2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD) | 978-1-6654-4594-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/METROCAD51599.2021.00015

PlateGuard: License Plate Privacy Protection for Internet of Vehicles

Michael Nutt
Dept. of Computer Science and Engineering
University of North Texas, Denton, USA
MichaelNutt2@my.unt.edu

Qing Yang and Song Fu
Dept. of Computer Science and Engineering
University of North Texas, Denton, USA
{Qing.Yang, Song.Fu}@unt.edu

***Abstract*—The prevalence of affordable cameras and computing power has given rise to Automatic License Plate Reader (ALPR) systems that are able to easily collect and catalog license plate (LP) data from passing cars, providing a database of locations for drivers at specific times. These systems are often unsecured and little research has been done to provide a solution for protecting the privacy of citizens. This paper presents a system of encrypting the data gathered by these ALPR systems in a way that is easily implemented and able to reliably retrieve the encrypted data given legitimate cause. To improve the efficiency, reliability and scalability of the system, we design an edge based system to distribute the processing responsibilities of the system, considering that the main use cases require the system to service a large number of cameras in a city. The proposed system is evaluated using a set of metrics considered by the requirements of the system, with results indicating that the system can feasibly protect the privacy of anyone whose data is captured and retrieve specified LP data while maintaining secrecy on unspecified plates in the captured footage.**

I. Introduction

The rapid increase of Internet of Things (IoT) devices has driven development of systems that are more easily able to run on minimal hardware. For example, Automatic License Plate Reader (ALPR) [17] devices currently range from industry standard hardware, with ALPR technology built in to the camera, to hobby-grade systems built using Raspberry Pis and free and open source software. As such, we have witnessed an increasing number of cities around the world adopting Internet connected video systems, for a variety of traffic planning and law enforcement purposes. At the same time, these systems are continuously gathering personally identifiable information (PII), e.g., the LPs of passing vehicles. Many of these systems use specialized cameras and software, to log every passing LP number, the time, and the location of its passing, compromising the privacy of every passing individual. This poses a challenge to strike a balance between preserving the information gathered by these systems, as needed for legitimate uses by government entities, and maintaining the privacy of those whose information is gathered.

This can be achieved with a system that provides safeguards, adapted onto existing systems by adding an additional layer of edge devices between the existing sensing devices and the cloud storage system. Due to the strict privacy requirements posed by various ALPR applications, it is possible to transmit image/video data to the cloud, and process them on centralized servers; however, compliance is easier when the edge device acts as the front-line for privacy protection and policy enforcement. In addition, implementing privacy protection at the edge layer can save network bandwidth, and add privacy protection closer to the sources of data. This additional layer is able to provide the security and privacy protections that are currently lacking or in some cases non-existent on many of these ALPR systems.

A. Motivations

Privacy advocates are pushing for better security and privacy protections covering camera data, while understanding a need for the recoverability of the information gathered. "Privacy advocates do not oppose the use of the technology during an active investigation, but they say that maintaining a database of LP locations for months or years provides too much opportunity for abuse by police." [14] How to achieve a trade-off between the privacy protection and utility of camera data becomes a profound yet important research problem.

This issue was recently brought before the Virginia courts in a case where the American Civil Liberties Union (ACLU) sued the government regarding the use of these cameras. The ACLU won an injunction where the county judge ordered the county police to stop the use of these cameras as passive use of data from the ALPR cameras violates Virginia privacy law. [14]. This case specifically calls for a solution that provides means to the police to protect the sensitive information captured by these cameras, enforcing privacy protections.

Systems in place often have limited or no security measures to prevent unauthorized access of the information gathered, presenting a security and privacy risk to everyone whose information is captured by this system. In an investigation of open ALPR cameras connected to the Internet, TechCrunch security editor Zack Whittaker found that over 150 ALPR cameras were searchable online and easily accessible, with many also listing default passwords in the user guides [21].

B. Proposed Solution

In order to protect the privacy of citizens, a system is needed to detect and encrypt the LP information gathered, especially considering the rising prevalence of ALPR cameras and ease of use to set up these systems. Examining this need, we design the PlateGuard system which can retrieve and encrypt PII and

978-1-6654-4594-8/21/$31.00 ©2021 IEEE
DOI 10.1109/MetroCAD51599.2021.00015

hide the private information in captured data. While designing this system, we consider several requirements to ensure its adoption: ease of setup, accuracy of LP detection, privacy protection of LP information, and ability of recovering specific information if provided legitimate cause.

To meet these requirements, our solution takes a three-layer approach, containing the sensor layer, processing layer, and application layers. The sensor layer is comprised of all the cameras that are feeding data into the system, the processing layer is a network of edge devices servicing the sensor layer by isolating and encrypting the LP data in the saved videos and providing storage for them, and the application layer is deployed on a cloud server allowing end users to query and retrieve relevant footage for specified LPs. In captured images, the regions that contain LP information will be encrypted and only non-sensitive information, e.g, color and type of vehicles, is accessible to end users.

C. Contributions

To the best of our knowledge, we are the first to propose a system of protecting LP data using ALPR to detect, isolate and protect the plate areas; preserving the main video data using the LP number as key for encryption to allow only specified plates to be recovered. Through our experimentation and evaluation, we show that PlateGuard provides an improved scope of security for individuals, while also creating a distributed network to allow for lower network usage for transferring video, and a faster response time when querying a range of cameras across the edge network. The proposed framework does not limit the choice of encryption algorithms for protecting LP information, instead, it supports a wide range of encryption solutions, including AES, Chacha, Salsa20, etc. These algorithms can be replaced by others, considering different system requirements and levels of security protection.

The structure of this paper is as follows, Section II explains the challenges faced and our design of PlateGuard to overcome these challenges. Section III discusses the results of our testing and the feasibility of the system. Section IV examines other work in this area and how PlateGuard builds upon those ideas. Section V presents conclusions and future directions that this work could be taken.

II. PlateGuard: License Plate Privacy Protection

The PlateGuard system aims to protect vehicles' licence plate data by hiding such sensitive information in videos/images captured by roadside cameras or those installed on autonomous vehicles. Although the cameras on autonomous vehicles are mainly used to help vehicles detect objects [6] and recognize road signs, they can potentially leak other vehicles' private information. On the other hand, to facilitate LP searching, e.g., in tracking suspicious vehicles, the PlateGuard system also allows users to quickly identify whether target vehicles (or certain licence plate numbers) appear in the captured videos. To ensure the PlateGuard system works effectively in practice, the system must be secure, scalable and responsive. In our approach to this system, we have three challenges to overcome in ensuring that (1) when searching for LPs we have an acceptable response time, (2) we maintain privacy of those whose LP data is captured, (3) and we can deploy the system in a large-scale setting, e.g., covering urban areas of a large city.

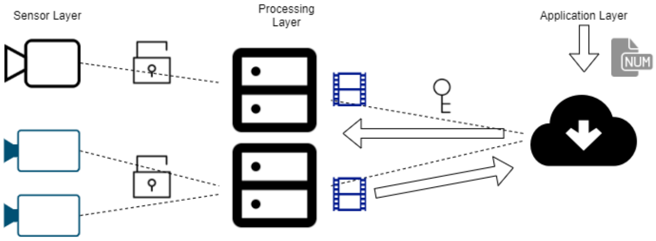

Fig. 1. System Architecture of PlateGuard. Video recorded from the Sensing layer is transferred to the Processing layer. The Application layer is where requests for decryption are generated. Once the request is sent the edge device sends back the video that has been decrypted.

To address these challenges, we develop the PlateGuard system using edge computing as a way to reduce processing load on the cloud and better protect privacy. Once captured by sensor devices, video data is transferred to edge devices for processing. The cloud service provides an interface for users to search and retrieve relevant data. Each of these parts will be explained in detail in the following sections.

A. System Architecture

To better understand the PlateGuard system, we decouple PlateGuard into three layers: sensing, processing, and application layer. As shown in Fig. 1, the sensing layer composes of various types of image-based sensors which are installed on roadside infrastructure (e.g., traffic lights) and/or autonomous vehicles. The processing layer lies in between the sensing and application layers, which provides the data processing capability so that videos captured by the sensing layer are processed and protected to hide vehicles' private information. Here, we assume the wireless communication capability is available on vehicles [22], and videos captured by vehicle cameras can be transmitted to edge servers wirelessly. The top layer is the application layer which is usually deployed on cloud services where users (e.g., from the law enforcement departments) can search and retrieve information from the encrypted database. The system design is rooted from the edge computing paradigm which has been proven to be able to offer better response time and scalability [23, 7], as well as stronger security and privacy protection of captured data [20].

B. Privacy Protection

The main challenge faced in developing the PlateGuard system is how to ensure the privacy of those whose plates are captured by the cameras installed on roadside infrastructures or in autonomous/conventional vehicles. The privacy protection of licence plate information is realized on two different levels. The videos captured from cameras are first encrypted locally using stream cipher techniques [11], and only encrypted data is transferred to nearby edge servers. Then, after edge servers receive the encrypted videos, they decrypt the data to obtain the original video for further processing. From the original videos, an edge server executes the ALPR algorithm [9] to detect possible licence plates. After locating the areas of each frame of video that contain LPs and extracting the pixel data, the edge server encrypts that data into a separate file along with the pixel location data to ensure the information can be retrieved later. This process is scalable, for instances where multiple plates are found in the same frame, each separate plate area will be retrieved from the frame and saved in individual per plate files. The key used to encrypt the data is the same as the license number detected. As such, if application layer users want to search a licence plate, the plate number itself will be used as the key to decrypt the captured videos. During the decryption process the separate files of LP data are scanned and decrypted on key successfully matching, then re-integrated back into the playback footage for review. Clearly, only frames that contain the target plate number can be decrypted successfully, i.e., the privacy of vehicles is protected.

Although the adversary could potentially launch a brute-force attack by searching all possible licence plates in the PlateGuard system, we believe a strict access control mechanism is in place on the cloud to prevent such attacks. Various access control solutions, e.g., role-based access control or attribute-based access control [19, 12], could limit the access of the PlateGuard system to only authorized users, e.g., police, Department of Transportation (DOT), etc.

The proposed design offers two additional advantages which make the system more user friendly. First, the PlateGuard system ensures should someone be able to gain access to the footage on the edge servers, he/she is only able to see the cars, not the LP data. As the edge servers are usually maintained by different entities, e.g., state DOTs or cellular companies, to provide various value-added services, it is possible that the videos are used for different purposes. For example, state DOTs may use the data to monitor traffic volumes or incidents; therefore, being able to view the vehicles is essential. As the PlateGuard system only protects the LP information, it would not hurt the traffic monitoring or incident detection applications. Second, because we store the footage on the edge servers, instead of transferring it directly to the cloud, the entire system's response time and scalability is significantly improved. In this way, we are reducing the number of attacks to the footage, as it is recorded by cameras and transferred to the edge device where it is processed, instead of a second transfer to the cloud through the public Internet. This also ensures that should one edge node become compromised, the footage kept on separate edge devices still remains secure as it is stored separately and vehicles' private information is protected.

C. Response Time

Given the massive amount of videos generated, it is crucial to design a system that can handle continuous image data in an effective manner. It is prohibitively expensive to transfer all videos to the cloud, processing data on edge servers is a more suitable solution as, not only will data transmission cause long network delay, the cloud may also become the point of failure of the entire system. To achieve a fast response time in processing/encrypting the captured videos, we offload the data processing tasks from the sensing layer to the processing layer. This allows the sensing layer to process the videos with a stream cipher encryption and then upload to the processing layer, where dedicated edge devices can work to process the videos fully for LP detection and encryption.

Although edge servers typically offer higher computing powers, it is a profound issue to achieve the best trade-off between fast response and less computation needed. As the data process task requires executing ALPR to identify LPs from videos and then encrypting regions in frames to protect users' privacy, the processing time could be very long. To address this issue, we design the PlateGuard system to dynamically execute ALPR based on previous data gathered. In PlateGuard, we use a combination of the openALPR algorithm and image trackers. The trackers used are initialized to the coordinates of the LP in the frame, then track the changes in the next frame to determine the new coordinates of the plate. To begin PlateGuard runs ALPR on the first frame, then uses trackers to monitor the LP location and skip frames where ALPR is run, increasing the amount of frames skipped by a factor of three each time ALPR is run.

On the ALPR run, if we detect a different number of LPs from the previous run, or if one of the trackers drops the LP in between ALPR runs, then the ALPR algorithm's run period is reset. For our testing the video we were processing was recorded at 30 frames/sec, PlateGuard prevents skipping more than 30 frames between runs to ensure new LPs are not missed. With the process of skipping ALPR runs in place we next need to ensure the accuracy of capturing plate numbers. Most cases for ALPR on a given frame produce an 80% accuracy rate on the LP number. To increase the accuracy, PlateGuard saves the detected plate numbers and then computes the mathematical mode of the LP numbers gathered to determine the correct plate number from the most frequently found number, using that as the key for the encryption process.

This method introduces another potential issue, with each run of the ALPR algorithm we need to determine if the plates detected are new plates or if they were previously captured. If they were previously captured, we need to ensure that the plate number is tracked correctly with the previously collected data for that plate. To accomplish this, we adapt the K Nearest Neighbor algorithm to PlateGuard. It compares the last location of each plate with the new locations detected by ALPR and appends the matches to the previously recorded data.

D. Scalability

Cities have access to dozens or even hundreds of cameras that record, whether they are stationary cameras or mobile cameras attached to vehicles. Some of these cameras may have ALPR functionality included, if that is the case, the licence plate information and locations in videos will be transmitted (in encryption) to the edge servers. Otherwise, encrypted videos will be transmitted to edge servers which then conduct the needed data processing. Due to the distributed nature of the computing process, e.g., data is processed and encrypted on either cameras or edge servers, the computation is implemented in a parallel manner. As such, the system's scalability is guaranteed by allocating adequate computing and network resources on edge servers. The required resources on an edge server could be estimated from traffic volume, which is highly predictable, allowing the resources to be dynamically allocated to meet real-time demands. For the extreme cases where the amount of videos suddenly increases in a certain area, due to traffic incidents, the serving edge server can mitigate its computing task to nearby edge servers [5].

The system is not only scalable in processing data, it also provides an efficient service to the end users. Once encrypted, the searching process to find specific LPs would introduce a large amount of latency as it works through this data storage. PlateGuard instead keeps the data distributed, i.e., encrypted videos are stored on the edge servers. When file retrieval is necessary the LP number needed is sent from the cloud to edge devices which then do the searching in parallel, signaling when the plate is found. As soon as the target LP is detected by an edge server, the trajectory of the vehicle can be predicted based on the vehicle's current movement. As such, only a subset of edge servers, along the vehicles path, will be triggered to continue searching the vehicle. In this way, as fewer edge servers get involved in the searching process, more searching requests can be served by the PlateGuard system, allowing our system to be scalable in terms of serving sheer amount of requests in a short period.

III. Performance Evaluation

When designing PlateGuard as a system to ensure privacy of the collected data and the ability to recover the data with legitimate cause, we design a set of metrics to test if these goals are achieved. For the system to be feasible it needs to process the data in a timely manner. We also check the memory footprint of the system through each of the buffer sizes to show the system could be run across different edge device configurations.

A. Dataset

While developing the PlateGuard system, we created a data set of videos recorded at a 4K resolution, recording cars in a parking lot, during a peak busy time. This allows capture of plates from cars that are entering and exiting, as well as for cars remaining in place while other cars drove past. This dataset provides scenarios to test for the entrance and exit of multiple plates per frame as well as handling lost tracking of obscured plates.

B. Experiments

As part of our evaluation of the PlateGuard system, and the benefits of deploying on an edge system instead of through a cloud platform, we utilize the dataset videos in a variety of experimental scenarios. We test the scalability of the system to handle multiple incoming video streams on cloud versus a distributed network of edge devices. We also evaluate the capabilities of the system to process frames quickly and accurately. Finally, we test a variety of encryption algorithms to determine the best one for performance and security protection.

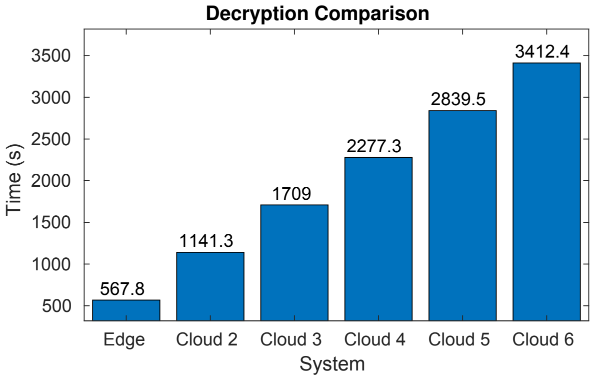

Fig. 2. Decryption on edge and cloud of a two-minute video. Simulating multiple cameras, cloud device decrypting N separate two-minute videos back to back. These tests were conducted using single core processing to provide comparative analysis of loads between base processing scenarios.

For our test system, we use a Raspberry Pi 3B+ to simulate the sensor layer, transmitting data to our server that acts as the edge device. The server is set up with 32 GB of RAM with a Intel Xeon quad-core processor at 3.3 GHz. The systems were running Ubuntu 16.04 LTS and OpenALPR v2.3.0.

C. Edge vs Cloud

To determine the effectiveness of the system on a cloud based system versus the distributed edge system, we look at the load on the cloud system when receiving multiple video streams, simulating processing information for multiple cameras in a service area. These tests prove inconclusive on our testing hardware as additional streams quickly overfill available system resources and cause the test system to crash. Further testing in this area is needed, however testing done for the decryption of multiple video streams can provide us with insight into the performance of the cloud based system versus the distributed edge network.

In the scalability tests for decryption, we find that the response time for the cloud system when tasked with searching multiple archived camera streams increases linearly in time to finish processing, as shown in Fig. 2. These results in a real-life scenario would depend on the number of camera streams to search as well as the length of the videos to process through. While this could be improved with optimizations to limit the videos to search based on outside knowledge of potential paths taken for the target car, we find that by using the distributed edge network of devices to process videos and act as parallel searches we are able to achieve a much faster response time on locating footage with the required plate.

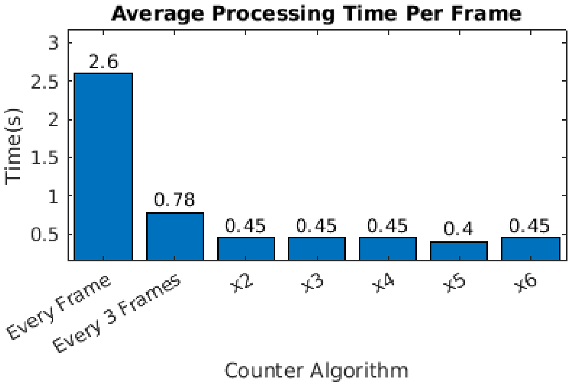

Fig. 3. The average processing time per frame is listed based on the counting algorithm used to determine how frequently ALPR runs, ranging from running every frame to delaying exponentially up to a factor of 5.

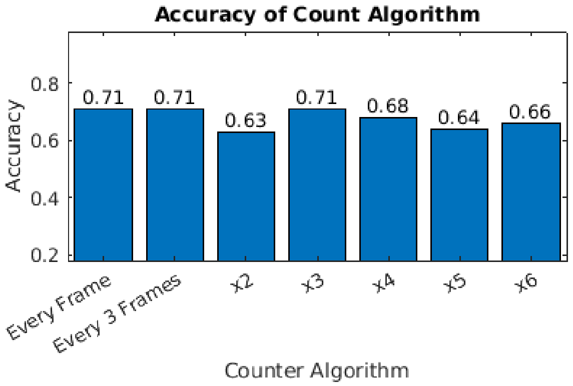

Fig. 4. Here we are examining how accurate each counting algorithm is with the delay between running ALPR on frames. We can see that using the exponential factor of 3 remains accurate while accuracy decreases at higher factors.

D. Frame Processing

As discussed in section II-C, running ALPR every frame is impractically slow, running an average of 2.6 seconds per frame to process. To determine the best rate to use for our dynamic ALPR method, we look at running ALPR every third frame, as well as exponentially increasing the delay between running ALPR by a factor of two, three, four, five and six, as see in Fig. 3. Running it every third frame shows significant improvement in processing time, cutting the time down to less than one second per frame. The exponential delay yields the best results, showing an average of .44 seconds per frame.

Using these results, we then look at the accuracy, ensuring that we retrieve the correct plate number so that decryption can occur if needed. As can be seen in Fig. 4, the exponential delay by a factor of three performs the same in accuracy as running ALPR every frame, at 71%.

Based on these two sets of experiments we decided to use the exponential increase by a factor of three to achieve the lower processing time while still keeping the highest possible accuracy of plate recognition. To further improve this, we next examine different buffer sizes to use on the system, in order to gather a large enough pool of data to determine the mode of the LPs gathered and correctly assign the plate number as key, while also ensuring the system can reasonably be deployed on edge devices comprised of different specifications.

We implement six different buffer sizes to gauge the accuracy benefit provided by each. We found that in buffer sizes of as little as 150 frames we were able to achieve a 100% accuracy of plate recognition, showing our system is able to accurately record and store the data, and be deployed on a variety of hardware specifications. For the bulk of our testing, we use a 300 frame buffer size as our test system is equipped with enough memory to maintain that buffer limit, but the system could easily be implemented on a system with less memory using the 150 frame buffer and still reliably process the video streams.

E. Encryption Processing

For our system we benchmark three encryption types to determine which provides the best performance while maintaining the security we needed. For our tests, we use the AES, ChaCha and Salsa20 encryption algorithms, looking at the average encryption time taken per frame of data. From our tests, Salsa20 performs the worst, averaging over $27ms$ per frame while ChaCha is the fastest, averaging just over $25ms$ per frame. We decide to use AES for in our system as the time taken is slightly over that of ChaCha, around $25.5ms$, while providing superior protection over the stream ciphers.

IV. Related Work

There has been significant work into the improvement of LP recognition, and the protection of PII collected by vehicle surveillance systems. In [2], work was done examining the prevalence of cameras in traffic scenarios finding that edge computing layers were needed to meet response time requirements. [1, 9] both examine the current state of LP recognition work, looking at various recording conditions as well as testing different algorithms on still image recording. [18, 16] look at novel methods of LP recognition, with [16] achieving an an 86.0% recognition rate and while [18] was able to achieve between 97 and 98% recognition accuracy on the plates using a CNN approach.

[15] explored the widespread use of video surveillance by law enforcement and the need to provide privacy protections on the data collected. [10, 4, 8] tested obscuring PII by distorting the images in an unrecoverable manner, leaving the non-PII portions of the image intact. In [3] research was done on methods of obscuring LP data with a pseudo-random generated number key for recovery of the data in the video stream, and in [13] looked at possible replacements for the physical LP, using electronic plates that would allow privacy of the individual with more accurate identification by law enforcement as needed.

Our work builds on this research by utilizing plate recognition methods for initial plate number recognition and, through the buffer system, generate a large enough sample of plate numbers for a more accurate final plate number reading to then be used as the key for encryption, allowing recovery of the LP later, while maintaining the privacy of others.

V. Conclusion and Future Work

The PlateGuard system we develop is able to reliably detect and encrypt the LP areas of supplied video streams. While this is achieving the goal we set out with, there is room to improve the system, with the LP recognition system being the main bottleneck in our system.

Our solution uses the open-source version of openALPR. We find in our initial tests of this system that the LP recognition confidence rate is typically 80%, and the processing time for accurate reads is much slower than is practical. At this time there are industry standard implementations of ALPR, including specialized cameras that implement ALPR directly, allowing more fine tuning of the camera to achieve optimal conditions for recognition. Utilizing these systems would improve the speed and accuracy of the PlateGuard system, allowing more focus on the privacy security of the data once detected.

Acknowledgement

The work has been supported in part by the National Science Foundation grants SaTC-2037982, CNS-1852134, OAC-2017564, and ECCS-2010332.

References

[1] Christos Nikolaos E. Anagnostopoulos et al. "License plate recognition from still images and video sequences: A survey". In: *IEEE Transactions on Intelligent Transportation Systems* 9.3 (Sept. 2008), pp. 377–391.

[2] Ganesh Ananthanarayanan et al. *Real-time Video Analytics time Video Analytics time Video Analytics –the killer appfor edge computing*. Tech. rep.

[3] Ikechukwu Azogu and Hong Liu. "Privacy-preserving license plate image processing". In: *2011 IEEE GLOBECOM Workshops, GC Wkshps 2011*. 2011, pp. 34–39.

[4] Margherita Bonetto et al. "Privacy in mini-drone based video surveillance". In: *Proceedings - International Conference on Image Processing, ICIP*. Vol. 2015-December. IEEE Computer Society, Dec. 2015, pp. 2464–2469.

[5] Lucas Chaufournier et al. "Fast transparent virtual machine migration in distributed edge clouds". In: *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. 2017, pp. 1–13.

[6] Qi Chen et al. "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds". In: *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2019, pp. 514–524.

[7] Qi Chen et al. "F-cooper: feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds". In: *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*. 2019, pp. 88–100.

[8] Liang Du and Haibin Ling. *Preservative License Plate De-identification for Privacy Protection*. Tech. rep.

[9] Shan Du et al. "Automatic license plate recognition (ALPR): A state-of-the-art review". In: *IEEE Transactions on circuits and systems for video technology* 23.2 (2012), pp. 311–325.

[10] Andrea Frome et al. "Large-scale privacy protection in Google Street View". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2009, pp. 2373–2380.

[11] Jovan Dj Golić. "Cryptanalysis of alleged A5 stream cipher". In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 1997, pp. 239–255.

[12] Vipul Goyal et al. "Attribute-based encryption for fine-grained access control of encrypted data". In: *Proceedings of the 13th ACM conference on Computer and communications security*. 2006, pp. 89–98.

[13] Jean Pierre Hubaux, Srdjan Čapkun, and Jun Luo. *The security and privacy of smart vehicles*. May 2004.

[14] Tom Jackman. *Judge orders Fairfax police to stop collecting data from license plate readers*. en. Apr. 2019.

[15] Bryce Newell. "Local Law Enforcement Jumps on the Big Data Bandwagon: Automated License Plate Recognition Systems, Information Privacy, and Access to Government Information". In: *Maine law review* 66 (May 2014), p. 398.

[16] Christos Nikolaos et al. "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications". In: *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS* 7.3 (2006).

[17] Murat Ozer. "Automatic licence plate reader (ALPR) technology". English. In: *The Police Journal: Theory, Practice and Principles* 89.2 (Apr. 2016), pp. 117–132.

[18] Rohith Polishetty, Mehdi Roopaei, and Paul Rad. "A Next-Generation Secure Cloud-Based Deep Learning License Plate Recognition for Smart Cities". In: Institute of Electrical and Electronics Engineers (IEEE), Feb. 2017, pp. 286–293.

[19] Ravi S Sandhu et al. "Role-based access control models". In: *Computer* 29.2 (1996), pp. 38–47.

[20] Weisong Shi et al. "Edge computing: Vision and challenges". In: *IEEE internet of things journal* 3.5 (2016), pp. 637–646.

[21] Zack Whittaker. *Police license plate readers are still exposed on the internet*. en-US. Apr. 2019.

[22] Qing Yang et al. "ACAR: Adaptive connectivity aware routing for vehicular ad hoc networks in city scenarios". In: *Mobile Networks and Applications* 15.1 (2010), pp. 36–60.

[23] Shanhe Yi et al. “Lavea: Latency-aware video analytics on edge computing platform”. In: *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. 2017, pp. 1–13.