POSTER: Bloom Filter based Certificate Validation for VANET

Hongyu Jin
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
hongyuj@kth.se

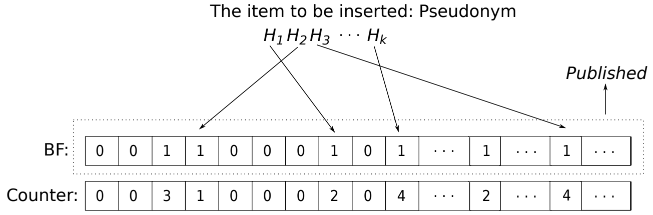
Panos Papadimitratos
Networked Systems Security Group
KTH Royal Institute of Technology
Stockholm, Sweden
papadim@kth.se

ABSTRACT

Security and privacy are important properties that have to be considered for the adoption of Vehicular Ad-hoc Networks (VANETs). Short-lived credentials, termed *pseudonyms*, are used to ensure message integrity and authentication while preserving vehicle (thus, their passengers') privacy. However, this introduces extra communication and computation overhead: pseudonyms have to be attached to the messages and signatures on pseudonyms and messages need to be verified before they can be accepted. In this poster, we are concerned with computation overhead for pseudonym validation. We preload vehicular On-Board Units (OBUs) with a Bloom Filter (BF) to facilitate pseudonym validation while traditional approach (i.e., signature verification on pseudonyms) can still be preserved as a fallback approach. We evaluate our scheme on automotive testbed with a preliminary implementation. Our scheme provides low processing delay for pseudonym validation at a cost of communication overhead for pre-downloading the BF.

1 INTRODUCTION

Vehicular Communication (VC) systems ensure safety for vehicles and their passengers. This is achieved through safety beacons broadcasted at high rate by the vehicular OBUs. This provides awareness of nearby vehicles (e.g., their speeds and directions) and surrounding environment (e.g., obstacles and accidents). Safety beacons are signed accordingly so that authentication and integrity of the messages are protected. However, to protect privacy of the passengers, the unlinkability of the messages needs to be preserved [5]. Therefore, short-term/pseudonymous certificates, termed *pseudonyms*, are used to ensure message authentication and integrity while preserving privacy [2]. However, this comes with costs of communication and computation overhead: certificates have to be attached to the messages to facilitate message verification and the signatures on the pseudonyms and the messages have to be verified before they can be accepted. In this poster, we are concerned with computation overhead, especially the processing delay for pseudonym validation. We propose a BF-based pseudonym validation scheme. A BF, built based on the pseudonyms the Pseudonymous Certification Authority (PCA) has issued, is downloaded from the PCA by vehicles [1]. The BF can be used to validate pseudonyms at a lower processing delay (with hash computations) than signature verification. We evaluate our scheme on automotive testbed with a preliminary implementation. We show that our scheme can be implemented on off-the-shelf vehicular OBU and the processing delay for pseudonym validation can be significantly decreased.



BF:	0	0	1	1	0	0	0	1	0	1	···	1	···	1	···
Counter:	0	0	3	1	0	0	0	2	0	4	···	2	···	4	···

Figure 1: BF Construction with k hash functions [1]

2 OUR SCHEME

In this section, we explain the BF-based pseudonym validation scheme, that first proposed and described in [1]. We start with preliminaries for our scheme and explain our scheme in detail.

2.1 Preliminaries

Counting Bloom Filter: BFs [3] are used for efficient membership checking, that built based on elements of a dataset. It can be used for membership checking for a given element. Each element in the set is hashed with k hash functions, while the output of each hash function is a position in an m-bit vector and these k positions are set to 1. However, if any of them is already set to 1 upon a previous insertion, these bits are simply kept as 1 and ignored. For a membership checking, the element is hashed with the same k hash functions, and the derived k positions are compared against the BF. If all k positions are 1, then the element has passed the membership test. A BF reduces spatial overhead at the expense of a false positive rate. An element not included in the original dataset could pass the BF test if all k positions for this element were set to 1 by other elements. For a standard BF (the type we consider in our paper), m and k are chosen based on the number of the dataset elements, n, and the false positive rate to minimize spatial overhead, m [3].

A standard BF supports insertions of new elements, but no deletions: a bit in the BF might be needed by multiple elements. A new BF has to be built from scratch if elements are deleted. Counting BFs [3] maintain a counter for each bit, indicating the times it was set to 1. Therefore, when an element is deleted, for each of its k bits, the counter is decreased by 1. If a counter is decreased to 0, then the corresponding bit in the BF is also set to 0. The size of a counter should be chosen properly [3].

Compressed BF-Delta: Compressed BF-deltas [3] can be used to publish updates when a few of the BF elements are changed

WiSec '17 , Boston, MA, USA
© 2017 Copyright held by the owner/author(s). 978-1-4503-5084-6/17/07…$15.00
DOI: 10.1145/3098243.3106020

(e.g., inserted or deleted). This provides an efficient way to publish differences (in terms of each bit value) between old and new BFs with minimum overhead.

Note: An *alternative* to BFs could be a concatenation of hash values for elements in the dataset, published as a hash list. For membership checking, the hash value of the element is computed and searched in the hash list. However, for a large dataset, BFs are far superior in terms of spatial overhead [4]. Moreover, searching in a hash list requires $O(n)$ time complexity, while a BF-based checks require $O(k)$ time complexity (with $k \ll n$, typically, for any sizeable dataset).

2.2 Scheme Details

Without loss of generality, we assume the majority of vehicles (e.g., local vehicles) have been preloaded with pseudonyms for a period, Γ (e.g., 24 h), thus covering $[t_{start}, t_{start} + \Gamma]$. We assume these pseudonyms are requested well in advance before t_{start}. The PCA generates a BF that includes the pseudonyms covering $[t_{start}, t_{start} + \Gamma]$. We do not dwell on the selection of t_{start}; e.g., a point during the night could be chosen, so that vehicles request pseudonyms and download the new BF while parked.

Pseudonyms can either have overlapping (e.g., 100 pseudo-nyms for each vehicle, all valid for 24 h) or non-overlapping (e.g., 144 pseudonyms for each vehicle, each valid for 10 *min*) lifetimes. In the former case, an element in the BF is the public key of a pseudonym; while for the latter case, an element is the combination of a public key and its corresponding lifetime. Fig. 1 shows the construction of the BF based on the pseudonyms. Although the PCA maintains a counting BF, only a standard BF is published, because the counters are not necessary for pseudonym validation; counters are used to support insertions and deletions to the BF. While a larger counter size results in higher storage overhead (for the PCA), this does not affect the size of downloaded BF (thus the communication overhead for the vehicles).

Vehicles that did not request pseudonyms from the PCA before t_{start} could request pseudonyms throughout the day. This can be, e.g., due to non-predictable trips or new vehicles joining from other domains. As these vehicles request pseudonyms from the PCA, the BF has to be updated to cover these new pseudonyms. A vehicle could update the BF either proactively, when the vehicle is parked, or reactively, when it starts receiving a considerable amount (above a protocol-selectable threshold) of pseudonyms not included in the BF. We use compressed BF-deltas to minimize the communication overhead for updating the BF.

Validation process: In order to validate a pseudonym, the receiver first tests the pseudonym against the currently available local version of the BF. If the BF test is successful, the pseudonym is checked against the Fake Pseudonym List (FPL): the pseudonym is validated if it is not included in the FPL. If the pseudonym did not pass the BF test, the signature on the pseudonym has to be verified (i.e., the baseline scheme). To ensure resilience to clogging Denial of Service (DoS), the fraction of such baseline validation should be conservative and adaptive. In order to mitigate the effect of fake pseudonyms, for each pseudonym that passed BF test and FPL check, the receiver could verify probabilistically (with a low probability) the signature on the pseudonym. If this pseudonym cross-verification fails, the fake pseudonym is reported to the Vehicular Public-Key Infrastructure (VPKI) and added to the FPL.

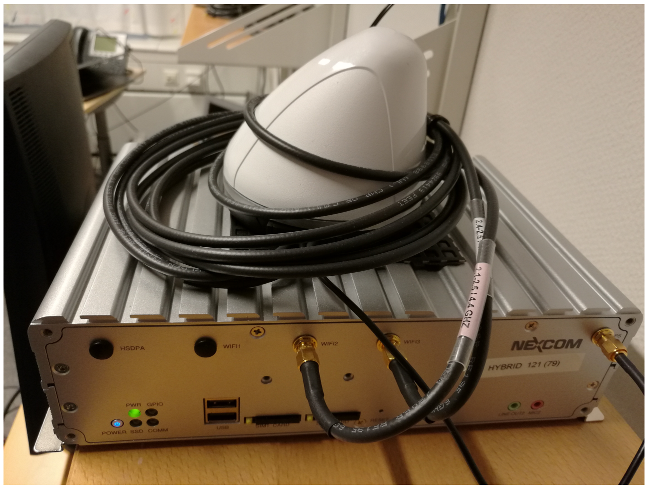


Figure 2: Vehicular OBU used in the evaluation

3 EVALUATION

In this poster, we mainly evaluate the performance of our scheme on automotive testbed in a lab environment. We refer the reader to [1] for security and privacy analysis. Fig. 2 shows the NEXCOM vehicular OBU used in our experiment, which support IEEE 802.11p. We implement our scheme in C++ and ECDSA algorithm is used for message signing and verification. We show that our scheme can provide low processing delay for pseudonym validation with hash computations using preloaded BF. This trades off communication overhead for downloading the BF from the PCA and storing the BF locally. As described in Sec. 2, the BF needs to be updated when there are new vehicles joining [1]. However, we do not evaluate BF update in this poster, while we assume all the received valid pseudonyms are included in the downloaded BF. Our scheme is orthogonal to the traditional approach, because all the pseudonyms are still attached with PCA signatures. Thus, the traditional approach is available as a fallback approach when the BF is not available or a pseudonym that is not included in the BF is received.

REFERENCES

[1] H. Jin and P. Papadimitratos. Proactive certificate validation for vanets. In *IEEE VNC*, Columbus, OH, Dec. 2016.

[2] M. Khodaei, H. Jin, and P. Papadimitratos. Towards deploying a scalable & robust vehicular identity and credential management infrastructure. In *IEEE VNC*, Paderborn, Germany, Dec. 2014.

[3] M. Mitzenmacher. Compressed bloom filters. *IEEE/ACM Transactions on Networking (TON)*, 10(5):604–612, 2002.

[4] M. Nielsen. Why bloom filters work the way they do. http://www.michaelnielsen.org/ddi/why-bloom-filters-work-the-way-they-do/. Accessed 2016-11-05.

[5] P. Papadimitratos, V. Gligor, and J.-P. Hubaux. Securing vehicular communications-assumptions, requirements, and principles. In *ESCAR*, Berlin, Germany, Nov. 2006.