Proactive Certificate Validation for VANETs

Hongyu Jin and Panos Papadimitratos
Networked Systems Security Group, KTH Royal Institute of Technology, Sweden
{*hongyuj, papadim*}@kth.se

***Abstract*—Security and privacy in Vehicular Ad-hoc Networks (VANETs) mandates use of short-lived credentials (pseudonyms) and cryptographic key pairs. This implies significant computational overhead for vehicles, needing to validate often numerous such pseudonyms within a short period. To alleviate such a bottleneck that could even place vehicle safety at risk, we propose a proactive pseudonym validation approach based on Bloom Filters (BFs). We show that our scheme could liberate computational resources for other (safety- and time-critical) operations with reasonable communication overhead without compromising security and privacy.**

***Index Terms*—Bloom Filter, Pseudonym, Security and Privacy**

I. Introduction

Vehicle-to-Vehicle (V2V) communication improves road safety and traffic efficiency with safety beacons, broadcasted at a high rate to provide cooperative awareness. Short-term credentials, i.e., pseudonyms obtained from the Vehicular Public-Key Infrastructure (VPKI) through protocols as in, e.g., [3], [11], [6], are used for message (beacon) authentication and integrity while protecting user privacy. Pseudonyms with overlapping or non-overlapping lifetimes can be preloaded, for a long period (e.g., 1 year) or be requested on-demand (e.g., on a daily basis). In a multi-domain Vehicular Communication (VC) system, pseudonyms in a domain are generally issued by the Pseudonymous Certification Authority (PCA) dedicated to that domain, and a vehicle that wishes to enter another (foreign) domain should request pseudonyms from the corresponding PCA. For ease of explanation, we assume the domains are separated geographically in the rest of the paper.

Pseudonyms are changed over time for message unlinkablity. Due to mobility of vehicles, the neighborhood of a vehicle can be volatile, thus, having new pseudonyms received practically continuously. The challenge is that all such digitally signed new pseudonyms must be validated in order to verify messages. Certificate omission [2], and optimistic or probabilistic message validations [5], [1], [7] have been proposed, but they do not reduce pseudonym validation overhead. In some situations, a vehicle could receive a very large number of new pseudonyms within a short period (e.g., around a mix-zone [4], where all vehicles would change their pseudonyms).

We propose a Bloom Filter (BF) based pseudonym validation scheme. Instead of verifying the PCA signature for each and every pseudonym, the pseudonyms are validated through a BF published by the PCA, which includes all pseudonyms valid within a protocol selectable period. Once the BF is verified and stored, a vehicle can efficiently validate the pseudonyms based on cheap hash computations with reasonably low false positive rate.

We require that all the pseudonyms are still signed by the PCA and the messages be signed under the pseudonyms. This ensures that a fallback approach (i.e., PCA signature verification on each and every pseudonym) can be invoked when suspicious behavior is detected. We show that our scheme could reduce computational overhead. Although an attacker could launch a brute false attack targeting the false positive rate of the BF (attempting to inject messages signed under fictitious pseudonyms), we show that such an attack is expensive and could cause minimal harm to the system.

In the rest of the paper, we describe the adversary model (Sec. II), present our pseudonym validation scheme inspired by [10] (Sec. III), provide a security and privacy analysis (Sec. IV), and a preliminary evaluation of our scheme (Sec. V) before some concluding remarks (Sec. VI).

II. Adversary Model

We consider (external or internal) adversaries that attempt to insert false messages, without using a legitimate private/public key pair and the corresponding pseudonym in order to affect other vehicles. Such an attack could inject, for example, a false event (e.g., a non-existing accident). In addition, we consider adversaries interested in launching clogging Denial of Service (DoS) attacks, i.e., sending out messages with fake signatures at a high rate, in order to consume resources of benign vehicles (and leave them with scarcer resources for processing legitimate, and potentially critical, messages).

Internal adversaries could threaten the network by sending out false information with valid signatures under valid pseudonyms. Misbehavior detection would then lead to their identification and eviction; this is orthogonal to this paper.

III. Our Scheme

A. Preliminaries

Counting Bloom Filter: BFs [8] are used for efficient membership checking in Internet applications. A BF is built based on elements of a dataset, and the published BF 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 k hash functions, and the derived k positions are compared with the BF. If all k positions are 1, then the

978-1-5090-5197-7/16/$31.00 ©2016 IEEE

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 [8].

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 [8] 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 [8].

Compressed BF-Delta: Compressed BF-deltas [8] can be used to publish updates when a few of the BF elements are changed (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 [9]. 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).

B. Scheme Overview

We propose an efficient pseudonym validation scheme based on BFs. The PCA generates a counting BF based on all the currently valid pseudonyms issued to vehicles. Compressed BF-deltas are used to publish updates in case of insertions (e.g., new pseudonyms provided in response to recent requests) and deletions (e.g., revocation of pseudonyms). Vehicles can download the BF (without the counters) from the PCA once it is built, and download periodically newer versions (or deltas). Once the BF is downloaded, vehicles could validate received pseudonyms with the BF, at a processing cost that is a tiny fraction of that to validate a digital signature by the PCA. If a pseudonym does not pass the BF test, e.g., in the event recently issued pseudonyms are not yet included in the BF, a receiving/validating vehicle can always choose the fallback approach (referred as the *baseline* scheme in the rest of this paper): verify the PCA signature on the pseudonym.

C. Bloom Filter based Pseudonym Validation

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.

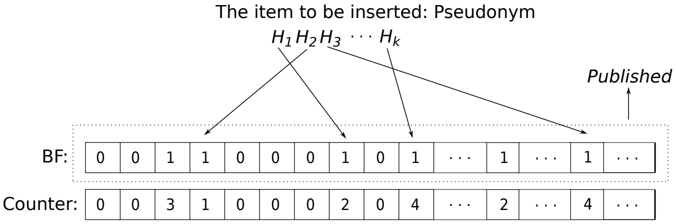

Fig. 1: BF Construction with k hash functions

Pseudonyms can either have overlapping (e.g., 100 pseudonyms 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.

As BF exhibits a false positive rate, a fake pseudonym discovered by a brute force search (with very low probability if we choose m appropriately) could be accepted even if it were not issued by the PCA (see Sec.IV for more details). This can be mitigated by applying a probabilistic verification even if a pseudonym passed the BF test. If such a double-checked pseudonym is proven fake, it is reported to the VPKI and published in a Fake Pseudonym List (FPL). An FPL is a list of detected fake pseudonyms that could pass BF tests.

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 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 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

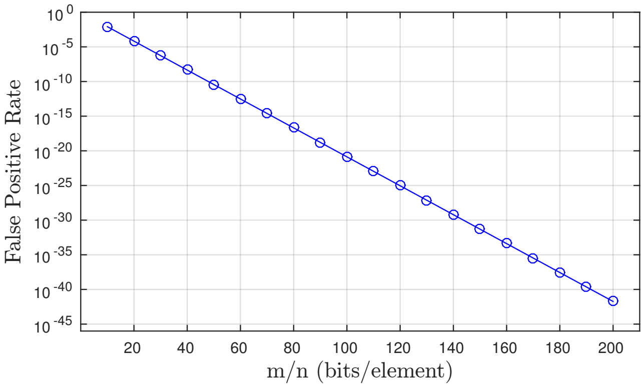

Fig. 2: False positive rate as a function of m/n

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 VPKI and added to the FPL.

IV. Security & Privacy Analysis

Non-repudiation, authentication and integrity: We require that all valid pseudonyms and messages be signed by the PCA and their senders respectively. Thus, our scheme does not affect the non-repudiation of the messages. The BF and its deltas are authenticated and cannot be repudiated. If the received pseudonyms are not included in the local BF or, in case, any suspicious actions are detected, vehicles can always validate pseudonyms using the baseline scheme.

Fake pseudonyms: An adversary could target the false positive rate of a BF, shown in Fig. 2 as a function of m/n (bits per element) [8]. For example, when $m/n = 80$ bits, the false positive rate is around 2×10^{-17}. Consider the case the pseudonyms have the same lifetime (e.g., valid for $24\,h$). On average, an attacker would have to generate around 10^{17} public/private key pairs to find a fake one passing the BF test, each test needing k hash computations. If the pseudonyms have non-overlapping lifetimes, a pair of public/private key could be tested with different lifetimes, thus less key pairs would be needed to find one passing the BF test. However, such a fake pseudonym can only be used for a short period (i.e., within its lifetime). Moreover, probabilistic verification further shortens the period a fake pseudonym can be used before being detected. Last but not least, a limited number (depending on the processing power of the attacker) of fake pseudonyms would not significantly affect the system if vehicles with valid pseudonyms are the majority within the neighborhood. Overall, such a brute force attack is expensive, and it may have a negligible effect.

Privacy of newly joining vehicles: A PCA would update its BF as new vehicles join the domain (and request pseudonyms). However, this raises a privacy concern that pseudonyms of new vehicles could be easily linked. For example, if the BF is updated for three new vehicles which appear in different parts of the domain; a global passive attacker could easily link the sets of pseudonyms that were not included in the old BF. This can be mitigated by updating the BF only when a considerable amount of new vehicles joins the domain, essentially, creating a larger anonymity set for those new comers.

Thwarting clogging DoS: No invalid pseudonyms (or messages) would be accepted by the baseline scheme. However, this makes clogging likely: an attacker could generate arbitrary strings as public keys and attach arbitrary strings as signatures, and broadcast at a high rate. This kind of attack is cheap, while consuming resources of benign vehicles to verify the fake signatures. Optimistic message verifications have been widely studied while some of them rely on short-term linkability of the messages signed under the same pseudonym. For example, [7] proposes to use TESLA for the verification of following messages after a signature verification on the first message. However, it would be pointless to thwart this kind of attack by sacrificing linkability among the pseudonyms. Our scheme can efficiently thwart such an attack, while an attacker needs significant effort to find false positive pseudonyms as we discussed earlier. If increasing amount of pseudonyms with fake signatures (that do not pass BF tests) are received; the fraction of CPU time assigned for verifying signatures on pseudonyms (including both new legitimate pseudonyms and randomly generated pseudonyms) can be reduced and vehicles should update their BFs in order to properly validate legitimate pseudonyms.

V. Performance Evaluation

A. Communication Overhead

Consider a BF with a false positive rate of 10^{-20}. If $n = 14\,400\,000$ (100 000 vehicles equipped with 144 pseudonyms valid for $24\,h$ each), $m \approx 164.5\ Mbytes$ ($m/n \approx 96$ bits, see Fig. 2). The original BF can be downloaded before a trip starts, which takes, e.g., around $1\,min$ with a bandwidth of $20\ Mbps$. The size of the BF is acceptable considering the volumes that could be provided by off-the-shelf hard-drives nowadays.

The use of compressed BF-delta [8] decreases the communication overhead to update the BF. The compression rate can be computed, with $q = p(1 - p^f)$:

$$\textit{Compression Rate} \ = -q \log_2 q - (1-q) \log_2 (1-q), \quad (1)$$

where p is the probability that a bit in the original BF is 1, and q is the probability that a bit in the BF-delta is 1 after a fraction, f, of pseudonyms are added to the original BF. For a standard BF, which we use in our scheme, m/n and k are chosen so that $p \approx 1/2$ [8].

Fig. 3 shows the compression rate of a BF-delta as a function of f. If 144 000 new pseudonyms (for 1 000 new vehicles) were added, then the size of the compressed BF-delta is around 6.2 Mbytes (with a compression rate of 0.0045). For 10 000 new vehicles, it is 34.8 Mbytes. We should note that the false positive rate of the BF increases as pseudonyms are added, because more bits are set to 1. The probability that a bit in the updated BF is 1 can be computed as: $p' = p + (1-p)q$. For example, if 10 000 (i.e., $f = 0.1$) new vehicles would join everyday, $p' = 0.6$ after adding the pseudonyms. Then, the

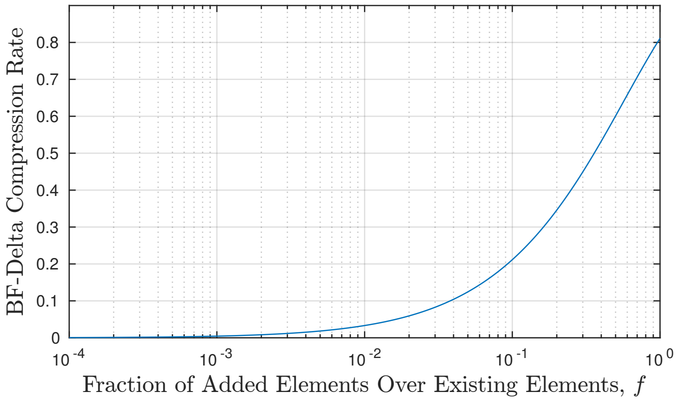

Fig. 3: Compression rate of BF-delta as a function of the fraction of added elements.

false positive rate of the updated BF is around 3.29×10^{-20}: a slight increase from 10^{-20}. We refer the reader to [8] for the false positive rate calculation.

B. Computation Overhead

In our scheme, validation of a pseudonym requires k hash computations, which is much cheaper than a signature verification. Consider the following case: N vehicles are within a vehicle's communication range and the neighborhood refresh/change (in terms of new pseudonyms) ratio is c per second. Each vehicle broadcasts γ beacons per second. We assume ECDSA-256 for both the pseudonyms and the PCA certificate, and signature verification delay $\tau = 4\ ms$ (a typical value from the literature [2]). For simplicity, we assume the delay of a BF test is $0\ ms$ (in reality, it introduces a tiny delay for k hash computations, which can be in the order of μs) and all the pseudonyms from N neighbors are included in the BF.

We consider a two-class $M/D/1$ queue for message verifications, as in [2]. The first class includes messages signed under new pseudonyms and the second class includes messages signed under stored pseudonyms. The *average system time*, $\bar{T}$ (total time in the queue until a message is verified), can be represented as:

$$\bar{T} = \bar{S} + \frac{\lambda_1 S_1^2 + \lambda_2 S_2^2}{2(1-\rho)}, \tag{2}$$

where $\bar{S}$ is the average service time, and λ_i and S_i are the arrival rate and the service time of ith class. We can derive that $\lambda_1 = cN$, $\lambda_2 = \gamma N - \lambda_1$ and $S_2 = \tau$, while S_1 for the baseline and the BF-based schemes are 2τ and τ respectively. From *Little's law*, we know that $\rho = \rho_1 + \rho_2$, $\rho_i = \lambda_i S_i$ and $\bar{S} = \frac{\rho}{\lambda_1+\lambda_2}$.

Fig. 4 shows $\bar{T}$ as a function of c when $N = 50$. As expected, c does not affect $\bar{T}$ for the BF-based scheme, because BF tests introduce negligible delay. However, for the baseline scheme, $\bar{T}$ increases as c increases because more signature verifications are needed for the new pseudonyms. For example, when $\gamma = 3$ and $c = 0.6$, $\bar{T}$ with the baseline scheme is almost double of that with the BF-based scheme (without proactive cross-verification).

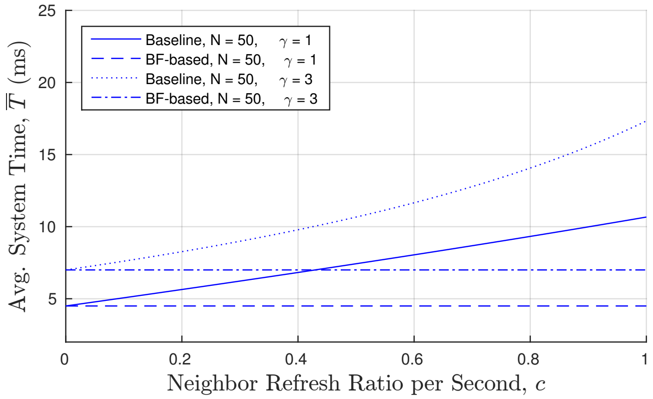

Fig. 4: Average waiting time as a function of neighbor refresh ratio per second

VI. Conclusion and Future Work

We presented a BF-based pseudonym validation scheme. We showed that the BF can be downloaded with acceptable overhead and it can be used to validate pseudonyms efficiently with a reasonably low false positive rate. Even though an attacker could launch a brute force attack on the BF, this would be expensive and likely to cause minimal harm to the system.

In this paper, we consider and analyzed the scheme with one BF per domain. The immediate extension is to generalize, e.g., with one BF per PCA (presuming multiple PCAs exist in a domain), and download the BFs from all the PCAs in the domain or even download BFs from PCAs in neighboring foreign domains (to facilitate "roaming").

References

[1] A. Al-Momani, F. Kargl, C. Waldschmidt, S. Moser, and F. Slomka. Wireless channel-based message authentication. In *IEEE VNC*, Kyoto, Japan, Dec. 2015.

[2] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy. On the performance of secure vehicular communication systems. *IEEE TDSC*, 8(6):898–912, 2011.

[3] Car-to-Car Communication Consortium Pilot PKI. http://www.car-2-car.org/.

[4] J. Freudiger, M. Raya, M. Félegyházi, P. Papadimitratos, and J.-P. Hubaux. Mix-zones for location privacy in vehicular networks. In *ACM Win-ITS*, Vancouver, Canada, Aug. 2007.

[5] H. Jin and P. Papadimitratos. Scaling VANET security through cooperative message verification. In *IEEE VNC*, Kyoto, Japan, Dec. 2015.

[6] 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.

[7] C. Lyu, D. Gu, Y. Zeng, and P. Mohapatra. PBA: Prediction-Based Authentication for Vehicle-to-Vehicle Communications. *IEEE TDSC*, 13(1):71–83, 2016.

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

[9] 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.

[10] K. Ren, S. Yu, W. Lou, and Y. Zhang. Multi-user broadcast authentication in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 58(8):4554–4564, 2009.

[11] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn. A security credential management system for V2V communications. In *IEEE VNC*, Boston, MA, Dec. 2013.