POLITECNICO
MILANO 1863

Security Exercises

Alessandro Margara
alessandro.margara@polimi.it
https://margara.faculty.polimi.it

Exercise 1

- Consider a server for secure group communication using the centralized flat table and supporting up to 16 members
- The following members are currently connected:
  - 1, 4, 5, 8, 10, 12, 13, 15.
- Describe the state of the server and the clients
  - The current table for the server
  - The set of keys for each client
- Member 9 joins and member 5 leaves
  - Describe which keys have been revoked and which keys have been (re)generated after both operations.

Exercise 1

	bit 0	bit 1	bit 2	bit 3
0	K 0, 0	K 1, 0	K 2, 0	K 3, 0
1	K 0, 1	K 1, 1	K 2, 1	K 3, 1

+ DEK

1	(0001)	DEK + K0,0 + K1,0 + K2,0 + K3,1
4	(0100)	DEK + K0,0 + K1,1 + K2,0 + K3,0
5	(0101)	DEK + K0,0 + K1,1 + K2,0 + K3,1
8	(1000)	DEK + K0,1 + K1,0 + K2,0 + K3,0
10	(1010)	DEK + K0,1 + K1,0 + K2,1 + K3,0
12	(1100)	DEK + K0,1 + K1,1 + K2,0 + K3,0
13	(1101)	DEK + K0,1 + K1,1 + K2,0 + K3,1
15	(1111)	DEK + K0,1 + K1,1 + K2,1 + K3,1

Exercise 1

	bit 0	bit 1	bit 2	bit 3
0	K 0, 0	K 1, 0	K 2, 0	K 3, 0
1	K 0, 1	K 1, 1	K 2, 1	K 3, 1

+ DEK

- Member 9 joins

- DEK is dropped and DEK' is added
  - DEK' is sent to everyone using the old DEK
  - Member 9 receives DEK' and its own keys encrypted e.g., with its public key
  - 9 (1001) DEK' + K0,1 + K1,0 + K2,0 + K3,1
  - This does not ensure backward secrecy if member 9 intercepted the previous delivery of DEK!
    - To ensure backward secrecy we need to create new K0,1 + K1,0 + K2,0 + K3,1

Exercise 1

	bit 0	bit 1	bit 2	bit 3
0	**K 0, 0**	K 1, 0	**K 2, 0**	K 3, 0
1	K 0, 1	**K 1, 1**	K 2, 1	**K 3, 1**

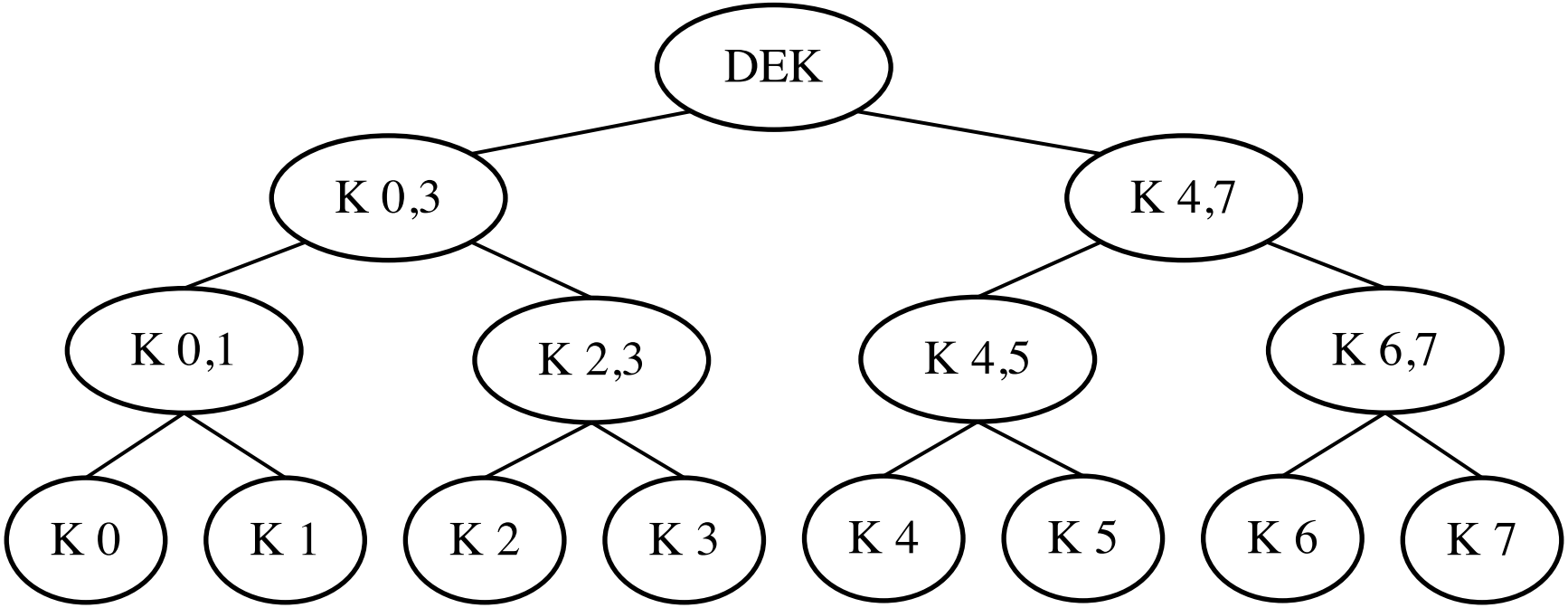
+ DEK

- Member 5 leaves
- DEK" is encrypted with the 4 remaining keys
  - It is sent to everyone, and all the remaining members will have at least one valid key to open the message
- The new KEKs will be encrypted twice and sent to all members
  - E.g., K 0,0" is encrypted as DEK"[K 0,0[K 0,0"]])

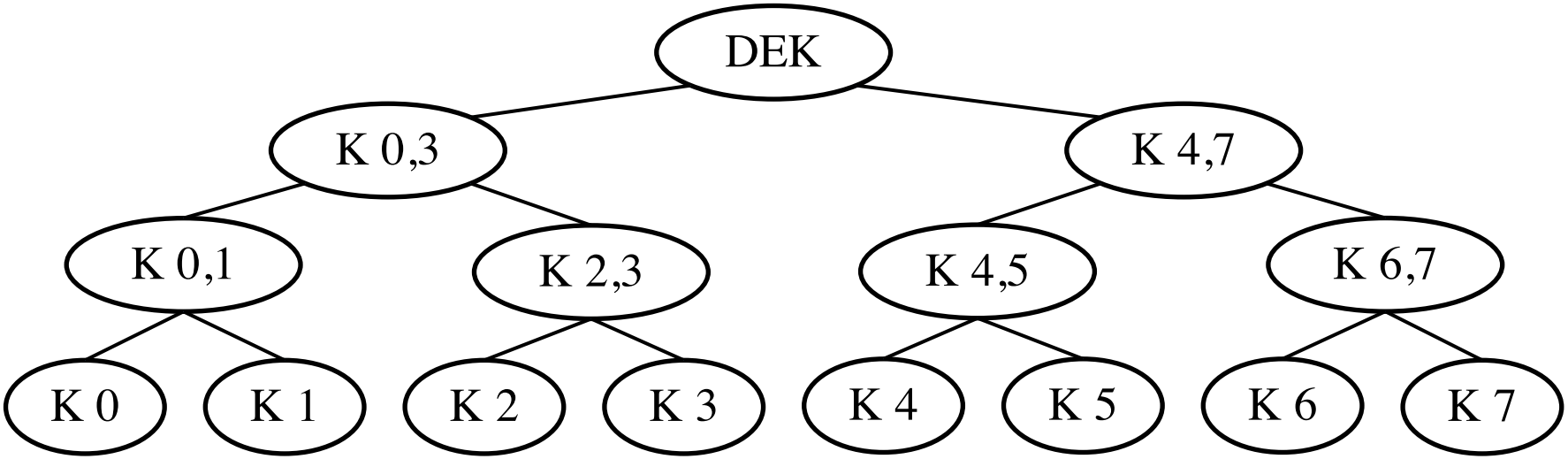
Exercise 2

- Consider a server for secure group communication using a logical key hierarchy (tree).
- The following members are currently connected:
  - 0, 1, 4, 5, 6
- Describe the state of the server and the clients
  - The keys of the server
  - The set of keys for each client
- Member 2 joins and member 5 leaves
  - Describe which keys have been revoked and which keys have been (re)generated after both operations.

Exercise 2

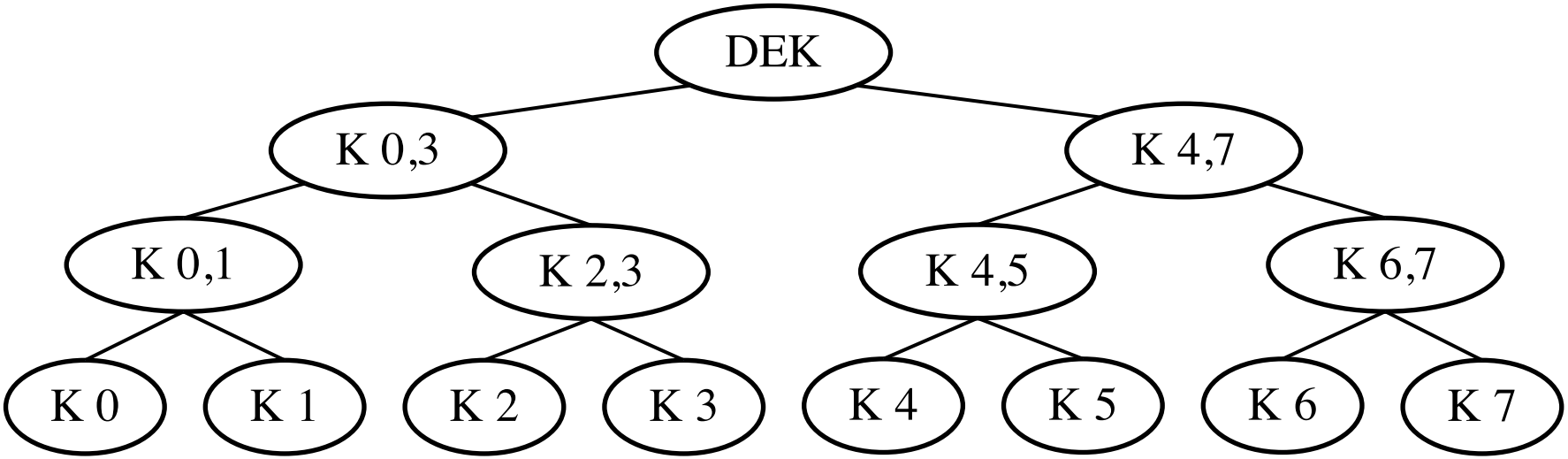


Exercise 2



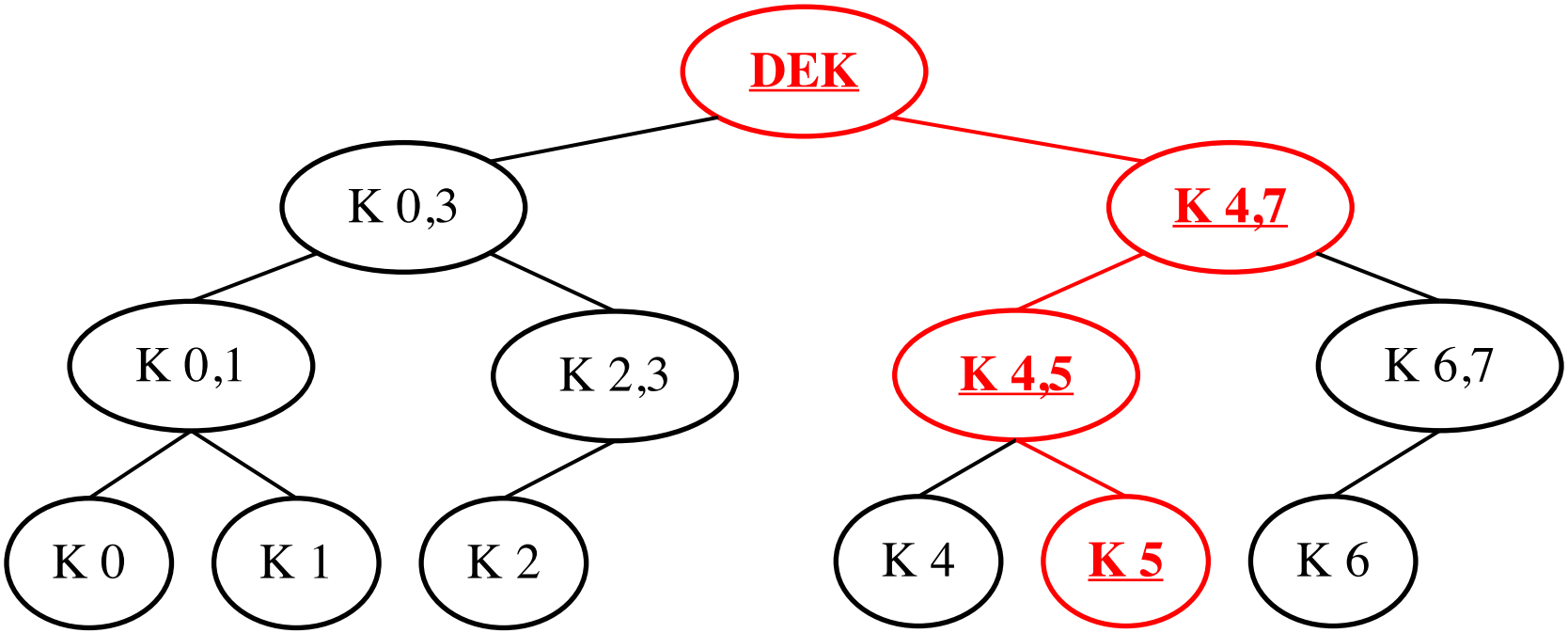
- Connected members: 0, 1, 4, 5, 6
  - Server has all the keys in the picture
  - Node 0 has K 0 / K 0,1 / K 0,3 / DEK
  - Node 1 has K 1 / K 0,1 / K 0,3 / DEK
  - Node 4 has K 4 / K 4,5 / K 4,7 / DEK
  - Node 5 has K 5 / K 4,5 / K 4,7 / DEK
  - Node 6 has K 6 / K 6,7 / K 4,7 / DEK

Exercise 2



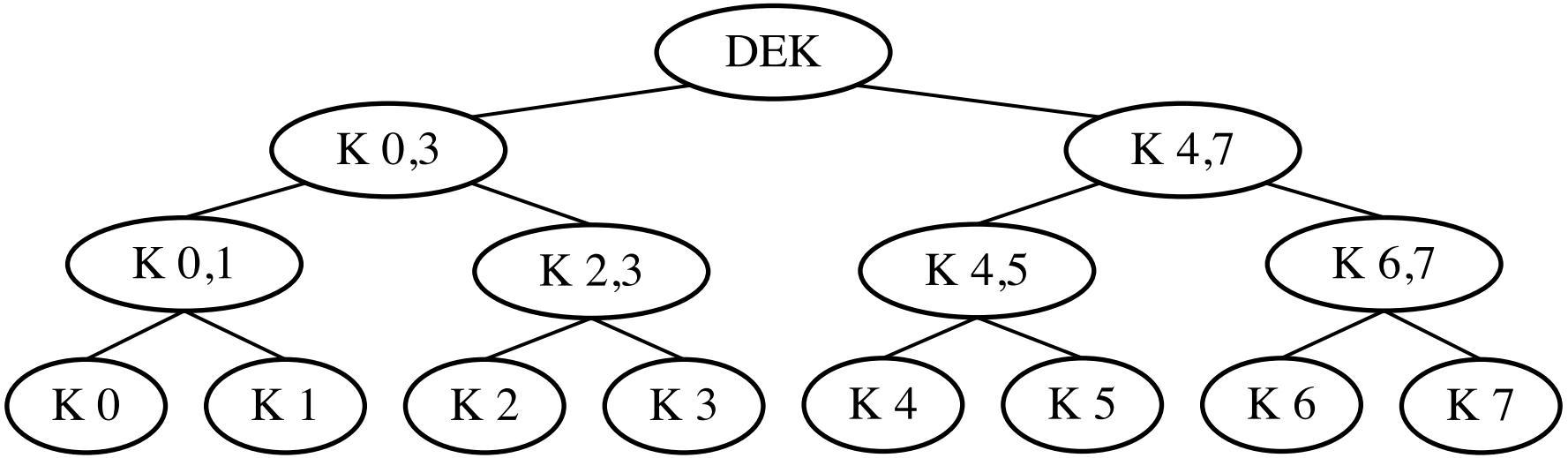
- Member 2 joins. It gets:
  - K 2 (this can be the public key that Node 2 already owns)
  - K 2 (K 2,3)
  - K 2,3 (K 0,3)
  - K 0,3 (DEK')
- Note: this does not ensure backward secrecy!
  - If we want to also ensure backward secrecy we need to invalidate all the keys from K2 to the root, not only DEK, similar to the case of a centralized flat table

Exercise 2



- Member 5 leaves
- The following keys are invalidated
  - K 5 / K 4,5 / K 4,7 / DEK

Exercise 2



- New distribution of DEK and KEK as follows
  - K 4 (K' 4,5)
  - K' 4,5 (K' 4,7)
  - K 6,7 (K' 4,7)
  - K 0,3 (DEK')
  - K' 4,7 (DEK')