The Diffie-Hellman Algorithm

Overview

- Introduction
- Implementation
- Example
- Applications
- Conclusion

Introduction

- Discovered by Whitfield Diffie and Martin Hellman
  - “New Directions in Cryptography”
- Diffie-Hellman key agreement protocol
  - Allows two users to exchange a secret key
  - Requires no prior secrets
  - Real-time over an untrusted network

Introduction

- Security of transmission is critical for many network and Internet applications
- Requires users to share information in a way that others can't decipher the flow of information

"It is insufficient to protect ourselves with laws; we need to protect ourselves with mathematics."

-Bruce Schneier

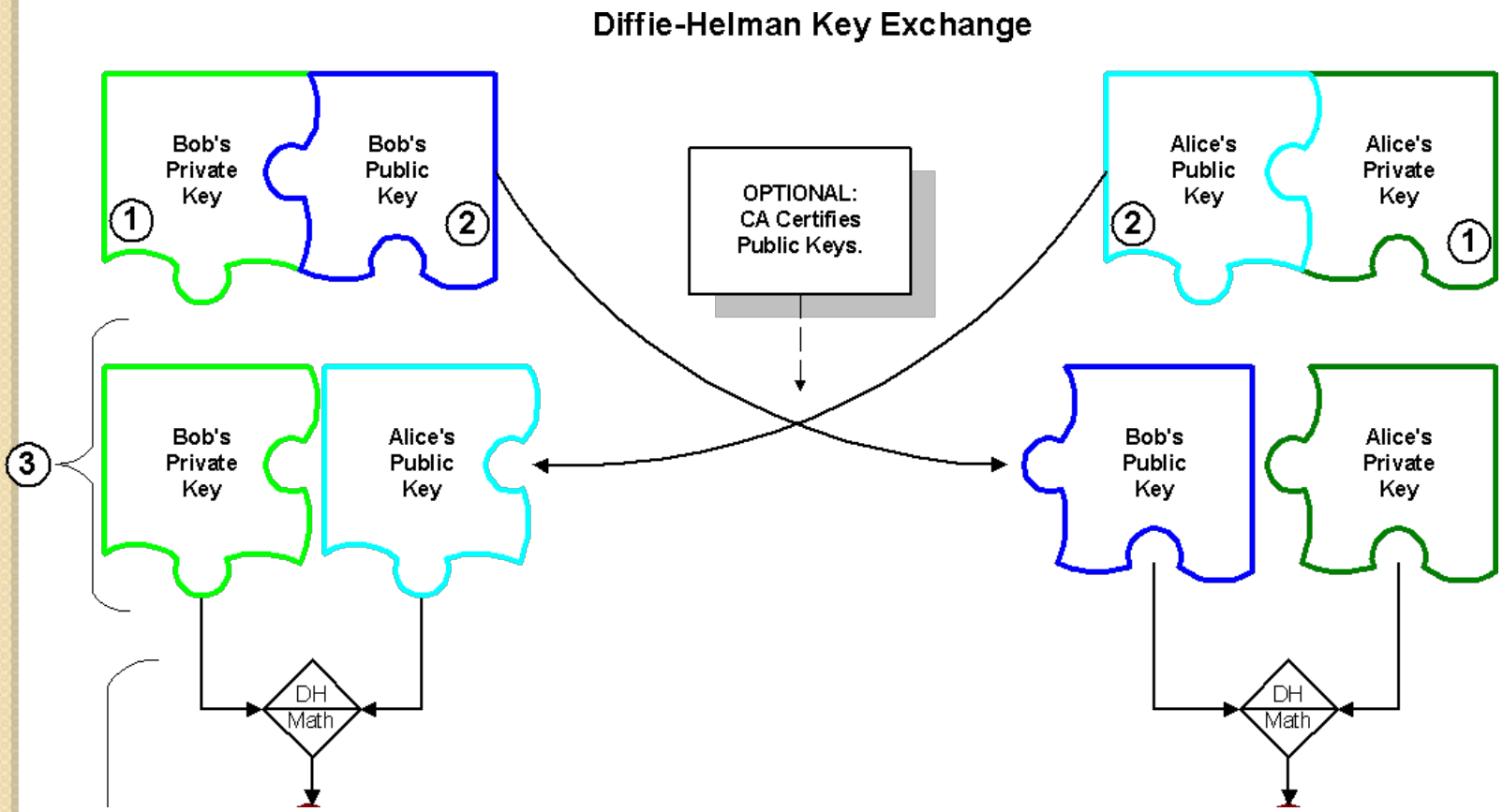
Introduction

- Based on the difficulty of computing discrete logarithms of large numbers.
- No known successful attack strategies*
- Requires two large numbers, one prime (P), and (G), a primitive root of P

Implementation

- P and G are both publicly available numbers
  - P is at least 512 bits
- Users pick private values a and b
- Compute public values
  - $x = g^a \bmod p$
  - $y = g^b \bmod p$
- Public values x and y are exchanged

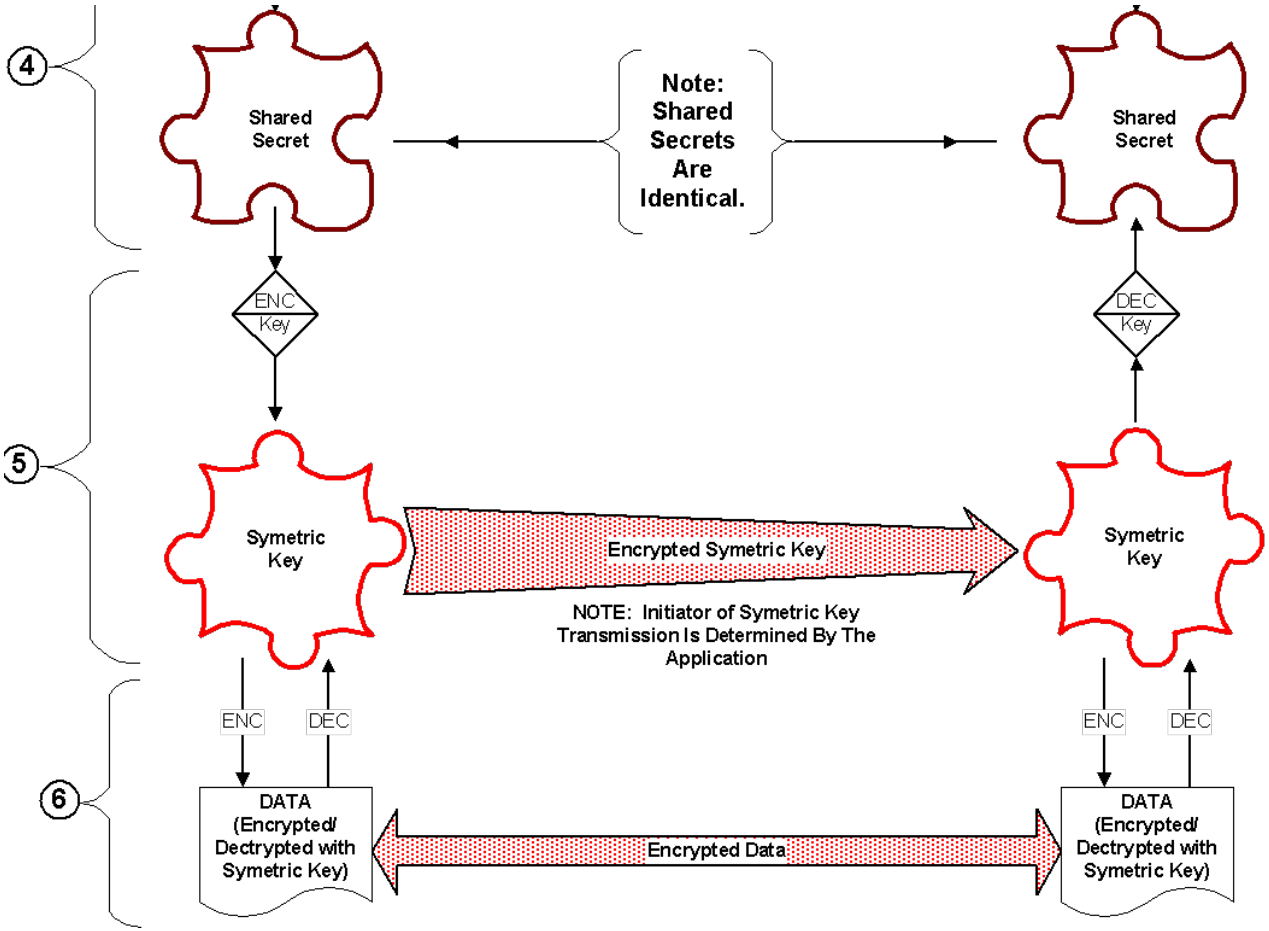
Implementation



Implementation

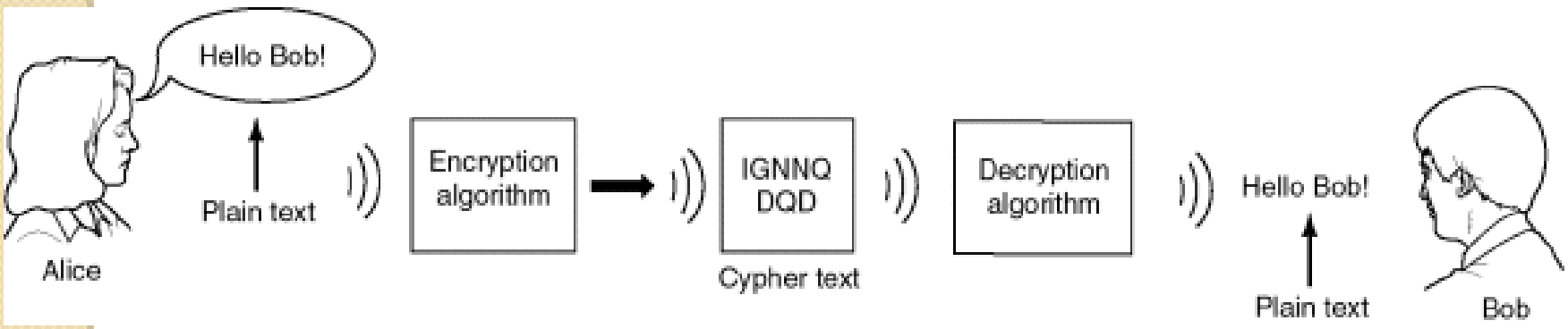
- Compute shared, private key
  - $k_a = y^a \bmod p$
  - $k_b = x^b \bmod p$
- Algebraically it can be shown that $k_a = k_b$
  - Users now have a symmetric secret key to encrypt

Implementation



Example

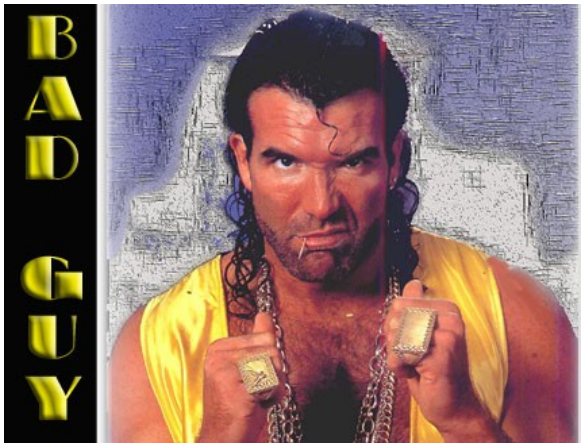
- Two Internet users, Alice and Bob wish to have a secure conversation.
  - They decide to use the Diffie-Hellman protocol

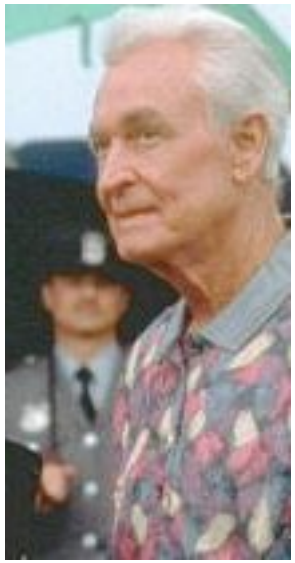


Example

- Bob and Alice are unable to talk on the untrusted network.

–Who knows who's listening?





Example

- Alice and Bob get public numbers
  - P = 23, G = 9
- Alice and Bob compute public values
  - $X = 9^4 \bmod 23 = 6561 \bmod 23 = 6$
  - $Y = 9^3 \bmod 23 = 729 \bmod 23 = 16$
- Alice and Bob exchange public numbers

Example

- Alice and Bob compute symmetric keys
  - $= 16^4$
  - $= 6^3$
  - now

Applications

- Diffie-Hellman is currently used in many protocols, namely:
  - Secure Sockets Layer (SSL)/Transport Layer Security (TLS)
  - Secure Shell (SSH)
  - Internet Protocol Security (IPSec)
  - Public Key Infrastructure (PKI)

Man-in-the-Middle Attack

- Alice and Bob wants to communicate.
- Darth is a middle man.
- Darth prepares for the attack by generating two random private keys K_{d1} and K_{d2} and then computing the public keys Y_{d1} and Y_{d2}.
- Darth intercepts Y_A and transmits Y_{d1} to Bob.
- He also calculates K_2 (secret key between Alice and Darth).
- Bob receives K_{d1} and calculates K_1.
- Bob transmits Y_B to Alice.
- Darth intercepts Y_B and transmits K_{d2} to Alice and also calculates K_1.
- Alice receives K_{d2} and calculates K_2.

Digital Signatures

- have looked at message authentication
  - but does not address issues of lack of trust
- digital signatures provide the ability to:
  - verify author, date & time of signature
  - authenticate message contents
  - be verified by third parties to resolve disputes
- hence include authentication function with additional capabilities

Digital Signature Properties

- must depend on the message signed
- must use information unique to sender
  - to prevent both forgery and denial
- must be relatively easy to produce
- must be relatively easy to recognize & verify
- be practical save digital signature in storage

Direct Digital Signatures

- involve only sender & receiver
- assumed receiver has sender's public-key
- digital signature made by sender signing entire message or hash with private-key
- can encrypt using receivers public-key
- important that sign first then encrypt message & signature

Arbitrated Digital Signatures

- involves use of arbiter A
  - validates any signed message
  - then dated and sent to recipient
- requires suitable level of trust in arbiter
- can be implemented with either private or public-key algorithms
- arbiter may or may not see message