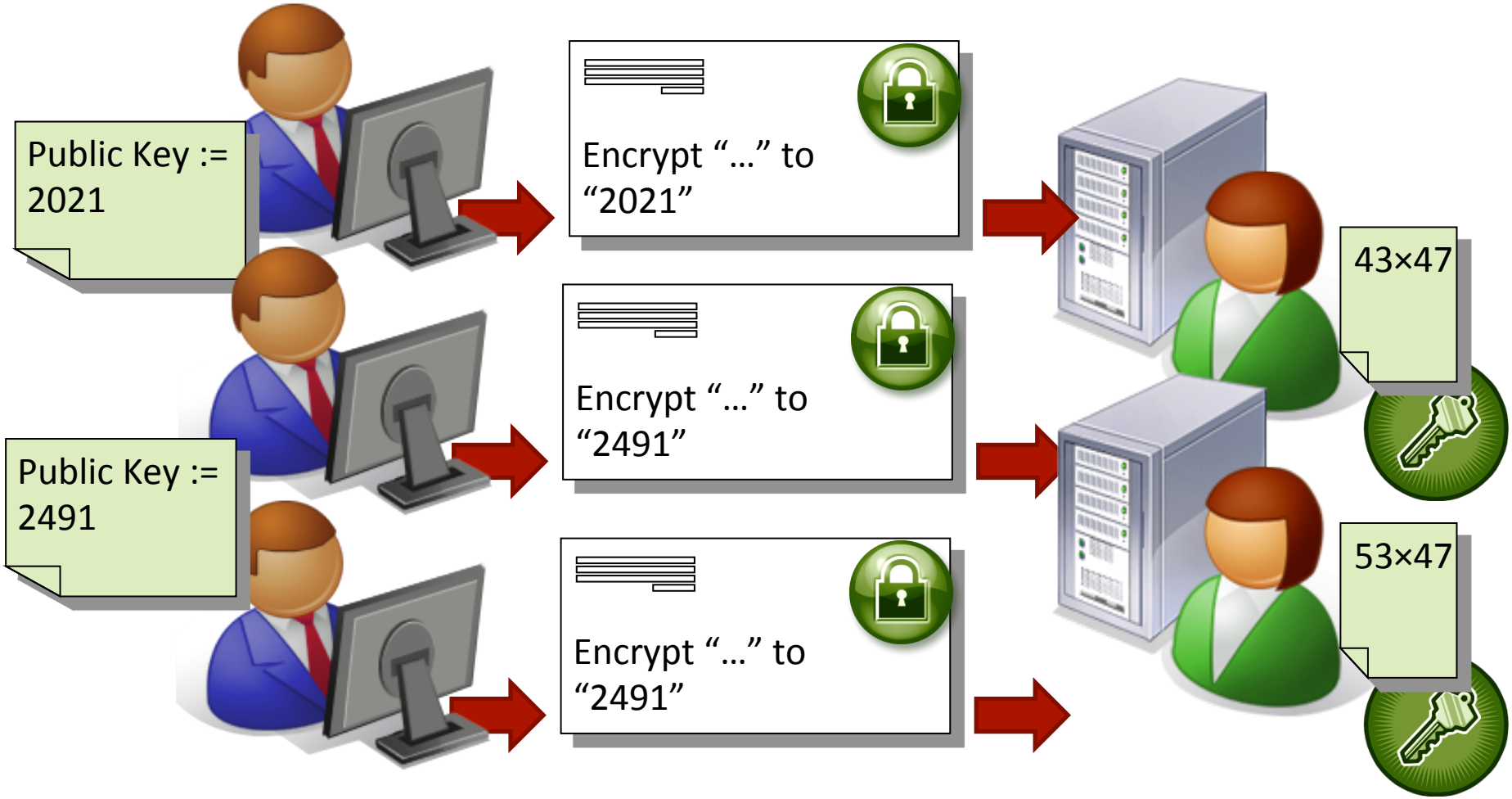
Public Key Cryptography (PKC)

- Every participant has a key-pair: Public Key and Private Key
- Public key is published or sent to everyone else openly
- Private key is kept secret by its owner
- Plaintext encrypted by Alice's public key can only be decrypted by Alice's private key
- Valid signatures (which make verification returns valid) under Alice's public key can only be created by Alice's private key

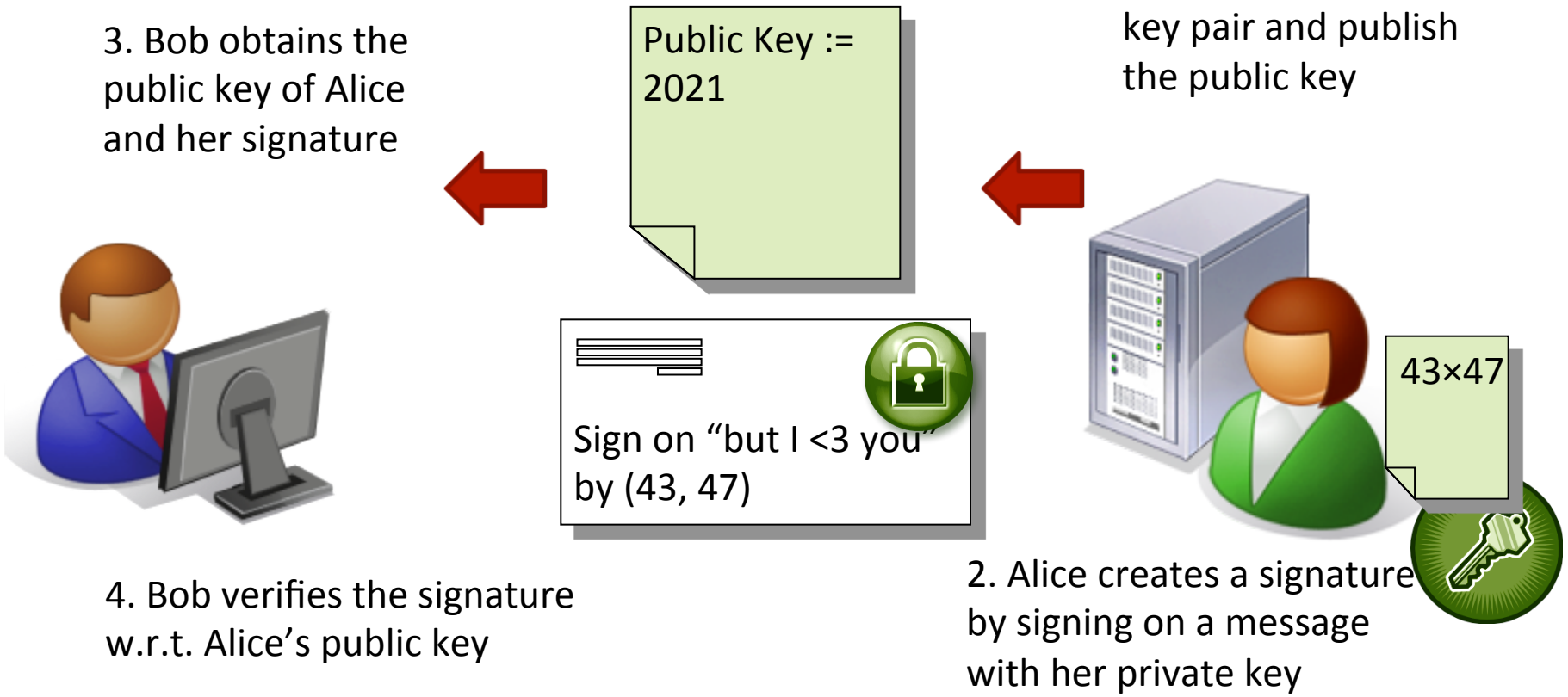
Public-Key Encryption (PKE) illustrated



Public-Key Encryption illustrated (cont.)



(Public-Key) Signature illustrated



History of PKC in the "Public World"

- Diffie was a graduate student in Stanford, working with Prof. Hellman on solving the "key distribution problem".
- They proposed the concept of "Public-Key Cryptosystem" (PKC) (developed jointly with Merkle).
  - Even more amazingly, introduce the *notion* of digital signature
- They cannot realize it; yet, they were able to find a way for communication parties to establish a *shared secret* via *open communications* only (Diffie-Hellman Key exchange, stay tuned)
- http://www.gchq.gov.uk/History/Pages/PKE.aspx

From DH to RSA

- Nov '76, Diffie and Hellman published their ideas and findings in "New Directions in Cryptography" (ACM Turing award 2015)
- Open problem of realizing PKC, *i.e.*, finding D(), E() s.t.,
  - D(E(*m*)) = *m* (= E(D(*m*))) and
  - D( ) can be kept secret while E( ) is known to the public
- Ron Rivest was intrigued by DH's paper. He then enlisted the help of Adi Shamir and Leonard Adleman, all from MIT, and came up with a solution (*i.e.*, RSA algorithm) in '77 (ACM Turing award 2002)
- Diffie, Hellman, Merkle, Rivest, Shamir, Adleman were commonly recognized as the founders of PKC

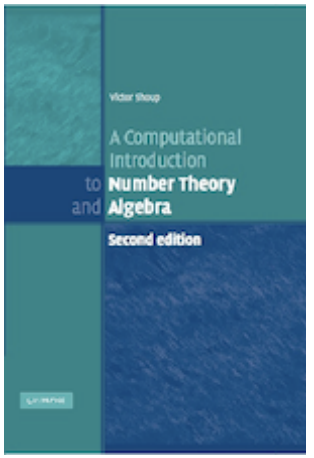
Diffie, Hellman, Rivest, Shamir in 2011

Leftmost: Ari Juels, Chief Scientist of RSA

Rightmost: Dickie George, Technical Director, Information Assurance, NSA

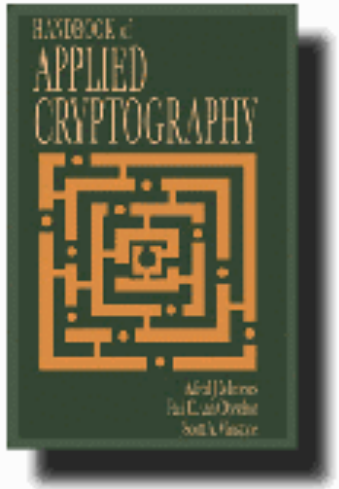
Some Suggested Readings

- http://www.shoup.net/ntb
- http://cacr.uwaterloo.ca/hac
- http://www.ams.org/notices/199902/boneh.pdf



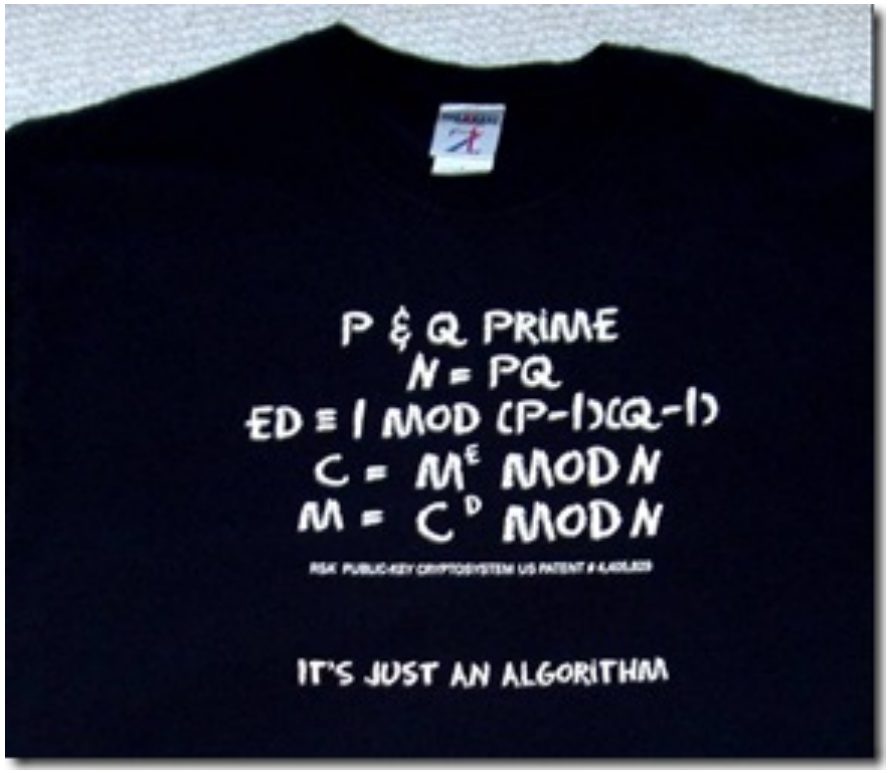
Twenty Years of Attacks on the RSA Cryptosystem

Dan Boneh



RSA Algorithm

- Most widely accepted and implemented approach to PKE
- "Block cipher" where $0 \leq m, c \leq N - 1$ for some N



RSA Key Generation

- Choose two large prime numbers p, q (*e.g.*, 1024 bits each)
- Compute $N = pq$, $\Phi(N) = (p-1)(q-1)$ // we use shorthand Φ below
- Choose e ($< N$) that has no common factors with Φ
  - *i.e.*, e, Φ are "relatively prime"
- Choose d such that $ed-1$ is exactly divisible by Φ
  - *i.e.*, $ed - 1 = K\Phi$ for some integer K
  - which means $ed \bmod \Phi = 1$ // remainder of ed divided by Φ is 1
- Public key is (N, e)
- Private key is (N, d) // or just d (since N is public anyways)

RSA Encryption and Decryption

- Enc(m) → c --- $c := m^e \bmod N$
- Dec(c) → m --- $m := c^d \bmod N$
- Fermat's Little Theorem: For any prime p, and any x such that p does not divide x (*e.g.*, $1 \le x \le p\text{-}1$), we have $x^{p-1} = 1 \bmod p$
- Euler's Theorem: For any integer N, and x in $\mathbf{Z}_N^*$, $x^{\Phi(N)} = 1 \bmod N$
  - Treat "x in $\mathbf{Z}_N^*$" as "x is relatively prime to N"
  - $\Phi(N) = (p-1)(q-1)$
- Correctness: $c^d \bmod N = m^{(K\Phi+1)} \bmod N = m^{K\Phi} m \bmod N = m \bmod N$

RSA KeyGen: Toy-Example

- $p = 5$, $q = 7$ // random prime generation (details later)
- so $N = 5 \times 7 = 35$
- $\Phi = (5-1) \times (7-1) = 24$
- Pick $e = 5$ (so e, Φ are relatively prime)
- We have $d = 29$
  - Computed by extended Euclidean algorithm (details later)
  - Check: $ed - 1 = 5 \times 29 - 1 = 144$, and we have $144 / 24 = 6$
- We can encrypt {2, 3, …, 34}, more than enough for English alphabet

RSA E and D: Toy-Example

- Public key = *(N* = 35, *e* = 5). Private key = *d* = 29, b.t.w. *Φ* = 24
- Let's encrypt *m* = 4, *i.e.*, 4^5 = mod 35 = 1024 mod 35 = 9
- Let's decrypt *c* = 9, note that (*a* mod *N*)×(*b* mod *N*) = *ab* mod *N*
  - 9^{29} mod 35 = {(9^{10} mod 35)×(9^{10} mod 35)×(9^{9} mod 35)} mod 35
  - (16 × 16 × 29) mod 35 = 4
  - OR Repeated squaring, 9^29 = 9× (9^14)^2 = 9× ((9^7)^2)^2
  - = 9×((9×(9^3)^2)^2)^2 = 9×((9×(9×9^2)^2)^2)^2 // $29_{10} = 11101_2$

Security of RSA relies on RSA assumption

- Security relies on the following "RSA assumption"
- Namely, given (N, e) and $c := m^e$, cannot compute m

- One can break it by finding p and q, and thus d from e
- Underlying assumption: cannot factor N
- These two assumptions turn out to be equivalent

How secure is RSA?

- Brute force: try all possible keys – larger *d* => more secure
- Larger *d*, the slower the decryption (repeated squaring)

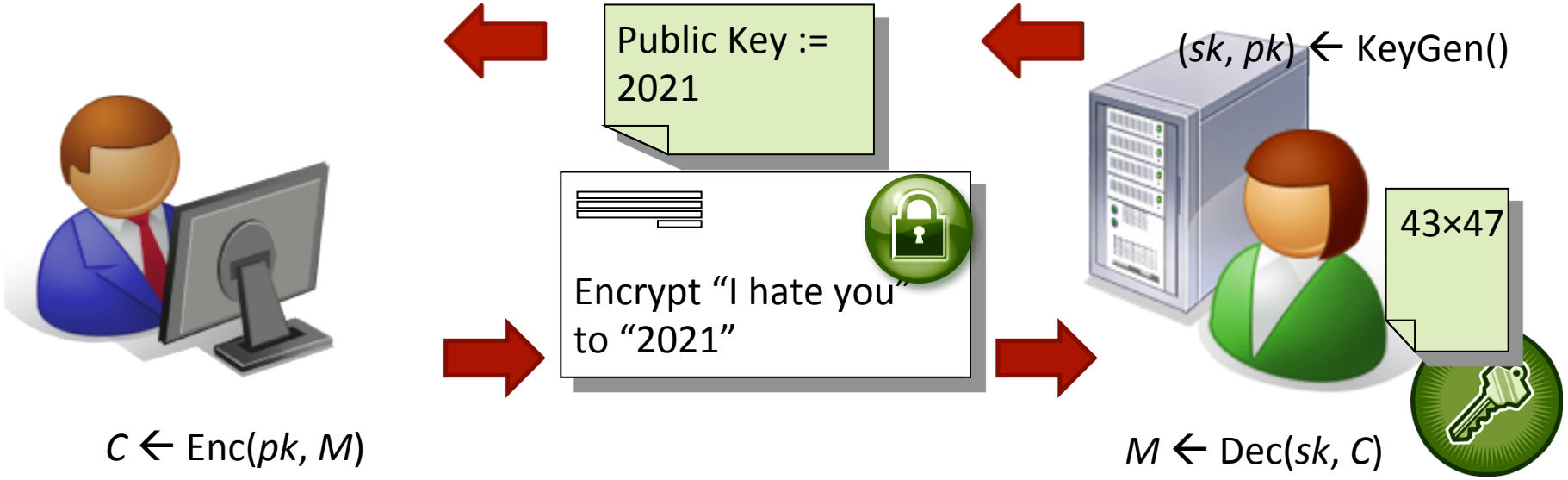
- A 430-bit key was cracked in 1996 => Prize: USD $14,527
- 1024-bit (bit-length of *N*) is considered strong enough, **for now**
  - *https://en.wikipedia.org/wiki/RSA_Factoring_Challenge*

RSA Signature

- E() and D() are "symmetric" for RSA in the following sense:
  - D(E(m)) = m = E(D(m))
  - Recall Enc(m) --- $c := m^e \bmod N$, Dec(c) --- $m := c^d \bmod N$

- Sig(s) --- $s := m^d \bmod N$
- Ver(m) --- Is $m := s^e \bmod N$? // verifying, not recreating (MAC)
- This is "textbook RSA signature": Insecure, we'll fix it later

PKE and RSA (Summary)



- PRIME p and q
- $N = pq$
- $ed \equiv 1 \bmod (p - 1)(q - 1)$
- $C = M^e \bmod N$
- $M = C^d \bmod N$

Modular Addition and Multiplication

- Consider mod 9, what are 1+8, 2+5, and 11+41?
- Additive inverse of *x* is the number we need to add to *x* to get 0.
  - *e.g.*: What is the additive inverse of 2 mod 9?
- Consider mod 9, $2 \times 5 = 10 \bmod 9 = 1$
- Multiplicative inverse: if $xy = 1 \bmod n$, *x* & *y* are each other's inverse
- Relatively prime: no common factors other than 1
- Does multiplicative inverse always exist?
  - 3 mod 9? 4 mod 9?

Extended Euclidean Algorithm

- Based on Greatest Common Divisor
- Theorem: Getting $\gcd(a, b) = au + bv$ is poly. time (in len. of a, b)
- 1) Division: $a = bq_1 + r_1$, $0 \leq r_1 < b$
- 2) Repeat: $b = r_1q_2 + r_2$, $0 \leq r_2 < r_1$
- 3) Keep repeating until $r_{s+1} = 0$
- We have $\gcd(a, b) = \gcd(b, r_1) = \ldots = \gcd(r_{s-1}, r_s) = r_s$

Finding Multiplicative Inverse

- Input: (a, b)
- Output: (u, v) s.t. gcd(a, b) = $au + bv$
- Existence of multiplicative inverse:

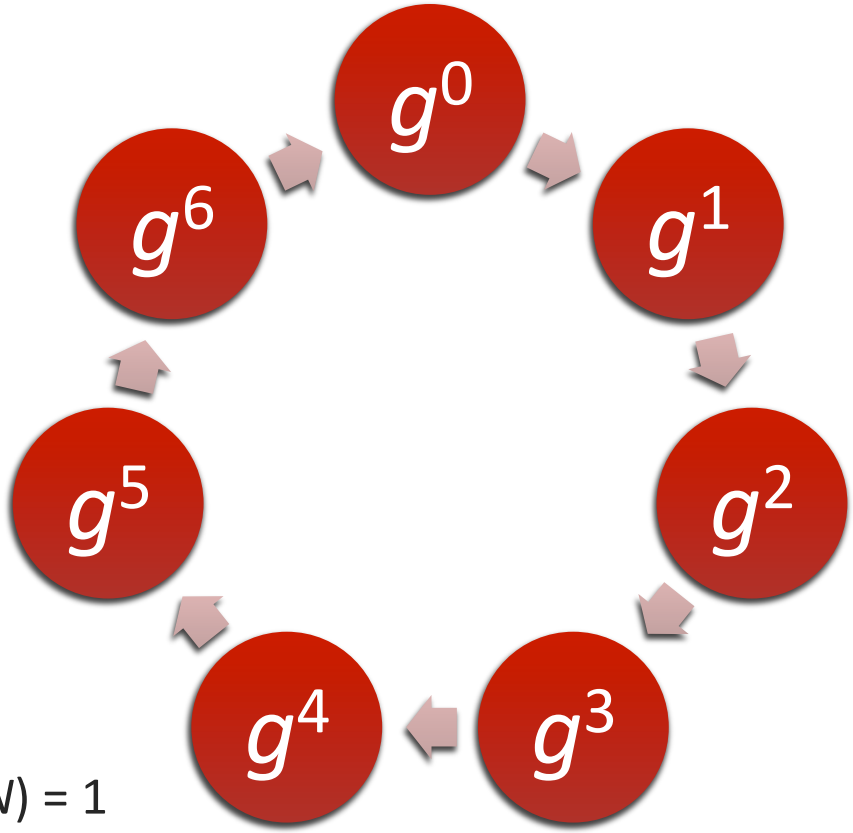
  x has multiplicative inverse mod n *iff* x is relatively prime to n
- $1 = au + bv$
- $u \times a = -v \times b + 1$
- $u \times a \equiv 1 \bmod b$

Groups

- A (finite) set **G** and a "group operation" (say *) that is
  - closed ($\forall$ *a*, *b* in **G**, *a* * *b* is also in **G**)
  - associative ($\forall$ *a*, *b*, *c* in **G**, (*a* * *b*) * *c* = *a* * (*b* * *c*) holds)
  - equipped with identity *e*, ($\forall$ *a* in **G**, *e* * *a* = *a* * *e* = *a* holds)
  - invertible ($\forall$ *a* in **G**, $\exists$ *b* in **G** s.t. *a* * *b* = *b* * *a* = *e* holds)
  - commutative for Abelian group ($\forall$ *a*, *b* in **G**, *a* * *b* = *b* * *a* holds)
- Examples:
  - **Z** (integers, with addition operation; infinite group)
  - $\mathbf{G}^n$ (**G**, a group; coordinate-wise operation)

Finite Cyclic Group

- Order of a group
  - denoted by $|\mathbf{G}|$
  - number of elements in **G**
- Cyclic group
  - (in multiplicative notation)
  - there is one element g s. t.
  - $\mathbf{G} = \{g^0, g^1, g^2, \ldots, g^{|\mathbf{G}|-1}\}$
- *E.g.* $(\mathbf{Z}_N, +)$
  - $\mathbf{Z}_N = \{0, 1, \ldots, N-1\}$
  - consider $g = 1$, or any g s.t. $\gcd(g, N) = 1$



Multiplicative Group of $\mathbf{Z}_N$

- $\mathbf{Z}_N^*$*:* set of elements from $\mathbf{Z}_N$ that are relatively prime to N
- For any *N*, 0 is not in $\mathbf{Z}_N^*$
- For any prime *p*, $\mathbf{Z}_p^* = \mathbf{Z}_p \setminus \{0\}$ // Set-minus denoted by \
- *e.g.*: Consider $\mathbf{Z}_N^* = \{1, 2, 3, 4\}$ // $N = 5$
- = {2^0, 2^1, 2^3, 2^2} // 2 can generate the group
- = {3^0, 3^3, 3^1, 3^2} // 3 can also generate the group

Fermat's Little Theorem

- Order of a (in $\mathbf{Z}_p^*$) is the smallest x s.t. $a^x = 1 \bmod p$
- FLT: For any prime p, and x in $\mathbf{Z}_p^*$, $x^{p-1} = 1 \bmod p$
- All elements' orders are $(p - 1)$?
- *e.g.*: Consider $p > 3$ and $(p - 1)^2$

Euler's Theorem, and RSA

- $f_{e,N}(x) = x^e \bmod N$, where $N = pq$, x *in* $\mathbf{Z}_N^*$ and e *in* $\mathbf{Z}_{\Phi(N)}^*$
- Euler phi-function: For any positive integer m, $\Phi(m)$ is the number of +ve int. < m that are relatively prime to m.
- Euler's Theorem: For any int. m, & x in $\mathbf{Z}_m^*$, $x^{\Phi(m)} = 1 \bmod m$
- What should $\Phi(N)$ be for $N = pq$, p and q are prime?

Finding big primes p and q

- Pr[a randomly chosen number n to be prime] is $1 / \ln n$
  - So, how many time we need to try on average?
  - 230 for n of 100-digit
- Test whether a random number n is a prime
- Trial divisions of n (how many should we try?)
- Fermat's Theorem: if n is a prime and $0 < a < n$, $a^{n-1} = 1 \bmod n$
  - Carmichael numbers: non-primes but satisfy the above condition
  - For a 100-digit non-prime, Pr[false positive / Carmichael #] $\approx 10^{-13}$
- Use Miller-Rabin algorithm (for your reference, details omitted)

Convenient *e*

- *e* = 3
  - btw, can we set *d* = 3?
- 65537
  - $2^{16} + 1$
- 3 → 2 multiplications
- 65537 → 17 multiplications

Repeated Squaring and Multiplication, *e.g.*:

$$x^{277} = x^{(00100010101)_2} = \prod_{i=0}^{10} b_i \cdot x^{2^i}$$

Inconveniences of $e = 3$

- What do we expect from e again?
  - $e = 3$ should be relatively prime to $(p - 1)(q - 1)$
  - Easier to choose eligible primes for 65537
- Why d cannot be 3 again? (Too small → Security problem)
- Likewise, what if $m < N^{1/3}$? (Too small → Security problem)
  - $c = m^3$, recover m by exhaustive search?
  - NO! Modular arithmetic became "null", just do cube-root!
  - Solution: "Pad" m s.t. it is larger than $N^{1/3}$
- What if m is broadcast to three recipients?
  - $c_1 = m^3 \bmod N_1$, $c_2 = m^3 \bmod N_2$, $c_3 = m^3 \bmod N_3$
  - Still, cube-root attack, but how?

鬼谷算

- 「孫子算經」：「今有物，不知其數，
三三數之，剩二，五五數之，剩三，七七數之，剩二，
問物幾何？」
- 「韓信點兵」：「兵不知數，
三三數之剩二，五五數之剩三，七七數之剩二。」

- $a = 2 \bmod 3, 3 \bmod 5, 2 \bmod 7$
- What is a? (or smallest such a)?

The Accolade by Edmund Blair Leighton

Chinese Remainder Theorem (CRT)

- Consider mapping elements in $\mathbf{Z}_{15}$ to $\mathbf{Z}_3$ and $\mathbf{Z}_5$
- {0, 6, 12, 3, 9; 10, 1, 7, 13, 4; 5, 11, 2, 8, 14}
- The pair (a mod 3, a mod 5)
  uniquely determines (a mod 15)
- CRT: For $N = PQ$ (P, Q relatively prime),
  $a \longmapsto$ (a mod P, a mod Q)
  maps the N elements to the N distinct pairs
- Extends to product of > 2 (relatively prime) #'s.

Z_{15}	Z_3	Z_5
0	0	0
1	1	1
2	2	2
3	0	3
4	1	4
5	2	0
6	0	1
7	1	2
8	2	3
9	0	4
10	1	0
11	2	1
12	0	2
13	1	3
14	2	4

CRT and $\mathbf{Z}_N$

- Addition can be done coordinate-wise
- $(a, b) +_{(\text{mod } N)} (a', b') = (a +_{(\text{mod } P)} a', b +_{(\text{mod } Q)} b')$
- CRT: $\mathbf{Z}_N \cong \mathbf{Z}_P \times \mathbf{Z}_Q$ (group isomorphism)

$\mathbf{Z}_{15}$	$\mathbf{Z}_3$	$\mathbf{Z}_5$
0	0	0
1	1	1
2	2	2
3	0	3
4	*1*	*4*
5	2	0
6	**0**	**1**
7	**1**	**2**
8	2	3
9	*0*	*4*
10	*1*	*0*
11	2	1
12	0	2
13	**1**	**3**
14	2	4

CRT and $\mathbf{Z}_N^*$

- No multiplicative inverse iff (0, *x*) or (*x*, 0)
- Multiplication (and identity, inverse)
  - Coordinate-wise
- $\mathbf{Z}_N^* \cong Z_P^* \times Z_Q^*$
- Isomorphism is easy to compute if know *P*, *Q*
  - (in both directions)

Z_{15}	Z_3	Z_5
~~0~~	~~0~~	~~0~~
1	1	1
2	2	2
~~3~~	~~0~~	~~3~~
4	1	4
~~5~~	~~2~~	~~0~~
~~6~~	~~0~~	~~1~~
7	1	2
8	2	3
~~9~~	~~0~~	~~4~~
~~10~~	~~1~~	~~0~~
11	2	1
~~12~~	~~0~~	~~2~~
13	1	3
14	2	4

Back to (2 mod 3, 3 mod 5, 2 mod 7)

- Consider $\mathbf{Z}_3$ and $\mathbf{Z}_5$ first:
  - $a = (2)(x)(5) + (3)(y)(3)$
  - $5x \equiv 1 \bmod 3 \rightarrow x = 2$, $3y \equiv 1 \bmod 5 \rightarrow y = 2$
  - $a = 38 \bmod (3\times5) = 8$
- $a = 8 \bmod 15, 2 \bmod 7$
- Consider $\mathbf{Z}_{15}$ and $\mathbf{Z}_7$:
  - $a = (8)(x)(7) + (2)(y)(15)$
  - $7x \equiv 1 \bmod 15 \rightarrow x = 13$, $15y \equiv 1 \bmod 7 \rightarrow y = 1$
  - $a = 758 \bmod (7\times15) = 23$ (actually, $a = 23 + 105k$)

Single Step Treatment

- a = (2 mod 3, 3 mod 5, 2 mod 7)
- $a = (2)(x_1)(5\times7) + (3)(x_2)(3\times7) + (2)(x_3)(3\times5)$
  - $35\ x_1 \equiv 1 \bmod 3 \rightarrow x_1 = 2$
  - $21\ x_2 \equiv 1 \bmod 5 \rightarrow x_2 = 1$
  - $15\ x_3 \equiv 1 \bmod 7 \rightarrow x_3 = 1$
- $a = 140 + 63 + 30 = 233 \bmod 105 = 23$

Nested detour…

- $c_1 = m^3 \bmod N_1$, $c_2 = m^3 \bmod N_2$, $c_3 = m^3 \bmod N_3$
- $c = m^3 \bmod N_1N_2N_3$
  - So what?
- $m = c^{1/3}$
- Why? We still don't know $\Phi(N_1N_2N_3)$
- m "became small" relative to $N_1N_2N_3$
- Solution: Public-key-specific padding of message

No More Number Theory!

- OK, now “Crypto Theory” ☺
- What if we are just encrypting “Yes” or “No”?
- Recall ECB
- Trial encryption for each possible plaintext
- Note that (textbook) RSA encryption is deterministic

RSA Signature (Review)

- $S \leftarrow \text{Sig}(M)$ --- $S := M^d \bmod N$
- $1/0 \leftarrow \text{Ver}(M, S)$ --- Is $M := S^e \bmod N$?

Chosen-Ciphertext Attack (CCA)

- CCA: An attacker can choose the ciphertext to be fed to a decryption oracle
  - Why we have a decryption oracle?
  - "Lunch-time" attack
- Don't using the same (RSA) key for both confidentiality and authenticity!
  - Eve, the attacker, records an encrypted letter sent to you by someone else.
  - Eve then asks you to sign this recorded message
    - (and of course, return the signed result to her.)
  - If you follow Eve's request, you are decrypting your own secret letter for Eve.

Textbook RSA

- Isn't it obvious I got a ciphertext to "sign"?
  - Will be addressed shortly afterwards
- The algorithm you have seen is often called "Textbook RSA"
- Probabilistic encryption
  - Randomly pad the plaintext before encryption
- Optimal asymmetric encryption padding (OAEP)
  - A form of Feistel network!
  - For your reference only, details omitted

Factoring Milestones

- RSA-4, IBM quantum computer, ’01, successfully factored 15 into 3 × 5
- Quantum Factorization of 143 (’11)
- RSA-768, USD $50k, Dec ’09. You can try RSA-896, or RSA-1024, or…
- RSA-2048 (USD $200,000) =
  2519590847565789349402718324004839857142928212620403202777713783
  6043662020707595556264018525880784406918290641249515082189298559
  1491761845028084891200728449926873928072877767359714183472702618
  9637501497182469116507761337985909570009733045974880842840179742
  9100642458691817195118746121515172654632282216869987549182422433
  6372590851418654620435767984233871847744479207399342365848238242
  8119816381501067481045166037730605620161967625613384414360383390
  4414952634432190114657544454178424020924616515723350778707749817
  1257724679629263863563732899121548314381678998850404536402352738195137863656439121201039712282212072035 7

Performance of RSA

- For hardware/software implementation, ~1000/100 times slower than DES
- Decryption on a 1 MIPS VAX requires ~30s when it was invented in late 70's
- Special-purpose implementation was needed (*e.g.* special circuit board)
- IBM PC debuts in '81, with Moore's Law, s/w now runs 2000x faster…
- Speed differs on types of op. (enc/verify vs. dec/sign) and exponent size
- With $e = 3$, encryption and signature verification are typically 5-10 times faster than decryption and digital signing resp.

Blind RSA

- $C = M^e \bmod N$
- $C' = (r^e)C = r^e M^e = (rM)^e \bmod N$, for a random r in [1, N]
- Decrypt and get $rM \bmod N$
- Get back M by multiplying it with multiplicative inverse of r

- Enough attack! Give me useful application!
- More generally, multiplicative homomorphism
  - $E(a) * E(b) = E(a * b)$, computable without knowing private key

Blind RSA Signature

- $S \leftarrow \text{Sig}(M)$ --- $S := M^d \bmod N$
- I want Alice to sign M for me without letting her to know M
- $M' = r^e M \bmod N$
- $S' = (r^{ed})M^d = rM^d$
- Get back $S = M^d \bmod N$

Textbook-RSA Signature is Insecure

- $S_1 := M_1^d \bmod N$, $S_2 := M_2^d \bmod N$
- $S' := S_1 S_2 \bmod N = M'^d \bmod N$
  - where $M' = M_1 M_2$

- We expect $S^e = M \bmod N$
- Let's try to create a signature on a "random" message.

Public-Key Cryptography Standard (PKCS)

- A list of Standards (PKCS#1 to PKCS#15) on how to use RSA in practice, regarding
  - message formatting,
  - information encoding scheme,
  - choice of parameters, *etc*.
- Protected against the following "improper use" or attacks, *e.g.*:
  - plaintext guessing
  - chosen ciphertext attack
  - $m^3 < N$
  - sending the same message to multiple people

Next Lecture

- Diffie-Hellman Protocol
- Discrete-Logarithm-based systems
  - Encryption: ElGamal
  - Signature: Schnorr
- Overview of Elliptic Curve Cryptography (ECC)
  - ECC vs. RSA
- Public-Key Infrastructure
- Authentication protocols