CAESAR: A Criticality-Aware ECDSA Signature Verification Scheme with Markov Model

Chao Chen[1], Sang Woo Lee[2], Tim Watson[1], Carsten Maple[1], and Yi Lu[1]

[1]Cyber Security Centre, University of Warwick, United Kingdom
[2]ETRI, Electronics and Telecommunications Research Institute, South Korea
[1]{c.chen.27, tw, cm, y.lu.16}@warwick.ac.uk, [2]ttomlee@etri.re.kr

***Abstract*—Intelligent transport systems (ITS) facilitate road traffic by periodically exchanging messages with neighbouring vehicles, road side units (RSUs) and ITS stations. For security reasons these messages will be encapsulated with security credentials to form secured messages (SMs) and will be inoperative until the authentication completes. This creates a challenge in a dynamic and dense road network where many SMs are awaiting authentication. To address this problem, we propose CAESAR, a criticality-aware Elliptic Curve Digital Signature Algorithm (ECDSA) signature verification scheme that utilizes multi-level priority queues (MLPQs) and Markov model to dispatch and schedule SMs. Simulation results verify the accuracy of CAESAR and the enhancements in terms of several safety awareness metrics compared with the existing schemes.**

***Index Terms*—ITS, Authentication, Security;**

I. Introduction

Intelligent transport systems (ITS) is a core component of the future traffic system. Equipped on-board units (OBUs) and wireless data exchanging are essential requirements for cooperative awareness, emergency warning notification, efficient route guidance and entertainments. ITS applications broadcast their information to surrounding vehicles, RSUs and ITS stations. These broadcast messages are known as basic safety messages (BSMs) in the WAVE standard [1], cooperative awareness messages (CAMs) and decentralized environmental notification messages (DENMs) in ETSI standard [2]. Exchanged messages facilitate vehicles and stations to extend their control beyond the non-line-of-sight (NLOS) and develop a local dynamic map (LDM) containing a guidance of road traffic [3], such as intersection collision warning, wrong way driving warning, approaching emergency vehicle warning application, etc. Because of the importance of LDM, exchanging messages are an easy target for malicious users. Hence, authentication is a key procedure in the transmission of messages. Once a message is generated at the application layer, a digital signature is added and it is encapsulated as a secured message (SM) before transmission [4]. At the receiver side, the received SM is placed into a message queue where it will be verified before passing it to the relevant application. The content of message cannot be aware by the ITS application before it has successfully been verified.

For the road safety, vehicles are required to respond to arrival SMs as soon as possible. However, this authentication incurs additional communication and processing overheads that decrease the quality of service (QoS) of ITS [5] [6]. Both ESTI and WAVE recommended the usage of ECDSA instead of RSA algorithm for signing and verifying SMs to improve the QoS. In addition, [7] [8] adopt the approach to verify SMs nearby firstly, to achieve the road safety. But it could delay the processing of critical SMs (CSMs) and hazard the safety at the following three scenarios in Fig. 1.

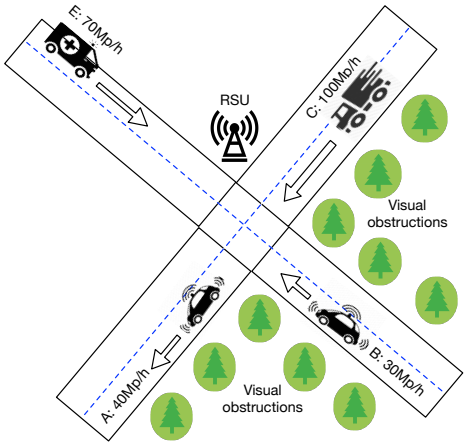

Fig. 1: The exemplar of critical scenarios

- Vehicle A (40Mp/h) and vehicle B (30Mp/h): despite vehicle A and B are geographically close compared to all others, they cannot meet each other or cause any dangerous issue, because they are in different directions of the crossroad. However, if vehicle A verifies SMs from vehicle B or vice versa, it could delay the CSMs from other areas where cause dangerous.
- Truck C (100Mp/h) and vehicle B: since vehicle B is far away in distance from truck C, vehicle B might process the verification from truck C lately. But truck C is in a high speed moving towards vehicle B, and trees amongst in the middle of the road as a visual obstruction that becomes a NLOS condition. This will cause a severe accident if its CSMs cannot be verified in time.
- Vehicle E (70Mp/h) and others: Vehicle E as an emergency vehicle is approaching in this area from a long distance. Considering this is a busy area where vehicles verified and processed SMs nearby firstly, before vehicle E is closing, none of these vehicles could have enough time to adjust their positions.

In this paper, we propose a new criticality-aware authentication framework, CAESAR. It prioritizes the signature

verification of CSMs, rather than simply verifying the geographically close area. CAESAR classifies the received signal strengths of SMs into several ranges and dispatches them into MLPQs. This enables SMs from different areas to be placed in different priority queues and hence verified rapidly. We analyse the performance of CAESAR and compare it with existing schemes using simulations. Results show that the criticality-aware authentication framework can significantly improve various performance metrics of safety applications.

II. Related work

A number of schemes have been proposed to reduce verification processing overhead [4], [7]–[9]. These can be regarded as two main aspects: the random, and the distant-based signature verification.

The random based verification schemes [4] [9] choose a few of random SMs for verification to reduce the congestion at the security queue. The drawback of these random verifications cannot guarantee the nearby situations or important SMs will be verified and processed in time. [8] prioritizes the security queue based on the distance between transmitter and receiver. [7] also adopts the distance-based method with the K-means clustering to classify the received SMs, which requires the off-line training and a larger time complexity compared to ours. In addition, as we mentioned in Section I, critical cases are inevitably happening various areas in the real traffic. Authentication solely relying on the distance could delay the verification and processing of CSMs and it causes unnecessary accidents.

A Markov chain (MC) is a sequence of random values, X_t, whose value at a time interval, t. The value of a MC variable at the present time is called its *state*. For any $t \in$ time intervals, the probability distribution of the state at time $t+1$ depends on the state at time t, and does not depend on the previous states leading to states at time t [10].

$p_{i,j} \equiv \vec{P}$ is the probability that state j at time t to the system is in state i. If the system has a finite number of states (1, 2, $\cdots$, S), then the MC model can be defined by the transition probability matrix as Eq. 1, where $\sum_{j=1}^{s} p_{i,j} = 1$.

$$\vec{P} = \begin{bmatrix} p_{1,1} & p_{1,2}, & \cdots & p_{1,s} \\ p_{2,1} & p_{2,2}, & \cdots & p_{2,s} \\ \vdots & \ddots & & \vdots \\ p_{s,1} & p_{s,2}, & \cdots & p_{s,s} \end{bmatrix} \quad (1)$$

Therefore, we can define the transition matrix after $m+n$ steps using Eq. 2 [10].

$$P_{i,j}^{n+m} = \sum_{k \in S} P_{i,k}^{n} P_{k,j}^{m} \quad (2)$$

III. CAESAR: Criticality-Aware ECDSA Signature Verification Framework

In this section, we unfold CAESAR in details. The key idea is to prioritize the verification of receiving SMs based on its distribution of CSMs on different message queues as shown in Fig. 2. Equipped sensors in the vehicle can receive SMs from other surrounding vehicles or ITS-stations. Some of SMs

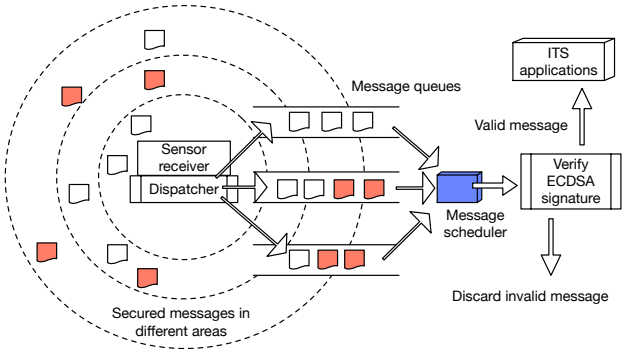

Fig. 2: CAESAR system overview

are extremely critical in the road situation that we marked them as the red colour, their criticality is unknown for ITS applications before the verification. After the receiver gathers its surrounding SMs, it then dispatches SMs into MLPQs based on their signal strengths for being scheduled, verified and processed. Experimental studies [11], [12] demonstrate sensors can partition these received messages by its signal strengths. Each one of message queues represents its geographical region by one interval of signal strength. The MLPQ allows the vehicles to schedule the verification of SMs based on the assigned priority. Using MLPQ and MC, CAESAR schedules SMs from each message queue to verify based on the distribution of CSMs amongst MLPQs. As a consequence, CSMs are verified with improved outcomes compared to others.

A. Dispatching SMs based on the physical distance into multilevel queues

The first feature of CAESAR consists of dispatching SMs to their corresponding message queue. Each one of message queues has an interval of signal strength that indicates a SM will be allocated to which message queue. For example, there are k message queues, and each interval of signal strength is $[(k_i - 1) \cdot \theta,\ k_i \cdot \theta)$, where k_i is the index of message queue, and θ is the interval of signal strength that we can obtain from Eq. 3, where $\vec{\mu}$ is the vector of signal strengths for all received SMs.

$$\theta = \frac{\max(\vec{\mu}) - \min(\vec{\mu})}{k} \quad (3)$$

Using the received signal strengths and the number of message queues, we can determine and dispatch SMs into relative message queues, respectively. It is worth noting the time complexity of this classification is $O(n)$ compared to $O(n^{dk+1} \cdot logn)$ in [7] by k-means clustering, which is more practicable in the real case.

B. Criticality distribution by modelling Markov Chain

CAESAR's second feature is about determining the priority of MLPQ by modelling MC. In terms of MC, we consider each state in the MC as the ratio of CSMs from all received CSMs in MLPQs at one time slot, which is achievable at the application layer. At each one of time slot, i.e., one second, a vehicle receives SMs with various signal strengths

and dispatches them in message queues. Each one of MLPQs contains a portion of CSMs. A transition between two states represents the probability/portion of CSMs in MLPQs between two sequential time slots. For example, a three message queue has one state $[0.3, 0.1, 0.5]$ means 30% of CSMs from all received CSMs are in the first message queue, q_1, 10% CSMs in q_2 and 50% CSMs in q_3. Given the observations of received CSMs, $X_0, X_1, \cdots, X_t$, we can obtain $p_{i,j}$, the probability of transition from state i to state j by Eq. 4.

$$p_{i,j} = Pr[X_j|X_i] = Pr[q_j \leftarrow q_i \mid t\,] = \frac{C_j}{C} \leftarrow \frac{C_i}{C} \qquad (4)$$

where C is the total number of received CSMs within MLPQs at time slot t, the number of received CSMs at message queues, q_i and q_j, are represented as C_i and C_j, respectively.

Based on ITS safety applications, we can use a series of observations regarding to the distribution of CSMs over message queues to calculate the transition probability matrix. Therefore, based on the current distribution of received CSMs as the state and the transition probability matrix, we can use Eq. 2 to calculate the distribution of CSMs of next state.

C. SM Queue Scheduler and ECDSA Signature Verification

The third feature of our proposed framework contains the SM multi-level queue scheduling algorithm which aims at extracting SMs from different queues based on the distribution of CSMs to verify their signatures (using ECDSA), as listed in Algorithm 1. During the first initial step (line 2 in Algorithm 1), all received SMs have been dispatched into relevant MLPQ based on its signal strengths and ready to be scheduled. For each one of scheduling, a random number between 0 and 1 is generated, p_d, for determining the selected queue (line 3). Eq. 2 is then applied to calculate the next distribution of CSMs for MLPQs (line 4).

Then, roulette selection is adopted to select a queue, q_i, based on the calculated distribution of CSMs over MLPQs and the determined number, p_d (line 5). For the roulette selection procedure (line 14), it will firstly iterate all probabilities to generate the ratio of CSMs over MLPQs as Cumulative Distribution Function (CDF). Then, to iterate each one of CDF and compare it with p_d. If p_d is bigger than one of probabilities, this queue is selected to be returned.

Once the selected queue, q_i, is determined, message scheduler extracts the SM from the chosen queue to apply ECDSA algorithm (line 6). If this SM is verified successfully, it will pass to ITS applications to further process, such as build up LDM. Otherwise, if it cannot be verified (due to malicious data injection attacks), this message will be regarded as invalid and discarded it, or a reputation module [13] can be implemented to isolate the malicious nodes in the network, which is outside the scope of this paper.

Algorithm 1 Scheduling SMs to ECDSA signature verification

```
1:  procedure SECUREDMESSAGE SCHEDULE
2:      while True do
3:          The determined number, p_d ← Random(0, 1)
4:          The probability distribution, P_i ← Eq. 2
5:          Selected queue, q_i ← Roulette selection (p_d, P_i)
6:          Verification flag ← ECDSA_verify(q_i)
7:          if Verification flag ≠ True then
8:              Invalid message and discard it.
9:          else
10:             Valid message and pass it to ITS applications.
11: procedure ROULETTE SELECTION(p_d, P_i)
12:     for p_i ∈ P_i do
13:         CDF of message queues, F(x) += p_i
14:     for x ∈ F(x) do
15:         if p_d < x then
16:             select x as q_i
17:             return q_i
```

Hardware	ECDSA_Sign (ms)	ECDSA_Verify (ms)
OBU-ARMv7	27.9	33.7
Intel-i5	5.46	7.32
Intel-i7	1.85	2.21

TABLE I: The comparisons of ECDSA on different hardwares

IV. Performance evaluation

A. Simulation Setup

We develop a simulation model to analyse the performance of CAESAR. This simulation is used for SMs transmission exchange between vehicles and RSUs. The packet size of a SM including the security overhead is taken as 300 bytes. Each vehicle generates 10 SMs per second with 6Mpps data rate. We adopt ECDSA-256-SHA-256 algorithm from the ETSI to sign and verify SMs. We evaluated the signature and verification at three hardware platforms (OBU ARM v7, Intel i5 and Intel i7), the results display in Table I. Due to the road safety, specialized hardware security module or trusted platform module will be utilized to accelerate the cryptographic operations by using the higher CPU instead of directly using OBU [6]. For simplicity, we choose 5ms and 7ms for signing and verifying, respectively. SMs that could not get verified within 100ms time interval are dropped from the security queue and this loss is termed as the cryptographic loss.

To generate mobility traces, we choose the scenarios as shown in Fig. 1. The critical cases are emerged from different areas that based on three major distributions (Exponential, Gaussian and Uniform) through five MLPQs within the simulation. A road network of 2km $\times$ 2km is used. The vehicle density is set to 150 vehicles/km^2 to create a dense network.

B. Simulation Results

We first evaluate the accuracy of CAESAR compared to our generated traffic, which is composed of different distribution of CSMs for incoming SMs in different areas. Then, CAESAR compares the loss percentage of CSMs with three existing techniques. The first one is the single queue first-come-first-server (SQ-F), which is the default signature verification mechanism in the WAVE and ETSI standards. The second technique is the MLPQ first-come-first-server (MQ-F) [7]

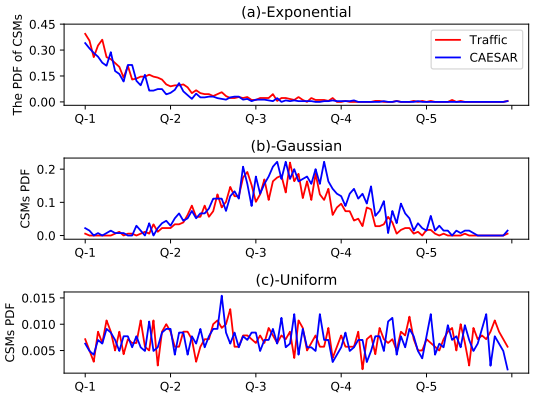

Fig. 3: The comparisons of CAESAR and road traffic in PDF

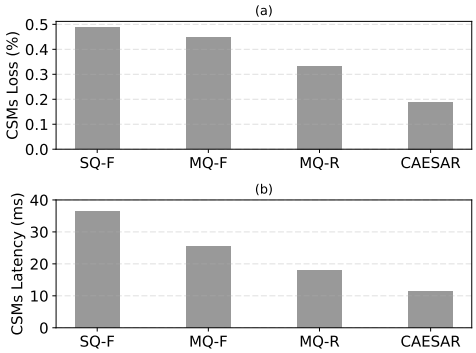

Fig. 4: The performance comparisons of CAESAR with others

and the third technique is MLPQ random signature verification (MQ-R) that randomly picks SMs from its MLPQs. In the end, we compare the end-to-end latency for CSMs and all received SMs with three techniques.

1) Accuracy of CAESAR: The experimental results for the accuracy of CAESAR on the prediction of CSMs as in probability density function (PDF) are shown in Fig. 3, which breaks down the accuracy by five message queues (Q-1, Q-2, Q-3, Q-4 and Q-5). It can be seen from this figure that CAESAR is fairly accurate. A further observation shows that as the traffic distribution from the exponential to the uniform, the average ratio of difference are float around 0.45. Although a few of ups and downs at Q-4 in the Gaussian distribution have some differences, CAESAR captures most sharp changes through 5 message queues amongst these three major distributions.

2) The performance comparisons of CAESAR with others: The experiments in this subsection investigate the loss of CSMs within a dense network. Results are presented in Fig. 4(a). CAESAR significantly reduces the loss of CSMs compared with other three approaches by around 30%, 26% and 14%.

Moreover, we compare CAESAR in the end-to-end latency for CSMs in Fig. 4(b). CAESAR improves the latency by 25ms, 14ms and 6.6ms, respectively. In addition, the latency of all received SMs are primarily identical for MQ-F, MQ-R and CAESAR, which proves the efficiency of CAESAR at critical situations whilst it can guarantee the latency of total received SMs.

V. Conclusion

We have described a priority based queuing approach to reduce the signature verification time for CSMs in the ITS. CAESAR adopts the message classification by the signal strengths to dispatch into MLPQ. With the help of Markov model, CAESAR assigns the relevant priority to MLPQ that ensures CSMs have a rapid verification. Then, we have compared our approach with the existing schemes and shown its accuracy and a significant improvement in terms of cryptographic packet loss and end-to-end delay.

Acknowledgement

This work is conducted under international technology R&D collaboration program which is supported by the Ministry of Trade, Industry & Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT) (N0001710).

References

[1] D. Jiang and L. Delgrossi, "Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments," in *IEEE VTC*, 2008.

[2] A. Festag, "Cooperative intelligent transport systems standards in europe," *IEEE communications magazine*, vol. 52, no. 12, 2014.

[3] ETSI. (2011) Intelligent transport system (its); basic set of applications; local dynamic map (ldm). [Online]. Available: https://goo.gl/U4gYU2

[4] M. Raya, P. Papadimitratos, and J.-P. Hubaux, "Securing vehicular communications," *IEEE Wireless Communications*, vol. 13, no. 5, 2006.

[5] E. B. Hamida, H. Noura, and W. Znaidi, "Security of cooperative intelligent transport systems: Standards, threats analysis and cryptographic countermeasures," *Electronics*, vol. 4, no. 3, 2015.

[6] M. A. Javed, E. Ben Hamida, and W. Znaidi, "Security in intelligent transport systems for smart cities: From theory to practice," *Sensors*, vol. 16, no. 6, 2016.

[7] E. B. Hamida and M. A. Javed, "Channel-aware ecdsa signature verification of basic safety messages with k-means clustering in vanets," in *AINA*. IEEE, 2016.

[8] Z. Li and C. Chigan, "On resource-aware message verification in vanets," in *Communications (ICC)*. IEEE, 2010.

[9] S. Biswas and J. Mišić, "Location-based anonymous authentication for vehicular communications," in *Personal Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2011.

[10] S. Karlin, *A first course in stochastic processes*. Academic press, 2014.

[11] R. Meireles, M. Boban, P. Steenkiste, O. Tonguz, and J. Barros, "Experimental study on the impact of vehicular obstructions in vanets," in *Vehicular Networking Conference (VNC)*. IEEE, 2010.

[12] E. B. Hamida and G. Chelius, "Investigating the impact of human activity on the performance of wireless networks—an experimental approach," in *IEEE WoWMoM*, 2010.

[13] S. K. Dhurandher, M. S. Obaidat, A. Jaiswal, A. Tiwari, and A. Tyagi, "Vehicular security through reputation and plausibility checks," *IEEE Systems Journal*, vol. 8, no. 2, 2014.