HW 1

Ray Li - CSCI-UA.0480-63

September 30, 2025

1. Threat modeling for Citi Bike

Here are three security policies for New York's Citi Bike system, each targeting a distinct adversary type (thieves, terrorists, trolls). For each policy, we specify concrete enforcement mechanisms and rationale.

Policy A (Thieves): Ensure bikes cannot be used or resold if unlawfully removed

Goal: Reduce economic incentive by making stolen bikes valueless and hard to monetize.
Solutions:

- *Tamper-resistant electronic locks with authenticated release*: Require station or on-bike lock to perform mutual authentication with backend before unlock; deny offline unlocks; sign unlock events to enable audit. Implement geo-fencing so a bike that leaves a service area without an authorized session hard-locks. (Defense-in-depth policy/mechanism separation discussed in Notes 2025.01.01.)
- *On-bike immobilizer with cryptographic pairing*: Pair controller to per-bike keys; if controller or battery is replaced without re-provisioning via service tool, firmware refuses to engage motor/gears. (Avoid security by obscurity; rely on secret keys rather than hidden design; cf. Kerckhoffs principle, Notes 2025.01.04.)
- *Active recovery and deterrence*: GPS/Cellular beaconing with periodic attestations; stolen-mode triggers brighter lighting patterns and server-side location beacons for recovery in collaboration with NYPD.

Why this helps: Removes resale utility and increases recovery likelihood, shifting attacker utility (Notes 2025.01.01 on attacker/defender utilities).

Policy B (Terrorists): Maintain rider safety and rapid incident response

Goal: Preserve safety and availability in the face of attempts to cause violent disruption.
Solutions:

- *Station and fleet integrity monitoring*: Backend analytics for simultaneous abnormal unlocks, sudden mass rebalancing anomalies, or tamper alerts indicating coordinated sabotage; automated incident playbooks to disable unlocks in affected zones.
- *Physical hardening and surveillance at high-risk sites*: Bollards, tamper-evident fasteners, and CCTV coverage at major transit hubs to deter placement of hazards and to support forensics.

- *Emergency broadcast and geo-fenced shutdown*: Capability to pause rentals and push safety notices within a radius; integrate with city emergency systems for coordinated response.

Why this helps: Defense-in-depth across detection, prevention, and response reduces likelihood and impact (Notes 2025.01.01 on defense-in-depth).

Policy C (Trolls): Maintain service availability and fair access

Goal: Limit nuisance attacks (dock blocking, false reports, mass reservation griefing).
Solutions:

- *Strong user authentication and rate-limiting*: Bind accounts to verified payment tokens; apply per-user and per-IP quotas on reservations, unlock attempts, and incident reports; require proof-of-ride (e.g., short post-unlock movement) to keep a dock reserved.
- *Anomaly scoring and graduated friction*: Add friction (CAPTCHAs, SMS recheck) when behavior deviates from normal usage to deter automated griefing.
- *Dock occupancy attestation*: Use on-dock sensors plus bike IMU to verify a dock is truly occupied; auto-release ghost holds.

Why this helps: Increases attacker cost for nuisance actions while minimizing friction for legitimate riders (Notes 2025.01.01 on policy vs. mechanism and attacker utilities).

2. Hash functions and privacy

Why plain hashing phone numbers does not protect privacy

Phone numbers live in a tiny, structured space (e.g., 10–15 digits with known national formats). An adversary can simply enumerate all plausible phone numbers, hash each, and match the server's values. This defeats confidentiality despite hash one-wayness (one-wayness resists inverting a random preimage, not exhaustive enumeration of low-entropy domains). It also leaks users' entire address books to the server via set-membership queries.

3. MACs vs. PRFs

(a) PRF implies MAC (short proof)

Assume there exists an adversary A_1 that wins the MAC existential unforgeability game against F with non-negligible probability. We construct A_2 to win the PRF distinguishing game using A_1 as a subroutine:

- A_2 receives oracle $O(\cdot)$ which is either $F(k,\cdot)$ or a truly random function $R(\cdot)$.
- A_2 runs A_1 and answers each MAC query m by returning $t = O(m)$.
- When A_1 outputs a purported forgery (m^*, t^*) with m^* not previously queried, A_2 outputs "real" iff $O(m^*)=t^*$, else "random".

Analysis: If $O=F(k,\cdot)$ then A_1 sees a perfect MAC oracle and, by assumption, produces a valid new (m^*, t^*) with non-negligible probability, so A_2 distinguishes PRF-from-random with the same advantage. If $O=R(\cdot)$, then t^* is independent of $R(m^*)$ and matches with probability at most 2^{-t} (negligible). Hence any MAC forger yields a PRF distinguisher, proving a secure PRF is also a secure MAC.

(b) A MAC that is not a PRF

Let F be a secure PRF. Define F'(k, m) that outputs the pair (F(k, m), 0^t) (or equivalently, append a fixed trailer bit). Verification accepts a tag t' iff its first t bits equal F(k, m). Then:

- **MAC security**: Given oracle access, producing a fresh (m, t') requires predicting F(k, m) on a new m, which is hard if F is a PRF. The fixed trailer does not help forgery.
- **Not a PRF**: F' is trivially distinguishable from random because its output always ends with 0^t. A random function produces that pattern with probability 2^{-t}, yielding a huge distinguishing advantage.

4. Semantic security for the Vigenère cipher

The Vigenère cipher with a fixed period p=10 is not semantically secure. We give a no-query adversary that wins the IND-CPA game with probability 1.

Attack idea

1. Choose two equal-length plaintexts m0, m1 longer than 10 so that for some position i, the letters at all indices congruent to i mod 10 differ between m0 and m1 but are identical within each plaintext across those positions. For example, for English letters let the positions i, i+10, i+20, ... in m0 all be 'A' and in m1 all be 'B'; keep all other positions equal between m0 and m1.
2. Submit m0, m1 as the challenge messages. Let the challenger encrypt mb under a uniformly random 10-letter key k and return c.
3. Inspect ciphertext letters at positions i, i+10, i+20, ...: under Vigenère, each of these positions is a shift by the same key letter k[i]. Because the plaintext letters at those positions are constant within a message, the corresponding ciphertext letters will also be constant (but shifted) within each message. Compute the shift between c[i] and c[i+10]. If they are equal shifts for 'A', output b=0; if they align to the shift for 'B', output b=1.

This succeeds with probability 1 since the repeated key letter induces identical shifts at every i+10j position, letting the adversary tell which constant letter ('A' vs 'B') was encrypted.

Why this breaks semantic security

IND-CPA requires that no efficient adversary can distinguish encryptions of chosen equal-length messages with probability non-negligibly greater than 1/2. Here, the periodic key leaks class information (the congruence class modulo 10), enabling frequency/structure-based distinguishing on repeated positions, violating semantic security. The weakness is exactly the key-reuse periodicity. In contrast, a one-time pad (non-repeating key) would be semantically secure, and modern randomized modes ensure semantic security by using nonces/IVs.

5. Block cipher modes of operation: plaintext block chaining (PBC)

Let PBC be defined by $c_0 = E_K(m_0) \oplus IV$ and for $i \geq 1$: $c_i = E_K(m_i) \oplus m_{i-1}$. This lets the sender evaluate all $E_K(m_i)$ in parallel, then xor with the known previous plaintext (or IV for the first block).

Attack when $m_1 = m_2 = x$ (known)

Given public IV, c_0, c_1, c_2 and a known block x with $m_1 = m_2 = x$: 1) From $c_2 = E_K(m_2) \oplus m_1 = E_K(x) \oplus x$, compute $E_K(x) = c_2 \oplus x$.

2) From $c_1 = E_K(m_1) \oplus m_0 = E_K(x) \oplus m_0$, recover the first block:

$$m_0 = E_K(x) \oplus c_1 = (c_2 \oplus x) \oplus c_1.$$

This uses only public values and the known x. Hence PBC is not semantically secure: repetitions of plaintext blocks leak $E_K(x)$ and reveal neighboring plaintexts.

6. CBC-MAC

Alice computes a MAC by running CBC encryption and keeping only the final block (tag). Let the block cipher be E_K, IV be public, and messages be multiples of the block size: for a message with blocks $(m_0, \ldots, m_{\ell-1})$:

$$c_{-1} = IV, \quad c_i = E_K(m_i \oplus c_{i-1}), \quad \mathrm{Tag}(M) = c_{\ell-1}.$$

This scheme is insecure if the IV is not fixed to zero and/or message lengths are not fixed. We show existential forgeries using only observed tags.

Forgery from two known (IV, Tag) pairs

Suppose we know (IV_1, T_1) for message $M_1 = (m_0^{(1)}, \ldots, m_{a-1}^{(1)})$ and (IV_2, T_2) for message $M_2 = (m_0^{(2)}, \ldots, m_{b-1}^{(2)})$. Consider the crafted message

$$M_3 = M_1 \parallel (m_0^{(2)} \oplus IV_2 \oplus T_1) \parallel m_1^{(2)} \parallel \cdots \parallel m_{b-1}^{(2)}.$$

Then CBC chaining over M_3 with IV$= IV_1$ yields $c_{a-1} = T_1$ and the next block input equals $m_0^{(2)} \oplus IV_2$, so the remainder of the computation exactly reproduces the path that produced T_2. Hence

$$\mathrm{Tag}(M_3) = T_2, \qquad \text{using } (IV_1, T_2).$$

This is a valid existential forgery unless M_3 was previously MACed.

One-block forgery (when a single-block tag is known)

If an observed message M has a single block x with pair (IV, T), then $T = E_K(x \oplus IV)$. For any chosen block y, define

$$M' = x \parallel (y \oplus IV \oplus T).$$

Its tag under IV is $E_K(y \oplus IV)$, i.e., the tag that would be produced for the one-block message y. Thus we can forge a valid tag for y without the key.

Why this fails

CBC-MAC is only secure when the IV is fixed (often zero) and the message length is fixed and included in the computation; otherwise, the chaining malleability above enables splice-and-extend forgeries.