Bit-Flipping Attack Exploration and Countermeasure in 5G Network

Joon Kim
UC Berkeley
EECS

Chengwei Duan
University of Florida
ECE

Sandip Ray
University of Florida
ECE

Main Contributions

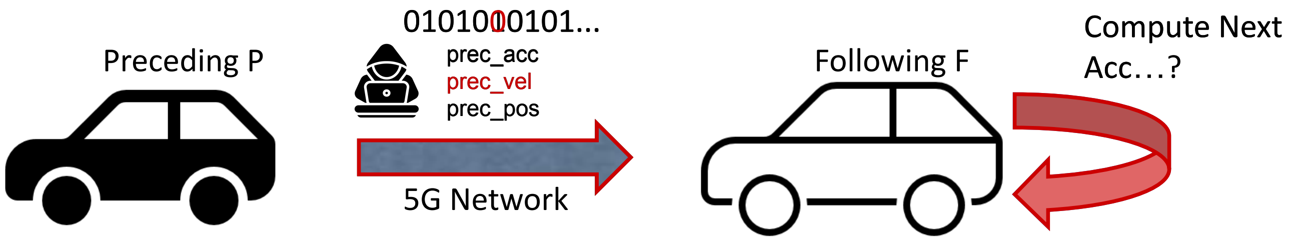
- Identified a Man-in-the-Middle bit-flipping attack on 5G network without integrity protection enabled ⇦ **Offense!**

- Proposed an alternative keystream-based shuffling protection against the bit-flipping attack ⇦ **Defense!**

- Proved that both the bit-flipping attack and the shuffling algorithm works with real datasets ⇦ **Experiments!**

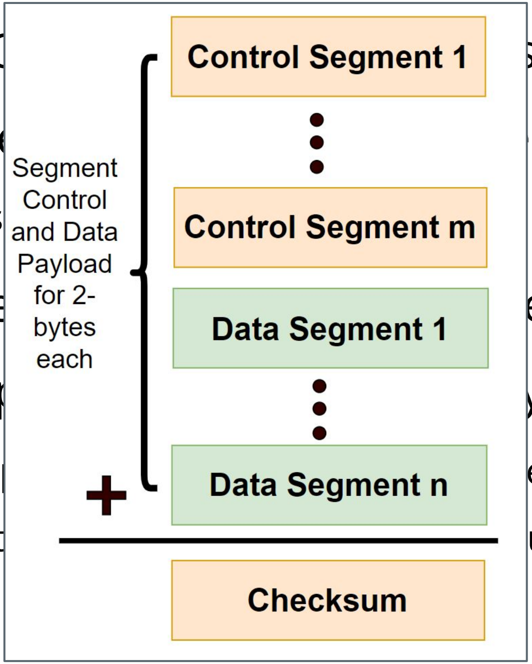
Background: Why 5G Security?

- 5G is widely used for its low latency and high data rate
- 5G enables many layers of security measures, but time-sensitive applications have to consider the cost of employing them
    - ex) Cooperative Adaptive Cruise Control (CACC)



Background: Encryption and Integrity in 5G

- Checksum: Bitwise addition of 2-byte words appended to the payload
  - For detecting corruption in network channels, not equipped to detect adversaries
- NEA (Encryption): XO sion with bit strings generated from a private seede rating function
  - Requires consensus initial seed
- NIA (Integrity): Appe enerated by a cryptographic hash function with y as inputs
  - The overhead of ap e performance critical to some time-sensitive real-t urned off)



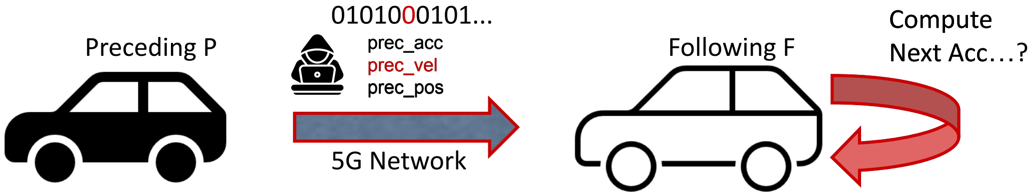
Main Contributions

- **Identified a Man-in-the-Middle bit-flipping attack on 5G network without integrity protection enabled**

- Proposed an alternative keystream-based shuffling protection against the bit-flipping attack

- Proved that both the bit-flipping attack and the shuffling algorithm works with real datasets

Threat Model



An adversary, **A**, acts as a Man-in-the-Middle (MITM) attacker between a sender (**S**) and a receiver (**R**).

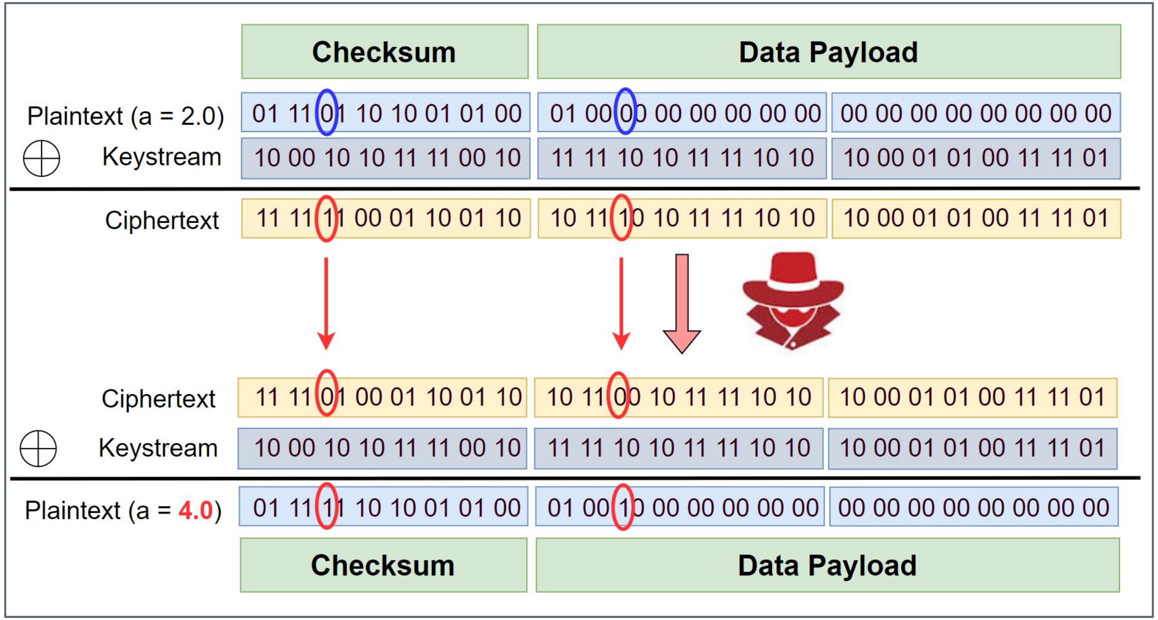
A can:

- Intercept the physical layer signal.
- Reconstruct the encrypted PDCP-layer bitstream.
- Flip any bits in the checksum and data payload fields.
- Re-encode and forward the modified message to **R**.

A cannot:

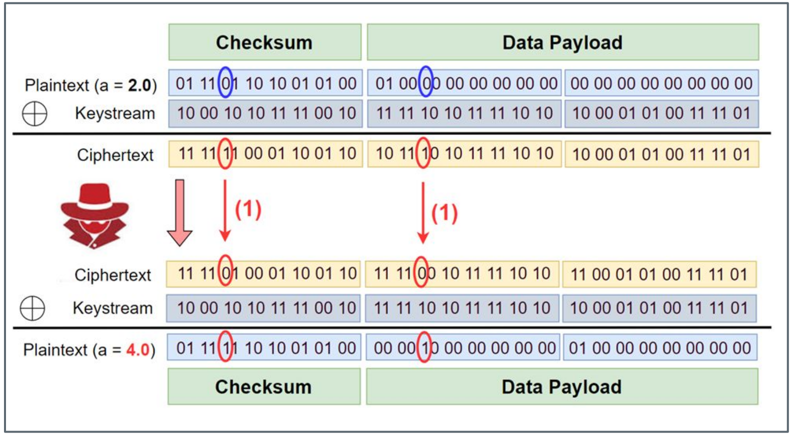
- Decrypt the NEA-encrypted ciphertext or know the secret key.

Bit-Flipping Attack



Bypasses Checksum+NEA protection **without knowledge of the keystream**!

Checksum Bit-Flipping



The attacker flips **two bits**:

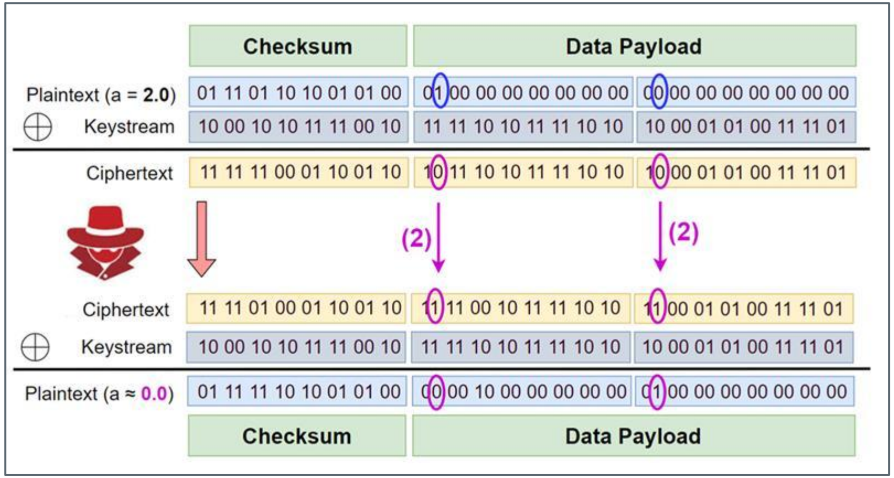
1. One bit in the data payload.
2. One bit in the checksum field at an *aligned position*. (i.e., in the same column when divided into 2-byte words for the checksum calculation).

When does it succeed?

- The attack bypasses the checksum if the two flipped bits in the *original plaintext* have **even parity** (i.e., they are the same: 0 and 0, or 1 and 1).

Since the checksum is *nearly* independent of any single payload bit, this attack has a success rate of approximately **50%**.

Payload Bit-Flipping



Motivation: checksum bit-flipping can only affect one bit.

The attacker flips **two aligned bits**, both *within* the data payload.

When does it succeed?

- The attack succeeds if the two flipped bits in the *original plaintext* have **odd parity** (i.e., they are different: 0 and 1, or 1 and 0).

The success of this attack is highly dependent on the specific data being transmitted, unlike the checksum attack.

Main Contributions

- Identified a Man-in-the-Middle bit-flipping attack on 5G network without integrity protection enabled
- **Proposed an alternative keystream-based shuffling protection against the bit-flipping attack**
- Proved that both the bit-flipping attack and the shuffling algorithm works with real datasets

Playing Defense

The Problem: The attack works because the attacker knows the position of the bits they want to change (e.g., "the 5th bit of the acceleration value").

The Idea: What if we could **shuffle** the bits of the ciphertext unpredictably before sending it?

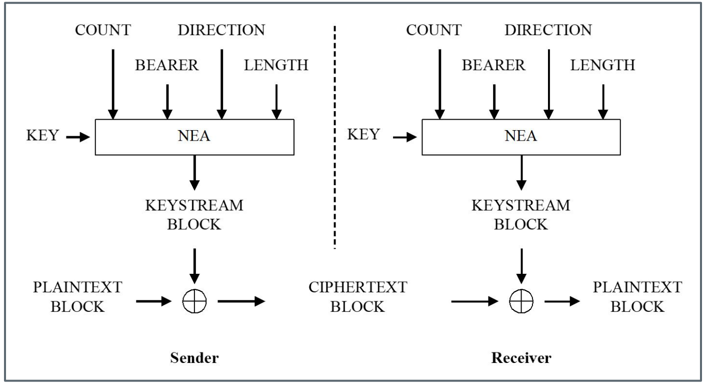
abcdefg → dgfcabe → dgfcabe → abcdefg

“I’ll flip the fifth bit!”

- If the attacker tries to flip the 5th bit, they are no longer hitting a specific, targeted bit in the plaintext, but a random one.
- We expect that multiple bit-flipping attacks in in differing positions will have an <u>exponential decay</u> in success rate. → Not too many flips!

Playing Defense

The Challenge: How can the receiver deterministically *unshuffle* the bits?
(Or, how do we coordinate the randomness between sender & receiver?)



The Solution: Use the **private keystream** already implemented in NEA!

→ Use the keystream as seed for pseudorandom permutation (Fisher-Yates)

Keystream-Based Shuffling

Sender Side:

- Generate the keystream K.
- Encrypt the plaintext: $C=P\oplus K$.
- Use the keystream K as a seed to generate a permutation table T.
- Shuffle the ciphertext C according to T to get C’ and transmit it.

Receiver Side:

- Generate the exact same keystream K and permutation table T.
- Unshuffle the received ciphertext C’ using the inverse of T to recover C.
- Decrypt: $P=C\oplus K$.

NIA vs Shuffling

	NIA	Shuffling
Protection	**Deterministic**	Probabilistic (fail w.p. <4%)
Overhead	32-bit MAC	**Zero overhead**
Coverage	**General corruptions**	Prevents targeted bit flips

- Use NIA when the system cannot afford any integrity attacks and 32-bit overhead is not significant to the system performance

- Use Shuffling when sporadic, rare attacks are acceptable but the 32-bit overhead from appending MAC is non-negligible. (CACC!)

Main Contributions

- Identified a Man-in-the-Middle bit-flipping attack on 5G network without integrity protection enabled
- Proposed an alternative keystream-based shuffling protection against the bit-flipping attack
- **Proved that both the bit-flipping attack and the shuffling algorithm works with real datasets**

Setup

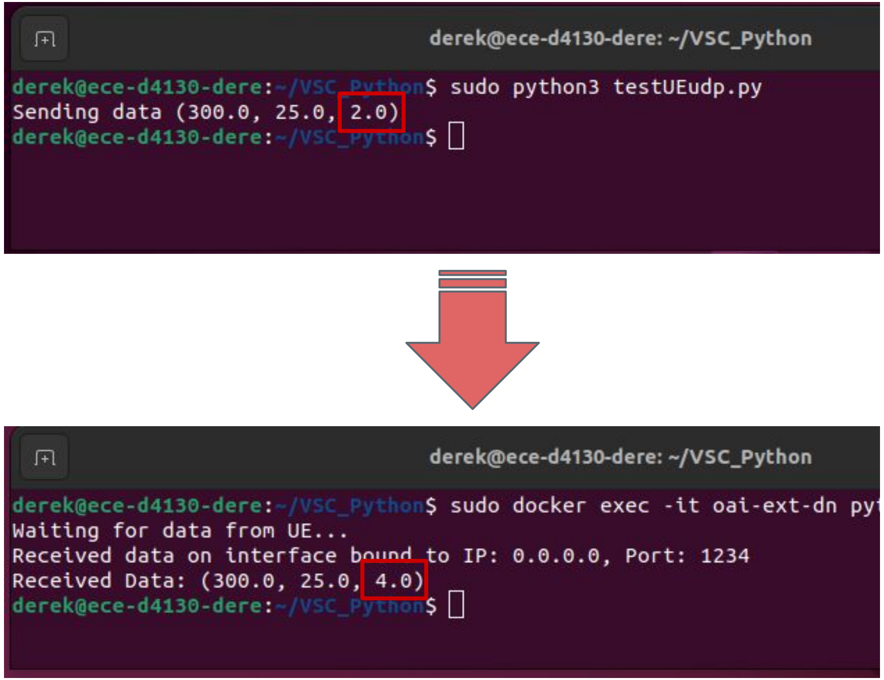
Platform: OpenAirInterface (OAI), a full-software 5G network simulation.

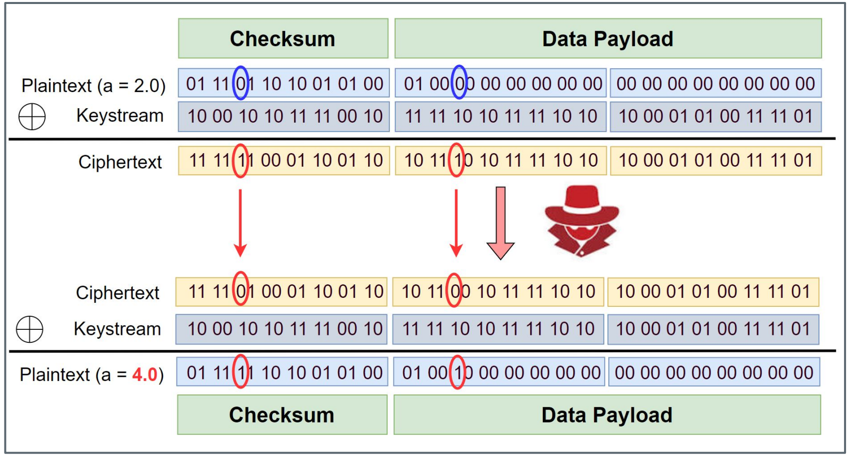
- Attacks and defenses were implemented by modifying the PDCP layer source code.

Scenario: Simulated vehicular communication (V2X).

- Transmitted Message: A vehicle's X-coordinate, velocity, and acceleration.
- Data Source: Real-world vehicle trajectories from the **NGSIM dataset**.

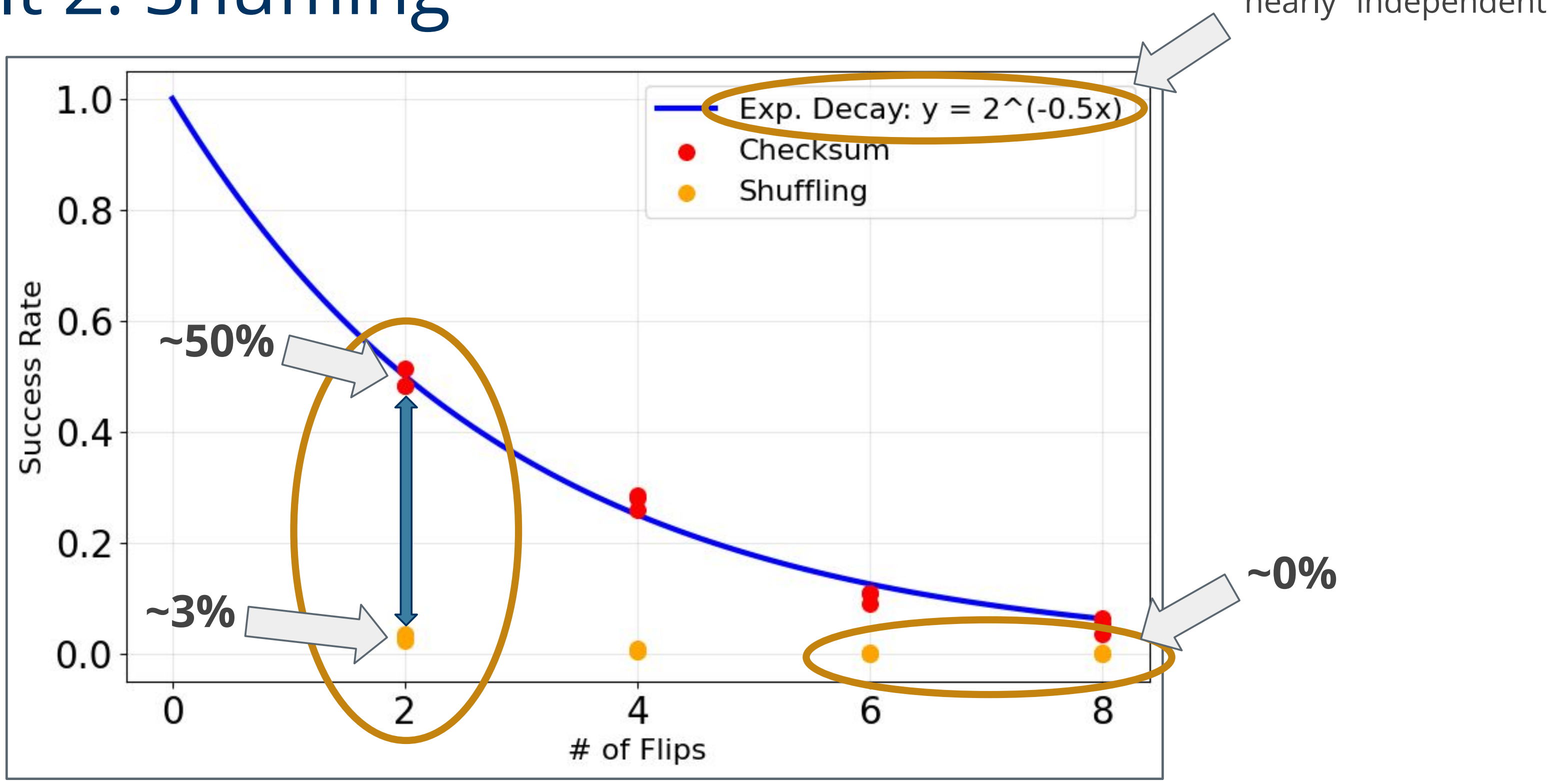
Result 1: Attack Feasibility





Flipping works as intended!

Result 2: Shuffling



Shuffling works as intended!

Conclusion

We demonstrated that **MITM bit-flipping attacks are a practical threat** in 5G, even when the attacker does not know the plaintext.

Simple checksum-based attacks can achieve a **~50% success rate** in mutating data while remaining valid.

We proposed a **keystream-based shuffling defense** that:

- Requires **no communication overhead**, unlike NIA.
- Effectively **mitigates attacks** by reducing the success rate to ~3%.
- **Prevents targeted manipulation** by obfuscating bit positions.