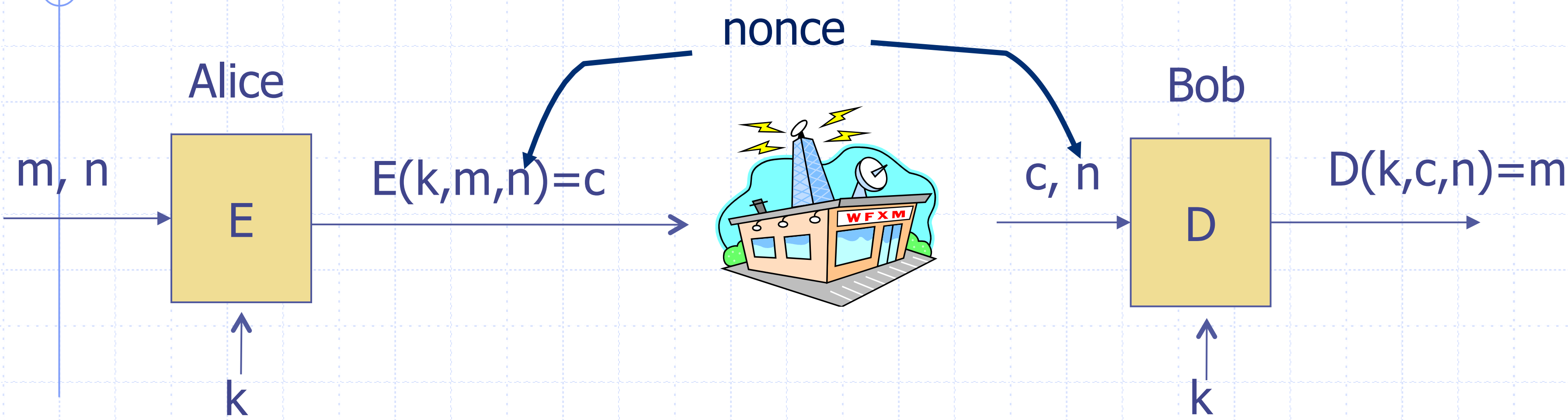
Public Key Cryptography

Vijay Ganesh

Motivation for Public-key Cryptography

- Issues with Symmetric cryptography
  - Assumes parties already share a secret key
  - How do the parties exchange keys in the first place?

Recalling Symmetric Cryptography



E, D: cipher k: secret key (e.g. 128 bits)

m, c: plaintext, ciphertext n: nonce (Initial Vector)

Encryption algorithm known publicly

- Never use a proprietary cipher

Symmetric Cryptography Example: One Time Pad (single use key)

- Vernam (1917)

Key:	0	1	0	1	1	1	0	0	1	0
										$\oplus$
Plaintext:	1	1	0	0	0	1	1	0	0	0
Ciphertext:	1	0	0	1	1	0	1	0	1	0

- Shannon '49:
  - OTP is "secure" against ciphertext-only attacks (COA)
  - Information-theoretically secure

One Time Pad: Perfect Security
(single use key)

- Vernam (1917)

Key:	0	1	0	1	1	1	0	0	1	0
Plaintext:	1	1	0	0	0	1	1	0	0	0
Ciphertext:	1	0	0	1	1	0	1	0	1	0

$\oplus$

- Secure against adversary with unlimited computational power
- $Pr(M) = Pr(M|C)$
- Cipher-text doesn't give attacker any additional power

Problems with One-time Pad

(single use key)

- Vernam (1917)

Key:	0	1	0	1	1	1	0	0	1	0
Plaintext:	1	1	0	0	0	1	1	0	0	0
Ciphertext:	1	0	0	1	1	0	1	0	1	0

$\oplus$

- Not secure if key is reused. Key must be truly random every use
- Key as long as the message
- No authentication
- Key exchange

Public-key Cryptography

Problems Solved using Public-key Cryptography

- Key exchange (Diffie & Hellman 1976. Merkle 1974 – 1978)
- Public key encryption system (e.g., RSA 1978)
- Digital signatures (e.g., RSA 1978)

Public-key Cryptography: Background Mathematics

Modular arithmetic and multiplicative inverse

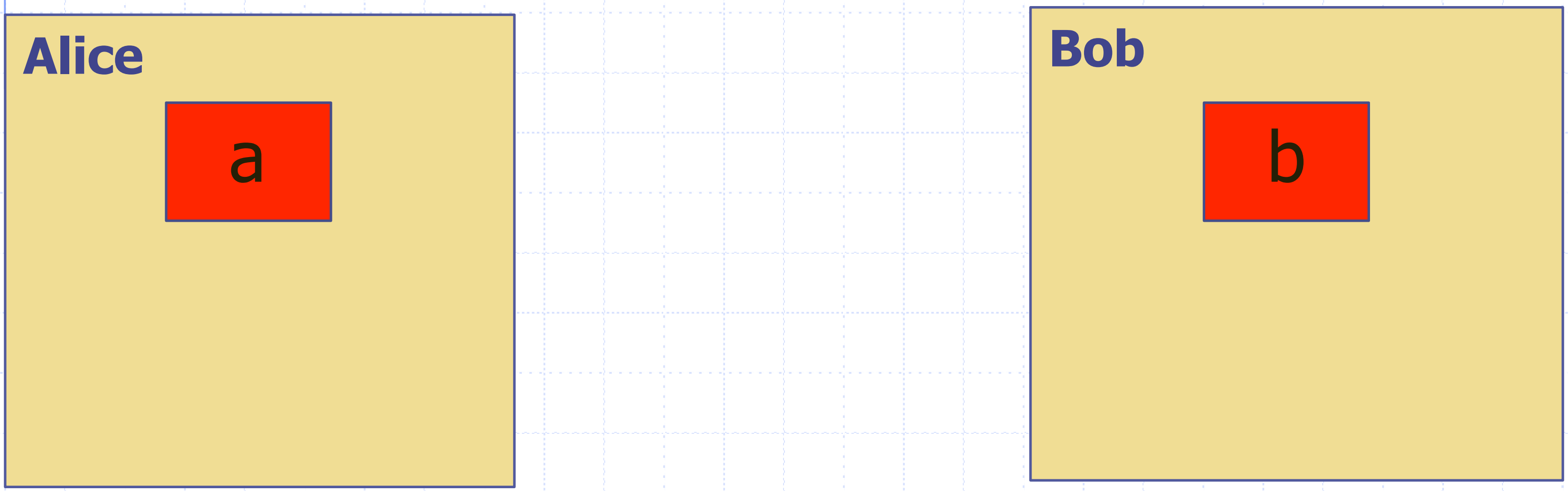
- Modular arithmetic mod p: $Z_p = \{0,1,\ldots,p-1\}$
  - $Z_9 = \{0,1,\ldots 8\}$
- Definition of multiplicative inverse
  - $x.x^{-1} = 1$
- Theorem
  - For any x: x^{-1} exists iff gcd (x,p) = 1
- Examples of Theorem
  - 4 is in Z_9. gcd(4,9) = 1. 7 is its inverse mod 9
  - 3 is in Z_9, gcd(3,9) = 3. No inverse

Diffie-Hellman-Merkle Key Exchange Protocol

- Problem
  - How to exchange a key between Alice and Bob that at the same time is hard for anyone else to learn the key.
- Solution (given in steps)
  - Step 1: The Setup Phase
    - Alice and Bob agree on a large prime p and a number called g, where, $0 <= g < p$

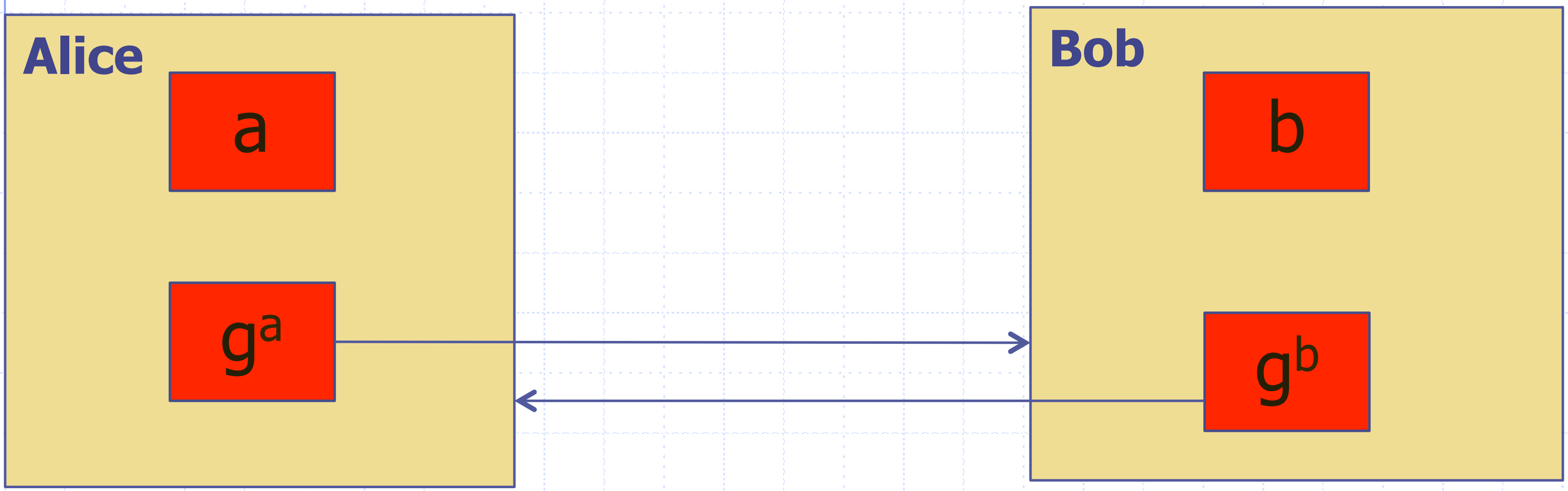
Diffie-Hellman-Merkle Key Exchange Protocol

- Step 2
  - Alice chooses randomly a number ‘a’ and Bob chooses randomly a number ‘b’ (order doesn’t matter)



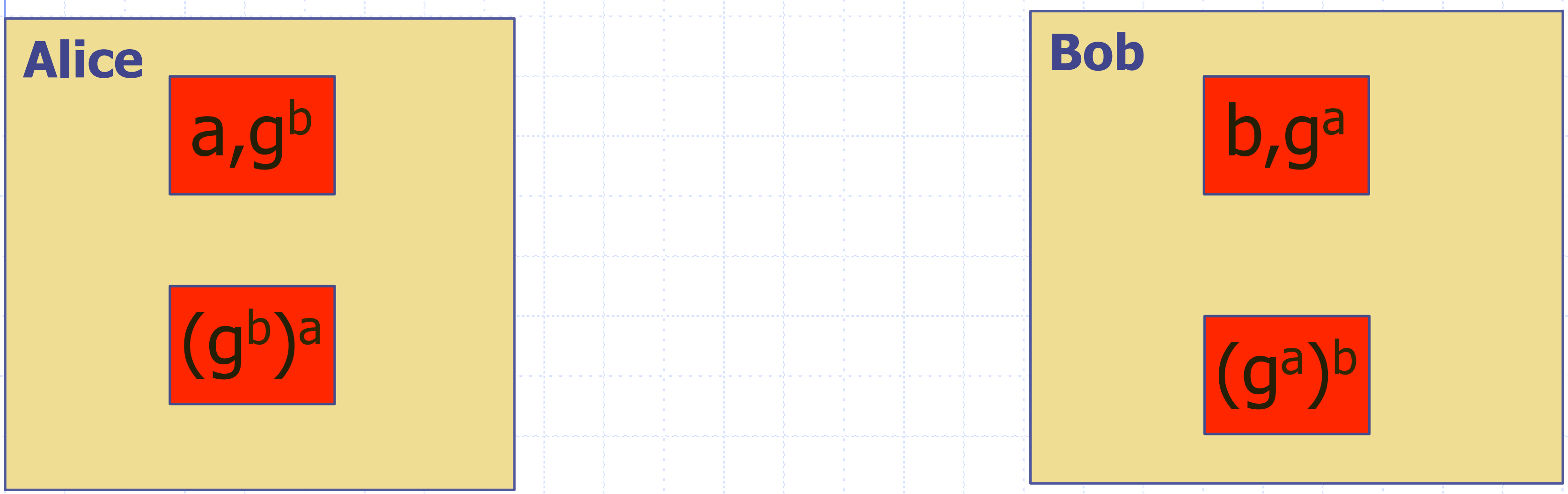
Diffie-Hellman-Merkle Key Exchange Protocol

- Step 3
  - Alice computes g^a and Bob computes g^b



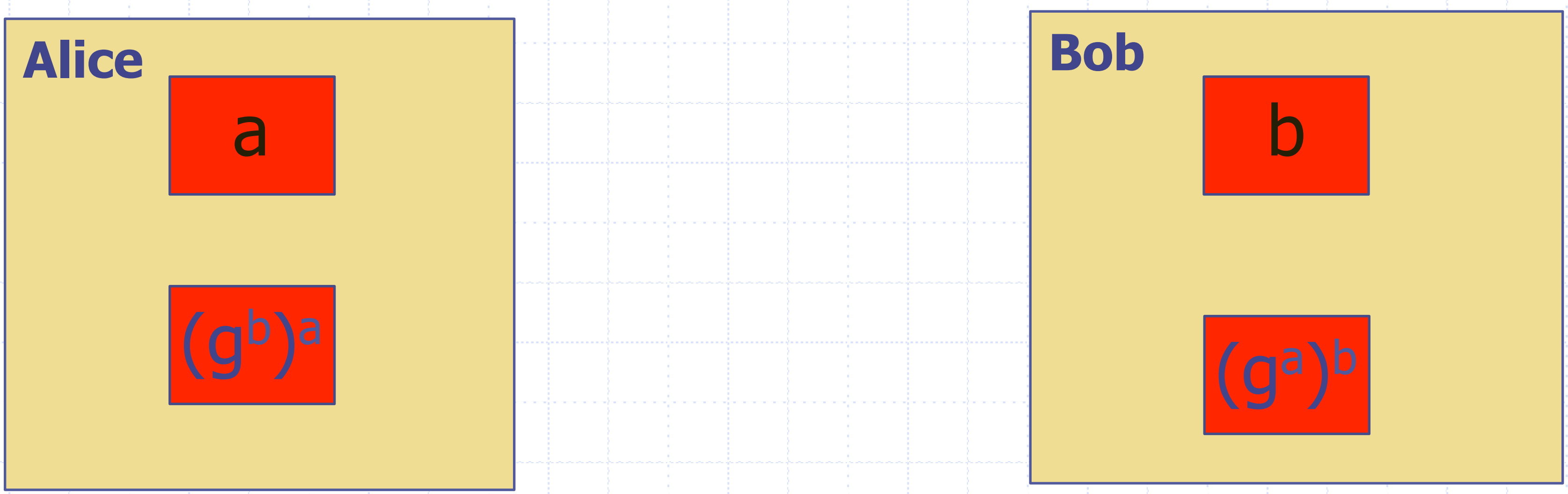
Diffie-Hellman-Merkle Key Exchange Protocol

- Step 4
  - Alice computes $(g^b)^a$ and Bob computes $(g^a)^b$
  - By laws of exponentiation $(g^b)^a = (g^a)^b = g^{ab}$ = key



Diffie-Hellman-Merkle Key Exchange Protocol

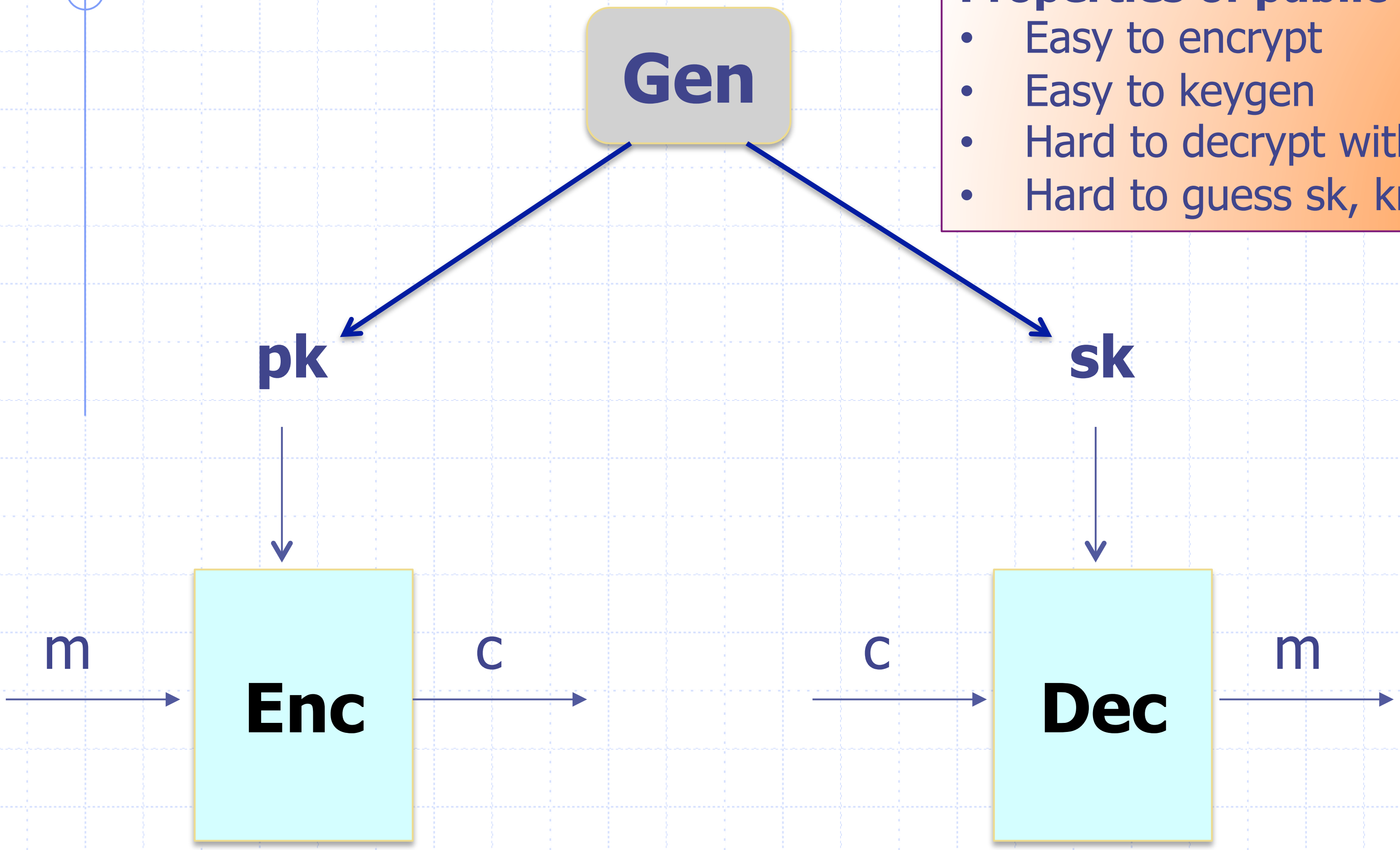
- Wait, why does this work?
  - CDH problem: Given g, g^a, g^b, it is hard to compute g^{ab} mod p
  - Attacker has g, p, g^a, g^b (by observing the communication between Alice and Bob), and yet cannot compute g^{ab}
  - Reduction: If you break DH key exchange protocol you can solve the CDH problem



Public key encryption: (Gen, Enc, Dec)

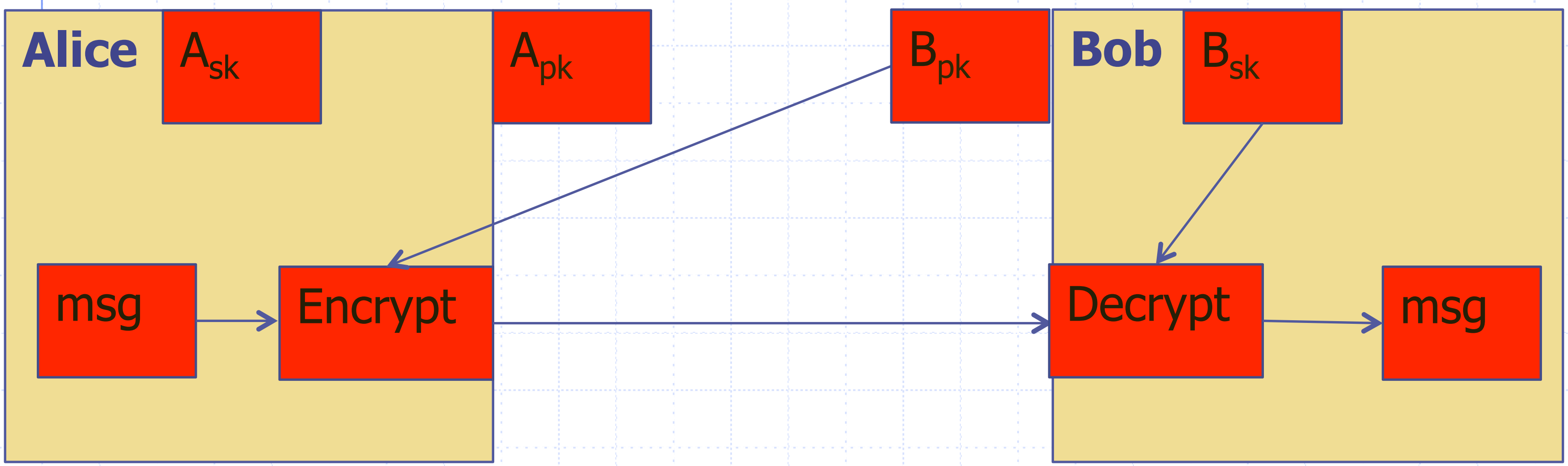
Properties of public key crypto:

- Easy to encrypt
- Easy to keygen
- Hard to decrypt without sk
- Hard to guess sk, knowing only pk



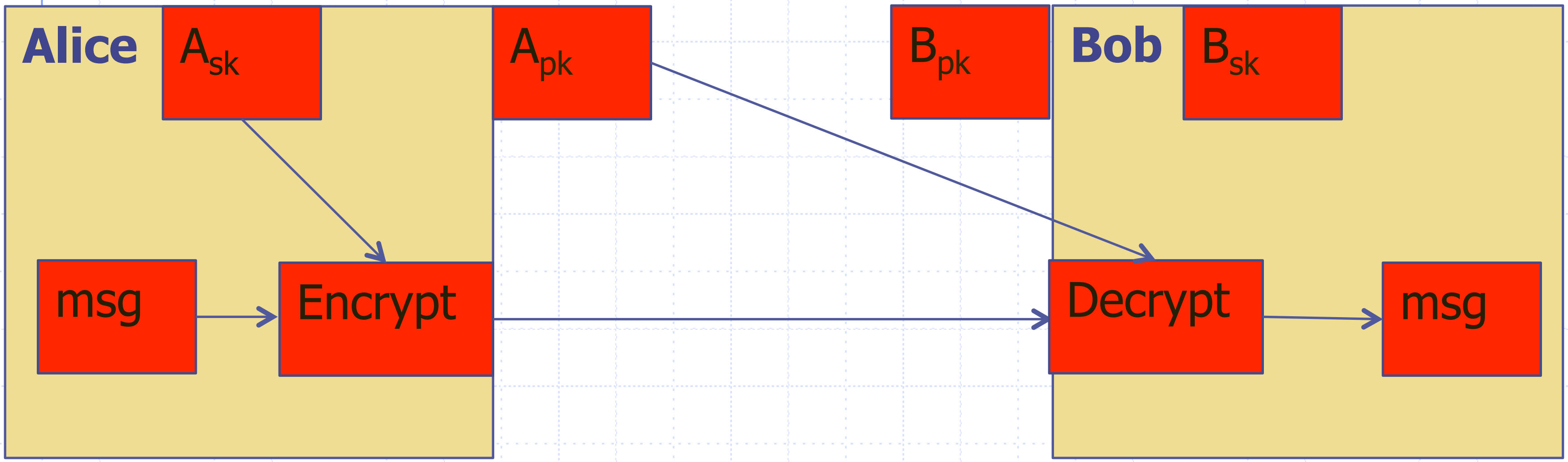
Public Key Encryption

- **Confidentiality: Alice wants to send msg to Bob over an open channel confidentially**



Digital Signatures

Authenticity: Alice wants to prove to Bob her identity



Public-key Cryptography: Background Mathematics

Modular arithmetic and multiplicative inverse

- Modular arithmetic mod p: $Z_p = \{0,1,\ldots,p-1\}$
  - $Z_9 = \{0,1,\ldots 8\}$
- Definition of multiplicative inverse
  - $x.x^{-1} = 1$
- Theorem
  - For any x: x^{-1} exists iff gcd (x,p) = 1
- Examples of Theorem
  - 4 is in Z_9. gcd(4,9) = 1. 7 is its inverse mod 9
  - 3 is in Z_9, gcd(3,9) = 3. No inverse

Public-key Cryptography: Background Mathematics

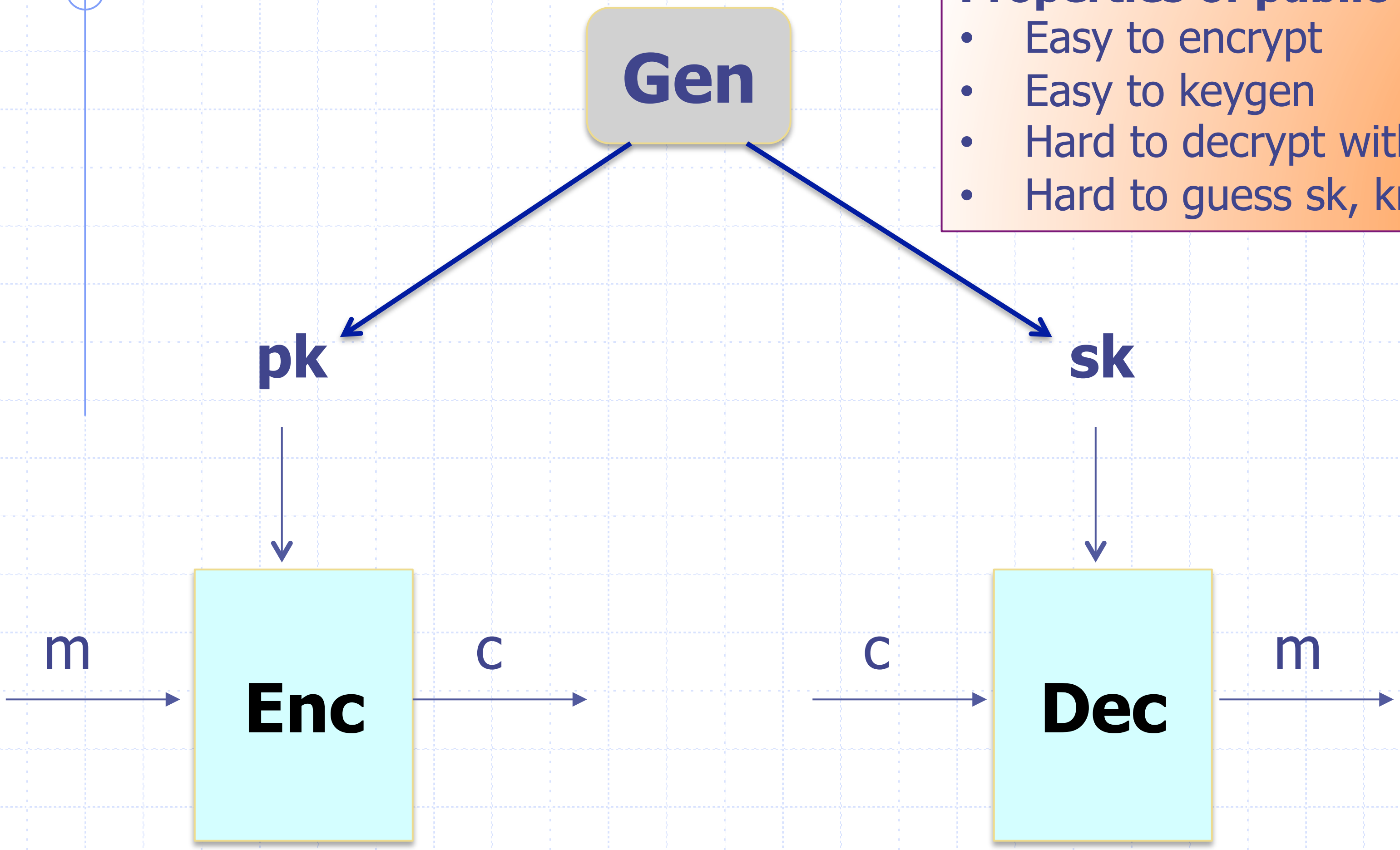
Euler's Totient

- Euler's Totient
  - Number of elements which have multiplicative inverse in a set modulo some integer p
  - Denoted as $\Phi(p)$
- Fact
  - For any prime p: $\Phi(p) = p-1$
  - Because for every x element of Z_p, $\gcd(x,p) = 1$
  - All numbers smaller than a prime p are relatively prime to p

Public key encryption: (Gen, Enc, Dec)

Properties of public key crypto:

- Easy to encrypt
- Easy to keygen
- Hard to decrypt without sk
- Hard to guess sk, knowing only pk



RSA: Background Mathematics

Key Generation

- **Step 1: Large Prime Number Generation**
  - Two large prime numbers *p* and *q* need to be generated. These numbers are very large: At least 512 digits, but 1024 digits is considered safe
- **Step 2: Modulus**
  - From the two large numbers, a modulus *n* is generated by multiplying *p* and *q*
- **Step 3: Totient**
  - The totient of *n*, $\Phi(n)$ is calculated to be (p-1)(q-1). Why is (p-1)(q-1) the Totient of p.q?
- **Step 4: Public Key denoted as pk**
  - A *prime number* is calculated from the range $[3, \Phi(n))$ that has a greatest common divisor of 1 with $\Phi(n)$
- **Step 5: Private Key (Secret key) denoted as sk**
  - Because the prime in step 4 has a gcd of 1 with $\Phi(n)$, we are able to determine it's inverse with respect to mod $\Phi(n)$

RSA: Background Mathematics
Encryption and Decryption

- **Encryption**
  - **Enc(plaintext,pk) = ciphertext = $(\text{plaintext})^{pk}$ mod n**
- **Decryption**
  - **Dec(ciphertext,sk) = $(\text{ciphertext})^{sk}$ mod n**
- **Can you invert the roles of pk and sk?**
  - Yes. We get digital signatures

RSA: Background Mathematics

Why RSA works?

- Why RSA is correct, i.e., why can you decrypt the encrypted message for the appropriate key pairs?
  - D(E(pt,pk),sk) = m
- Is it efficient?
- Why is it secure?
- Why the inverse of the key is calculated w.r.t the Totient?

Public-key Cryptography: Background Mathematics

Modular arithmetic and multiplicative inverse

- Modular arithmetic mod p: $Z_p = \{0,1,\ldots,p-1\}$
  - $Z_9 = \{0,1,\ldots 8\}$
- Definition of multiplicative inverse
  - $x.x^{-1} = 1$
- Fermat's Little Theorem
  - For any prime p that does not divide an integer 'a'
    $a^{(p-1)} = 1 \mod p$

RSA: Background Mathematics
Why RSA correct?

- Observe that
  - $Enc(pt, pk) = ct = (pt)^{pk} \bmod n$
  - $Dec(ct, sk) = (pt)^{pk.sk} \bmod n$
- Correctness means
  - $pt = (pt)^{pk.sk} \bmod n$
- Chinese remainder theorem
  - First observe that to show two quantities are equal mod n, it suffices to show they are equal mod p and mod q.

RSA: Putting it all Together

Why RSA is correct?

- Correctness means
  - $pt = (pt)^{pk.sk} \bmod n$
- Proof
  - First observe that to show two quantities are equal mod n, it suffices to show they are equal mod p and mod q.
  - $(pt)^{pk.sk} \bmod p = (pt)^{k\Phi(n) + 1} \bmod p$
  - $(pt)^{k\Phi(n) + 1} \bmod p = (pt)^{k(p-1)(q-1) + 1} \bmod p$
  - $(pt)^{k(p-1)(q-1) + 1} \bmod p = (pt^{(p-1)})^{k(q-1) + 1} \bmod p$
  - $(pt^{(p-1)})^{k(q-1) + 1} \bmod p = (pt^{(p-1)})^{k(q-1)} pt \bmod p$
  - $(pt^{(p-1)})^{k(q-1)} pt \bmod p = (1)(pt) \bmod p$ (By Fermat's Little Theorem)

RSA: Background Mathematics

Why RSA works?

- Why RSA is correct, i.e., why can you decrypt the encrypted message for the appropriate key pairs?
  - D(E(pt,pk),sk) = m
- Is it efficient?
  - Not really. Because of exponent computations with very large numbers
- Why is it secure?
- Why the inverse of the key is calculated w.r.t the Totient?

RSA: Putting it all Together

Why RSA works?

- Why is RSA secure?
  - RSA security lies in the fact that it is difficult to deduce what is private from what is public

Public information

n - the modulus

e - the public exponent (public key)

c - the cipher text

Private information

p - the prime factor of n

q - the other prime factor of n

$\varphi(n)$ - the totient of group *modulo n*

d - the private exponent (secret key)

RSA: The importance of Totient

Why RSA works?

- Learning the Totient is one way to break RSA
- We can show that one way to learn the Totient is to learn the factorization of $n = p.q$
- It is considered that factorization is hard (and there are no other good known ways to learn the Totient)
- Hence, the choice of Totient is critical to security of RSA
- RSA is not known to be provably secure

Digital Signatures

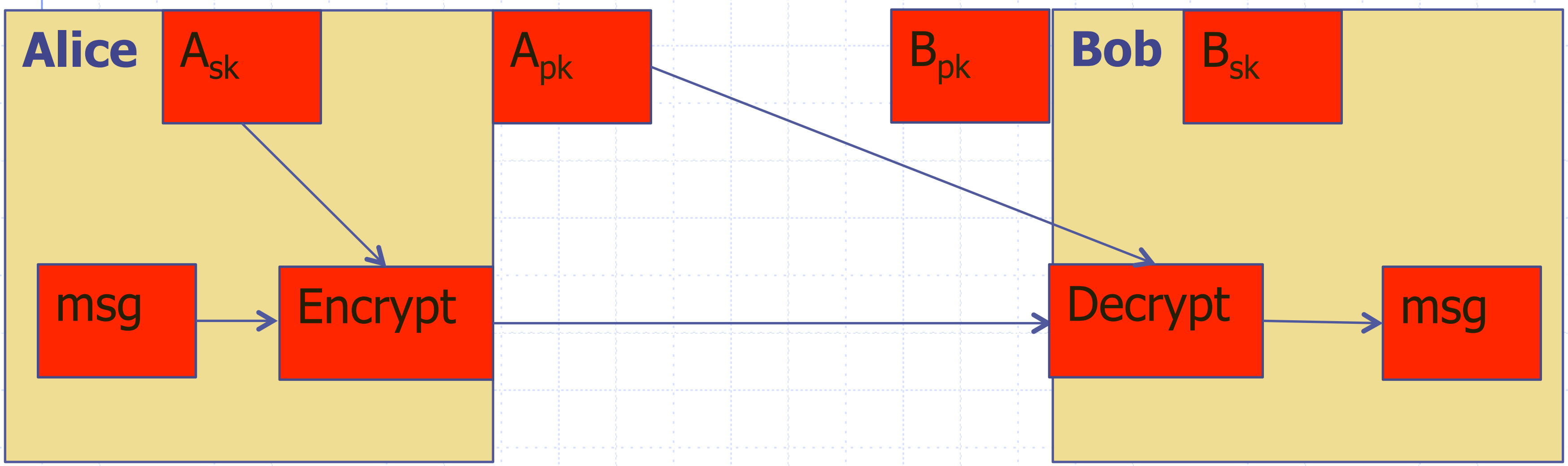
Digital Signatures

- Public-key encryption
  - Alice publishes encryption key
  - Anyone can send encrypted message
  - Only Alice can decrypt messages with this key

- Digital signature scheme
  - Alice publishes key for verifying signatures
  - Anyone can check a message signed by Alice
  - Only Alice can send signed messages

Digital Signatures

Authenticity: Alice wants to prove to Bob her identity

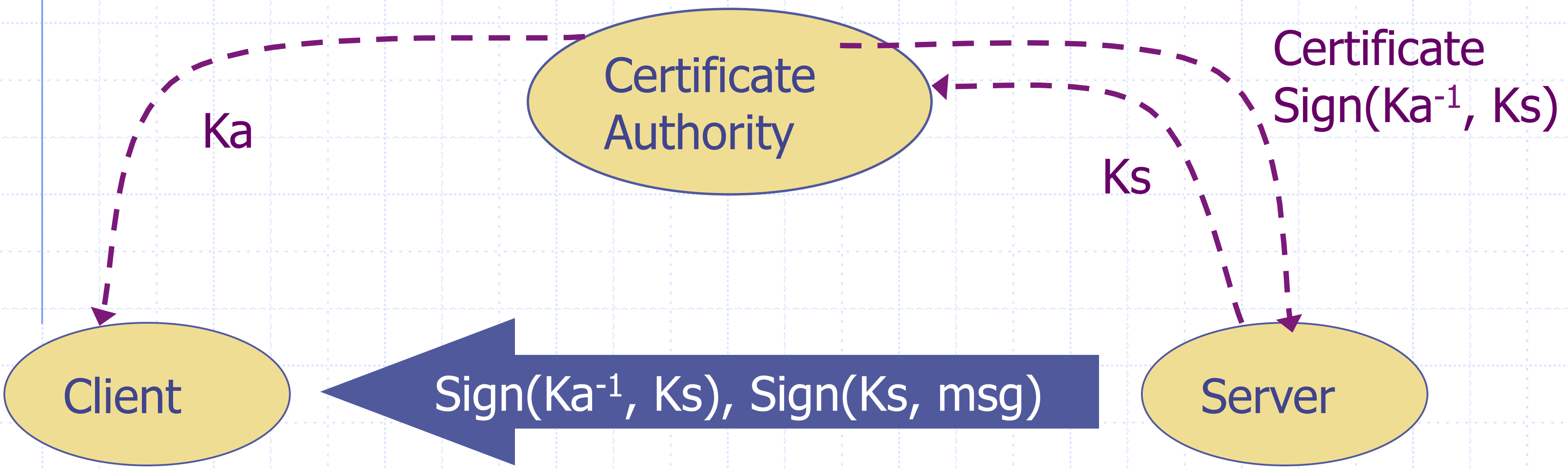


Public-Key Infrastructure (PKI)

- Anyone can send Bob a secret message
  - Provided they know Bob's public key
- How do we know a key belongs to Bob?
  - If imposter substitutes another key, can read Bob's mail
- One solution: PKI
  - Trusted root authority (VeriSign, IBM, United Nations)
    - Everyone must know the verification key of root authority
    - Check your browser; there are hundreds!!
  - Root authority can sign certificates
  - Certificates identify others, including other authorities
  - Leads to certificate chains

Public-Key Infrastructure

Known public signature verification key Ka



Server certificate can be verified by any client that has CA key Ka

Certificate authority is "off line"


Certificate Manager
Your Certificates
Other People's
Web Sites
Authorities
You have certificates on file that identify these certificate authorities:
Certificate Name
Security Device
Comodo CA Limited
Digital Signature Trust Co.
Entrust.net
Equifax
Equifax Secure
Equifax Secure Inc.
GTE Corporation
GeoTrust Inc.
GlobalSign nv-sa
Government Root Certification A...
IPS Internet publishing Services s.l.
IPS Seguridad CA
NetLock Halozatbiztonsagi Kft.
QuoVadis Limited
RSA Data Security, Inc.
RSA Security Inc
SECOM Trust.net
Sonera
View
Edit
Import
Delete
OK

The New York Times

An Attack Sheds Light on Internet Security Holes

By RIVA RICHMOND

Published: April 6, 2011

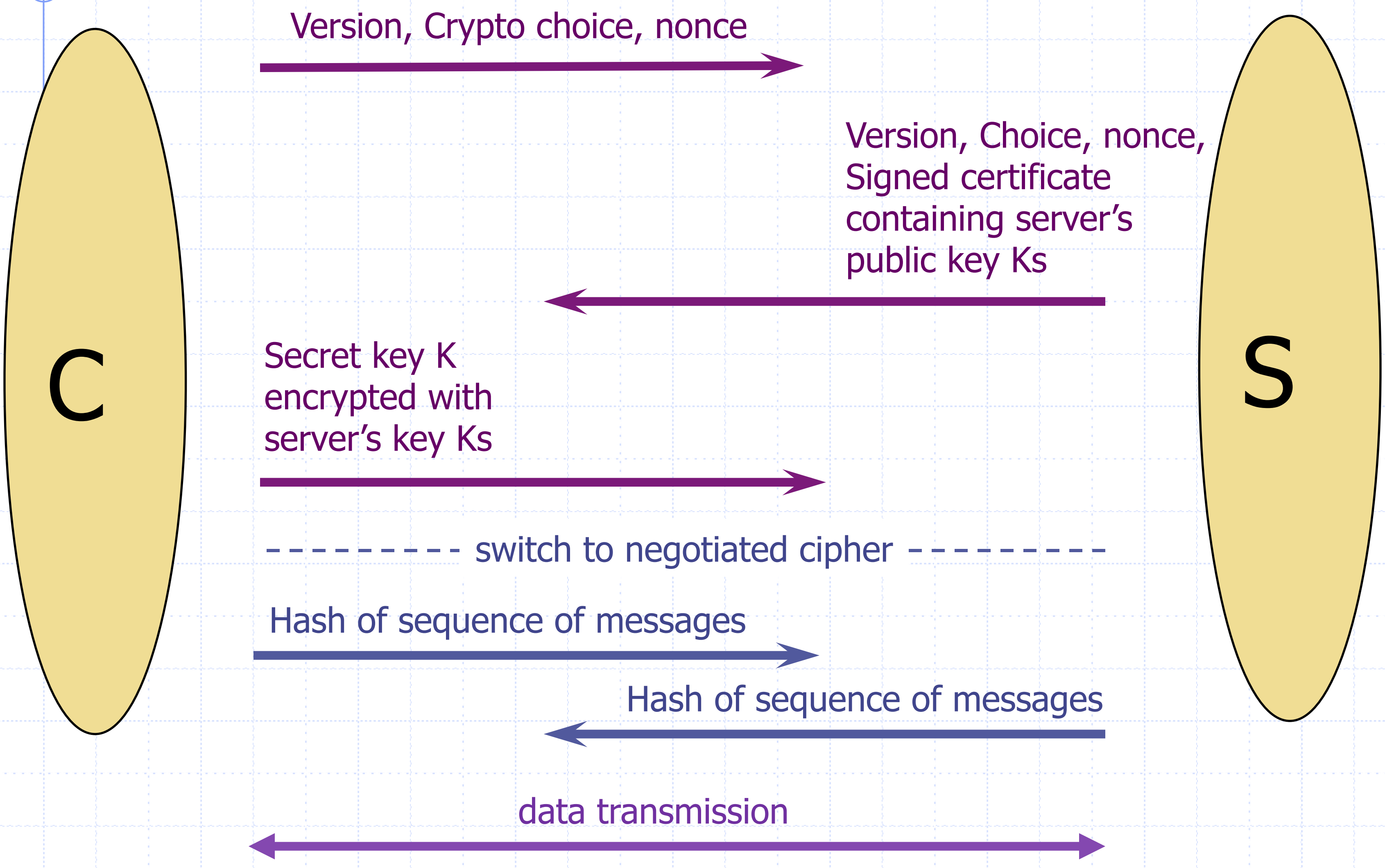
The Comodo Group, an Internet security company, has been attacked in the last month by a talkative and professed patriotic Iranian hacker who infiltrated several of the company's partners and used them to threaten the security of myriad big-name Web sites.
But the case is a problem for not only Comodo, ... It has also cast a spotlight on the global system that supposedly secures communications and commerce on the Web.

The encryption used by many Web sites to prevent eavesdropping on their interactions with visitors is not very secure. This technology is in use when Web addresses start with "https" (in which "s" stands for secure) and a closed lock icon appears on Web browsers. These sites rely on third-party organizations, like Comodo, to provide "certificates" that guarantee sites' authenticity to Web browsers.

But many security experts say the problems start with the proliferation of organizations permitted to issue certificates.

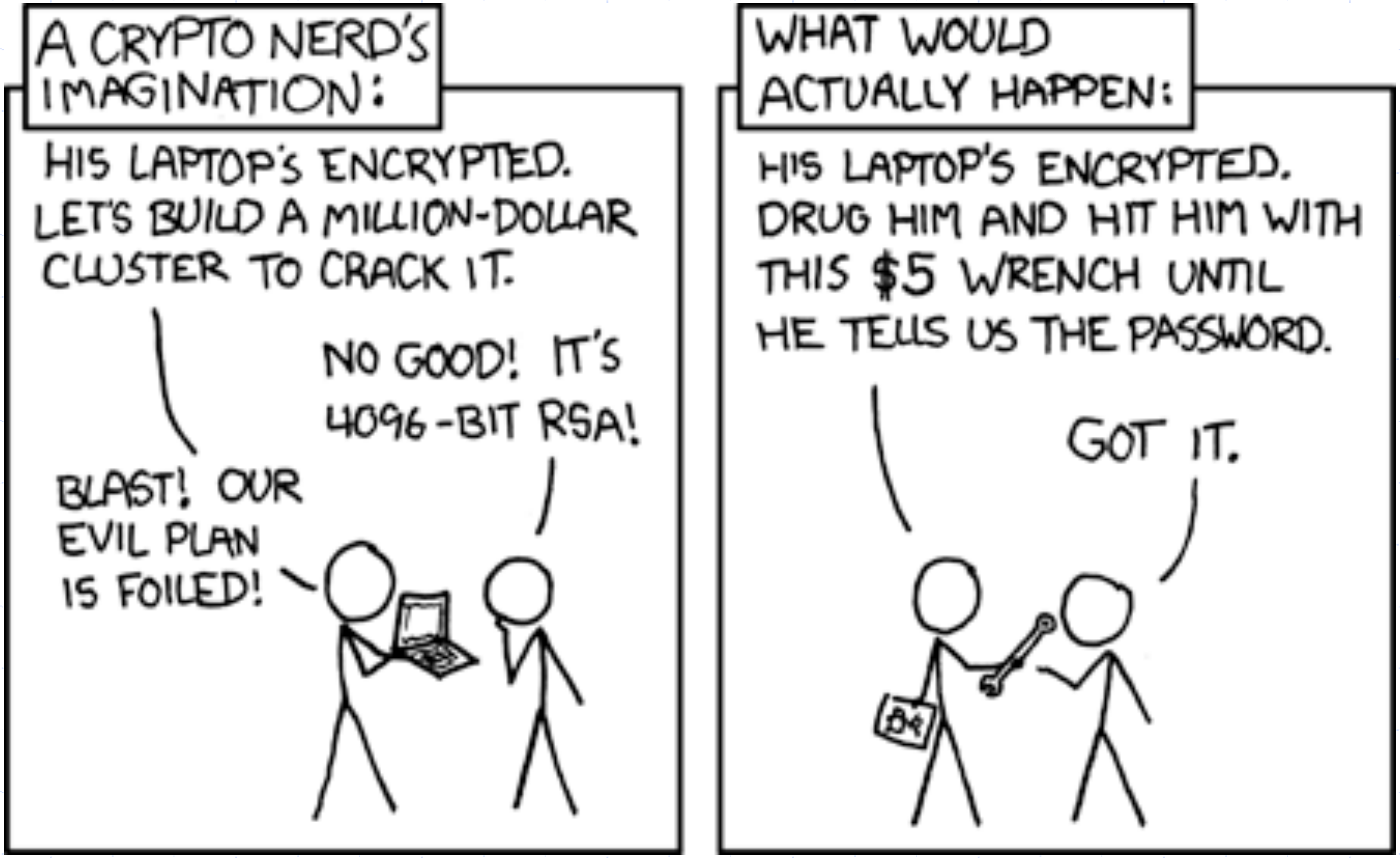
Browser makers like Microsoft, Mozilla, Google and Apple have authorized a large and growing number of entities around the world — both private companies and government bodies — to create them. Many private "certificate authorities" have, in turn, worked with resellers and deputized other unknown companies to issue certificates in a "chain of trust" that now involves many hundreds of players, any of which may in fact be a weak link.

Back to SSL/TLS



Limitations of cryptography

- Most security problems are not crypto problems
  - This is good
    - Cryptography works!
  - This is bad
    - People make other mistakes; crypto doesn't solve them
- Misuse of cryptography is fatal for security
  - WEP – ineffective, highly embarrassing for industry
  - Occasional unexpected attacks on systems subjected to serious review


A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.