UNSW AUSTRALIA

WK03-02: Key exchange mechanisms: DH, ECDH and Kerberos

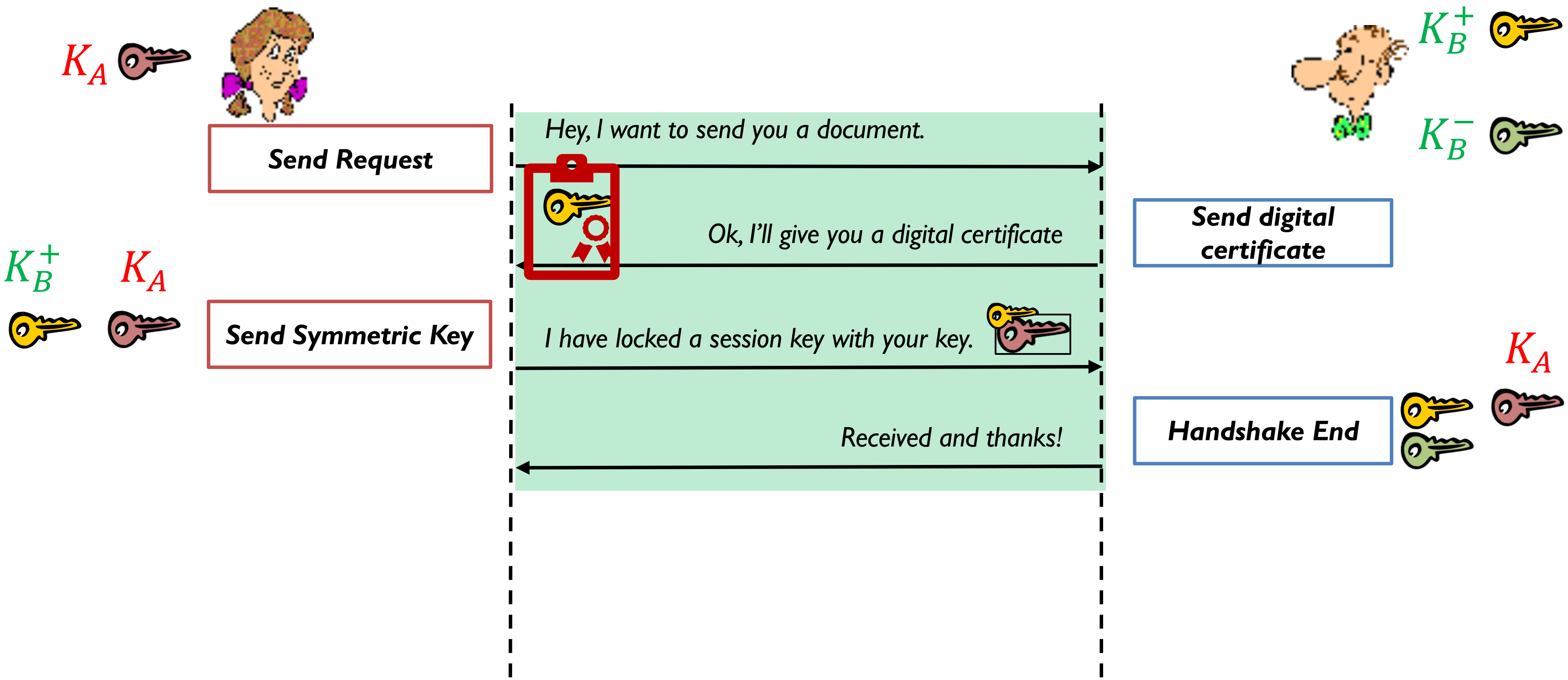
Securing Fixed and Wireless Networks, COMP4337/9337

Never Stand Still

Sanjay Jha, Nadeem Ahmed

RSA-Based Key Exchange - recap

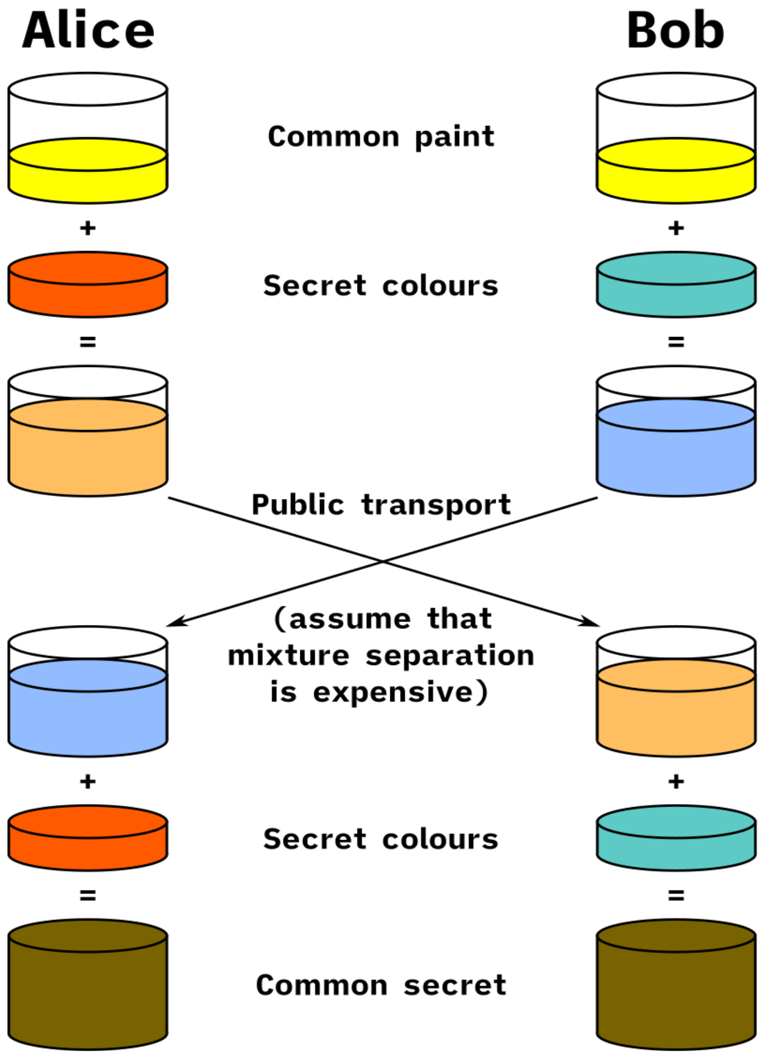
- Exchange session (symmetric) key with *asymmetric* encryption

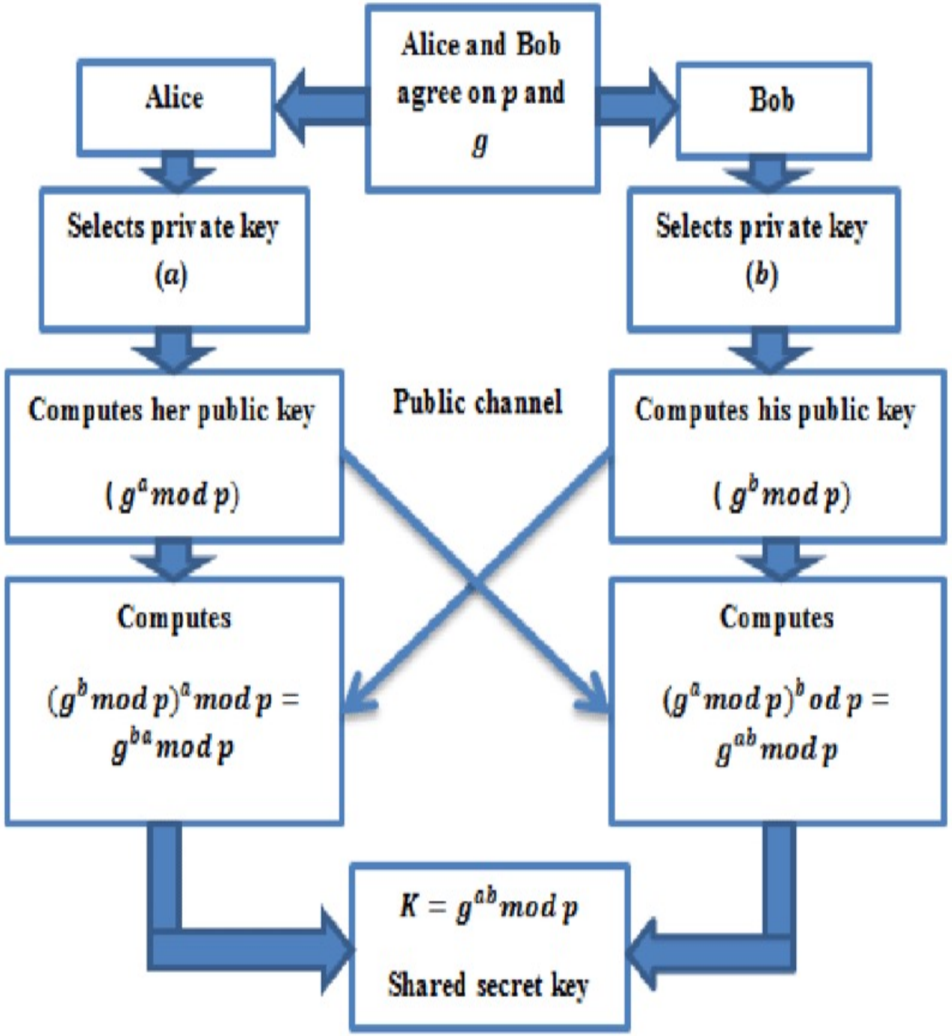


Diffie-Hellman Key Exchange

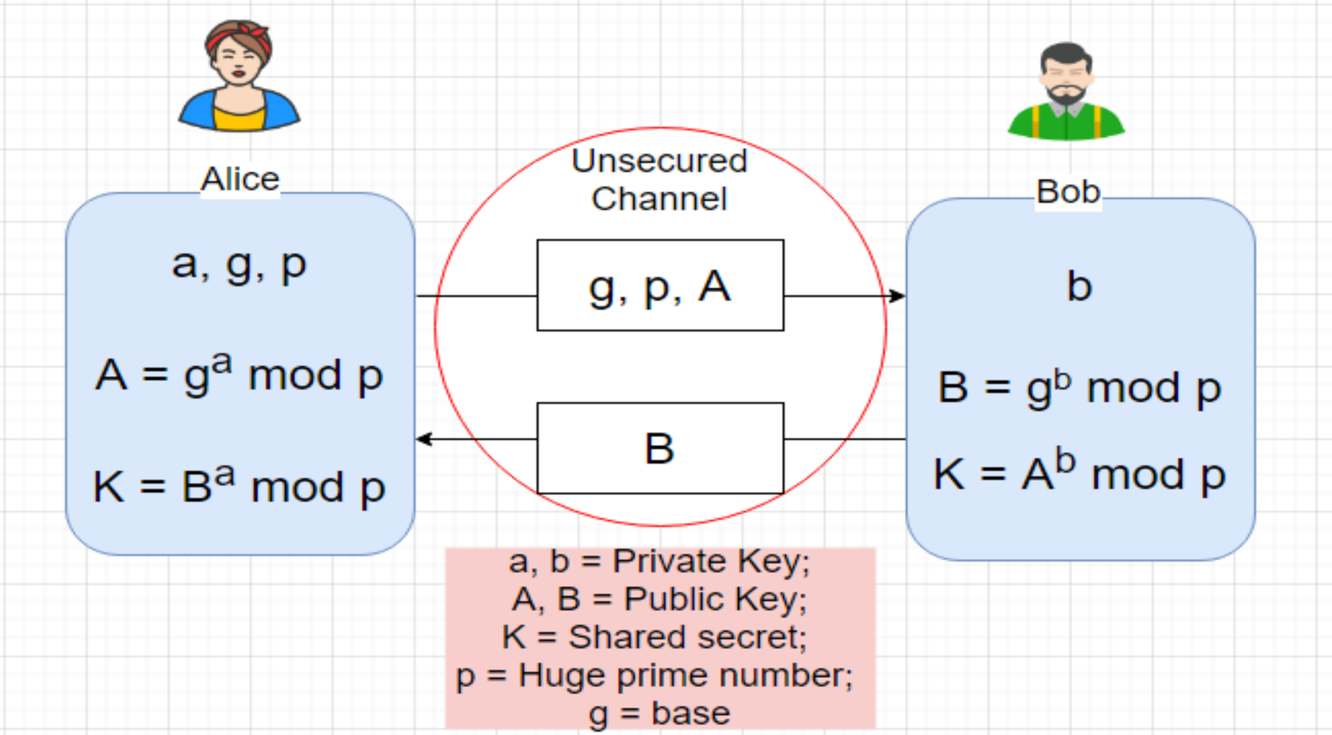
- DH uses a *private-public key pair* to establish a shared secret, typically a symmetric key.
  - The shared secret is then used for symmetric encryption or for further session/temporal key derivation
  - Keys are not exchanged but derived from common knowledge
- In RSA-based key exchange, the actual (encrypted) symmetric key is sent over the wire

Diffie-Hellman Key Exchange





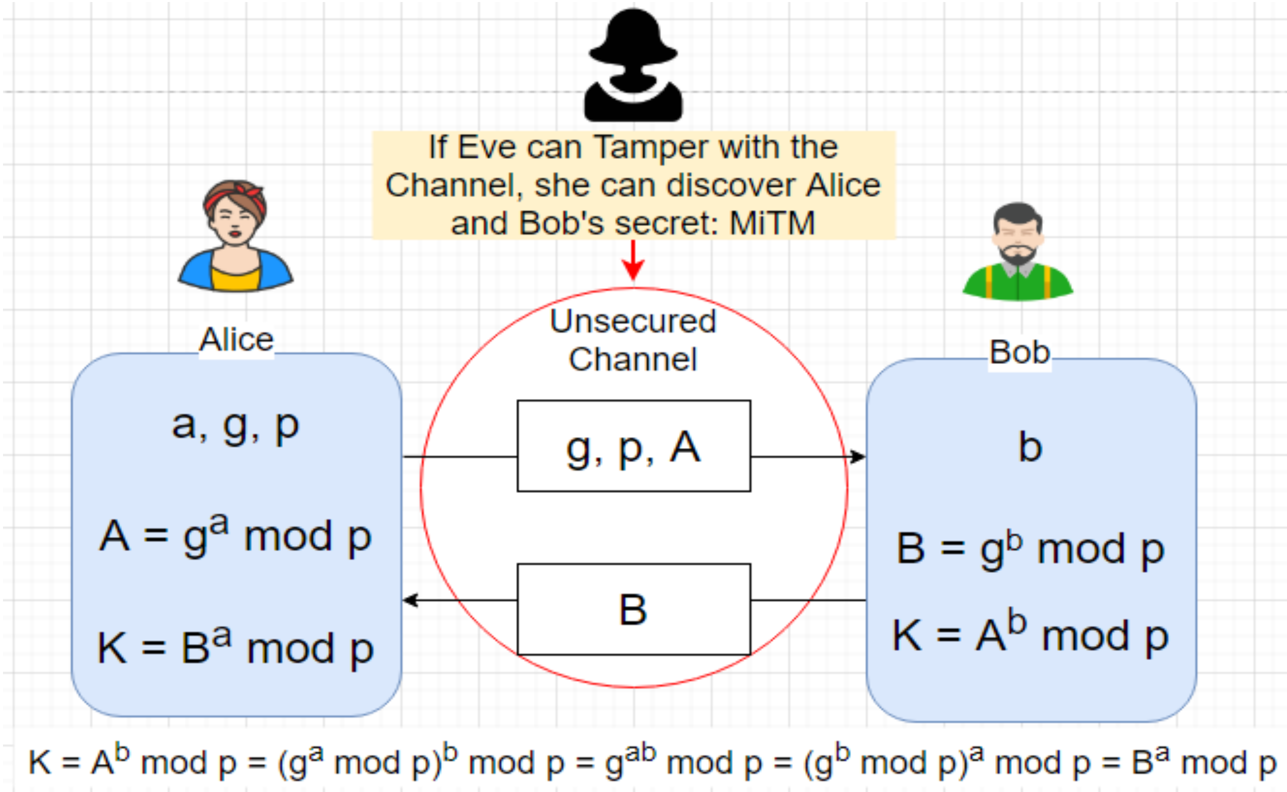
Diffie-Hellman Key Exchange



$K = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p$

- Alice's private key = 5, Bob's private key = 4, g=3, p=7
- Alice's public key = 3^5 mod 7 = 5, Bob's public key = 3^4 mod 7 = 4
- Alice's shared key = 4^5 mod 7 = 2, Bob's shared key = 5^4 mod 7 = 2

Diffie-Hellman Key Exchange - PiTM



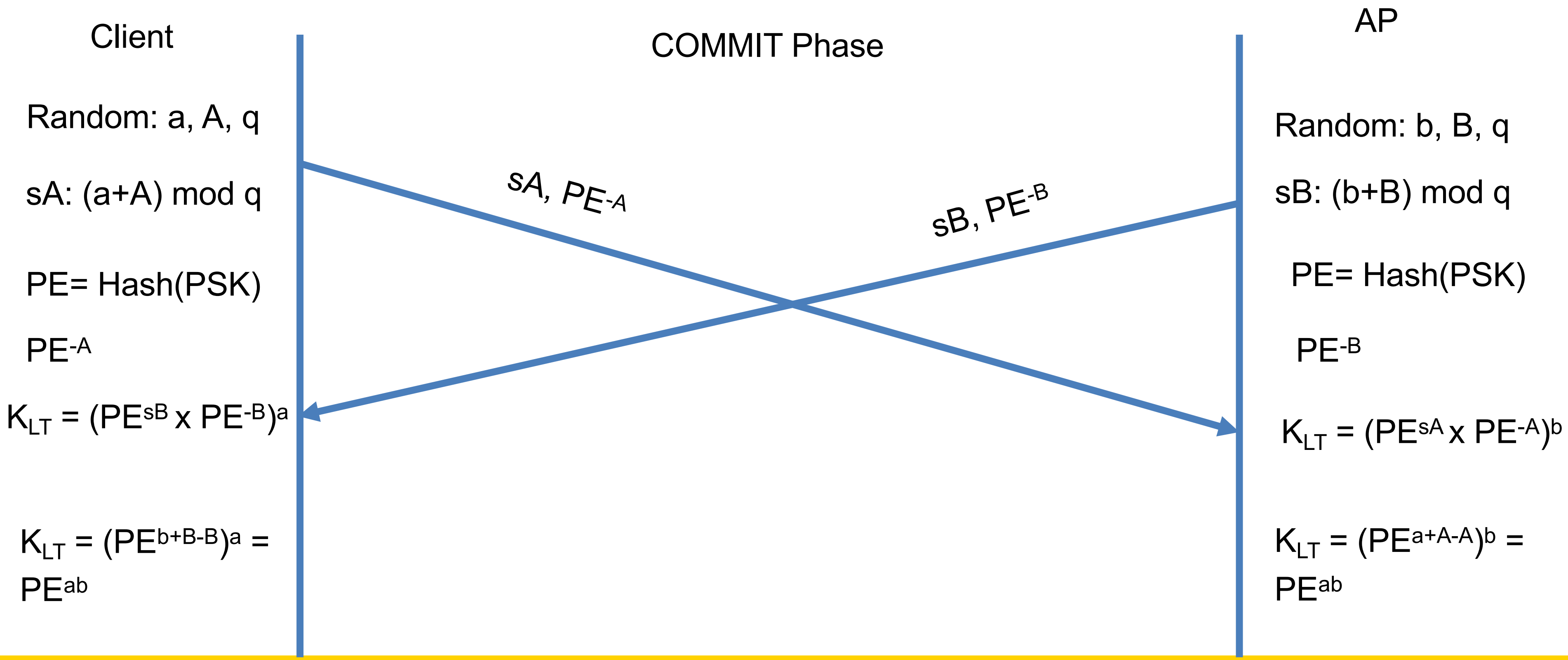
- DH is vulnerable to PiTM
- DH does not authenticate the parties; we need to provide authentication along with DH

Diffie-Hellman Key Exchange – PiTM

- Draw the protocol exchange and show how PiTM works in the previous scenario.
- Read about: Station-to-Station (STS) key exchange protocol developed by Diffie, Van Oorschot, and Wiener in 1992
  - Learn what counter measure is used to avoid the PiTM attack
- Public key changed for every connection between the two parties for forward secrecy
  - Ephemeral DH (DHE)
  - Fundamental design principle used in many real-world protocols e.g WPA-3

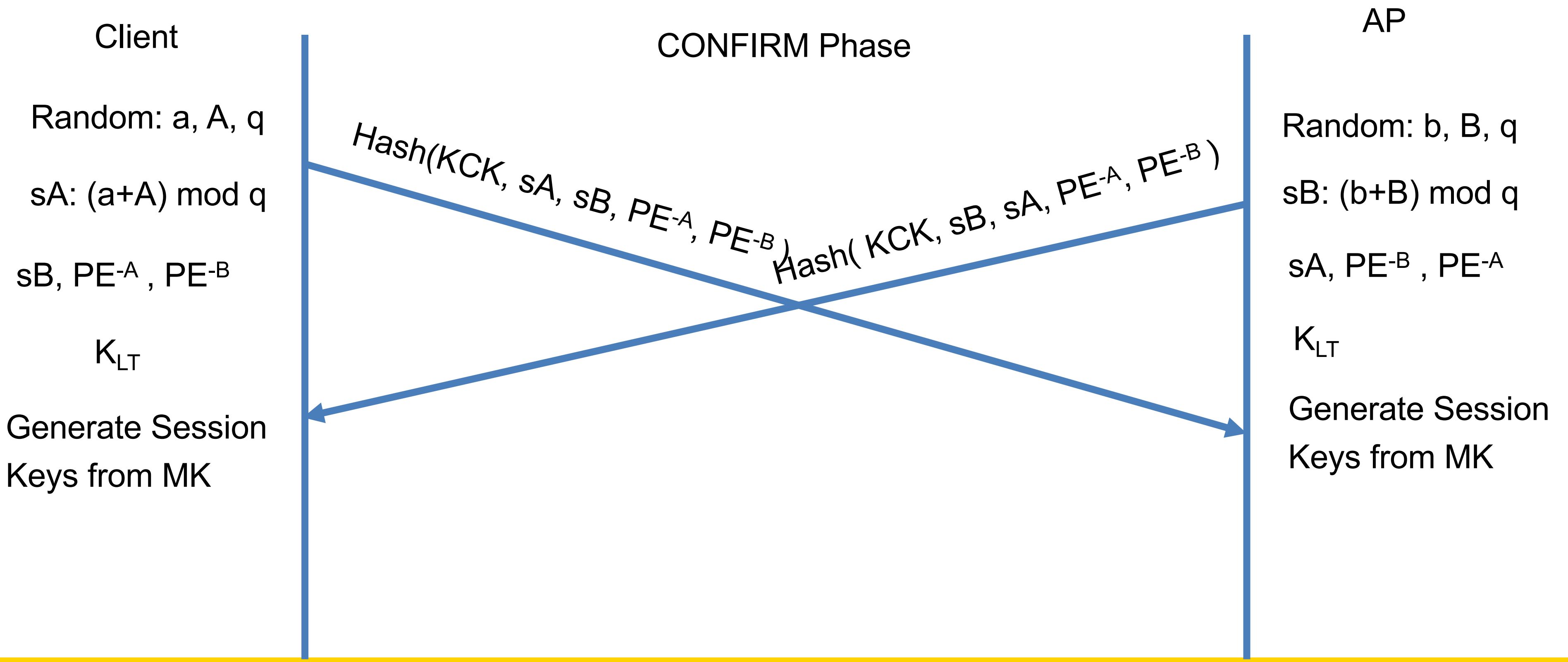
IEEE DragonFly Key Exchange SAE in WPA-3 Personal (802.11s)

- DH key exchange is unauthenticated
- SAE is based on DH but involves the PSK in deriving cryptographic keys



IEEE DragonFly Key Exchange (RFC 7664) SAE in WPA-3 Personal

- CONFIRM after the COMMIT Phase
- Generate Master Key (MK) and Key Confirmation Key (KCK) using K_{LT}



Elliptic Curve Cryptography (ECC)

- Elliptic curve cryptography is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite field.
- ECC presents various benefits over RSA such as:
  - fast computation
  - small key size
  - compact signatures
- For example, to provide equivalent security to 1024-bit RSA, an ECC scheme only needs 160 bits.

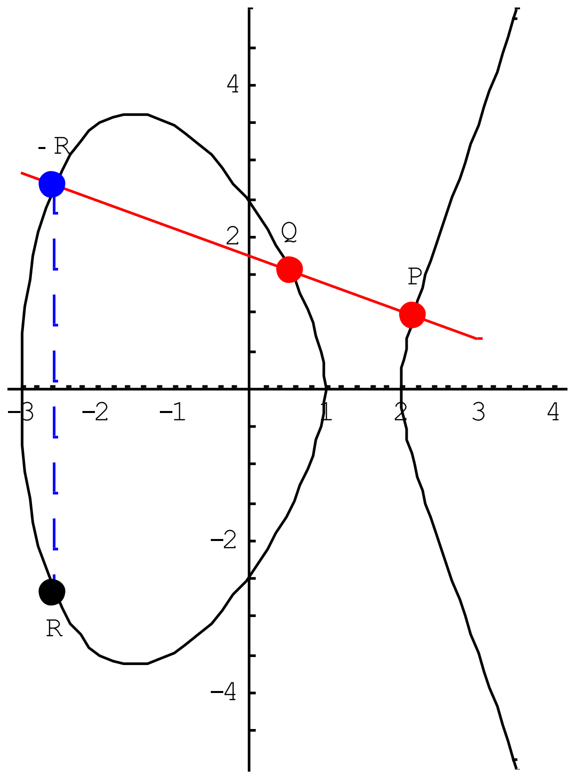
ECC Scheme

- Key Agreement through Elliptic Curve Diffie-Hellman (ECDH)
- Digital Signature: Elliptic Curve Digital Signature Algorithm (ECDSA), allows use of public/private key for signing a message and verification of signature, more efficient than RSA based DSA.

Adding Two Points (Geometrically): $xP \neq xQ$

- We skip maths/algebraic details (beyond scope)
- The line L through P and Q will intersect the curve at one other point.
- Call this third point -R.
- Reflect the point -R about the x-axis to point R.
- P+Q = R

- $y^2 = x^3-7x+6$

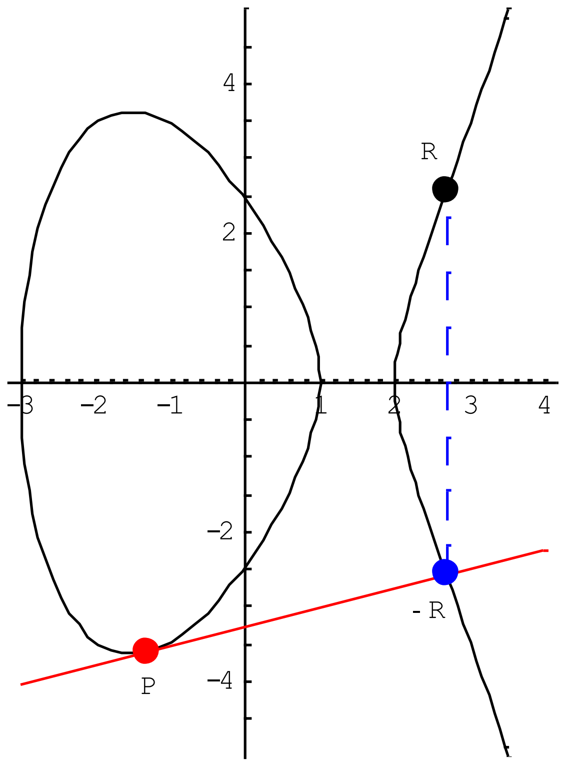


UNSW AUSTRALIA

Point Doubling: xP=xQ and yP = yQ

- Since P = Q, the line L through P and Q is tangent to the curve at P.
- Again L will intersect the curve at another point, -R.
- As in Case 1, reflect -R about the x-axis to point R.
- P+P = R
- Notation: 2P = P+P
- *Basically this computation (and variants) is more efficient than the standard Diffie-Hellman*
- *Crypto: Let P and Q be two points on an elliptic curve such that kP = Q, where k is a scalar. Given P and Q, it is hard to compute k.*

- $y^2 = x^3-7x+6$



Elliptic Curve Diffie-Hellman Key Exchange

1. Alice and Bob publicly agree on an elliptic curve E over a finite field Zp.
2. Next Alice and Bob choose a public base point B on the elliptic curve E.
3. Alice chooses a random integer $1<\alpha<|E|$, computes P = α B, and sends P to Bob. Alice keeps her choice of α secret.
4. Bob chooses a random integer $1<\beta<|E|$, computes Q = β B, and sends Q to Alice. Bob keeps his choice of β secret.

1. Alice and Bob choose E to be the curve $y^2 = x^3+x+6$.
2. Alice and Bob choose the public base point to be B=(2,4).
3. Alice chooses $\alpha = 4$, computes $P = \alpha B = 4(2,4) = (6,2)$, and sends P to Bob. Alice keeps α secret.
4. Bob chooses $\beta = 5$, computes $Q = \beta B = 5(2,4) = (1,6)$, and sends Q to Alice. Bob keeps β secret.

Maths not examinable

ECDH Key Exchange (cont.)

5. Alice computes
   $KA = \alpha Q = \alpha(\beta B)$.
6. Bob computes $KB = \beta P = \beta(\alpha B)$.
7. The shared secret key is $K = KA = KB$.

- Even if Eve knows the base point B, or P or Q, she will not be able to figure out α or β, so K remains secret!

5. Alice computes $KA = \alpha Q = 4(1,6) = (4,2)$.
6. Bob computes $KB = \beta P = 5(6,2) = (4,2)$.
7. The shared secret key is $K = (4,2)$.

Kerberos

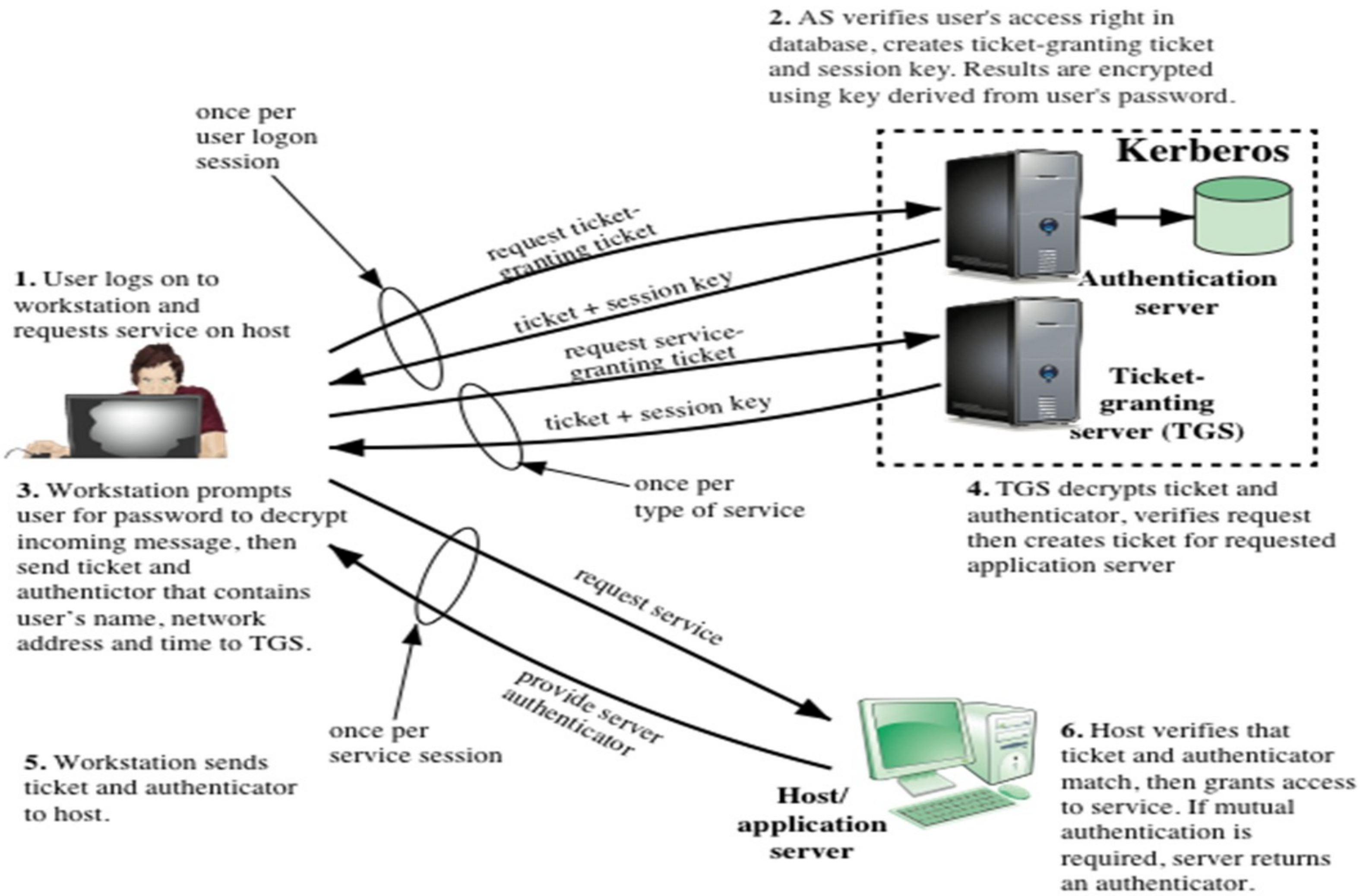
- Key distribution and user authentication service developed at MIT
- Provides a centralized authentication server whose function is to authenticate users to servers and servers to users
- Relies exclusively on symmetric encryption, making no use of public-key encryption

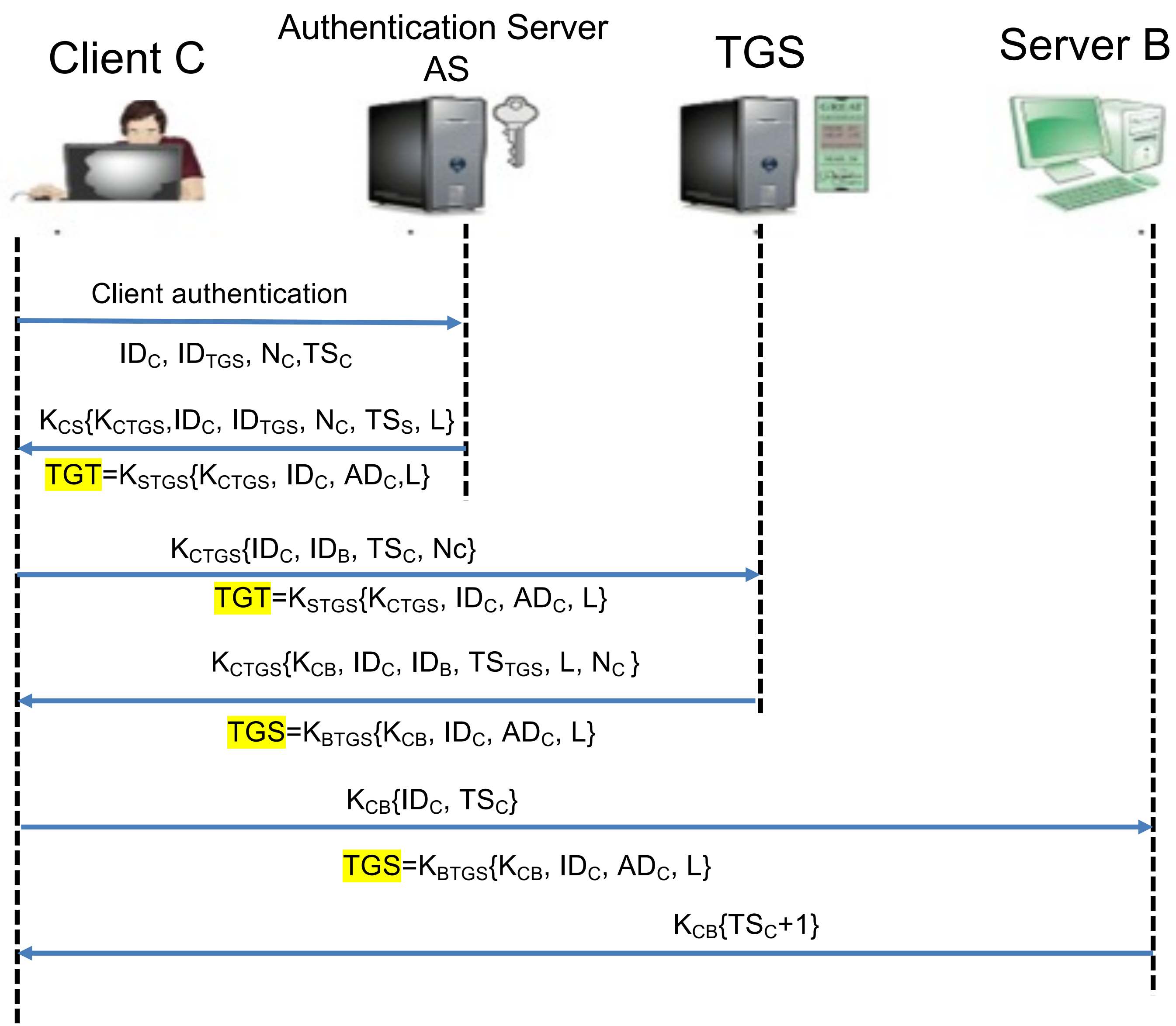
Two versions are in use

- Version 4 implementations still exist, although this version is being phased out
- Version 5 corrects some of the security deficiencies of version 4 and has been issued as a proposed Internet Standard (RFC 4120)

Kerberos Version 4

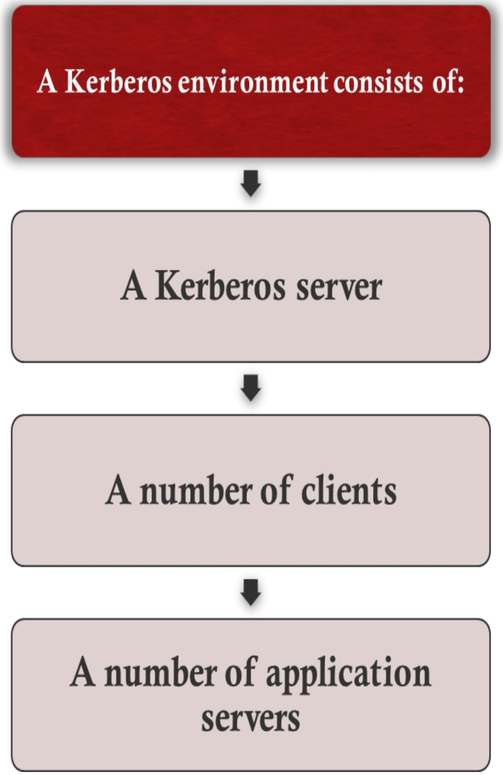
- A basic third-party authentication scheme
- Authentication Server (AS)
  - Users initially negotiate with AS to identify self
  - AS provides a non-corruptible authentication credential (ticket granting ticket TGT)
- Ticket Granting Server (TGS)
  - Users subsequently request access to other services from TGS on basis of users TGT


2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.
once per user logon session
Kerberos
request ticket-granting ticket
ticket + session key
Authentication server
1. User logs on to workstation and requests service on host
request service-granting ticket
ticket + session key
Ticket-granting server (TGS)
once per type of service
3. Workstation prompts user for password to decrypt incoming message, then send ticket and authentictor that contains user's name, network address and time to TGS.
4. TGS decrypts ticket and authenticator, verifies request then creates ticket for requested application server
request service
provide server authenticator
once per service session
5. Workstation sends ticket and authenticator to host.
Host/ application server
6. Host verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.


Client C
Authentication Server AS
TGS
Server B
Client authentication
ID_C, ID_{TGS}, N_C,TS_C
$K_{CS}\{K_{CTGS}$,ID_C, ID_{TGS}, N_C, TS_S, L}
TGT=$K_{STGS}\{K_{CTGS}$, ID_C, AD_C,L}
$K_{CTGS}\{ID_C$, ID_B, TS_C, Nc}
TGT=$K_{STGS}\{K_{CTGS}$, ID_C, AD_C, L}
$K_{CTGS}\{K_{CB}$, ID_C, ID_B, TS_{TGS}, L, N_C }
TGS=$K_{BTGS}\{K_{CB}$, ID_C, AD_C, L}
$K_{CB}\{ID_C$, $TS_C\}$
TGS=$K_{BTGS}\{K_{CB}$, ID_C, AD_C, L}
$K_{CB}\{TS_C+1\}$

Kerberos Realms

- A set of managed nodes that share the same Kerberos database
- The Kerberos database resides on the Kerberos master computer system, which should be kept in a physically secure room
- A read-only copy of the Kerberos database might also reside on other Kerberos computer systems
- All changes to the database must be made on the master computer system
- Changing or accessing the contents of a Kerberos database requires the Kerberos master password



Kerberos Version 4 vs 5 (self-read)

- Environmental shortcomings
  - Encryption system dependence
  - Internet protocol dependence
  - Message byte ordering
  - Ticket lifetime
  - Authentication forwarding
  - Inter-realm authentication

- Technical deficiencies
  - Double encryption
  - PCBC encryption
  - Session keys
  - Password attacks

Kohl, J.; Neuman, B. “The Evolution of the Kerberos Authentication Service”

Recap: How to use keys?

- Rule of thumb
  - Public key cryptography: slow
  - Symmetric key Cryptography: fast

- Do not encrypt large messages with public key cryptography
- Either use DH to arrive at fresh symmetric keys or encrypt a random, fresh symmetric key with public key cryptography
- For digital signatures, use private key to sign only the hash of the message

Acknowledgments

Lecture material covered from William Stallings, CRYPTOGRAPHY AND NETWORK SECURITY: PRINCIPLES AND PRACTICE, (Chapter 4).

Bryant, W. Designing an Authentication System: A Dialogue in Four Scenes. http://web.mit.edu/kerberos/www/dialogue.html

http://www.isi.edu/gost/info/kerberos/