Section 3: Week 6: IoT and Assisted Driving

Nate Bachmeier

TIM-7010: Computer Networking and Mobile Computing

August 4, 2019

North Central University

# IoT and Assisted Driving

“Vehicular collisions, which kill thirty thousand people in the US annually and injure almost a million more, maybe tackled by using embedded wireless sensors, monitors, and actuators in automobiles (Kanuparhi, Karri, & Addepalli, 2013).” Due to several open problems, society has not realized that vision. If we can successfully deploy IoT sensor networks and create smart roads, the potential gains to safety are enormous.

# Literature Review

## An Internet of Cars (2012).

If vehicles are going to share sensor data, then a notion of identity needs to exist for each participant (Speed & Shingleton, 2012). Without an identification service, it would not be possible to determine metrics are coming from ‘this or that’ car. Speed and Shingleton propose a system that relies on the car’s registration information, similar to a barcode. They describe an integration into smart cities that tracks the drivers and assesses common route patterns.

There are advantages to this approach; however, they do not address the drivers need for privacy. Perhaps there are multiple classes of drivers such as public buses, rideshare vehicles, and private commuters. Each of these classes requires a different level of privacy before they are willing to participate in that vision.

One solution could be to allow each vehicle to control the lifespan of their registration identifier. For instance, the public bus might be willing to base its identity on the VIN while the private commuter rotates with the engine.

This level of anonymity would protect the driver’s personal information and still enable smart roads to efficiently operate. Analogously, today we see the vehicle next to us is a Honda Prius—we do not know the driver is named John Snow.

## Smart Cities: an IoT-centric Approach (2014)

After solving identity, the next layer to address is the organization and publication of these open datasets. Smart cars will need to consume data feeds from traffic lights, road sensors, accident reports, and construction schedules, to name a few. One solution is to decompose smart cities into smart city *hubs*, where a hub is a logical unit such as a shopping district or residential neighborhood (Lea & Blackstock, 2014).

Lea and Blackstock examined the implementation of two smart cities, one in the United Kingdom the other in Canada. With both locations, an efficient data catalog was critical to the project’s success. They attributed this to reducing the learning curve, resulting in more developers creating more applications available.

Over the last five years, several American cities, such as Seattle and New York, have created open data platforms. The easy access to data has enabled dozens of high-quality applications; however, they are specific to that location.

Small development studios are willing to build dedicated apps for a given city, but more prominent corporations are unwilling to enter the space. Their userbase spans the entire country and needs it to be usable on a national level. If they must implement a data adaptor of every metropolitan area, the development cost will be too great to justify. Until businesses, city planners, and hardware manufacturers agree on open standards; there will be challenges sharing the data and making it fully discoverable.

## NDN IoT Content Distribution Model

After standardizing the data schemes, they might become discoverable through Name Data Networks (NDN). Under NDN, publishers create signed content under different namespaces, and then consumers can discover and fetch those resources (Shilton, Burke, Caffy, & Zhang, 2016). Software-Defined Networking (SDN) can cache these resources directly in the networking layer and expose them through forwarding rules.

A smart vehicle might know that it is driving down Jackson toward 7th Avenue, and use this information to request content for /USA/Washington/Seattle/Jackson/7th avenue. Through a separation of duties, the network could then return an array of data catalog entries relevant to that corner. If data is not available for the child directory, then the protocol can fall back to the parent directory. Such a scheme addresses discoverability concerns and provides extensibility for future requirements.

Having a simple system for anonymous content to be discoverable, published, and consumed enables several core scenarios. Though it also introduces complexities with data authenticity. What prevents a malicious user from reporting that a road is under construction and causing all cars to divert to other routes? Accomplishing this feat could reduce their travel time at the expense of the broader community.

## Hardware and Embedded Security in the Context of Internet of Things (2013)

Data integrity, identity management, trust management, and privacy are four areas of security that impact all IoT scenarios (Kanuparhi, Karri, & Addepalli, 2013). Kanuparhi et al. continue to provide examples across vehicular, elderly homes, and smart city verticles. They propose Physical Unclonable Functions (PUF) as potential mitigations.

Given the heterogeneous nature of the Internet of Things ecosystems, it might be unrealistic to assume the additional hardware is available. Commodity server software addresses these limitations by using Public Key Infrastructures (PKI).

PKI topologies allow chains of trust to authenticate the authenticity of data. For instance, a traffic light could publish data signed with its device certificate. The smart city hub signs the device certificate with its certificate. These cross signatures recurse outward to state or national authorities. If a security-critical event occurs, then the, e.g., traffic light’s signing identity can be revoked and its published data invalidated.

There are naturally complexities to operating a PKI that could contain records for every vehicle and device on the road, though it a solvable problem (Cincilla, Hicham, & Charles, 2016). Evaluations with simulation networks conclude that modern technologies could handle these massive workloads, provided inevitable trade-offs occur. For instance, the system administrators would need to balance costs with acceptable latencies.

## Lightweight Secure-Boot Architecture for RISC-V System-on-Chip (2019)

When devices transition from hardware-based security solutions toward software implementations, there are additional attack vectors. For example, an attacker can physically modify the traffic light’s disk image. Secure boot protections can detect these scenarios and abort the initialization of the device (Jawad, Ming, Vikramkumar, Shivan, & Anupam, 2019).

Jawad et al. provide a hardware-based implementation for System-on-Chip (SoC) systems to ensure every stage in the boot process is signed. Like previous designs, they rely on the Trusted Platform Module (TPM) as a mechanism to validate each signature. If code authentication fails, then the solution aborts the initialization.

Given the ubiquitous availability of high-speed wireless, these sensors could stream their operating system image during secure boot. Then it would not be possible for many physical attacks to compromise integrity, though there can be challenges scaling such a solution globally.

Consider the scenario where a storm disrupts wireless signal or rural areas; it might be unacceptable for entire sensor networks to be offline. If smart vehicles need to download their image from Ford or Tesla, then what happens when those sites cannot be reached? Most likely they would use a locally cached copy. The local cache introduces a new attack surface that would require signatures authenticated during secure boot.