Improved Crash Detection Algorithm for Vehicle Crash Detection

Byoungman An[*] and YoungSeop Kim[*†]
[*†] Electronic and Electrical Engineering, Dankook University, Korea Dankook

ABSTRACT

A majority of car crash is affected by careless driving that causes extensive economic and social costs, as well as injuries and fatalities. Thus, the research of precise crash detection systems is very significant issues in automotive safety. A lot of crash detection algorithms have been developed, but the coverage of these algorithms has been limited to few scenarios. Road scenes and situations need to be considered in order to expand the scope of a collision detection system to include a variety of collision modes. The proposed algorithm effectively handles the x, y, and z axes of the sensor, while considering time and suggests a method suitable for various real worlds. To reduce nuisance and false crash detection events, the algorithm discriminated between driving mode and parking mode. The performance of the suggested algorithm was evaluated under various scenarios, and it successfully discriminated between driving and parking modes, and it adjusted crash detection events depending on the real scenario. The proposed algorithm is expected to efficiently manage the space and lifespan of the storage device by allowing the vehicle's black box system to store only necessary crash event's videos.

Key Words : Crash Detection, Automotive, Crash Event, Black Box, Video Recording

1. Introduction

Vehicle crashes cause injuries and deaths to road users as well as enormous economic and social costs, and more than 50% of vehicle crashes are caused by careless driving. [1]. In order to ensure driving safety, in recent years, all countries have researched car crash avoidance technology. According to statistics, if you can give the driver an additional 0.5 second reaction time in a dangerous situation, you can reduce collision by 45%, so modern cars are equipped with all kinds of measurement and alarm systems to keep driving safety. Vehicle crash detection can be detected in a variety of ways, but g-sensors (accelerometer) are usually used. This method is commonly used because it is cheap and possible to easily determine the amount of change in the values of x, y, z. Today, being able to support humans in their daily work by developing autonomous systems is one of the biggest challenges of modern computer science. One example is an autonomous driving system that helps reduce fatalities from traffic accidents. A variety of new sensors have been used in the past few years for tasks such as recognition, navigation and manipulation of objects. [2]

A plenty of technology has been aimed to research the crash detection in vehicle. Collision probability data generated from Monte Carlo simulation taking driver behavior and vehicle dynamics into account, tracking algorithm using interactive multi-model particle filter, and threat assessment algorithm to estimate collision probability [2]. In another method, two models are considered: a model in which the follower maintains a safe distance and a model in which the follower maintains a safe time. Analyze distance delays and time delays caused by major vehicles' impact on followers. [3] There is also a way to develop new challenging benchmarks for stereo, optical flow, visual odometer/SLAM and 3D object detection tasks using autonomous driving platforms. [4]

The motion sensor method uses a complex motion processor to provide very accurate data and, if used near the engine, can also filter out vibrations. They used automotive

[†]E-mail: wangcho@dankook.ac.kr

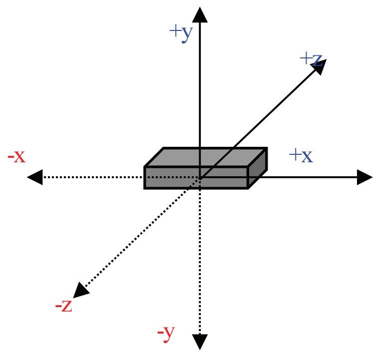

Fig. 1. G-sensor (accelerometer) axes.

sensors to reduce vibration and made readings very accurate. Algorithms that use data fusion between acceleration, deceleration, and tilt angles have a great success rate. No false positives or failed crashes were recorded in the test results. Collision detection is a very important feature for motorcycle occupants and is used by the E-call system, which can reduce the time between an accident and emergency service arrival by 50%. [5]

There is also a method that uses the image to recognize the license plate of the vehicle using Automatic license plate recognition (ALPR). [6] The distance is calculated from geometric derivation using additional descriptive data such as the distance between the cameras and other specific angles, such as the angle of view of the camera. This method achieves very high accuracy if the calculated distance between the vehicle and the camera is relatively accurate. The importance of measuring vehicle-to-vehicle distances lies in performing some important tasks in automotive vehicle systems, such as vehicle speed calculations and decision support (e.g. vehicle bypass, route change and speed control). In general, the work of the literature is divided into two types of distance measuring systems based on image processing: mono vision systems and stereo vision systems. [7]

In addition, there are algorithms using the ultrasonic sensor [8], method using vehicle-mounted device-based collision risk identification and warning system [9], pre-collision hot spot detection method using in-vehicle odometer recorder [10], magnetic resistance and sonar sensors to detect impending collisions in cars [11], a distance obstacle detection and safety distance calculation [12], adopting position control of an object using vision sensor [13], detecting the precise location of a moving car by using a design system of the gyro sensor [14] and investigating experimental security analysis of a modern automobile. [15]

In this paper, A novel algorithm for vehicle crash detection both driving and parking mode is presented. The purposed system concept includes a crash detection and safety distance calculation. The system detects the distance between the vehicle and the vehicle front (object) and uses vehicle CAN signal information from other devices. Moreover, by considering the situation over time, we devised a shock event algorithm that is more suitable for the real world. This paper is structured as follows: Section 2 describes the background with the major concept. Section 3 presents our preliminary work as well as purposed idea. Section 4 presents the experimental results from purposed designs. Finally, Section 5 concludes the paper before proposing the future of study.

2. Background

There are various methods for determining the crash event, using images or using autonomous driving techniques, but in this paper, we propose a method using g-sensor, which is a basic method.

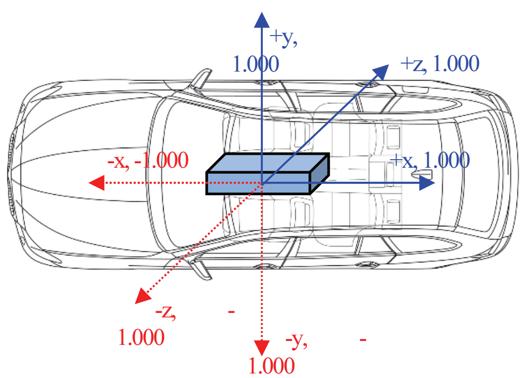

Fig. 2. Vehicle layout of g-sensor, the initial value is specified as 1.000 for convenience and understanding of calculation.

2.1 G-sensor

The g-sensor is commonly referred to as an accelerometer. They are used in various devices such as smartphones, vehicles and of course black boxes. The black box's g-sensor monitors the appropriate acceleration called G-Force. The 3-

axis accelerometer contains 3 accelerometers, one for each axis, which can measure the acceleration in the ±x, ±y, ±z, axes as shown in Fig. 1. The accelerometer output is highly dependent on the selected sensitivity expressed as G-force.

For example, an accelerometer with 2G sensitivity can output an acceleration of up to 2G. The value is read from 2 bytes register and the precision when using high sensitivity are sufficient for the consumer to use. Decreasing the accelerometer's sensitivity also decreases precision because the length of the register where the value is stored is the same. There are 3 registers where acceleration is stored, X acceleration register, Y acceleration register and Z acceleration register. The data collected on each axis is averaged and the values obtained are used to offset the output each time the sensor is read. Most accelerometers have offset registers, and writing the values obtained after calibration into these registers will offset the output.

For instance, the accelerometer's digital output has a full programmable range of ±2G, ±4G, ±8G and ±16GYou can select the appropriate sensitivity according to the application. In this paper, STMicroelectronics g-sensor was used and 2G was set up for measuring the accelerations of a vehicle.

2.2 Raw Data

The raw values of the accelerometer are read by the microcontroller and are obtained by using complementary filters to perform data fusion. In this case, the accelerometer is used to correct the drift of the gyroscope. Complementary filter is an equation that creates a weighted arithmetic mean between the values of the gyroscope and accelerometer.

$$angle = 0.90 \times (angle + gyroData \times \Delta t) + 0.10 \times accData \quad (1)$$

The weight chosen for the data coming from both sensors depends on the target application. Increasing the weight of the accelerometer data improves the responsiveness of the system, but increases its sensitivity to vibration, which makes the system unstable. If the gyro weighs more than 0.90, it makes a slow but very stable system. [5]

2.3. Car Crash Detection System

As shown in Fig. 2, the arrangement of g-sensors in a vehicle may vary depending on the vehicle's space and design. In general, the black box is placed in the center of the vehicle because it is considered to be placed in a safe location. In general, g-sensor uses the first value read after system booting as the default value and measures the change of the corresponding value. In this paper, the initial value is specified as 1.000 for convenience and understanding of calculation. There are countless crash detection algorithms.

In this paper, the method of detecting impact through absolute values is as follows.

$$\begin{aligned} |x_2| - |x_1| &> x\ axis\ threshold \\ |y_2| - |y_1| &> y\ axis\ threshold \\ |z_2| - |z_1| &> z\ axis\ threshold \end{aligned} \quad (2)$$

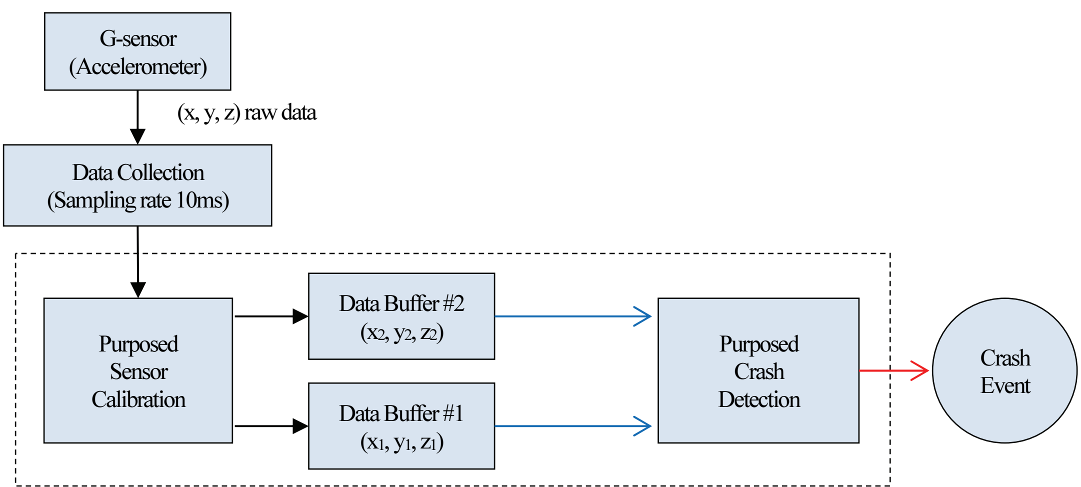

Fig. 3. Architecture of the proposed system.

The G-sensor value can reflect the period by sampling that is the changeable value. In this paper, the sampling period is 10ms for testing. x_1, y_1, z_1 are the previous values, and x_2, y_2, z_2 are the changed values after 10ms. When the previous value is subtracted from the changed value as in Equation 2, if the corresponding value is higher than each predetermined threshold value, it can be determined as the crash event. We will deal with it in more detail with the algorithm suggested in the chapter 3.

3. A Novel Algorithm of Vehicle Crash Detection

In this section, we present a novel algorithm for vehicle crash detection algorithm both driving mode and parking mode. Fig. 3 shows the architecture of the proposed system of vehicle crash event determining.

3.1 Purposed Sensor Calibration

Accurate sensor calibration is key to obtain reliable ground information. The calibration pipeline proceeds as follows: we calibrate the g-sensor intrinsically and extrinsically and rectify the input data. Existing techniques for this work were necessary because they were not accurate enough to calculate ground-based estimates. [4]

In this paper, we suggest a method to increase the accuracy of data. The sample rate plays a significant role in collecting accurate data. If the sample rate is low, the amount of collected data is increased as well as it causes computational complexity. On the other hand, On the other hand, if the sample rate is high, it is possible to miss a moment of crash event. The proposed method is presented in order to solve this problem as following. First, calculate the average sampling value of continuous sampling data as shown in Equation (3). Second, calculate the average of each odd and even sampled data as shown in Equation (4). Finally, by calculating the average of the two values as Equation (5), more reliable data were obtained than using each method (3, 4). The example below is a formula for calculating x_1. In this way, the values of y_1, z1 and x_2, y_2, z_1 can be obtained.

$$1^{st}\,avg.: (X1+X2+X3+X4+X5+X6)/6 \tag{3}$$

$$2^{nd}\,avg.: (X1+X3+X5)/3+(X2+X4+X6)/3 \tag{4}$$

$$x_1 = (1^{st}\,avg. + 2^{nd}\,avb.)/2 \tag{5}$$

The processing of the value calculated in the above equation will be dealt with in detail in the algorithm proposed in Section 3.2.

3.2 Purposed Crash Detection

The flow chart of the proposed crash detection algorithm is shown in Fig. 4. The proposed algorithm calculates the

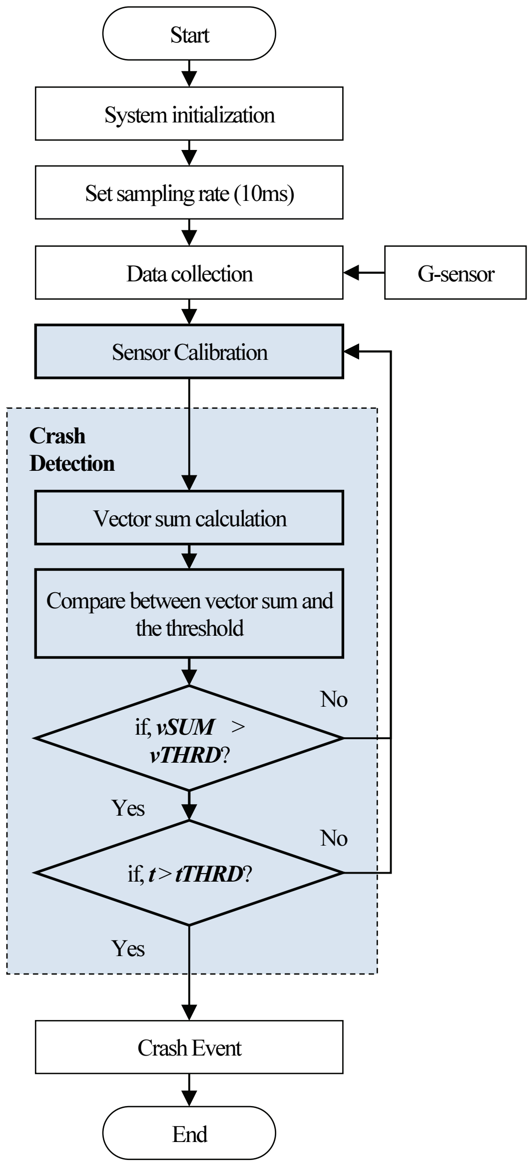

Fig. 4 Purposed flow chart of crash detection

vector sum of *x, y, z* and then calculates the difference between the previous value and the current value. In addition, we propose a more accurate crash detection method by considering the time at which the accumulated data were acquired, not only the instantaneous time when a crash is detected.

First, the equation for obtaining the vector sum and the difference between the previous and current values is as follows.

$$vSum = \sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2 - (z_2 - z_1)^2} \tag{6}$$

vSum > vTHRD → A possibility of crash

where, vSum = the vector sum
x2, y2, z2 = the current calibrated values
x1, y1, z1 = the previous calibrated values

if, *vSum > vTHRD*, it could be considered that there is a probability of crash in vehicle. *vTHRD* is a threshold values of vector sum. The *vTHRD* may vary depending on the vehicle environment. In this paper, the result of reviewing the vector sum of generally generated shocks is to be based on a value of about 2. The reason why it was selected as 2 in Chapter 4 was mentioned in detail. In a general situation, the impact was determined based on the corresponding value, and it is expected that it can be used by tuning according to the environment of the vehicle based on the value.

Time is always important. In many systems, time is considered, but most do not. Considering time, the system can be complicated, so it can be a poison rather than a benefit. In this paper, a simple relationship between time and impact was considered so as not to affect the design of the system as much as possible. It was a simple method, but what was considered and not considered showed a big difference in the results.

```
if ((vSum > vTHRD) && (t > tTHRD))
{
        Crash Detection → Crash Event          (7)
}
```

Table 1. Crash detection comparison

Scenario	Direction	C#1	C#2	Proposed
Driving	F	DET	DET	DET
	FR	DET	DET	DET
	R1	DET	-	DET
	R2	DET	-	DET
	RR	-	DET	DET
	R	DET	DET	DET
	LR	-	DET	DET
	L2	-	DET	DET
	L1	DET	-	DET
	FL	-	DET	DET
	T	DET	DET	-
	B	DET	DET	DET
Parking	F	DET	DET	DET
	FR	-	-	DET
	R1	DET	DET	DET
	R2	-	-	DET
	RR	-	DET	DET
	R	-	-	DET
	RL	DET	DET	DET
	L2	-	DET	DET
	L1	DET	-	DET
	FL	-	DET	DET
	T	DET	DET	-
	B	-	-	-

DET : Crash Detected
- : Not detected
C#1 : Commercial black-box device #1
C#2 : Commercial black-box device #2

where,
t = a crash detected time of vSUM > vTHRD
*tTHRD = sampling rate * calibration count * n*

tTHRD can be determined in various ways by n, and the calibration count is the amount of data used for the average used in equation (3,4,5). For example, if the sampling rate is 10ms, the calibration count is 6, and n is 10, the value of *tTHRD* can be 600ms. As confirmed through the experiment, it was confirmed that there was a considerable influence on

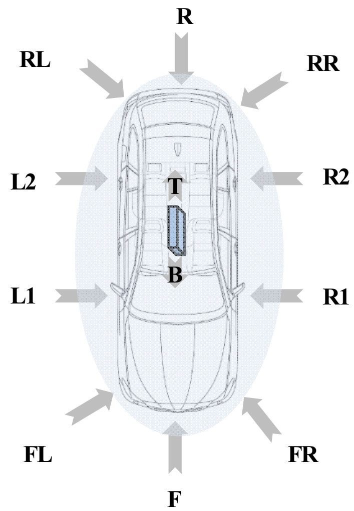

Fig. 5. Direction of vehicle crash detection during driving and parking mode

the values, and when the proposed method was applied, the impact could be judged more accurately than before.

If both of conditions were satisfied, it was determined as crash detection. The determined crash information occurred as an event and made it possible to process in the system. In this paper, when a crash event occurs, the system operates video recording, a system log file, an alarm sound to check whether it occurs, or not.

The next chapter 4 describes the experimental results of the method proposed in this algorithm.

4. Experimental Result

4.1 Performance Evaluation

The performance of the proposed crash detection algorithm was evaluated for two scenarios via offline simulations, including crash of various direction (see Fig. 5). To show the benefit of the proposed algorithm, driving, parking scenarios were considered. The driving scenario was considered to compare the crash detection between the proposed algorithm and a general black-box available in a market. The test environment was compared after installing the product proposed to the test vehicle and the product used, and the crash was applied in the actual road and parking environment to determine whether the crash was judged.

4.2 Crash Detection Comparison

The result of comparing the proposed algorithm and two commercial products are described in the **Table 1**. The experiment was carried out in the same environment for a fair test, and each experiment was performed 10 times for an accurate experiment, and the result by the average is indicated in the **Table 1**. The reason *vTHRD* mentioned in Section 3.2 was set to 2 is that when the initial value calibrated in most of the tests was 1.0. When the car is hit by a tennis ball that the value was around 2.0. Experiments are possible in a variety of ways, but in this paper, for safety, the vehicle is driving less than 10km during the driving test. In the test, a repeated test was conducted in which a tennis ball was lightly thrown and hit in the driving and parking state in each of the F (Front), FR(Front Right), R1 (Right Front Door), R2 (Right Rear Door), RR (Rear Right), R (Rear), RL (Rear Left), L1 (Left Front Door), L2 (Left Rear Door), FL (Front Left), T (Top), and B (Bottom). During driving, the bottom was tested over the bump, and during parking was omitted. Looking at the experimental results in Table 1, the proposed algorithm was able to detect impact in most directions of the overall average except for the top case. On the other hand, in the case of commercial products, when an impact is applied from a specific direction, there are many cases where the impact is relatively not recognized. In the proposed algorithm, the top crash event could be recognized by lowering the *vTHRD* value in the experimental vehicle. However, unnecessary shaking could be recognized as a crash in other points. Since the experiment considered the overall performance, tuning for the top was not possible, but if vTHRD and tTHRD are obtained through many experiments, it is expected that the impact on the entire area can be sufficiently detected. Overall, there was an improvement in the proposed method, and it is necessary to experiment in various vehicle environments.

5. Conclusion

This paper has proposed a novel crash detection algorithm applicable to general driving and parking mode scenarios that consider a various direction of car and that extracts as much information as possible from the calibrated data. The algorithm successfully detected in various directions of crash. It is expected that the proposed crash detection algorithm is reliable although further evaluations are required. The proposed algorithm can be used as an integrated collision detection algorithm by integrating tracking information from multiple sources for collision warning, avoidance and mitigation. Throwing new light on existing methods, we hope that the proposed algorithm will complement others and help to reduce overfitting to datasets with little test examples or training as well as contribute to the development of algorithms that work well in the crash detection system of vehicle.

Acknowledgements

This project would not have been possible without the kind support of many individuals. I would like to extend our sincere thanks to all of them. In addition, I would like to express our special gratitude and thanks to Management of Prof. Youngseop Kim for facilitating me by various resources and sufficient time whenever needed.

References

1. R. R. Knipling et al., "Assessment of IVHS countermeasures for collision avoidance systems," National Highway Traffic Safety Administration, Washington, DC, USA, Tech. Rep. DOT HS 807 995, May 1993.
2. T. Kim and H. Jeong, "A Novel Algorithm for Crash Detection Under General Road Scenes Using Crash Probabilities and an Interactive Multiple Model Particle Filter," in IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 6, pp. 2480-2490, Dec. 2014.
3. V. Naumov, "Analysis of Time and Distance Delays in Car Following Models," 2010 International Conference on Intelligent Systems, Modelling and Simulation, Liverpool, 2010, pp. 296-299.
4. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 3354-3361.
5. A. Cismas, M. Ioana, C. Vlad and G. Casu, "Crash Detection Using IMU Sensors," 2017 21st International Conference on Control Systems and Computer Science (CSCS), Bucharest, 2017, pp. 672-676.
6. S. Du, M. Ibrahim, M. Shehata and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 2, pp. 311-325, Feb. 2013.
7. A. Zaarane, I. Slimani, A. Hamdoun and I. Atouf, "Vehicle to vehicle distance measurement for self-driving systems," 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, 2019, pp. 1587-1591.
8. S. Haria, S. Anchaliya, V. Gala and T. Maru, "Car Crash Prevention and Detection System Using Sensors and Smart Poles," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 800-804.
9. Q. Zeng, C. Wu, L. Peng and H. Li, "Novel vehicle crash risk detection based on vehicular sensory system," 2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, 2015, pp. 622-626.
10. J. Lee and K. Jang, "Proactive Detection of Crash Hotspots Using In-Vehicle Driving Recorder," 2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), Nadi, 2016, pp. 193-198.
11. S. Taghvaeeyan and R. Rajamani, "Two-Dimensional Sensor System for Automotive Crash Prediction," in IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 1, pp. 178-190, Feb. 2014.
12. Y. Chen and C. Wang, "Vehicle Safety Distance Warning System: A Novel Algorithm for Vehicle Safety Distance Calculating Between Moving Cars," 2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring, Dublin, 2007, pp. 2570-2574.
13. E. Ha, G. Choi, "Position Control of an Object Using Vision Sensor," KSDET, vol.10, no.2, 25, 2011, pp. 49-56.
14. M. Kim, J. Kim, O, Yang, "Design of the Position Control System for Parabolic Antenna using Gyro Sensor," KSDET, vol. 12, no.2, 43, 2013, pp.85-91.
15. K. Koscher et al., "Experimental Security Analysis of a Modern Automobile," 2010 IEEE Symposium on Security and Privacy, Berkeley/Oakland, CA, 2010, pp. 447-462, doi: 10.1109/SP.2010.34.

접수일: 2020년 9월 6일, 심사일: 2020년 9월 16일,
게재확정일: 2020년 9월 22일