A Name-Based Secure Communications Architecture for Vehicular Networks

Christos Papadopoulos
University of Memphis
christos.papadopoulos@memphis.edu

Alexander Afanasyev
Florida International University
aa@cs.fiu.edu

Susmit Shannigrahi
Tennessee Technological University
sshannigrahi@tntech.edu

***Abstract*—Forthcoming automotive cybersecurity standards such as ISO 21434 and regulation such as WP.29, make it imperative that automakers establish cybersecurity-by-design practices. Vehicle communication cybersecurity (both in- and out-of-vehicle) is crucial in achieving this goal. With the adoption of automotive Ethernet, automakers are turning to the Internet protocol suite (IP) to achieve the desired cybersecurity properties. However, security was always an add-on to Internet protocols, resulting in well-known security weaknesses such as spoofing, denial of service attacks, lack of authentication and more. Such weaknesses may unwittingly be brought to the automotive space.**

In this position paper we take the position that the automotive industry should consider other networking architectures besides IP as they move away from existing architectures such as CAN. Specifically, we advocate *Named Data Networking (NDN)*, an architecture that incorporates unified security-by-design from the network to the application layers. While NDN has not yet been applied to the automotive domain, our position is that its superiority to IP, especially in security, makes it a strong candidate. Unlike IP, which secures the communication channel between two entities, NDN secures the content through digital signatures that cryptographically bind a name to the content, ensuring both authentication and integrity of the data. NDN is analogous to a pub-sub model and can be implemented directly over L1, L2 or L3 layers.

I. Introduction

In this position paper we take the position that the automotive industry should consider networking architectures other than IP as it moves past existing architectures such as CAN. Specifically, we propose that the industry should investigate *Named Data Networking (NDN)* [1] as a potential candidate. NDN [1] is an experimental Internet architecture that incorporates security by design. NDN is the result of over 10 years and more than $20M of funding by the National Science Foundation (NSF) and other agencies.

The automotive community has been hard at work to ensure future vehicles incorporate strong cybersecurity. The results are standards such as ISO 21434 and regulation such as WP.29 advocating cybersecurity by design. The adoption of automotive Ethernet enables automakers to adopt the Internet protocol suite and its cybersecurity properties. However, despite being around for decades and a plethora of standards, Internet protocols have well-known security weaknesses that may unwittingly be ported to the automotive space. These weaknesses range from lack of systematic authentication in the network control mechanisms to susceptibility to various types of denial-of-service (DoS) attacks, man-in-the-middle, spoofing, and replay attacks.

Security protocols such as MACsec, IPSec, and TLS have been developed to address some of these weaknesses. However, these protocols form a patchwork of security mechanisms that do not integrate well with each other. Since security is weakest at the seams, we need a communication architecture with security by design.

Unlike IP, which secures the communication channel, NDN directly secures the content through cryptographic signatures that bind a name to the content. A receiver requesting named content can cryptographically verify each received packet, ensuring integrity and authenticity of the content. NDN's feature of naming content, securing it at the time of creation, and stateful forwarding allows the network, among other properties, to (a) deliver content over any channel available without risking to compromise its integrity and authenticity, (b) realize efficient multicast/broadcast delivery, as multiple requests for the same named content can be aggregated (e.g., in gateway ECUs), (c) recover from transmission errors, e.g., fetching content from caches if the first attempt(s) failed, and (d) deduplicate transmission on lower layers, e.g., leveraging packet naming that is preserved across the layers. NDN can either replace IP or, when necessary to run over legacy infrastructure, can work on top of IP without compromising NDN's efficiency and security properties.

NDN has not yet been applied to the intra-automotive domain. In this work we present NDN as a vision for future of automotive communications technology that can either run on top of IP or (ideally) replace IP entirely. There are many reasons for exploring automotive applications of NDN, most notably *security by design* [2]. We also believe that Name-based communication vastly simplifies application development, especially with pub-sub app models [3], and leads to a better support for open standards.

II. Use Cases

NDN is a general communication architecture and thus applies to all communication inside the vehicle. We present one illustrative example, but the generality should be clear. Suppose various modules including ABS need the current rotation speed of the left rear wheel. Currently the information may be sent periodically by the wheel rotation sensor (or the module attached to it) to individual modules one by one. There is also no security in this model so a rogue module may spoof the signal. With NDN, a module would send a request directly to the network asking for the information trough its content *name* ("subscribe" request). Such a request may look like this completely made up name: "`/vehicle/chassis/RearAxle/LeftWheel/rotationSpeed/rpm`".

There are several things to note here. (1) The information is requested by its name, not from the module that may have produced it. This easily allows enabling publisher redundancy, as NDN can forward requests to multiple locations simultaneously and filter duplicate responses. (2) If there are multiple modules that require rotation speed information, all these modules can "subscribe to" (express NDN Interests for) the data at the same time. Whenever data is published, it will be efficiently "multicasted" back to each subscriber using NDN's stateful forwarding mechanisms. (3) The "rpm" in the content name asks for the data in revolutions-per-minute. The name could just as easily specify "angularVelocity" if a module supported that format. (4) The response is digitally signed by the module that produced it, thus enabling the receiving module to reject rogue responses.

An additional advantage of named data is the ability to cache responses in the network. For example, long lived data such as "`/vehicle/AmbientTemperature`" is typically valid for minutes or longer. A request for ambient temperature can be cached in the network for some time such that future requests will be satisfied from the cache. The cached data also serves retransmissions: a request for a lost packet will be served by the closest cache that has the data.

It is easy to see that by using a named data model applications are substantially simplified due to a network service model that better aligns with their needs. In the rest of the paper we expand on the NDN service model and argue that just like the general Internet, NDN is a great match for automotive networking needs and provides a substantially improved security model.

III. Security in IP

We begin by briefly describing important attacks in vehicular networks that frame the threat model. We then describe three IP security protocols typically used to secure intra-domain, inter-domain and application-to-application communication. The discussion is far from complete, but only meant to illustrate the security problems for this position paper.

Threats in Automotive Networks: We assume that an attacker can compromise any active component (e.g., ECUs, Domain Controllers, the Gateway, LiDARs) in the vehicle, and spoof packets, modify their content, launch denial of service (DoS), man-in-the-middle (MiTM) or replay attacks. Defending against such attacks requires protocols that support integrity, authentication, encryption, key management (including key update and revocation), sender verification, end-to-end protection, and replay protection.

Addressing threats in IP: Addressing threats in IP typically requires a patchwork of security protocols at different communication layers as we describe below.

1) MACsec: MACsec is an L2 protocol that creates an encrypted unicast channel between two Ethernet nodes by encapsulating an encrypted Ethernet frame within another Ethernet frame of EtherType 88E5 followed by MACsec SecTAG, which contains information that help the receiver identify the decryption key as well as a packet number (for replay protection), followed by the payload (possibly encrypted), and the ICV (Integrity Check Value). MACsec provides confidentiality, integrity, replay protection, and authentication for Ethernet frames. While MACsec protects intra-domain (L2) traffic including IP packets, ARP, neighbor discovery and DHCP, it does not protect inter-domain (L3) traffic. MACsec is not practical for intradomain multicast communication.

2) IPSec: IPSec protects communication at L3. In an automotive environment IPSec is used to secure interdomain communication by setting up encrypted channels between devices at different domains. IPSec is often used to set up VPNs, and provides IP packet encryption, along with authentication of the source. IPSec requires a key exchange between connected devices. It also adds several headers and trailers that contain authentication and encryption information. It authenticates each packet and it encrypts each packet's payload and IP header. Encrypted packets are transmitted using UDP. Similar to MACsec, IPSec is not practical for multicast because it has to create individual one-to-many associations.

3) TLS: Transport Layer Security (TLS) is a cryptographic protocol that provides end-to-end security of data sent between two applications over the Internet. It uses a combination of symmetric and asymmetric cryptography. TLS uses asymmetric cryptography for securely generating and exchanging session keys. A session key is used for encrypting the data transmitted by one party, and for decrypting the data received at the other end. It provides authentication, integrity and confidentiality between two applications running on any ECU regardless of the domain they run in. We are not aware of implementations of TLS that support multicast in automotive environments.

All three protocols (MACsec, IPSec and TLS are under consideration for automotive environments because they secure communication at different layers. Having multiple protocols, however, leads to complexity, which is expensive and results in fragility. As the saying goes, security breaks at the seams, and IP protocols have quite a few.

Beyond these basic protocols there are many other security techniques under consideration. These, however, increase the complexity and thus amplify the need for simpler security mechanisms.

IV. A Brief Introduction to NDN

In this section we briefly describe NDN [1] and contrast NDN's security with IP. NDN is the most prominent realization of the Information-Centric Networking (ICN) vision of future device-to-device and device-to-human communication. It is a new networking paradigm that focuses on retrieving named and secured data instead of pushing unsolicited IP packets to a destination hosts. NDN can be a full replacement of IP, e.g., run directly on top of Layer 2 links including Ethernet and automotive variants of Ethernet such as IEEE 802.X, or run on top of legacy infrastructure.

NDN Naming NDN names are hierarchical and use the concept of longest prefix match when forwarding. This results in a natural naming format. The following illustrative examples use signal names from VSS [4]:
"`Vehicle.Chassis.Axle.Row1.Wheel.Tire.Pressure`"
"`Vehicle.Drivetrain.FuelSystem.Level`"
To convert these signals from VSS to NDN notation we simply replace the "." with a "/":
"`Vehicle/Chassis/Axle/Row1/Wheel/Tire.Pressure`"
"`Vehicle/Drivetrain/FuelSystem/Level`"

These are valid NDN names and can be presented immediately to the network as we describe below. In contrast, with IP we would have to discover the IP address of the ECU providing this information, and possibly establish a (perhaps secure) connection before we can send a request for this data.

Interest Packets NDN communication follows a "subscription" model on the networking level. In other words, consumers explicitly request (subscribe) via *Interest* packets the data they want. State created in the network by each Interest, ensures that only consumers who requested this data packet, will receive it. Therefore, a consumer module simply cannot be flooded with unsolicited packets (e.g., DoS'ed or DDoS'ed). eliminating a large class of attacks because the network does not have a mechanism to deliver unsolicited data.

Forwarding of the requests towards the potential locations of data (be it the original producer, managed storage, or an in-network cache) is based on hierarchically structured names. When an Interest reaches a producer who can answer the query, the producer sends back a **Data packet**. This packet follows the reverse path (following network state) back to the sender of the Interest.

Here is a brief, illustrative example. When an on-board monitoring ECU needs to retrieve information on the rotation of the rear left wheel, it can subscribe to this data by sending an Interest that may look like this: "`/body/ABS/RearLeftWheel /rotation/rpm`" The Interest packet is forwarded by the network towards the appropriate sensor(s). When the Interest reaches the sensor, the latter publishes a *Data* packet with the requested information.

Directly Secured Content All content in NDN is signed and when needed encrypted at the time of publication (in our example, by the sensor). A Data packet contains both the name of the content and a cryptographic signature that binds the name with the (encrypted) content signed by the content producer. This provides both authentication, integrity, and confidentiality of the data regardless how data packets retrieved or where they were (temporarily) stored.

NDN Stateful Forwarding and Native Multicast NDN packet forwarders (domain controllers or the) forward Interest packets over one or more output ports toward publisher(s) that can satisfy the Interest. Forwarders record the name of the Interest in the Pending Interest Table (PIT). When an Interest reaches the data producer it generates a response in the form of one or more Data packets, which come back following the reverse path. With this, multiple Interests for the same Data can be aggregated and, when published, Data can be efficiently and **natively multicasted** back to the subscribers/consumers.

The NDN forwarders cache the returning Data packets in their Content Store (CS), an in-memory local cache that stores packets temporarily to satisfy future requests. Such in-network data retrieval is possible because content is named: a new Interest asking for the same data can be identified and serviced from the forwarder cache. Caching enables **fast recovery from transmission errors**. Upon receiving a new Interest, the forwarder tries to service the request from its local cache first. If the content is not available in the cache, the Interest is forwarded toward the data producer.

QoS and Real Time Named content also enables Quality of Service (QoS) and real-time guarantees to be implemented in the network by associating certain names with QoS classes. For example, Interests and Data that have strict timing requirements can be identified by the forwarders by their name and given higher priority. This can be done as follows: suppose that a video stream from the in-vehicle entertainment system (IVS) needs real-time guarantees, then Interest and Data packets with the prefix "`/vehicle/ivs/video`" would be moved to the front of the queue when forwarded by each NDN Domain Controller. Alternatively, if reservations are required Interests and Data packets can be used to setup reservations as described by TSN [5]

Forwarding Performance Current NDN implementations can reliably forward over 100Gbps using COTS equipment [6]. With resource-constrained automotive hardware forwarding speeds will be significantly less (although still in the hundreds of Mbps if not Gbps). Experiments have shown that the most significant factor affecting forwarding speed is the number of components in the name prefix [7], not the name length—long names can always be hashed. We suspect that in automotive environments the namespace prefix trees will be fairly shallow (although the number of leafs maybe large). Name prefixes only need to be long enough to reach the correct module.

NDN Limitations NDN is an experimental architecture and thus comes with limitations. Perhaps the main limitation for automotive applications is lack of standards. Unlike IP that has been around for decades, which allowed a rich set of standards to be developed, NDN is an experimental architecture. Another limitation that follows the lack of standards is lack of hardware implementations. These limitations mean that NDN cannot be immediately be put into production. However, it does not mean that NDN, once studied and evaluated carefully, cannot evolve into a robust automotive architecture, which is our position in this paper. We are in the process of doing exactly that by porting NDN into automotive modules and demonstrating its merits. Fortunately, there is significant existing work we can leverage: VSS for naming, SDKs for porting NDN into ECUs, testbeds to emulate real vehicle networks, CAN traffic simulators to test performance, and documented attacks to test security. We will leverage all these resources in our lab.

V. Security with NDN

An important architectural difference between NDN and IP that strongly impacts security is that in NDN all communicated

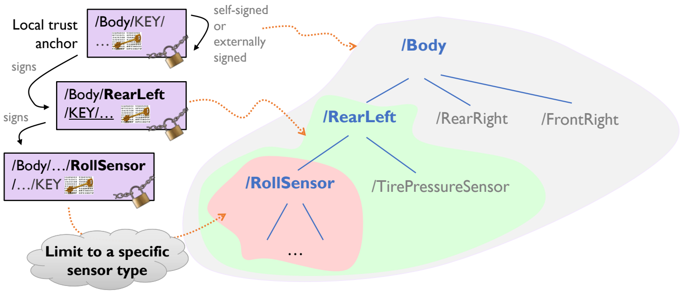

Fig. 1: Data authentication in NDN

Data is "directly secured". In other words, all Data packets are cryptographically protected and cannot be tampered with. The receiver-driven model (with proper naming strategy) essentially eliminates replay attacks: a receiver can receive only the Data that is requested by name and is legitimately produced by a corresponding sensor. In addition, as was already emphasized in previous sections, because the receiver needs to express an Interest in order to receive any packet back, it eliminates a large portion of denial of service attacks. A receiver can still be overwhelmed by Interests, but such attacks can be mitigated by leveraging the stateful forwarding model by keeping track of satisfied/unsatisfied interests and/or rate limiting and prioritizing specific Interests [8].

An additional security property made possible by NDN's Data naming approach is flexible privilege separation. Specifically, NDN can leverage relations of Data names and key names to reliably enforce not just that a specific Data is signed by a key, but that such a key is a legitimate signer for the Data (Figure 1). In other words, in addition to the signature, each data packet carries metadata including the name of the signing key, which identifies the cryptographic key to verify the signature. Here is an illustrative example. The name ("`/Body/RearLeft/RollSensor/.../KEY/...`") and the actual data name "`/Body/RearLeft/RollSensor/Degrees/v15/...`" Data) share a prefix, which provides the security context to determine whether the key is authorized to sign the data. Such context effectively enforces a least privilege principle, where an individual sensor key can only be used to authenticate this sensor data loss or compromise of such a key (e.g., by breaking into a single ECU) will not impact overall communication integrity for other components in the system. The name relations can be generalized in a *trust schema* [9], which can automate complex name-based authorizations.

When privacy is needed, NDN utilizes data encryption to prevent unauthorized access. Once a trust model is established, an entity can determine which other entities it can trust and provide decryption keys (perhaps temporarily) for accessing private data. The whole key exchange procedure is again based on names that require no additional infrastructure set up.

With name-based signatures NDN standardizes security mechanisms and largely eliminates the need for security mechanisms at other layers. The actual security implementation can vary from namespace to namespace, e.g., devoting stronger cryptography and resources to critical data. Similarly, a designer can optimize security, resource use and performance based on system needs. A designer can also fine tune mechanisms that provide integrity, authentication, encryption, key management (including key update and revocation), sender verification, end-to-end protection, and replay protection.

VI. Conclusions

As adoption of automotive Ethernet becomes a reality, the industry is leveraging long-lived IP technology for vehicular communication. However, despite its relative maturity, IP has well-known security limitations. Other models under consideration such as Data-Centric pub-sub communication models (DDS, CORBA, MQTT), still rely on and inherit the security limitations of IP.

In this position paper we argue that the automotive industry should pay attention to IP alternatives such as NDN that aim to improve on IP and incorporate security by design. NDN is a fundamentally better platform to implement Data-Centric communication because it secures all communication layers, not just the application. NDN can easily be adopted in automotive networks, and can be applied to both in-vehicle as well as V2X communication. NDN can run on top of IP, but maximum security benefits are achieved by replacing IP completely. NDN is actively supported by a substantial researcher and developer community, working on optimizing the implementation and supporting new applications. Code and libraries are available for download allowing anyone to experiment and evaluate NDN.

References

[1] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos *et al.*, "Named data networking (ndn) project," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, vol. 157, p. 158, 2010.

[2] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev, and L. Zhang, "An overview of security support in Named Data Networking," *IEEE Communications Magazine*, vol. 56, no. 11, pp. 62–68, 2018.

[3] K. Nichols, "Lessons learned building a secure network measurement framework using basic ndn," in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, 2019, pp. 112–122.

[4] "Overview :: Vehicle Signal Specification," Jul 2021, [Online; accessed 25. Jul. 2021]. [Online]. Available: http://genivi.github.io/vehicle_signal_specification/introduction/overview

[5] "IEEE 802.1 Time-Sensitive Networking Task Group," Jul 2017, [Online; accessed 25. Jul. 2021]. [Online]. Available: https://www.ieee802.org/1/pages/tsn.html

[6] J. Shi, D. Pesavento, and L. Benmohamed, "Ndn-dpdk: Ndn forwarding at 100 gbps on commodity hardware," in *Proceedings of the 7th ACM Conference on Information-Centric Networking*, 2020, pp. 30–40.

[7] Z. Li, Y. Xu, B. Zhang, L. Yan, and K. Liu, "Packet forwarding in named data networking requirements and survey of solutions," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1950–1987, 2018.

[8] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest flooding attack and countermeasures in Named Data Networking," in *Proc. of IFIP Networking 2013*, May 2013. [Online]. Available: http://networking2013.poly.edu/program-2/

[9] Y. Yu, A. Afanasyev, D. Clark, kc claffy, V. Jacobson, and L. Zhang, "Schematizing trust in Named Data Networking," in *Proceedings of 2nd ACM Conference on Information-Centric Networking*, Sep. 2015. [Online]. Available: http://dx.doi.org/10.1145/2810156.2810170