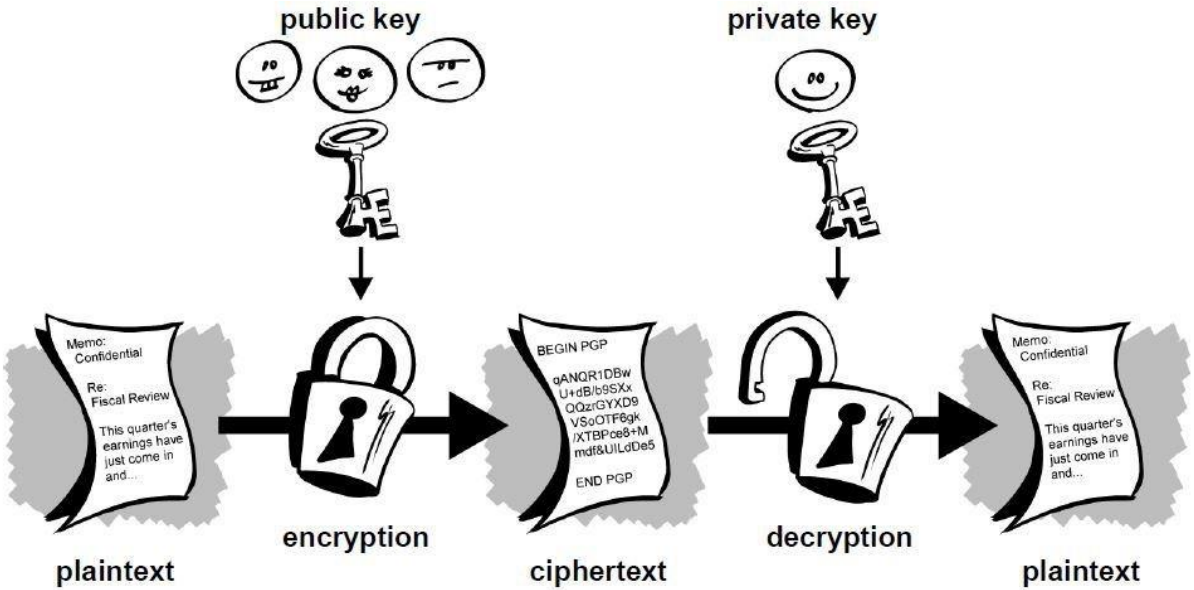
Introduction to Cryptography

By
Vipul Goyal

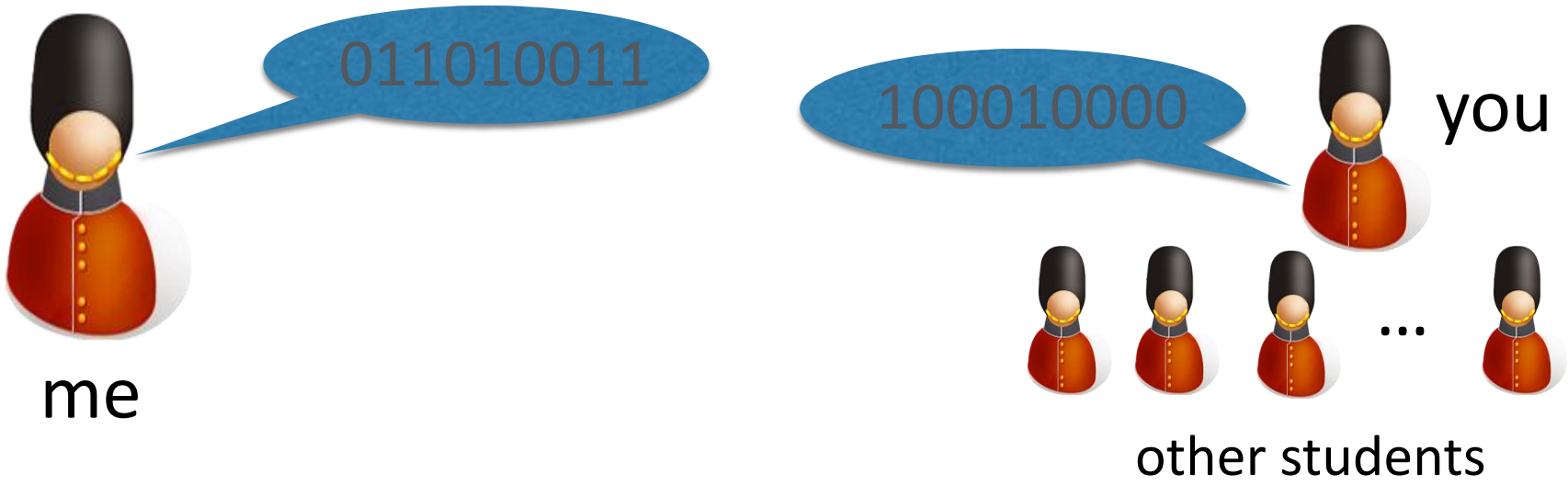


Key Exchange over the Internet

Key Exchange

Private key encryption relies on parties having a shared secret

Say I want to communicate with one of you securely. Never met. No private chat. Only speaking publicly on Zoom.

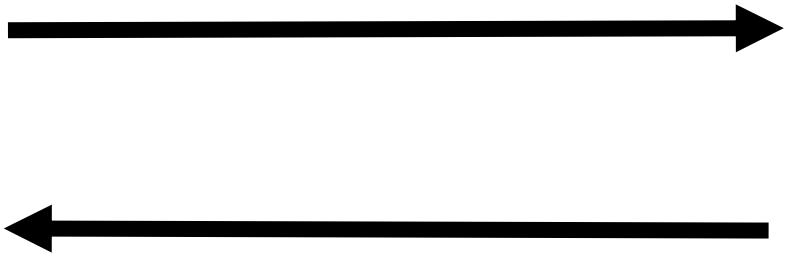


- You and I talk publicly
- You and I now have a shared key
- Other students listening can't compute it
- Impossible?

Key Exchange

Alice

Bob



Can Alice and Bob agree on a secret via a *completely public conversation*?

- Over the internet with adversary watching?

Diffie-Hellman Key Exchange

Alice

Bob

$(N, g), A = g^a$

$B = g^b$

sample a

sample b

Key $K = g^{ab} = A^b = B^a$

Alice: has a and $B = g^b$. Computes $K = B^a = g^{ab}$

Bob: has b and $A = g^a$. Computes $K = A^b = g^{ab}$

Diffie-Hellman Key Exchange

Alice

$(N, g), A = g^a$

$B = g^b$

Bob

sample a

sample b

Can Adv compute g^{ab}? Adv only has g^a and g^b

(but neither a nor b)

CDH/DDH say: this is hard!

After Key Exchange

Alice

Bob

$(N, g), A = g^a$

$B = g^b$

$C = M.g^{ab}$

sample a

sample b

To decrypt: compute $K = g^{ab}$, and K^{-1}

Recover message as $C.K^{-1}$

ElGamal Public-Key Encryption

Defining PKE

1) Gen: No input. Outputs PK and SK
2) Enc: Takes input *PK* and *M*. Outputs *C*.
3) Dec: Takes input *C* and SK. Outputs M.

Correctness: If (PK, SK) are output of Gen, must have

Dec(Enc(M, PK), SK) = M

Security: Should hide the message?

Security

Given C, probability of computing M is very small?

Given C, probability of computing M is at most ½?

Intuition: C should give “no information” about M

Security: Adv can’t tell apart encryption of M from encryption of a random message

Attempt at Building PKE?

Alice, who has never spoken to Bob, wants
to send him message m in encrypted form Enc(m)

Recovering m from Enc(m) should be a hard problem

How about $Enc(m) = g^m$

Discrete log hardness $\Rightarrow$privacy from eavesdropper?

But how will Bob figure out m?

- *He has to solve the same discrete log problem!*

Back to Diffie-Hellman Key Exchange

Alice

Bob

$(N, g), A = g^a$

$B = g^b$

$C = M.g^{ab}$

sample a

sample b

Maybe PK = (N, g, A), CT = (B, C)

ElGamal Public Key Encryption (1985)

Idea: Instead of sending $(N, g), g^a$ just to Bob,
Alice publishes this as her public key PK

$$PK = (N, g, g^a)$$

- Keeps a as her secret key SK

To encrypt M: Bob does exactly as in DH KE. Bob samples random b, computes g^{ab}, and uses it to mask the message

$$Enc(M, PK) = (g^b, M.g^{ab})$$

To decrypt: Alice computes g^{ab} using g^b and a. Computes its inverse. Recovers M from $M.g^{ab}$

Security

Adversary sees;

$PK = (N, g, g^a)$

$C = (g^b, M.g^{ab})$

DDH Assumption: Given (N, g, g^a, g^b), can't distinguish g^{ab} from random

One can show: looks same as random

ElGamal Encryption is Randomized

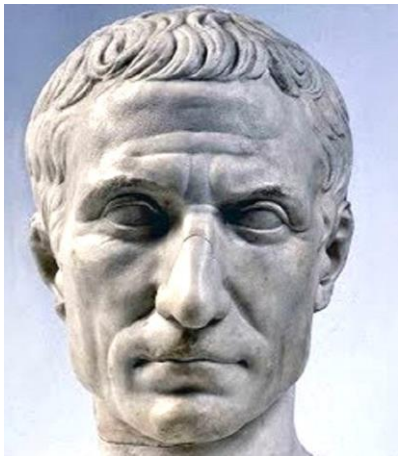
To encrypt M: Bob samples random b, computes g^{ab}, and uses it to mask the message

$$Enc(M, PK) = (g^b, M.g^{ab})$$

Everytime b will be different. Hence, even if you encrypt the same M, you might get different ciphertexts!

Randomized Encryption?

Randomized encryption is a
feature rather than a bug



jkds0923jd

3ljsf0321ld

jkds0923jd

Deterministic encryption = bad security

ElGamal Public Key Encryption (1985)

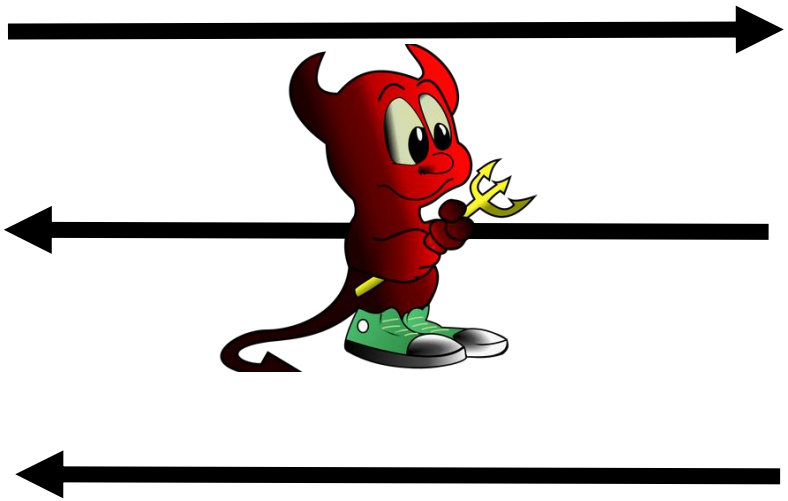
It took 8+ years from the Diffie-Hellman key exchange to the ElGamal encryption scheme

- In fact, this was not the first proposal for a PKE
- That honor belongs to the RSA scheme (1978)
- But ElGamal remains the simplest PKE

Putting Signatures and Encryption to Work: HTTPS/SSL protocol

How a new pair of parties could communicate?

You

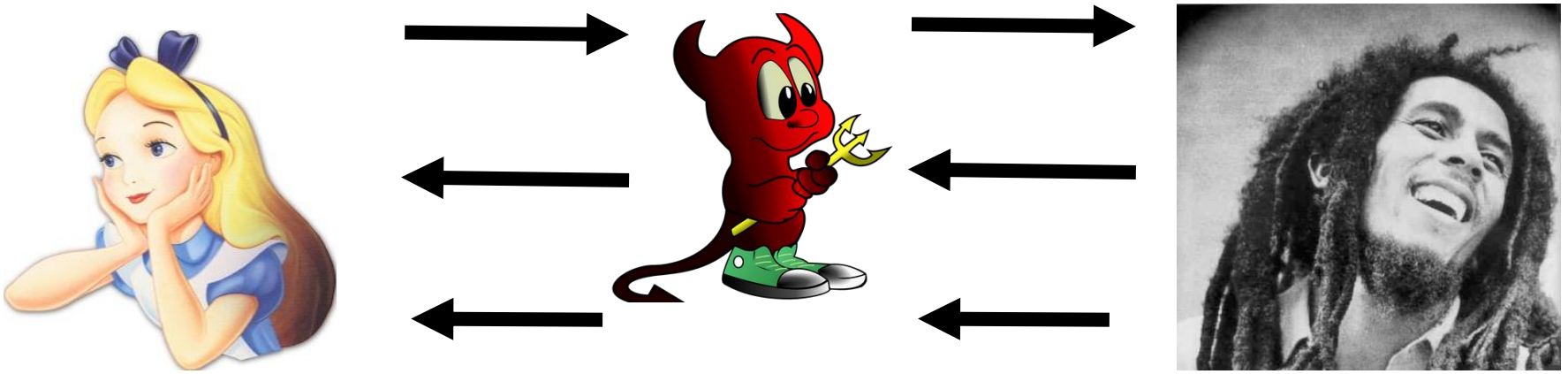


Google

DH KE? What if adv can modify messages?

Run key exchange with Alice and Bob separately!

How a new pair of parties could communicate?



K_1 K_2

Run key exchange with Alice and Bob separately!

Decrypt Alice's message using K_1, read, encrypt under K_2

and send to Bob!!

Use PKE?

You

PK

Enc(PK, m)

Google

Google sends you their PK, you encrypt?

Adv changes PK to PK_{adv}

Certificates and Certificate Authorities

Authority

"PK is the public key of Google.com"

Digitally signed by authority

Google

HTTPS

You

PK

Google

Enc(PK, m)

Adversary can’t change certificate since its digitally signed

How do You Verify the Certificate?

$VK_{authority}$

$VK_{authority}$ is inside your browser/OS

Questions?