UNIVERSITY of WASHINGTON

EE 418 Project: Attacks on RFID Mutual Authentication

Autumn 2025

Dept. of Electrical and Computer Engineering
University of Washington

DUE ON Dec 7th 11:59 pm.

Project Guidelines

- Due on **Canvas** at **11:59pm, Dec 7th (Sunday), 2025**
- Max. group size is **3**.
- Submit both **project report** and **source code:**
  - Python.
  - Provide in-line comments to help understand your code, and make sure your code is ready to run.
- On the front page of your project report, provide
  - Names and student IDs of group members.
  - Clear description of each member's contribution.
- Note: You are recommended to get started ASAP.

Outline

- Introduction to RFID Systems
- Cryptography in RFID – Requirements and limitations
- The MMAP and EMAP Protocols
  - Design
  - Cryptanalysis
  - Your assignment
- Summary

What is RFID?

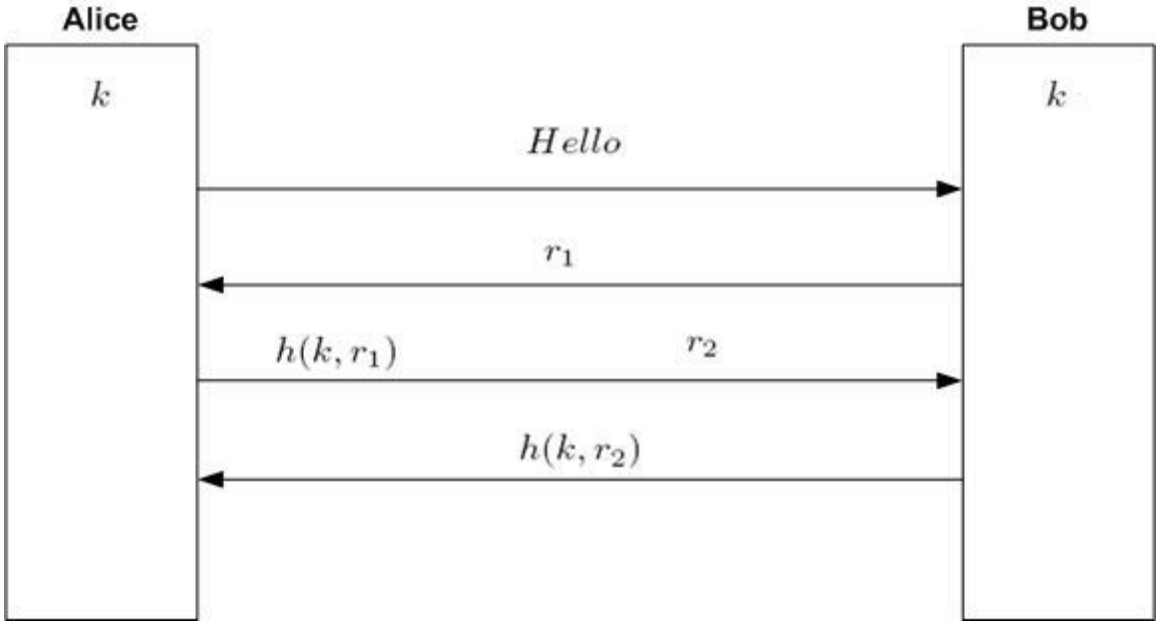
- RFID stands for **R**adio **F**requency **Id**entification
- Can identify objects via radio-frequency communication
- Consists of:
  - **Tags**: Integrated circuit with antenna, attached to an object
  - **Reader**: Handheld device to query the tag

RFID Security Issues

- Mutual Authentication
  - Tag and reader must be able to identify and verify one another
  - Possible concerns: vandalism, theft
- Privacy
  - The tag's secret information should not be leaked
  - The tag should not reveal information about the owner
  - Concerns: privacy in individuals, espionage in corporations/governments
- Cryptography is a vital part of many solutions!

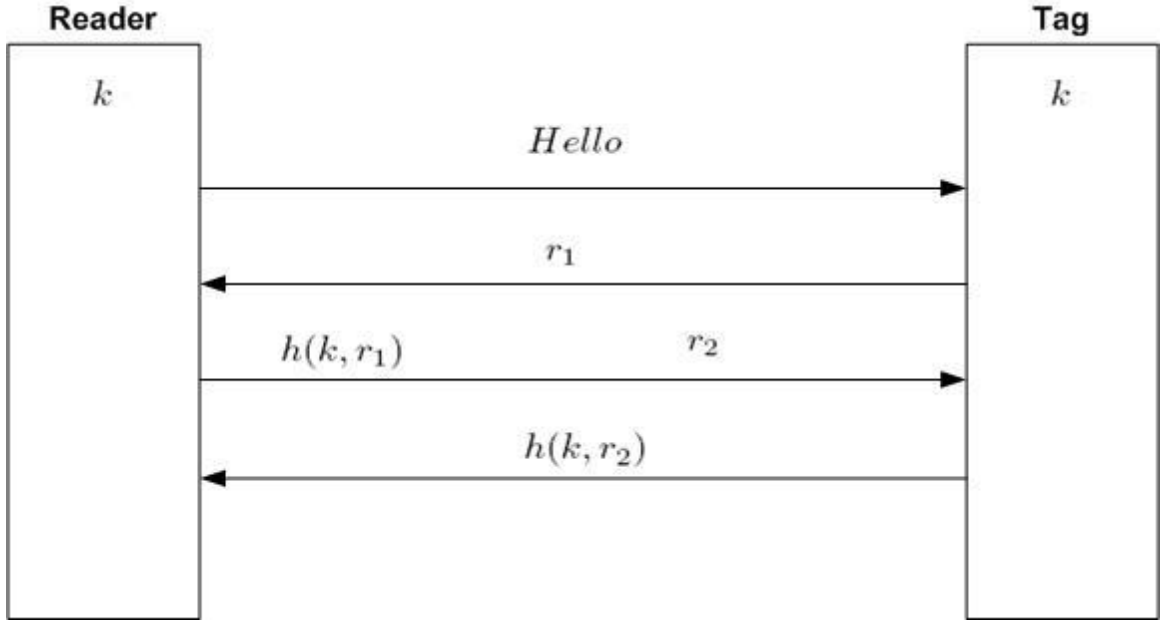
Background: Mutual Authentication

- Standard idea: **"Challenge-response protocol"**
- In order to answer Alice's challenge correctly, Bob must know k (and vice versa)
- The hash function protects k from being leaked



Cryptography in RFID Tags

- There are some problems with the challenge-response protocol in RFID.
  - The tag might be unable to generate random numbers
  - If there are many tags, each with a unique key, it will be difficult for the reader to answer the tag's query.
  - If the tag answers first, then privacy concerns – and search still takes a long time



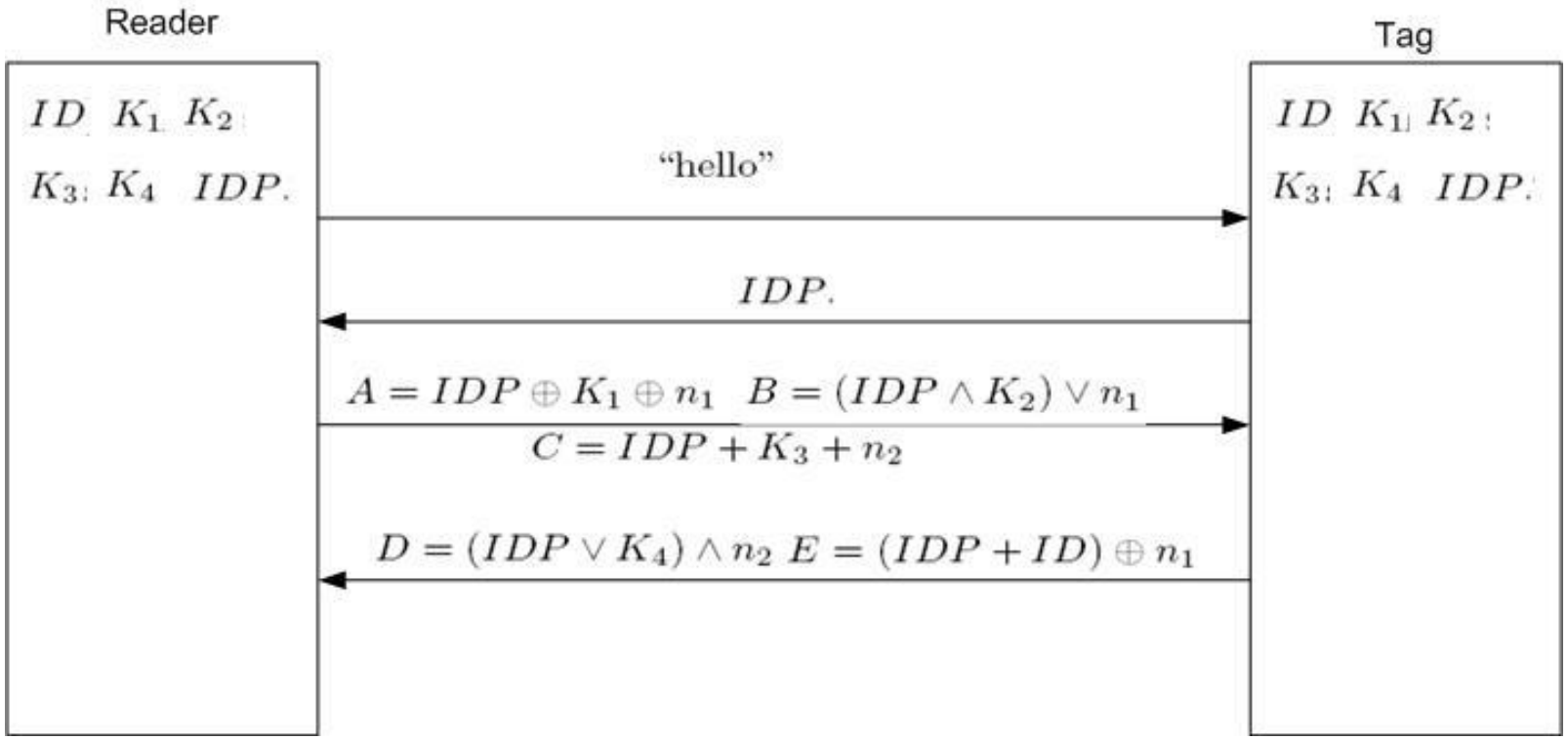
Recall: Bitwise Operations

- Bitwise operations take in two bits and output a single bit
- AND: 1 if *both* bits are 1, 0 otherwise
- OR: 1 if *either* bit is 1, 0 if both 0
- XOR: 1 if *one* bit is 1, 0 if both 0 or both 1
- Corollary: b XOR b = 0, b OR 0 = b, b AND 1 = b, b XOR 0 = b
- e.g., s1 = 011100, s2 = 100110
  - s1 AND s2 = 000100
  - s1 OR s2 = 111110
  - s1 XOR s2 = 111010

The MMAP Protocol

- Designed for identification and mutual authentication if one participant is very limited
- The tag has a secret, unique ID, a non-secret pseudonym IDP for identification, and four secret keys K1, K2, K3, and K4
- Challenge-response is done using bitwise operations only
- Tag and reader update tag's information (except ID) after the protocol is complete

The MMAP Protocol



Passive Attacks Against the MMAP Protocol

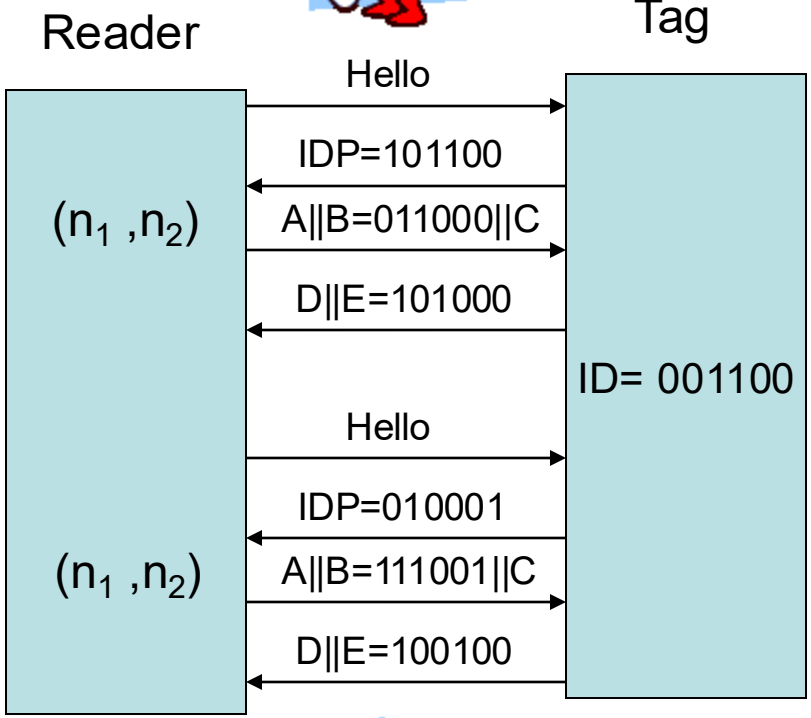
B = (IDP and K_2) or n_1
011000 = (101100 and K_2) or n_1
011000 = *0**00 or n_1
n_1 = *1**00
E xor n_1 = *1**00
ID = (E xor n_1) – IDP = *1**00 + 010100
ID = ****00

B = (IDP and K_2) or n_1
111001 = (010001 and K_2) or n_1
111001 = 0*000* or n_1
n_1 = 1*100*
E xor n_1 = 0*110*
ID = (E xor n_1) – IDP = 0*110* + 101111

ID = ****00 = 0*110* + 101111
ID = ****00 = 0*1101 + 101111
ID = **1100 = 0*1101 + 101111

ID = **1100 = *1**00 + 010100
ID = **1100 = *11000 + 010100
ID = *01100 = *11000 + 010100

ID = *01100 = 0*1101 + 101111
ID = *01100 = 011101 + 101111
ID = 001100 = 011101 + 101111



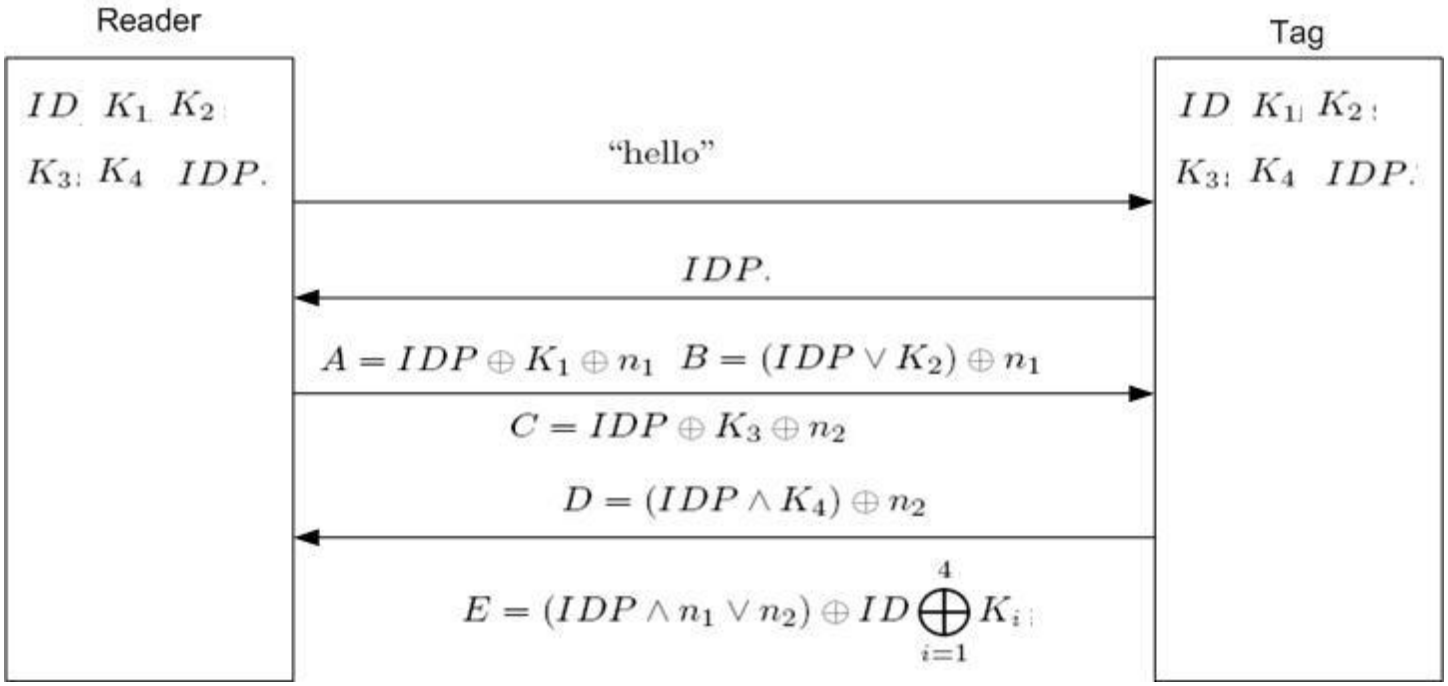
ID = 001100

The EMAP Protocol

- Very similar to MMAP – challenge-response using bitwise operations
- This time, the equations used to update the keys and IDP are important:

$$
\begin{aligned}
IDP^{(n+1)} &= IDP^{(n)} \oplus n_2 \oplus K_1, \\
K_1^{(n+1)} &= K_1^{(n)} \oplus n_2 \oplus (ID(1:48) \parallel F_p(K_4^{(n)}) \parallel F_p(K_3^{(n)})), \\
K_2^{(n+1)} &= K_2^{(n)} \oplus n_2 \oplus (F_p(K_1^{(n)}) \parallel F_p(K_4^{(n)}) \parallel ID(49:96)), \\
K_3^{(n+1)} &= K_3^{(n)} \oplus n_1 \oplus (ID(1:48) \parallel F_p(K_4^{(n)}) \parallel F_p(K_2^{(n)})), \\
K_4^{(n+1)} &= K_4^{(n)} \oplus n_1 \oplus (F_p(K_3^{(n)}) \parallel F_p(K_1^{(n)}) \parallel ID(49:96)),
\end{aligned}
$$

The EMAP Protocol



An Attack on EMAP

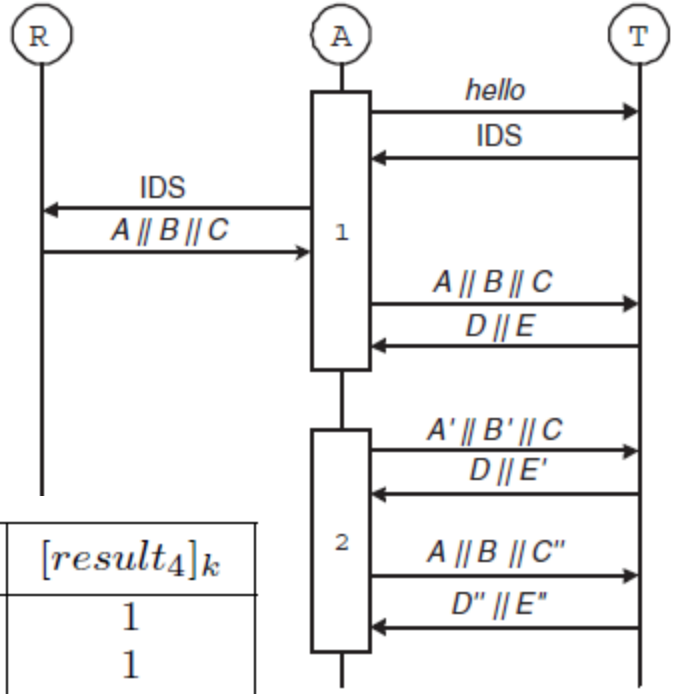
- The attack has three stages:
  - Compute half of the bits of the random numbers n1 and n2
  - Compute the other half of the bits by sending messages to the tag and observing response
  - Compute ID using the tag's output

Computing n1 and n2 – Part I

- The adversary can compute half the bits of n1 and n2 through passive monitoring alone
- Since $D = (IDP \wedge K_4) \oplus n_2$, the i-th bit of n2 can be recovered when the i-th bit of IDP is 0
- A similar approach yields half the bits of n1

Computing n1 and n2 – Part II

- The adversary computes the rest of n1 and n2 by interacting with the tag
- The adversary flips all bits of A and B that satisfy IDP = 1
- The adversary then observes the change in the tag's reply



$[IDS^{(n)}_{tag(i)}]_k$	$[n1]_k$	$[n1']_k$	$[n2]_k$	$[result_3]_k$	$[result_4]_k$
1	0	1	0	0	1
1	0	1	1	1	1
1	1	0	0	1	0
1	1	0	1	1	1

Computing ID

- The update rule $IDP^{(n+1)} = IDP^{(n)} \oplus n_2 \oplus K_1$ implies that the adversary can recover K1 using knowledge of n2
- K1 can then be used to compute the most significant bits of ID
- The remaining bits can be computed by observing additional protocol runs

Your Assignment

- Implement the Python classes MMAPoracle and EMAPoracle to simulate MMAP and EMAP protocols
- Implement Python functions to simulate MMAP attack and EMAP attack
- Answer the four questions

Summary

- RFID systems are a topic of current research interest
- In particular, security and privacy are great concerns
- MMAP and EMAP, while appearing to fulfill the RFID security requirements, are vulnerable to attack
- Your task is to simulate and analyze these attacks

Questions?


WHAT?
WHERE?
WHEN?
WHY?
HOW?
WHO?
WHICH?
WHOSE?
WHERE? WHICH? WHOSE? WHEN? WHY?
WHO? WHERE? WHAT? HOW?
WHAT? HOW? WHY?
WHY? HOW? WHERE?
WHO? WHOSE?
WHERE? WHAT? HOW?
WHEN? WHERE?
WHY? HOW?
HOW? WHO?
WHO? WHERE? WHAT? HOW?
WHAT? HOW? WHY?