# ECE 443/518 – Computer Cyber Security
# Lecture 12 Key Establishment with Symmetric Cryptography

Professor Jia Wang
Department of Electrical and Computer Engineering
Illinois Institute of Technology

September 24, 2025

# Outline

# Midterm Exam

- Lecture 1 $\sim$ Lecture 14, see Homework 1 and 2 for sample.
  - Points may be deducted if key steps are missing.
- Students registered for main campus section: Wed. 10/8, 11:25 AM – 12:40 PM, in class.
  - A physical calculator is allowed. Laptop or any other electronic device or calculator apps running on them are not allowed.
  - Closed book/notes. A letter-size page of cheat sheet is allowed.
- Online students may take the exam as above, or contact Charles Scott (scott@iit.edu) to make arrangement and confirm with me.
  - No make-up exam will be offered if you fail to do so.
- ADA Accommodations: contact Center for Disability Resource (disabilities@iit.edu)
- Emergency/extraordinary reasons for make-up midterm exams are accepted only with documented proof like docter's notes.

# Reading Assignment

- This lecture: UC 13
- Next lecture: UC 13

# Outline

# Key Establishment

- To establishing a shared secret between two or more parties.
  - Which could be used later for secure communication via symmetric cryptography.
- Key transport: one party securely transfers a secret value to others
- Key agreement: two or more parties derive the shared secret
  - Ideally, none of the parties can control what the secret will be.
- Key establishment assumes identification.
  - What about the $O(n^2)$ keys needed to support pair-wise communication among $n$ parties if we use symmetric cryptography?
  - What about the Man-in-the-Middle attack if we use public-key cryptography?

# Key Freshness

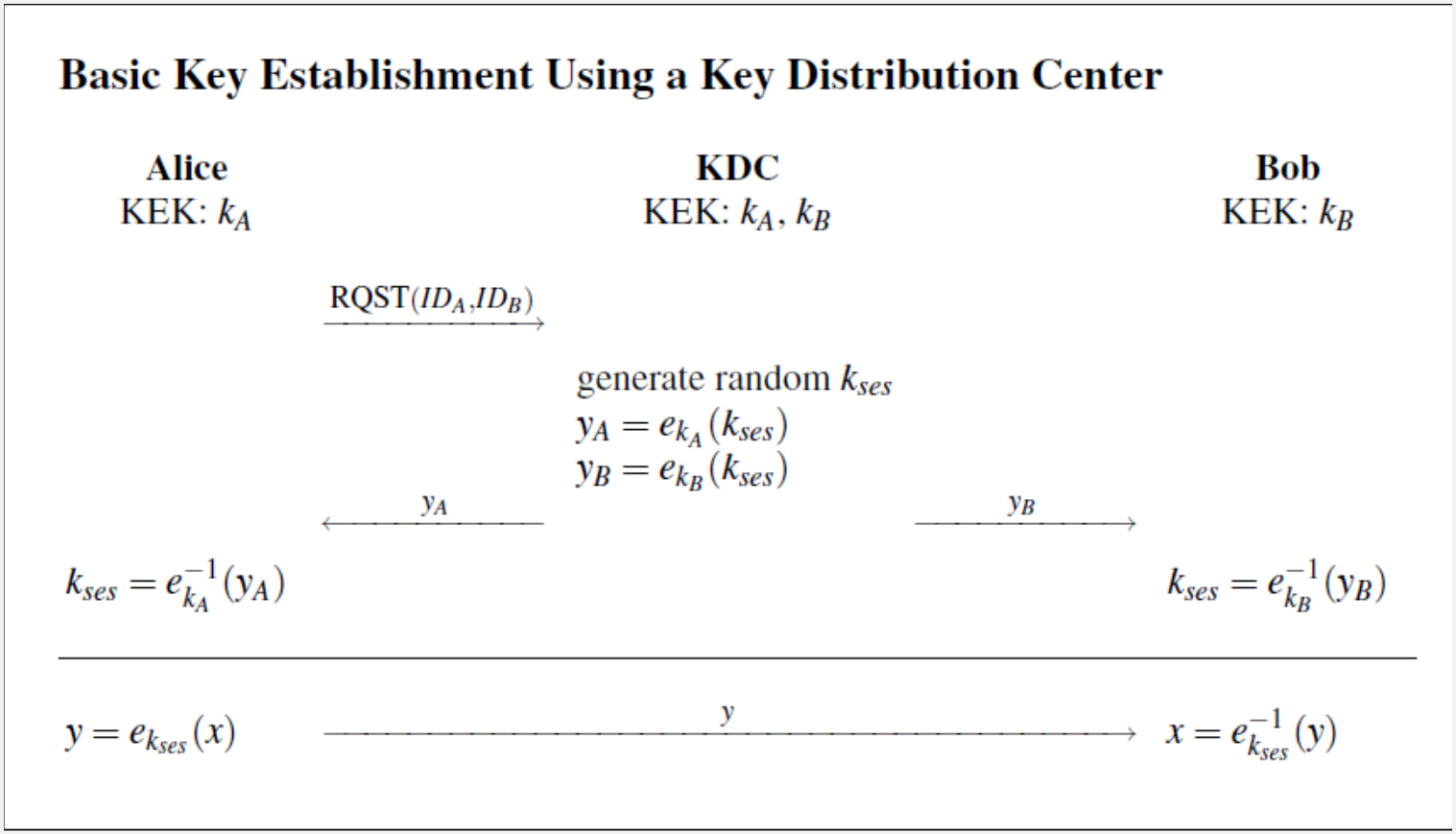
- Many security systems prefer to use one secret key only for a limited period of time.
  - Less damage if the key is exposed.
  - Less ciphertexts under the same key are available for attackers to analysis.
  - More works for attackers to decrypt same amount of ciphertexts.
- Session keys or ephemeral keys.
  - New keys are generated for each Internet connection, or within a matter of minutes, or sometimes even seconds.
- But how?
  - Need to be efficient in both computation and communication.

# Outline

# Kerberos

- Based on symmetric cryptography.
- Developed by MIT in 80's.
- Standardized as RFC 1510 in 1993, currently RFC 4120.

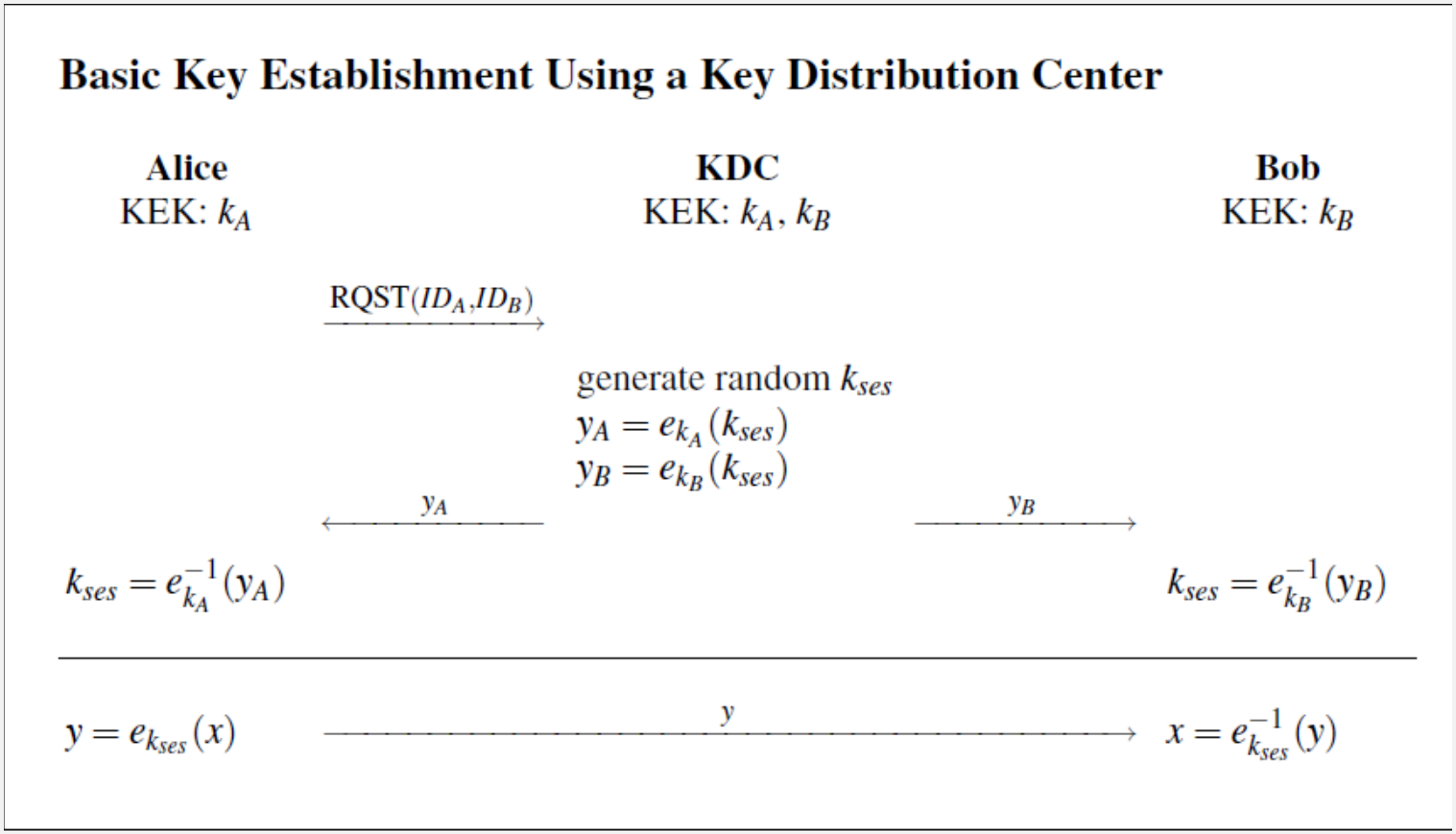
# Key Distribution Center (KDC)




(page 337, Paar and Pelzl)

- A trusted third-party.
  - Able to identify each party.
  - Won't leak secret and will follow protocol faithfully.
- KDC identifies each party with its Key Encryption Key (KEK).

# Basic Key Establishment using KDC




(page 337, Paar and Pelzl)

- Basic key establishment protocol
  - Alice requests to establish communication with Bob.
  - KDC generates $k_{ses}$ and distributes to Alice ($y_A$) and Bob ($y_B$).
  - KDC may ask Alice to distribute $y_B$ to Bob.
- Does any channel need to be secure or authentic?

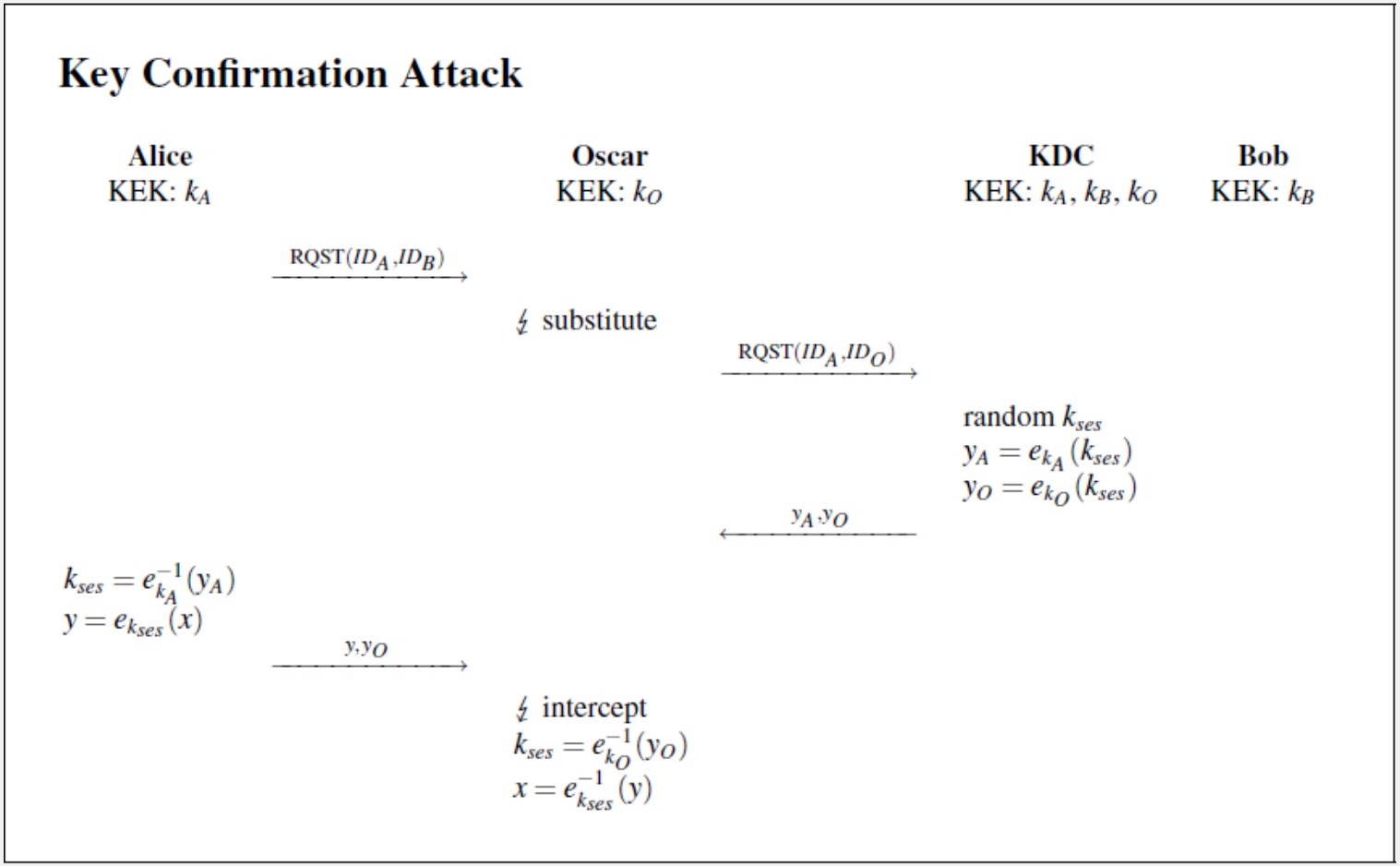
# Advantages of KDC

- System wide, only KEKs need to be stored in long term.
  - $O(1)$ storage per party.
  - $O(n)$ storage per KDC
- The $O(n^2)$ keys needed to support pair-wise communication are all generated on-the-fly and ephemeral.
- Party updates are all handled by KDC.
  - Leave
  - Join
  - KEK update

# Attacks

- Replay attack: Oscar may replace $y_A$ and $y_B$ from KDC.
  - With $y'_A$ and $y'_B$ that correspond to a previously compromised session key.
- Key confirmation attack: what is protected by KEKs?
  - The basic key establishment protocol: $k_{ses}$ only in $y_A$ and $y_B$.
  - Imply a valid session but not necessarily a session between Alice and Bob – how Bob confirms messages encrypted by $k_{ses}$ from Alice?
  - Give Oscar, a legitimate but malicious user, opportunities to attack the system.

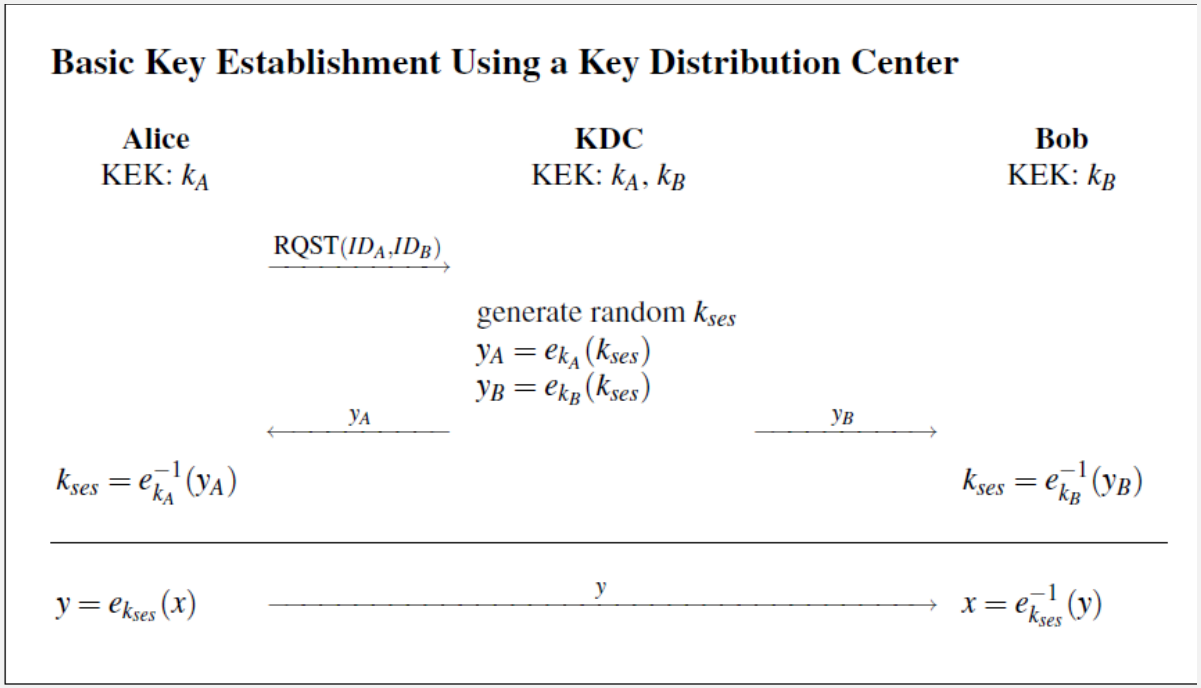
# Key Confirmation Attack




(page 339, Paar and Pelzl)

- Oscar could further establish a session with Bob and forward $x$ to Bob in order to impersonate Alice to Bob.
  - Consider this as Man-in-the-Middle attacks for symmetric cryptography when more than two parties are involved!

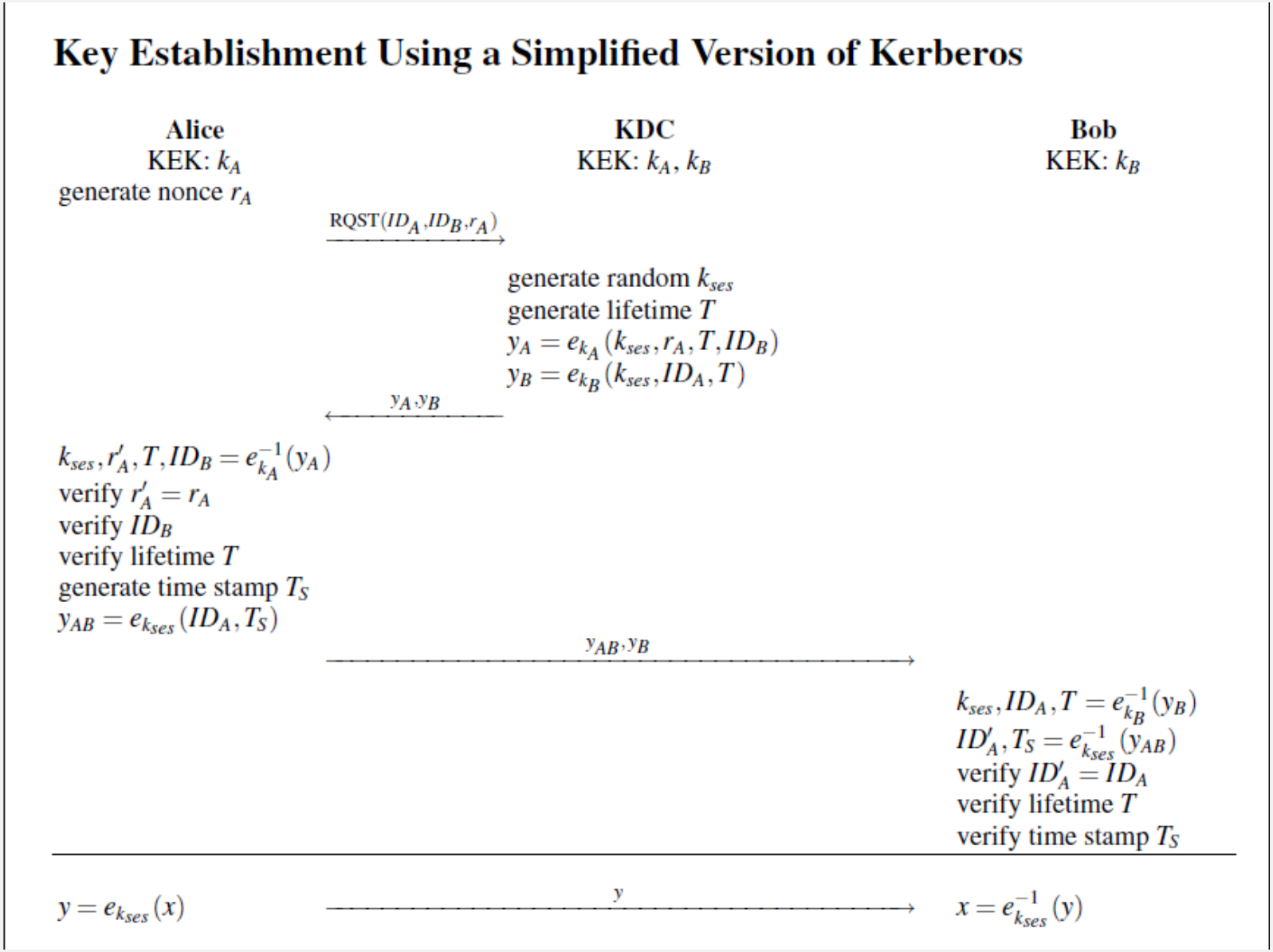
# Improving Basic Key Establishment




(page 337, Paar and Pelzl)

- Need to include session information in $y_A$ and $y_B$.
  - Challenge-response: no replay attack on Alice.
  - Participating parties: who are you talking to.
  - Time: no replay attack on Bob.

# Kerberos




(page 340, Paar and Pelzl)

# Remaining Issues

- KEK setup and update require secure channel to KDC.
  - As implied by symmetric cryptography.
- Communication requirements: KDC need to be online.
  - Performance concerns: need to response to every session.
  - Reliability concerns: no more sessions if KDC fails.
  - Should we add a secondary KDC?
- Single point of failure: security disaster.
  - A compromised KDC reveal all KEKs and thus all future communications.
  - And thus all past session keys if Oscar has recorded all sessions.
  - And this is highly possible since attackers know this weakness!
- Perfect forward secrecy (PFS): can we protect past session keys if KEKs are compromised?
  - KEKs are used to authenticate parties and to exchange session keys. Can we seperate these two purposes?

# Summary

- Kerberos: key establishment based on symmetric cryptography may work, but has a lot of potential issues.