International Journal of Advanced Smart Convergence Vol.10 No.3 10-16 (2021)
http://dx.doi.org/10.7236/IJASC.2021.10.3.10

IJASC 21-3-2

Creating Covert Channel by Harnessing Shapley Values from Smartphone Sensor Data

Jun-Won Ho

Associate Professor, Department of Information Security, Seoul Women's University, Korea
jwho@swu.ac.kr

Abstract

In this paper, we devise a Shapley-value-based covert channel in smartphones. More specifically, unlike ordinary use of Shapley value in cooperative game, we make use of a series of Shapley values, which are computed from sensor data collected from smartphones, in order to create a covert channel between encoding smartphone and decoding smartphone. To the best of our knowledge, we are the first to contrive covert channel based on Shapley values. We evaluate the encoding process of our proposed covert channel through simulation and present our evaluation results.

Keywords: *Shapley value, covert channel, smartphone sensor data.*

1. Introduction

The gain level of each coalition in cooperative game theory can be computed in the form of Shapley value. Unlike the original usage of Shapley value, however, we utilize Shapley value for building up covert channel in smartphones. More specifically, an encoding smartphone first computes a series of Shapley values from a series of sensor data that it collects, compares the calculated Shapley values to a preset threshold, and associates Shapley values with covert bit 0 or 1 in line with the comparison results. It then encodes a series of cover bits into a series of sensor data corresponding to Shapley values associated with these covert bits and sends these sensor data into a decoding smartphone. After receiving a series of sensor data to which a series of covert bits are encoded, a decoding smartphone obtains a series of covert bits by performing decoding process. We evaluate the encoding process of our devised covert channel through simulation and present our evaluation results.

2. Related Work

Let us look into the relevant work in the research field of covet channel. In [1], random number generator is utilized for covert channel setup. In [2] and [3], covert channels are investigated in air-gapped computers and android operating system, respectively. In [4], covert channel rooted on the Sequential Probability Ratio Test and sensor data is examined in IoT. In [6], covert channel rooted on user-behavior is designed in

Manuscript Received: June. 14, 2021 / Revised: June. 19, 2021 / Accepted: June. 20, 2021
Corresponding Author: jwho@swu.ac.kr
Tel: +82-2-970-5607, Fax: +82-2-978-7931
Associate Professor, Department of Information Security, Seoul Women's University, South Korea

smartphones. In [7], optimal strategy is studied in stealthy communication rooted on game theory. In [8], IP timing covert channel is proposed. In [9], covert channel rooted on sensor is developed in android. In [10], sound is utilized for covert channel setup in smartphone. In [11], covert channel without permission is explored in android. However, these relevant work do not consider covert channels based on Shapley values. In order to magnify the extent of covert channel research, we devise covert channels based on Shapley values, which are calculated from smartphone sensor data.

3. Setting up Covert Channel from Shapley Values Computed from Smartphone Sensor Data

For covert channel establishment, we consider two smartphones such that one smartphone plays a role of encoding covert bits to a sequence of sensor data with the aid of Shapley values and transmit that sequence of sensor data to the other smartphone, which plays a role of decoding covert bits from that sequence of sensor data with the help of Shapley values, where covert bits are defined as secret information bits that need to be shared between two smartphones.

Let us denote a series of 3*u* sensor data values acquired from smartphone by G_1, G_2, …, G_{3u}. For encoding process, we first compute a Shapley value with three sensor data in accordance with Definition 19.14 in [5], where a sensor data acts as a player and thus we calculate a Shapley value with three players. The specific procedure for calculating Shapley values is represented as the following pseudo-codes.

```
N = 3;
t[0] = 0;
i=0;
while ( i < 3u ) {
  S = $G_{i+1}$ + $G_{i+2}$ + $G_{i+3}$;
  t[1] = $G_{i+1}$ / S;
  t[2] = $G_{i+2}$ / S;
  t[3] = $G_{i+3}$ / S;
  t[4] = t[1] + t[2] ;
  t[5] = t[2] + t[3] ;
  t[6] = t[1] + t[3] ;
  t[7] = t[1] + t[2] + t[3] ;
  ShapleyValue[i] = ((t[1]-t[0])*2+t[4]-t[2]+t[6]-t[3]+(t[7]-t[5])*2) / (N * (N-1));
  ShapleyValue[i+1] = ((t[2]-t[0])*2+t[4]-t[1]+t[5]-t[3]+(t[7]-t[6])*2) / (N * (N-1));
  ShapleyValue[i+2] = ((t[3]-t[0])*2+t[6]-t[1]+t[5]-t[2]+(t[7]-t[4])*2) / (N * (N-1));
  i=i+3;
}
```

Figure 1. Pseudo-codes for computing Shapley values.

Each time a Shapley value is computed, we check whether a Shapley value is greater than or equal to a covet bit mapping parameter ε. If so, a Shapley value is associated with a bit 1. If a Shapley value is less than ε, it is associated with a bit 0. Let us denote a series of covert bits to be encoded by P_1, P_2, …, P_k. If P_1 matches a bit associated with a ShapleyValue[j], P_1 is encoded to a sequence of sensor data values ranging from sensor data value corresponding to ShapleyValue[0] to sensor data value corresponding to the currently matched ShapleyValue[j]. If P_v ($v≥2$) matches a bit associated with a ShapleyValue[j], P_v is encoded to a sequence of sensor data values ranging from sensor data value corresponding to ShapleyValue[j-m+1] to sensor data value corresponding to the currently matched ShapleyValue[j], where the previously matched Shapley value is ShapleyValue[j-m].

When an encoding smartphone wishes to send a decoding smartphone *w* sequences of sensor data, Z_1, Z_2, …, Z_{w-1}, Z_w to which *w* covert bits are encoded, *w*+1 special sequences of information, Y_0, Y_1, Y_2, …, Y_{w-1}, Y_w are used for a decoding smartphone to perform decoding process correctly. More specifically, an encoding smartphone actually sends a decoding smartphone Y_0, Z_1, Y_1, Z_2, Y_2, …, Z_{w-1}, Y_{w-1}, Z_w, Y_w. Note that Y_0 indicates the start of transmission of *w* sequences of sensor data and each Y_x ($1≤x≤w$) represents the completion of transmission of each Z_x. Note that *w*+1 special sequences of information, Y_0, Y_1, Y_2, …, Y_{w-1}, Y_w are generated in pre-determined random manner and shared between an encoding smartphone and a decoding smartphone. Hence, a decoding smartphone can discern and extract Z_1, Z_2, …, Z_{w-1}, Z_w from Y_0, Z_1, Y_1, Z_2, Y_2, …, Z_{w-1}, Y_{w-1}, Z_w, Y_w sent by an encoding smartphone.

After obtaining Z_1, Z_2, …, Z_{w-1}, Z_w, a decoding smartphone computes multiple sequences of Shapley values from Z_1, Z_2, …, Z_{w-1}, Z_w . It then acquires *w* covert bits associated with multiple sequences of Shapley values by comparing each Shapley value to a covet bit mapping parameter ε.

4. Simulation

We write a simulation program in order to evaluate the encoding process of our devised covert channel based on Shapley values. In our simulation, we set the number of sensor data to 1000 and each sensor data value is selected at uniformly at random out of numbers ranging from 0 to 1000. Moreover, we configure the number of covert bits to 8 such that each covert bit is chosen at uniformly at random between 0 and 1. We also make a sequence of games consisting of 3 sensor data and acquire a Shapley value per a sensor data in each game. We set a covet bit mapping parameter ε ranging from 0.4 to 0.8. If a Shapley value is greater than or equal to ε, it is associated with a bit 1. Otherwise, it is associated with a bit 0. Finally, we employ the metrics defined in Table 1 for our evaluation. We repeat our simulation 100 times and present an average result of 100 runs. Note that our simulation is terminated when all of 8 covert bits are encoded in our simulation.

Table 1. Simulation metrics.

Simulation metrics	Definition
Number of sensor data used for encoding a bit	An average number of sensor data used for encoding a bit
Sensor data value used for encoding bits in 1 (resp. 0)	An average value of sensor data used for encoding bits in 1 (resp. 0)
Shapley value used for encoding bits in 1 (resp. 0)	An average value of Shapley values used for encoding bits in 1 (resp. 0)

We show our simulation results in Figures 1, 2, 3, 4, 5. We notice that an average number of sensor data used for encoding a bit tends to increase as a covet bit mapping parameter ε rises from Figure 1. This indicates that the higher ε leads to the larger number of sensor data used for encoding a bit on an average. On the other hand, from Figures 2, 3, 4, 5, we perceive that a growth in ε leads to a decrease in average values of sensor data used for encoding bits in 1 and 0 and a reduction in average values of Shapley values used for encoding bits in 1 and 0. This means that the higher covert bit mapping parameter contributes to the smaller values of sensor data and Shapley values used for encoding bits on an average.

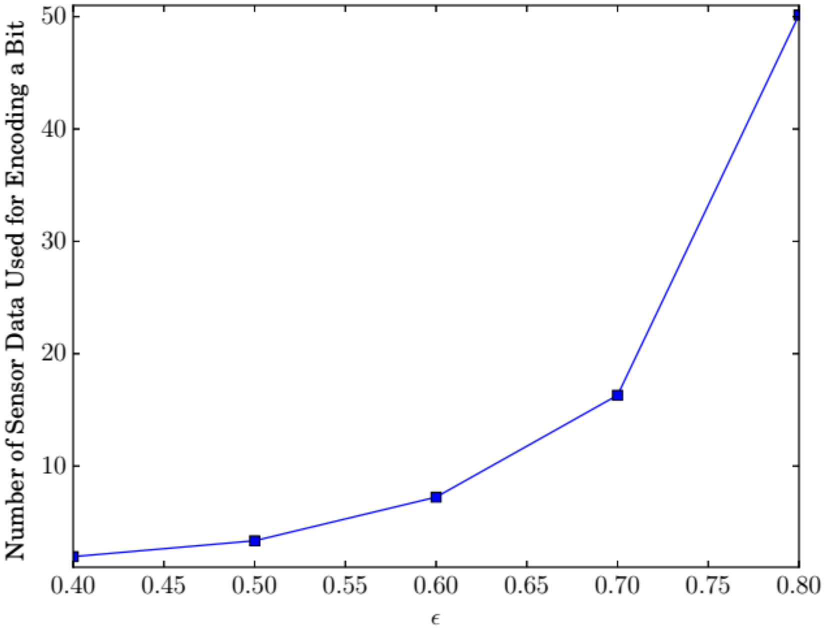

Figure 2. The effect of ε on an average number of sensor data used for encoding a bit.

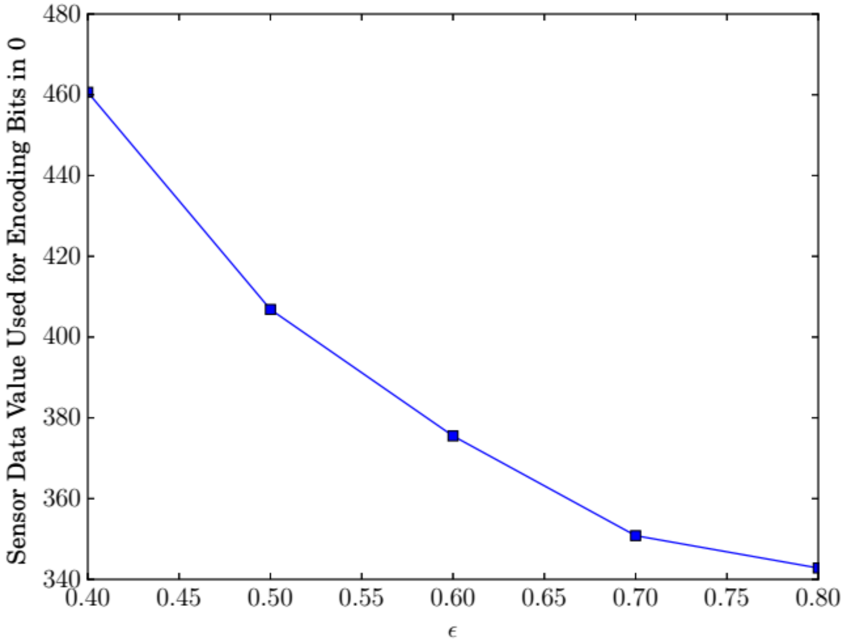

Figure 3. The effect of ε on an average value of sensor data used for encoding bits in 0.

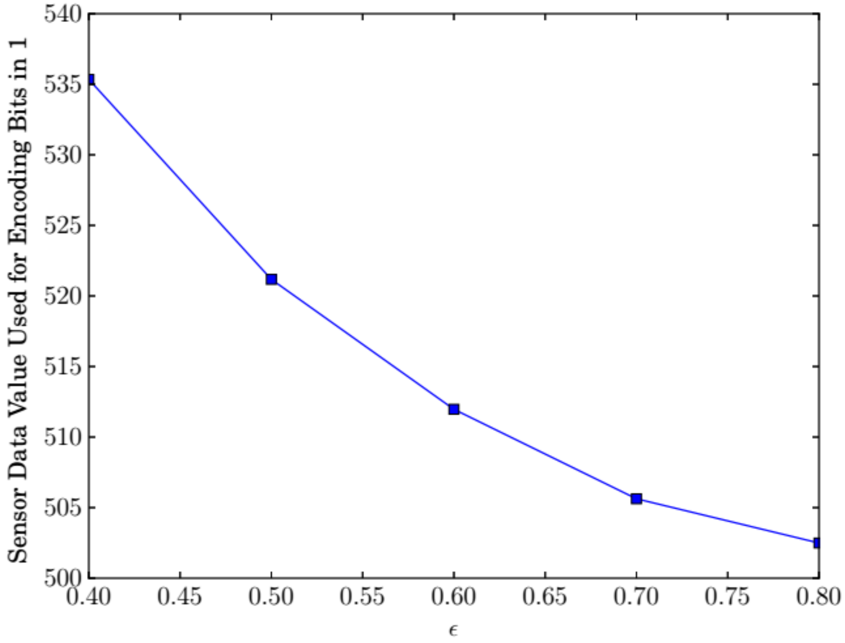

Figure 4. The effect of ε on an average value of sensor data used for encoding bits in 1.

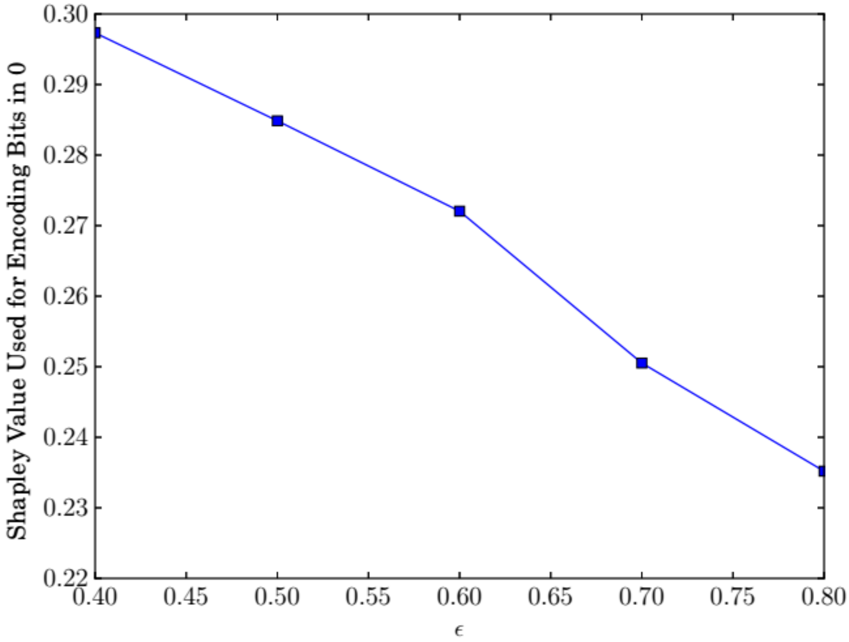

Figure 5. The effect of ε on an average value of Shapley values used for encoding bits in 0.

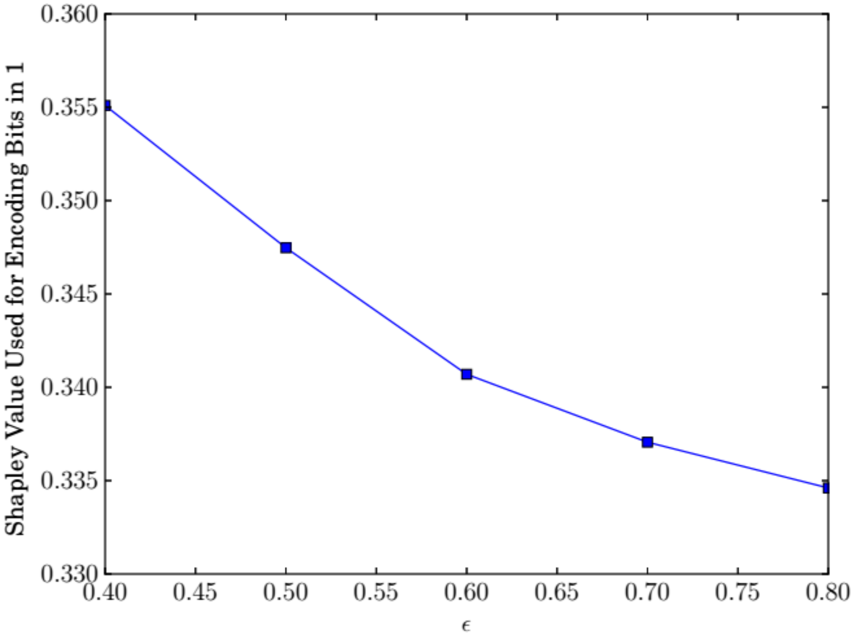

Figure 6. The effect of ε on an average value of Shapley values used for encoding bits in 1.

5. Conclusion

In this paper, we devise a covert channel rooted on Shapley values, which are calculated from sensor data collected from smartphone. We also evaluate the encoding process of our designed covert channel through simulation and present our evaluation results.

Acknowledgement

This work was supported by a research grant from Seoul Womenʹs University (2021-0092).

References

[1] D. Evtyushkin and D. Ponomarev, “Covert Channels through Random Number Generator: Mechanisms, Capacity Estimation and Mitigations,” In ACM CCS, 2016. DOI: https://doi.org/10.1145/2976749.2978374.

[2] M. Guri, Y. Solewicz, A. Daidakulov, and Y. Elovici, “Acoustic data exfiltration from speakerless air-gapped computers via covert hard-drive noise (diskfiltration),” In European Symposium on Research in Computer Security, 2017. DOI: https://doi.org/10.1007/978-3-319-66399-9_6.

[3] T. Heard, D. Johnson, and B. Stackpole, “Exploring a high-capacity covert channel on the Android operating system,” In IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2015. DOI: https://doi.org/10.1109/IDAACS.2015.7340765.

[4] J. Ho, "Covert Channel Establishment Through the Dynamic Adaptation of the Sequential Probability Ratio Test to Sensor Data in IoT," in IEEE Access, vol. 7, pp. 146093-146107, 2019. DOI: https://doi.org/10.1109/ACCESS.2019.2945974.

[5] M. Maschler, E. Solan, and S. Zamir, Game Theory, Cambridge University Press, Second Edition 2020.

[6] W. Qi, Y. Xu, W. Ding, Y. Jiang, J. Wang, and K. Lu, “Privacy Leaks When You Play Games: A Novel User-Behavior-Based Covert Channel on Smartphones,” In ICNP, 2015. DOI: https://doi.org/10.1109/ICNP.2015.40.

[7] J. Wang, W. Tang, X. Li and S. Li, "Optimal Strategy in Covert Communication based on Game Theory, " 2019 IEEE/CIC International Conference on Communications in China (ICCC), Changchun, China, 2019, pp. 189-194. DOI: https://doi.org/10.1109/ICCChina.2019.8855950

[8] S. Cabuk, C. Brodley, and C. Shields, “IP covert timing channels: Design and detection,” In ACM Conference on Computer and Communications Security, October 2004. DOI: https://doi.org/10.1145/1030083.1030108.

[9] A. Al-Haiqi, M. Ismail, and R. Nordin, “A New Sensors-Based Covert Channel on Android,” In The Sci entific World Journal, DOI: https://doi.org/10.1155/2014/969628, 2014.

[10] R. Schlegel, K. Zhang, X. Zhou, M. Intwala, A. Kapadia, and X. Wang, “Soundcomber: A Stealthy and Context Aware Sound Trojan for Smartphones,” In NDSS, 2011.

[11] K. Block, S. Narain, and G. Noubir, “An Autonomic and Permissionless Android Covert Channel,” In ACM WiSec, 2017. DOI: https://doi.org/10.1145/3098243.3098250.