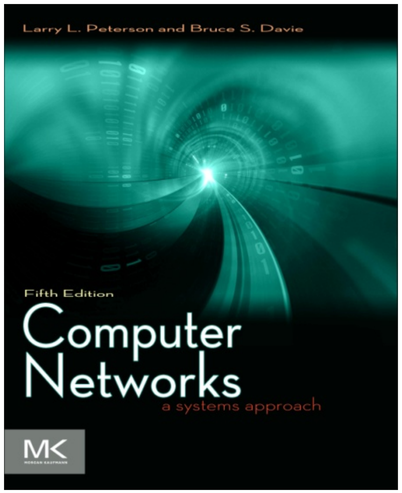
CPE348: Introduction to Computer Networks

Lecture #21: Chapter 8

Jianqing Liu
Assistant Professor of Electrical and Computer Engineering, University of Alabama in Huntsville

jianqing.liu@uah.edu
http://jianqingliu.net

*Some slides are borrowed from Dr. Kevin Butler at UF



Chapter 8

Network Security

Cryptosystem

A cryptosystem is a 5-tuple consisting of

$$(E, D, M, K, C)$$

Where,

E is an *encryption* algorithm

D is an *decryption* algorithm

M is the set of *plaintexts*

K is the set of *keys*

C is the set of *ciphertexts*

Cryptosystem – key

- A key is an input to a cryptographic algorithm used to obtain confidentiality, integrity, authenticity or other property over some data.
  - The security of the cryptosystem often depends on keeping the key secret to some set of parties.
  - The *keyspace* is the set of all possible keys
  - *Entropy* is a measure of the variance in keys
    - typically measured in bits
- Keys are often stored in some secure place:
  - passwords, on disk keyrings, ...
  - TPM, secure co-processor, smartcards, ...
- ... and sometimes not, e.g., certificates

Cryptosystem – algorithm

- Algorithm used to make content unreadable by all but the intended receivers

E(key,plaintext) = ciphertext

D(key,ciphertext) = plaintext

- *Algorithm is public, key is private*
- Block vs. Stream Ciphers
  - Block: input is fixed blocks of same length
  - Stream: stream of input

Cryptosystem – hardness

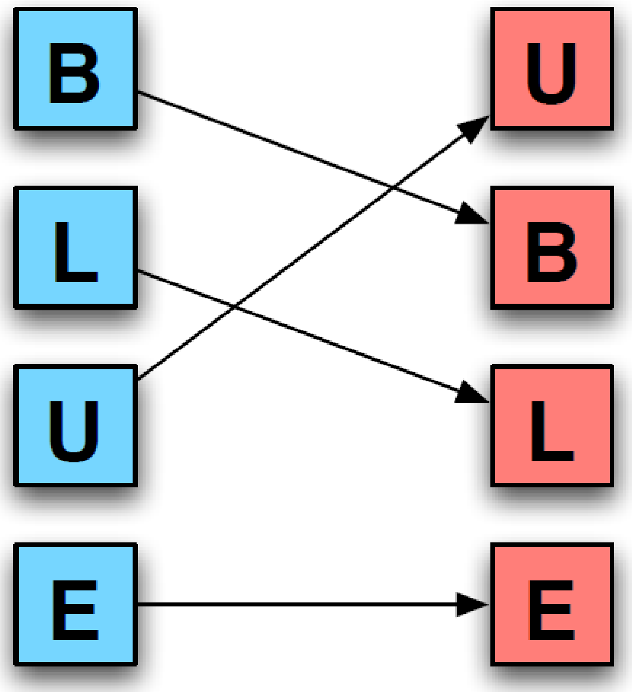
- Functions
  - Plaintext P
  - Ciphertext C
  - Encryption (E) key k_e
  - Decryption (D) key k_d

$$D(E(P, k_e), k_d) = P$$

- Computing P from C is hard, computing P from C with k_d
  - Is easy for all Ps (operation true for all inputs) ...
  - ... except in some vanishingly small number of cases

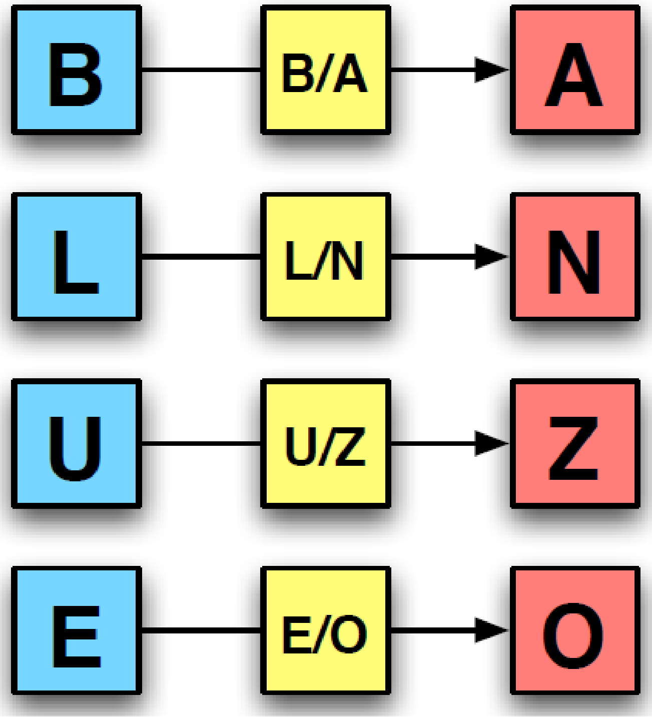
History of Cryptography – Crypto 0.0

- Transposition Ciphers



History of Cryptography – Crypto 0.0

- Substitution Ciphers

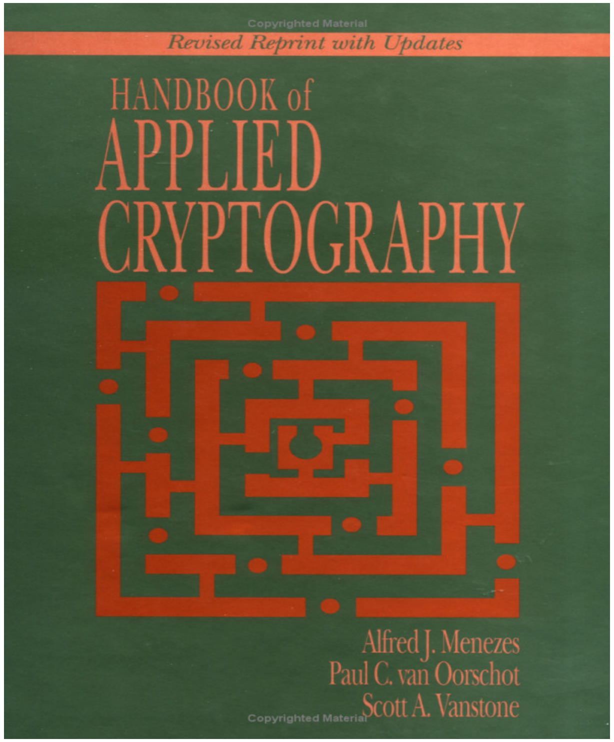


History of Cryptography – Crypto 1.0

- Crypto 1.0 concerns
  - encryption and authentication of data,
  - during communication and storage/retrieval
- Crypto 1.0 primitives:
  - Symmetric (secret key):
    -Stream/block ciphers
    -Message authentication codes
  - Asymmetric (public key):
    -Public-key encryption
    -Digital signatures
    -Key-exchange protocols
  - Keyless:
    -Cryptographic hash functions

History of Cryptography – Crypto 1.0

- http://cacr.uwaterloo.ca/hac/
- Handbook of Applied Cryptography
- By *Alfred J. Menezes*, *Paul C. van Oorschot* and *Scott A. Vanstone*



CRC Press
ISBN: 0-8493-8523-7
October 1996, 816 pages

Fifth Printing (August 2001)

History of Cryptography – Crypto 1.0

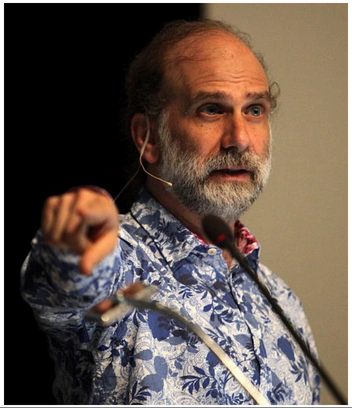
- Ch. 1 - Overview of Cryptography
- Ch. 2 - Mathematics Background
- Ch. 3 - Number-Theoretic Reference Problems
- Ch. 4 - Public-Key Parameters
- Ch. 5 - Pseudorandom Bits and Sequences
- Ch. 6 - Stream Ciphers
- Ch. 7 - Block Ciphers
- Ch. 8 - Public-Key Encryption
- Ch. 9 - Hash Functions and Data Integrity
- Ch. 10 - Identification and Entity Authentication
- Ch. 11 - Digital Signatures
- Ch. 12 - Key Establishment Protocols
- Ch. 13 - Key Management Techniques
- Ch. 14 - Efficient Implementation
- Ch. 15 - Patents and Standards

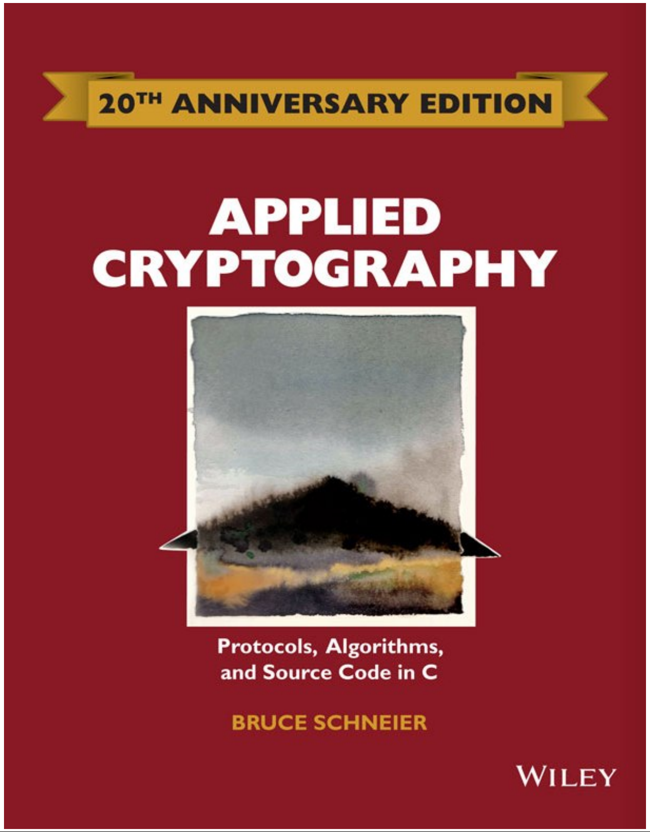
Handbook of APPLIED CRYPTOGRAPHY

History of Cryptography – Crypto 1.0

- Applied Cryptography: Protocols, Algorithms, and Source Code in C
- By *Bruce Schneier*

John Wiley & Sons
ISBN 978-1-119-09672-6
1996, 784 Pages



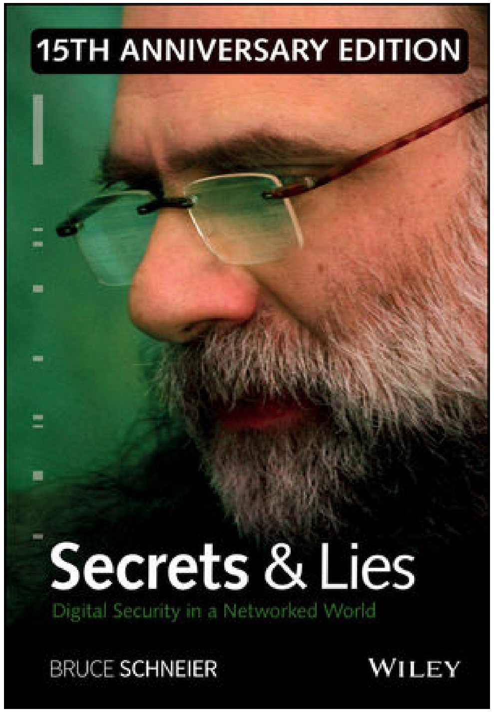


History of Cryptography – Crypto 1.0

- "A colleague once told me that the world was full of bad security systems designed by people who read *Applied Cryptography*."

  --- *Bruce Schneier*

- So, please be extra careful!



History of Cryptography – Crypto 2.0

- Crypto 2.0 additionally concerns
  - computing with encrypted data,
  - partial information release of data,
  - hiding identity of data owners or any link with them.
- Crypto 2.0 primitives:
  - homomorphic encryption
  - secret sharing
  - blind signatures
  - oblivious transfer
  - zero-knowledge proofs
  - secure two/multi-party computation
  - functional encryption
  - indistinguishable obfuscation

History of Cryptography – Crypto 2.0

- No books or graduate courses cover Crypto 2.0
- Only research papers, papers, papers
- A huge gap…

This lecture

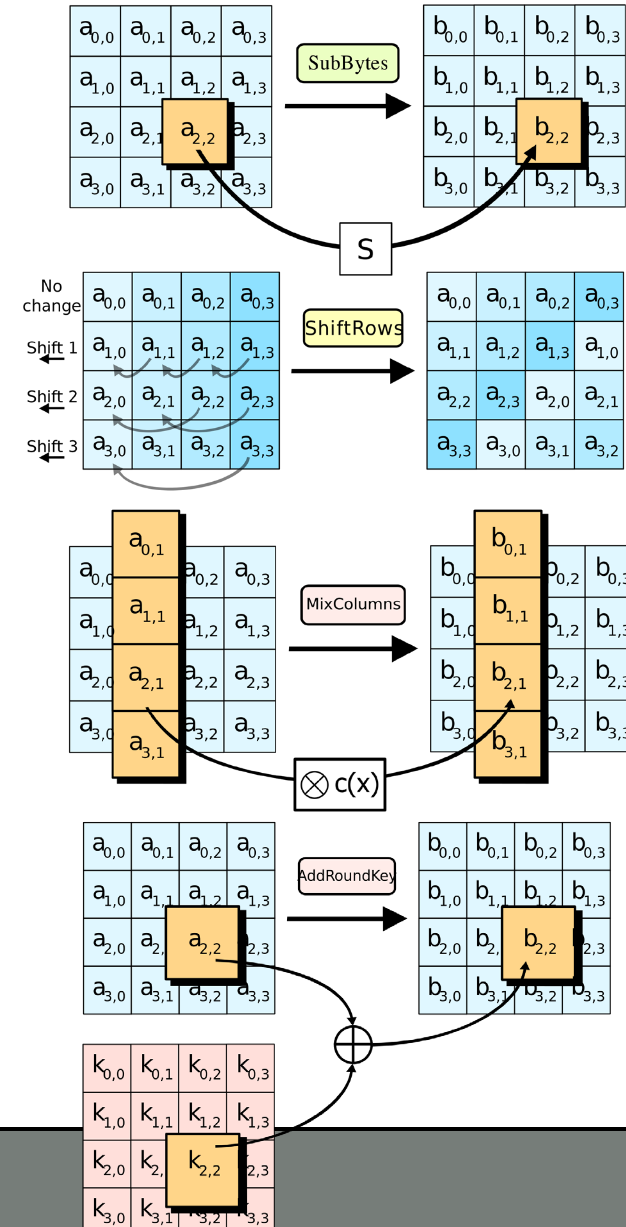
We will only talk about several Crypto 1.0 techniques…

…and some applied crypto in computer networks!

Advanced Encryption Standard (AES)

- Rijndael (pronounced "Rhine-dall")
- Currently implemented in many devices and software, but there are still DES holdouts

- AES takes 128, 192 or 256 bit keys;
- AES repeats many rounds (10,12,14) of transformation (a.k.a., substitution-permutation network) to encrypt the plaintext.



Hash Function

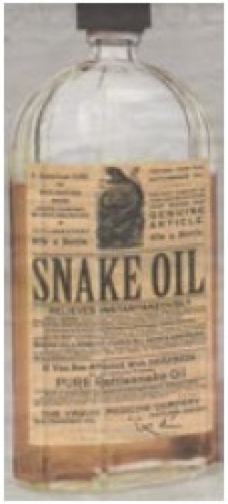
- Hash algorithm
  - Compression of data into a hash value
  - E.g., h(d) = parity(d)
  - Such algorithms are generally useful in systems (speed/ space optimization)
- … as used in cryptosystems
  - *One-way* - (computationally) hard to invert h() , i.e., compute $h^{-1}(y)$, where y=h(d)
  - *Collision resistant* hard to find two data x_1 and x_2 such that $h(x_1) == h(x_2)$

Hash Function

- How do you design a "strong cryptographic hash function?"
- No formal basis
  - Concern is backdoors
- MD2, MD4, MD5 (128bit):
  - Broken, Broken, Broken
  - MD4, MD5: Similar, but complex functions in multiple passes
- SHA-1 (160 bit)
  - "Complicated function"
  - Theoretical weaknesses
- SHA-2 (224, 256, 384 or 512-bit)
- SHA-3 (224, 256, 384 or 512-bit)

Basic truths of cryptography…

- Cryptography is not frequently the source of security problems
  - Algorithms are well known and widely studied
    - Use of crypto commonly is … (e.g., WEP)
  - Vetted through crypto community
  - Avoid any "proprietary" encryption
  - Claims of "new technology" or "perfect security" are almost assuredly snake oil



Common issues that lead to pitfalls

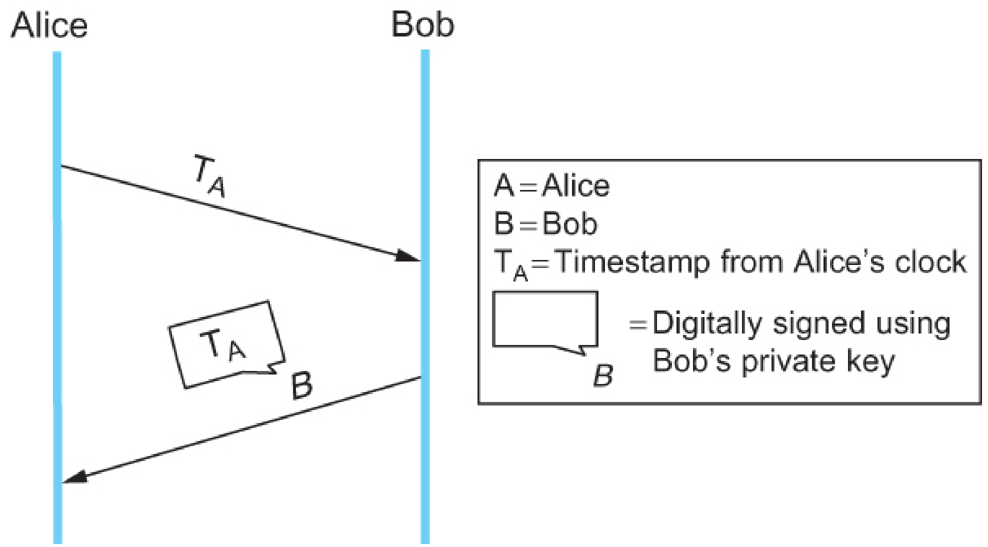
- Generating randomness
- Storage of secret keys
- Virtual memory (pages secrets onto disk)
- Protocol interactions
- Poor user interface
- Poor choice of key length, prime length, using parameters from one algorithm in another

Important Principles

- Don't design your own crypto algorithm
  - Use standards whenever possible
- Make sure you understand parameter choices
- Make sure you understand algorithm interactions
  - E.g. the order of encryption and authentication
    - Turns out that authenticate then encrypt is risky
- Be open with your design
  - Solicit feedback
  - Use open algorithms and protocols
  - Open code? (jury is still out)

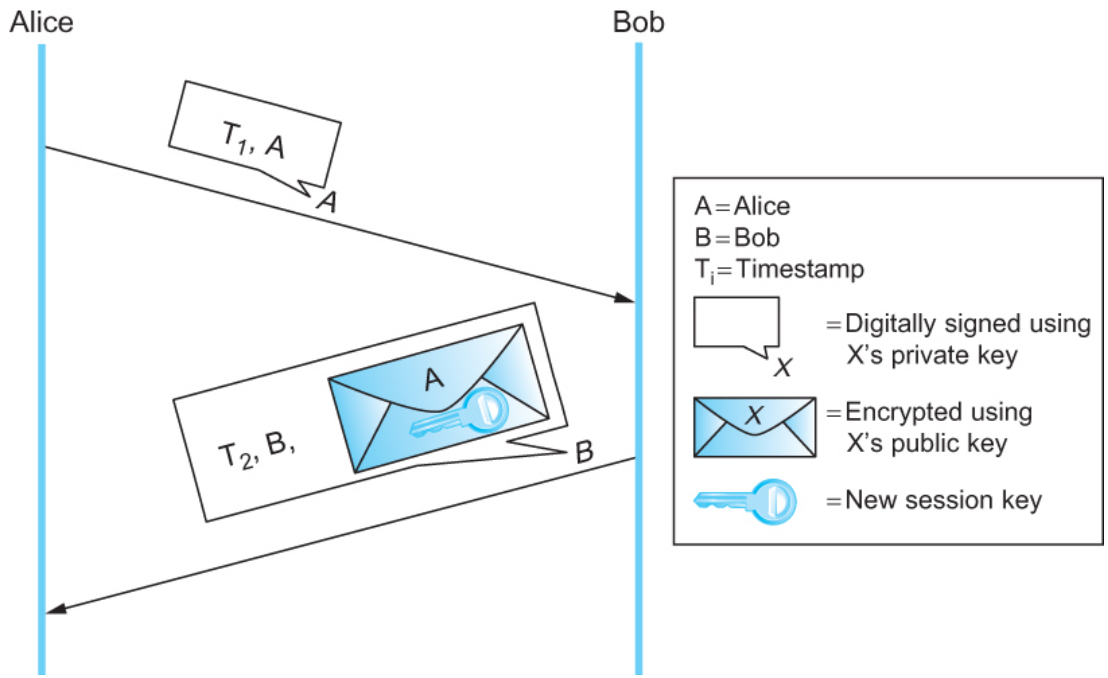
Network Authentication Protocols

- Originality and Timeliness
  - Challenge-response Protocol



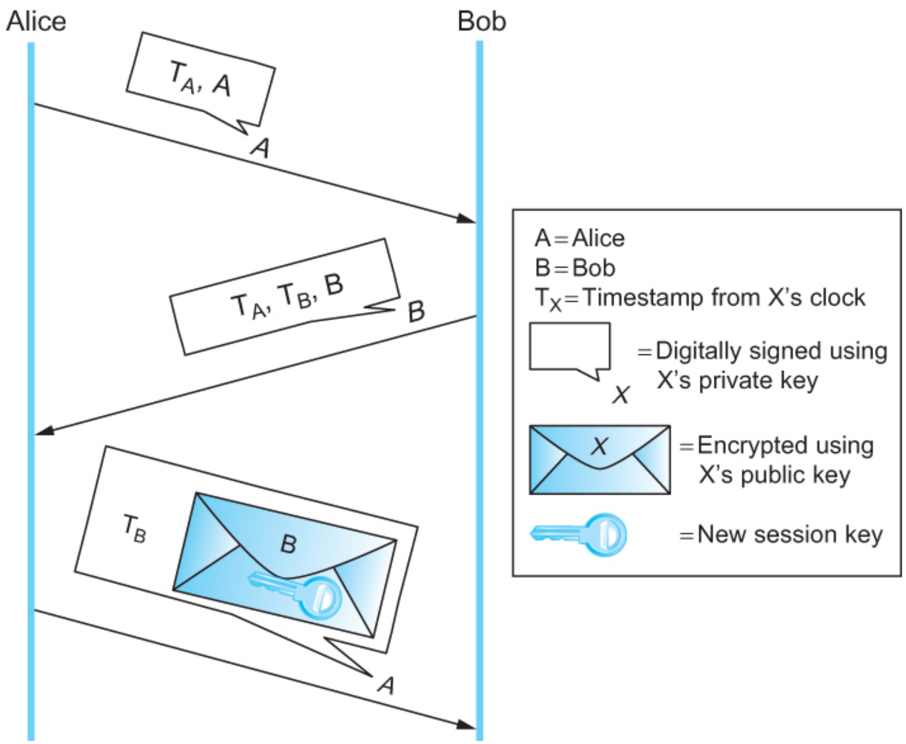
Network Authentication Protocols

- Public Key Authentication Protocols



Network Authentication Protocols

- Public Key Authentication Protocols

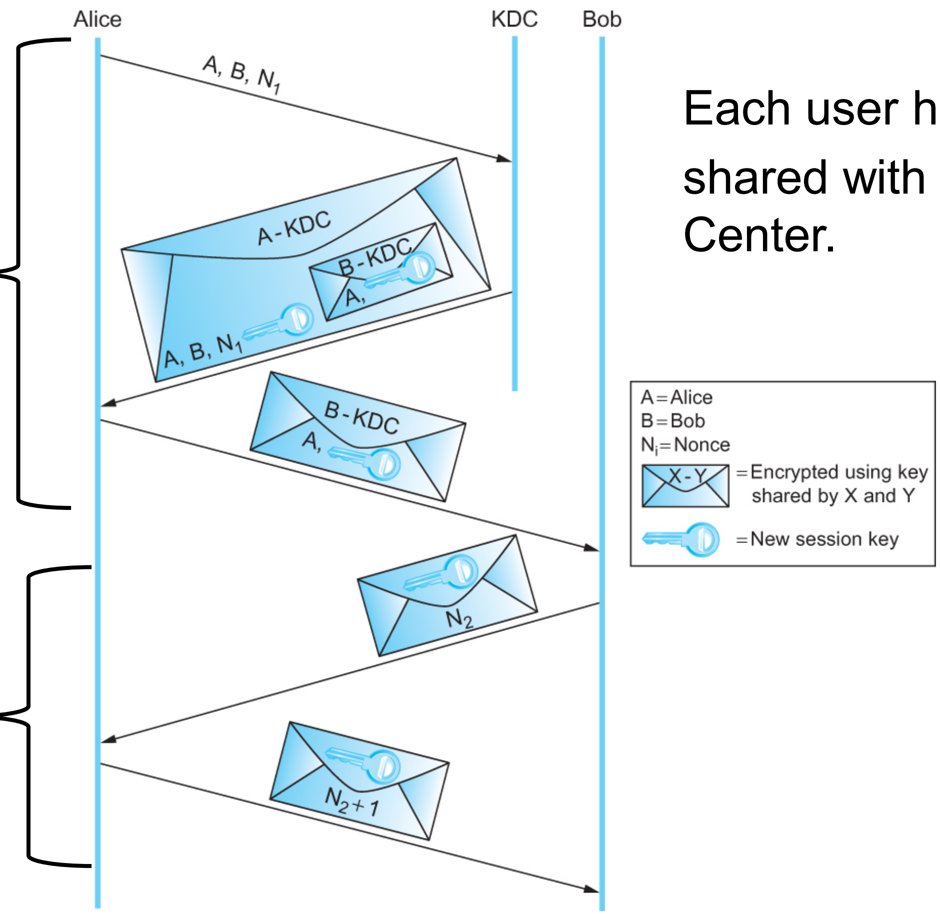


Network Authentication Protocols

- Symmetric Key Authentication Protocols

Key Distribution

Messages

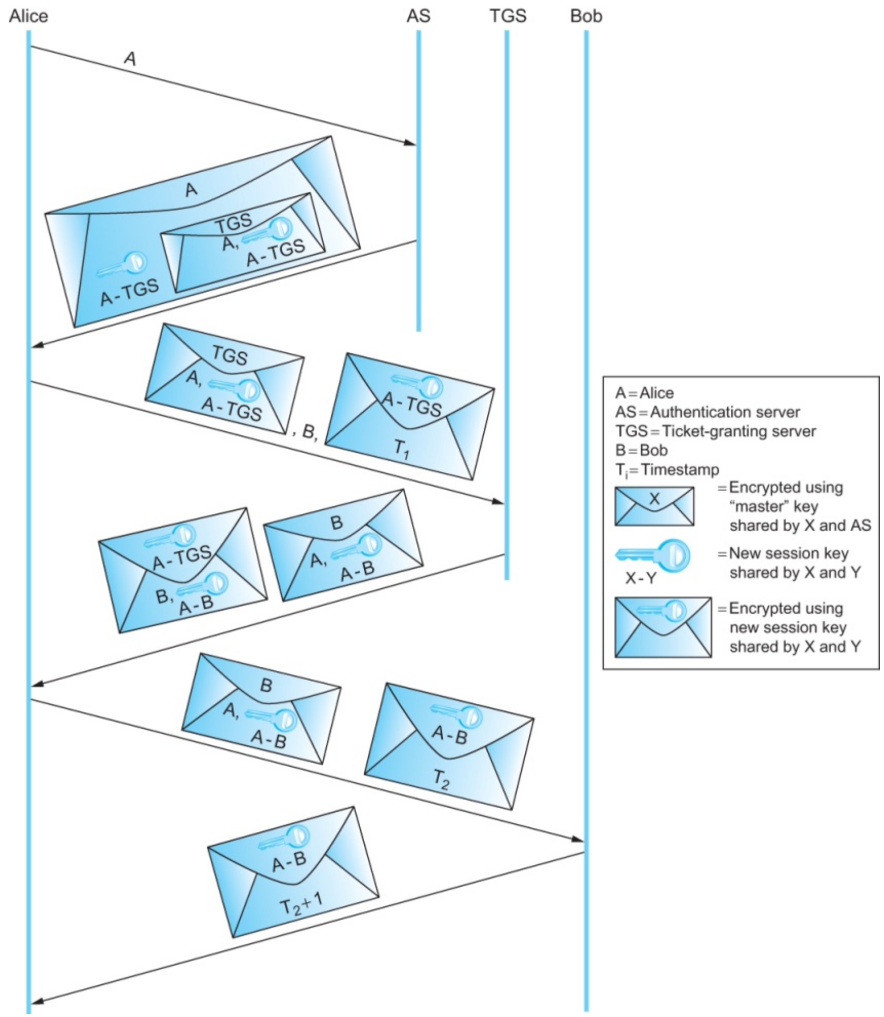


Each user has a master key shared with a Key Distribution Center.

The Needham-Schroeder authentication protocol

Authentication Protocols

- Symmetric Key Authentication Protocols - Kerberos



Summary

- General idea of cryptosystem
  - History
  - Building blocks
- Overview of a couple of crypto techniques
  - AES
  - Hash function
- One network authentication protocol
  - Kerberos