University of Amsterdam

Jaap van Ginkel

Security of Systems and Networks

September 19, 2024 Public Key Crypto

SSN Projects

- Approval content Today
- Approval/restrictions ECOS3 later

Recap Hashing

- Is SHA-1 still safe?
  A) Yes
  B) No
  C) It depends

Recap Hashing

- Is SHA-2 still safe?
  - A) Yes
  - B) No
  - C) It depends

Recap Hashing

Should we switch to SHA 3?

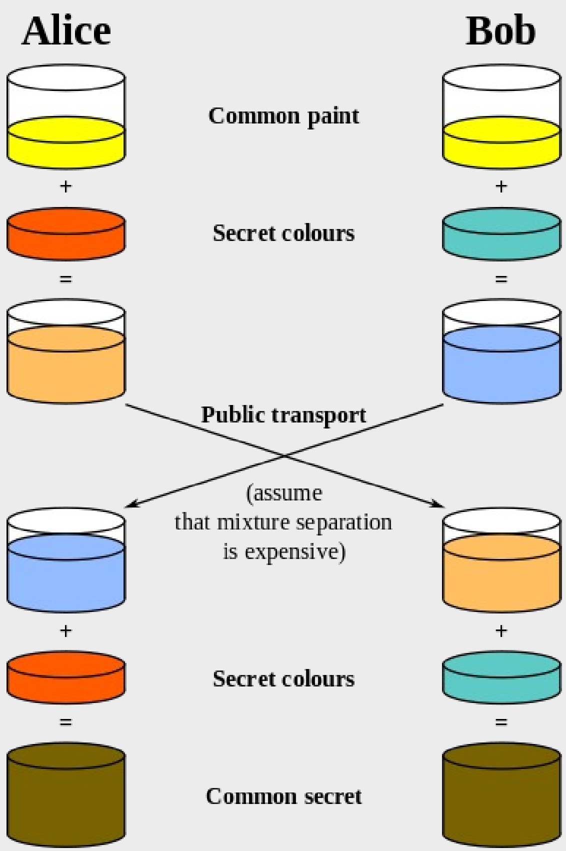
A) Yes

B) No

C) It depends

Public Key Cryptography

- Asymmetric encryption
- Expensive/Slow
- Diffie Hellmann
- RSA
- PGP



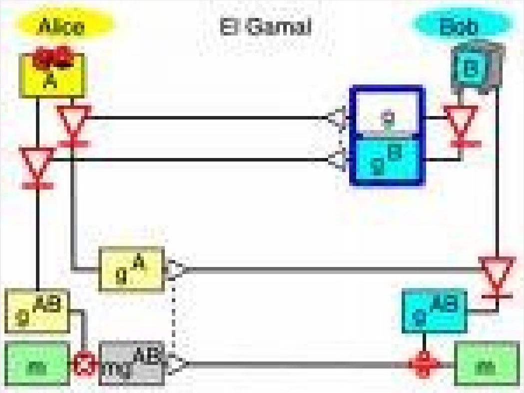
Diffie-Hellman Key Agreement Method

University of Amsterdam

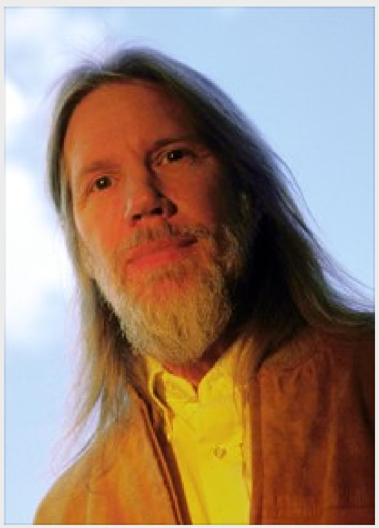
Chocolate key crypto

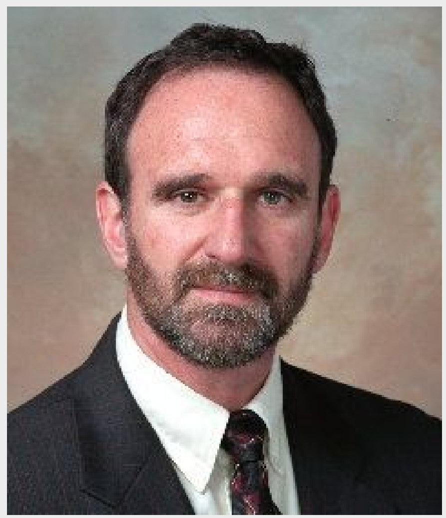
El Gamal

- Dr. Taher Elgamal
- طاهر الجمل
- Egyptian American cryptographer

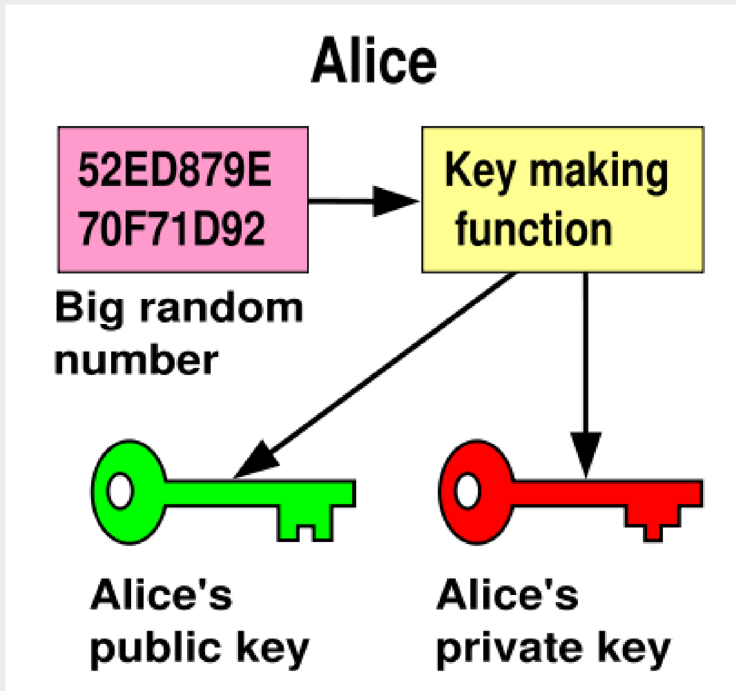


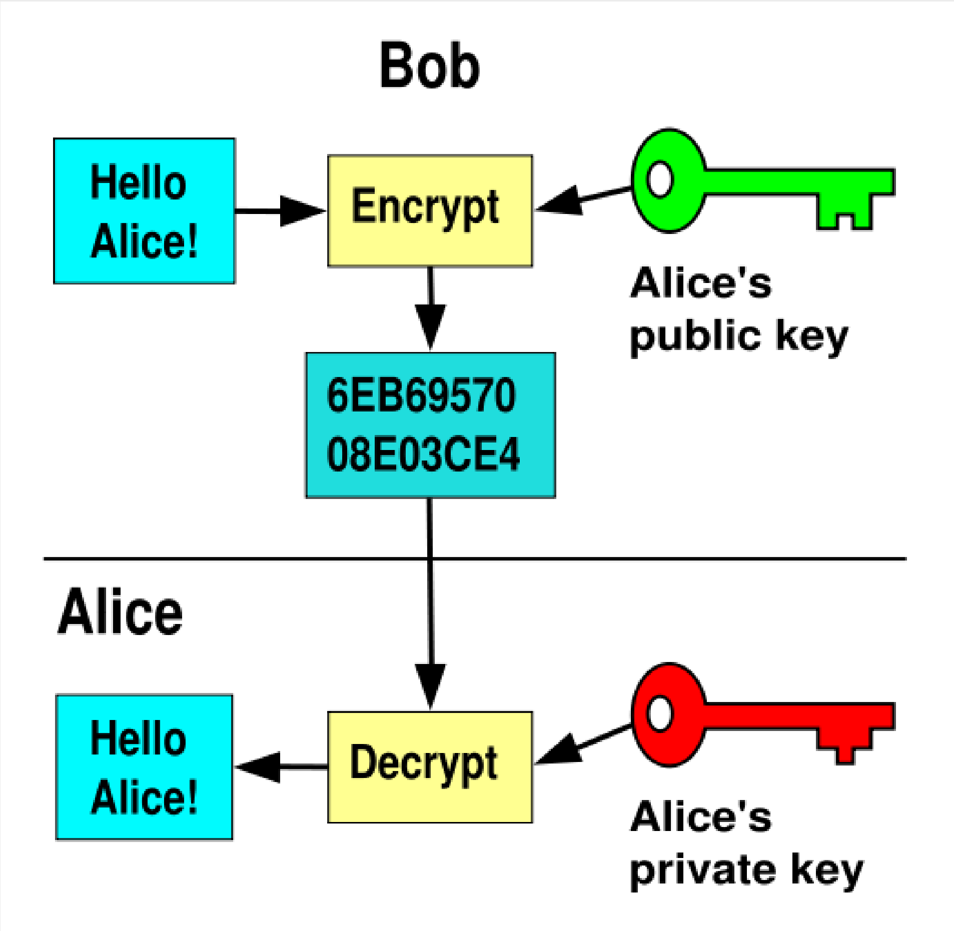
Diffie Hellman Merkle

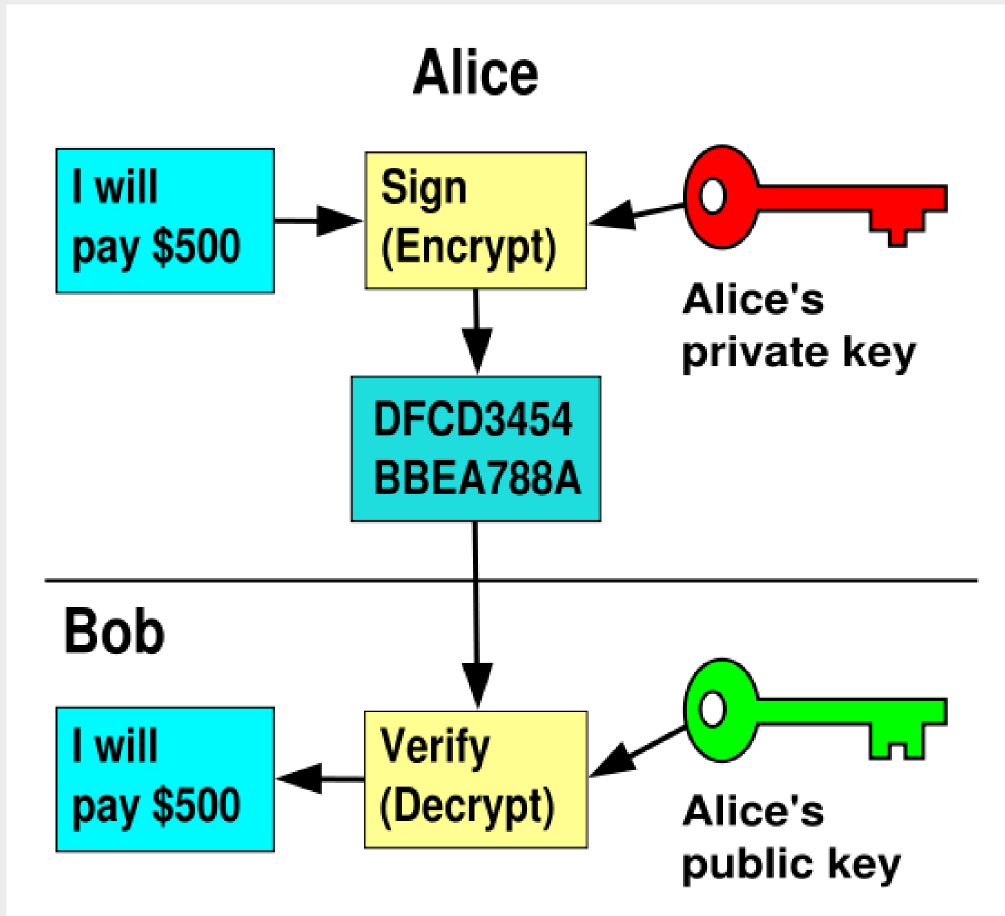




Public Key Encryption




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

William Stanley Jevons

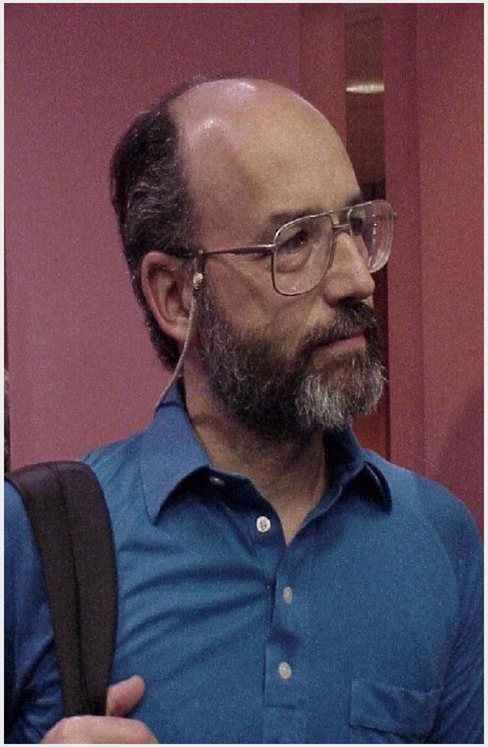
William Stanley Jevons (September 1, 1835 - August 13, 1882), English economist and logician,

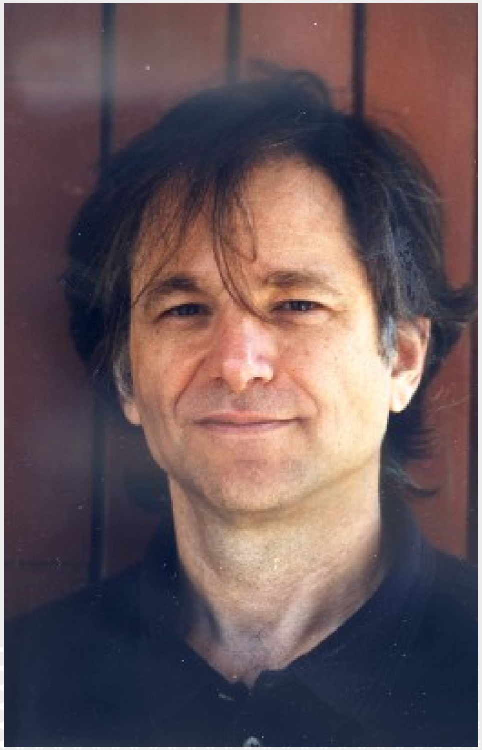
Non Secret Encryption

- James Ellis Clifford Cocks
- Secret research at GCHQ

RSA

- Ron Rivest, Adi Shamir en Len Adleman





Public Key Cryptography

You should not live one way in private, another in public.
— Publilius Syrus

Three may keep a secret, if two of them are dead.
— Ben Franklin

Public Key Cryptography

- Two keys
  - Sender uses recipient's **public key** to encrypt
  - Recipient uses **private key** to decrypt
- Based on "trap door one way function"
  - "One way" means easy to compute in one direction, but hard to compute in other direction
  - Example: Given p and q, product N = pq easy to compute, but given N, it's hard to find p and q
  - "Trap door" used to create key pairs

Public Key Cryptography

- Two keys
  - Sender uses recipient's **public key** to encrypt
  - Recipient uses **private key** to decrypt
- Based on "trap door one way function"
  - "One way" means easy to compute in one direction, but hard to compute in other direction
  - Example: Given p and q, product $N = pq$ easy to compute, but given N, it's hard to find p and q
  - "Trap door" used to create key pairs

Uses for Public Key Crypto

Non-non-repudiation

- Alice orders 100 shares of stock from Bob
- Alice computes **MAC** using symmetric key
- Stock drops, Alice claims she did ***not*** order
- Can Bob prove that Alice placed the order?
- **No!** Since Bob also knows the symmetric key, he could have forged message
- **Problem:** Bob knows Alice placed the order, but he can't prove it

Non-repudiation

- Alice orders 100 shares of stock from Bob
- Alice **signs** order with her private key
- Stock drops, Alice claims she did not order
- Can Bob prove that Alice placed the order?
- **Yes!** Only someone with Alice's private key could have signed the order
- This assumes Alice's private key is not stolen (revocation problem)

Public Key Notation

- **Sign** message M with *Alice's* **private key:** [M]Alice
- **Encrypt** message M with *Alice's* **public key:** {M}Alice
- Then

  {[M]Alice}Alice = M

  [{M}Alice]Alice = M

Public Key Infrastructure

Public Key Certificate

- **Certificate** contains name of user and user's public key (and possibly other info)
- It is ***signed*** by the issuer, a ***Certificate Authority*** (CA), such as VeriSign

    M = (Alice, Alice's public key), S = [M]CA

    Alice's Certificate = (M, S)

    - Signature on certificate is verified using CA's public key:

    Verify that M = {S}CA

Certificate Authority

- Certificate authority (CA) is a trusted 3rd party (TTP) — creates and signs certificates
- Verify signature to verify integrity & identity of **owner of corresponding private key**
  - Does **not** verify the identity of the **sender** of certificate — certificates are public keys!
- Big problem if CA makes a mistake (a CA once issued Microsoft certificate to someone else)
- A common format for certificates is X.509

PKI

- Public Key Infrastructure (PKI): the stuff needed to securely use public key crypto
  - Key generation and management
  - Certificate authority (CA) or authorities
  - Certificate revocation lists (CRLs), etc.
- No general standard for PKI
- We mention 3 generic “trust models”

PKI Trust Models

- Monopoly model
  - One universally trusted organization is the CA for the known universe
  - Big problems if CA is ever compromised
  - Who will act as CA???
    - System is useless if you don't trust the CA!

PKI Trust Models

- Monopoly model
  - One universally trusted organization is the CA for the known universe
  - Big problems if CA is ever compromised
  - Who will act as CA???
    - System is useless if you don't trust the CA!

PKI Trust Models

- Monopoly model
  - One universally trusted organization is the CA for the known universe
  - Big problems if CA is ever compromised
  - Who will act as CA???
    - System is useless if you don't trust the CA!

Confidentiality in the Real World

Symmetric Key vs Public Key

- Symmetric key +'s
  - **Speed**
  - No public key infrastructure (PKI) needed
- Public Key +'s
  - **Signatures** (non-repudiation)
  - No ***shared*** secret (but, private keys…)

Notation Reminder

- Public key notation
  - Sign M with Alice's **private key**

    [M]Alice
  - Encrypt M with Alice's **public key**

    {M}Alice
- Symmetric key notation
  - Encrypt P with symmetric key K

    C = E(P,K)
  - Decrypt C with symmetric key K

    P = D(C,K)